ٹیسٹ ایڈیشن

ٹیکسٹ بُک آف

کمپیوٹر سائنس

برائے جماعت دہم

Free From Government
NOT FOR SALE

خیبر پختونخوا ٹیکسٹ بُک بورڈ، پشاور

مومن تو آپس میں بھائی بھائی ہیں۔ پس اپنے دو بھائیوں میں صلح کرادیا کرو اور اللہ سے ڈرتے رہو تاکہ تم پر رحمت کی جائے۔

(سورۃ الحجرات : ۱۰)

Khyber Pakhtunkhwa Textbook Board

Phase – V, Hayatabad Peshawar

Phone: 091-9217714-15 Fax: 091-9217163, Website: www.kptbb.gov.pk

TEXTBOOK FEEDBACK FORM

S#	Statement	Remarks
1.	Overall the book is interesting and user friendly.	
2.	The language and content of the book is age / grade appropriate and the content is free of grammatical and punctuation errors.	
3.	Content is supported with examples from real life / culture.	
4.	Contents / texts are authentic and updated.	
5.	Pictures / diagrams / graphs / illustrations are informative, relevant and clear if not, then identify them.	
6.	Activities, projects and additional work is suggested for reinforcement of concepts.	
7.	Assessment achievements are thought provoking and comprise cognitive, psychomotor and effective skills.	
8.	The textbook is easy to be covered within academic year.	

Page No.	Observation/comments	Suggested amendment along with rationale

General Remarks: ____________________

____

Name & Address: ____________________

Contact No. & Email: ____________________

ٹیکسٹ بک آف

کمپیوٹر سائنس

برائے جماعت دہم

X

21/09/23

خیبر پختونخوا ٹیکسٹ بک بورڈ، پشاور

جملہ حقوق بحق لیڈنگ بکس پبلشرز، پشاور محفوظ ہیں۔

منظور کردہ: نظامت نصاب و تعلیم اساتذہ خیبر پختونخوا، ایبٹ آباد

قومی نصاب بمطابق: 2009

مراسلہ نمبر: No.4449-51/S-04/TBB/Comp-X-U مورخہ: 30.07.2021

مصنف: محمد خالد، خالد اسد

مترجم: سعدیہ بصر

نگران نظر ثانی: گوہر علی خان، ڈائریکٹر، نصاب و تعلیم اساتذہ خیبر پختونخوا، ایبٹ آباد

فوکل پرسن: ذوالفقار خان، ایڈیشنل ڈائریکٹر، نظامت نصاب و تعلیم اساتذہ خیبر پختونخوا، ایبٹ آباد

نظر ثانی:

صلاح الدین ایوبی، اسسٹنٹ پروفیسر، گورنمنٹ پوسٹ گریجویٹ کالج نمبر ا، ایبٹ آباد

فہیم احمد، لیکچرار، گورنمنٹ پوسٹ گریجویٹ کالج منڈیاں، ایبٹ آباد

سہیل انجم، ایس ایس ٹی (آئی ٹی)، گورنمنٹ ہائی سکول نمبر ۴، ایبٹ آباد

سفیان مطلوب، ڈیسک آفیسر، نظامت نصاب و تعلیم اساتذہ خیبر پختونخوا، ایبٹ آباد

شکیل اقبال، سابقہ ماہر مضمون، خیبر پختونخوا ٹیکسٹ بک بورڈ، پشاور

ایڈیٹر: ناصر کمال، ڈیسک آفیسر، خیبر پختونخوا ٹیکسٹ بک بورڈ، پشاور

طباعت زیر نگرانی: ظریف المعانی (چیئرمین)

عبدالولی خان، ممبر (ای اینڈ پی)، خیبر پختونخوا ٹیکسٹ بک بورڈ، پشاور

تعلیمی سال: 2023-24ء

ویب سائٹ: www.kptbb.gov.pk

ای میل: membertbb@yahoo.com

ٹیلی فون نمبر: 091-9217159-60

فہرست عنوانات

صفحہ نمبر	عنوان	باب نمبر
1	**باب 1: پروگرامنگ کا طریقۂ کار**	
2	1.1 مسئلے کو سمجھنا	
2	1.1.1 مسئلہ	
3	1.1.2 مسئلے کا تجزیہ	
4	1.1.3 مسئلے کے حل کی منصوبہ بندی کرنا	
5	1.1.4 مسئلے کے صریح حل	
5	1.1.5 بہترین حل کا چناؤ کرنا	
6	1.2 الگورتھمز	
6	1.2.1 الگورتھم کیا ہے؟	
6	1.2.2 مسئلے کے حل میں الگورتھم کا کردار	
6	1.2.3 الگورتھم کی کارکردگی کا معیار جانچنے کے پیمانے	
8	1.2.4 مسائل کے حل کے لیے الگورتھم کا استعمال	
13	1.3 فلوچارٹ	
13	1.3.1 فلوچارٹ کیا ہے؟	
13	1.3.2 مسئلے کے حل میں فلوچارٹ کی اہمیت	
14	1.3.3 فلوچارٹ کی ضروریات	
15	1.3.4 فلوچارٹ کی علامات	
16	1.3.5 چند مسائل کے حل کے لیے لکھے گئے فلوچارٹ	
21	اہم نکات	
22	مشق	
24	**باب نمبر 2: C پروگرامنگ**	
25	2.1 پروگرامنگ کا تعارف	
25	2.1.1 کمپیوٹر پروگرام	

NOT FOR SALE

صفحہ نمبر	عنوان	باب نمبر
25	2.1.2 پروگرامنگ لینگویجز	
29	2.1.3 ہائی لیول لینگویجز کی خصوصیات	
31	2.1.4 چند مقبول ہائی لیول پروگرامنگ لینگویجز	
32	2.1.5 کمپائلر اور انٹرپرٹر	
33	2.2 پروگرامنگ کا انوائرمنٹ	
33	2.2.1 انٹیگریٹڈ ڈیویلپمنٹ انوائرمنٹ (IDE)	
34	2.2.2 C پروگرامنگ ماحول کے مختلف اجزاء	
35	2.3 پروگرامنگ کی اساسیات	
36	2.3.1 ہیڈر فائل	
37	2.3.2 کلیدی الفاظ	
37	2.3.3 C پروگرام کا بنیادی ڈھانچہ	
40	2.3.4 C میں کمنٹس کا استعمال	
41	2.4 متغیرات اور مستقلات	
41	2.4.1 متغیر	
42	2.4.2 مستقل	
42	2.4.3 متغیر کا نام رکھنے کے لیے اصول	
43	2.4.4 C میں استعمال ہونے والی ڈیٹا کے اقسام	
45	2.4.5 ٹائپ کاسٹنگ	
46	2.4.6 const کا استعمال	
47	2.4.7 ویری ایبل بنانا (Declaration) اور ابتدائی قیمت رکھنا (Initialization)	
48	اہم نکات	
49	مشق	
51	اِن پُٹ اور آؤٹ پُٹ کا عمل	باب نمبر 3:
52	3.1 اِن پُٹ آؤٹ پُٹ فنکشنز	
52	3.1.1 آؤٹ پُٹ فنکشن	

NOT FOR SALE

صفحہ نمبر	عنوان	باب نمبر
55	3.1.2 اِن پُٹ فنکشن	
60	3.1.3 اسٹیٹمنٹ ٹرمینیٹر (Semicolon)	
60	3.1.4 فارمیٹ سپیسیفائر	
62	3.1.5 اسکیپ سکیونسز	
63	3.2 آپریٹرز	
64	3.2.1 اریتھمیٹک حسابی آپریٹر	
65	3.2.2 اسائنمنٹ آپریٹر (=)	
65	3.2.3 کمپاؤنڈ اسائنمنٹ آپریٹرز	
66	3.2.4 انکریمنٹ آپریٹر (++)	
68	3.2.5 ڈیکریمنٹ آپریٹر (--)	
69	3.2.6 نسبتی یا ریلیشنل آپریٹرز	
70	3.2.7 منطقی یا لاجیکل آپریٹر	
71	3.2.8 اسائنمنٹ آپریٹر (=) اور مساوی ہے (==) آپریٹر میں فرق	
71	3.2.9 یونری اور بائنری آپریٹر میں فرق	
72	3.2.10 کنڈیشنل آپریٹر (: ?)	
72	3.2.11 آپریٹرز اور انکی فوقیت	
	اہم نکات	
	مشق	
77		باب نمبر4: کنٹرول سٹرکچر
78	4.1 کنٹرول سٹرکچر	
78	4.1.1 کنٹرول اسٹیٹمنٹس	
78	4.1.2 کنڈیشنل / سیلیکشن اسٹیٹمنٹس	
79	4.1.3 if اسٹیٹمنٹ	
81	4.1.4 if-else اسٹیٹمنٹ	
83	4.1.5 switch اسٹیٹمنٹ	

NOT FOR SALE

باب نمبر	عنوان	صفحہ نمبر
	4.1.6 نیسٹڈ سیلیکشن سٹرکچر کا استعمال	86
	4.1.7 سیلیکشن اسٹرکچر کے درمیان فرق	88
	4.1.8 یونٹ 1 کے فلوچارٹس کے لئے پروگرام کوڈز	88
	اہم نکات	90
	مشق	91
باب نمبر 5: لُوپ سٹرکچر		93
	5.1 تعارف	94
	5.1.1 لُوپ سٹرکچر	94
	5.1.2 for لُوپ	95
	5.1.3 while لُوپ	97
	5.1.4 do-while لُوپ	99
	5.1.5 break اسٹیٹمنٹ	102
	5.1.6 continue اسٹیٹمنٹ	103
	5.1.7 مختلف لُوپ سٹرکچرز میں فرق	104
	5.1.8 نیسٹڈ لُوپ	104
	5.1.9 باب 1 کے فلوچارٹس کے لئے کوڈز	106
	اہم نکات	108
	مشق	[illegible]
باب نمبر 6: کمپیوٹر لاجک اور لاجک گیٹس		110
	6.1 کمپیوٹر میں ڈیٹا کا اظہار	
	6.2 لاجک گیٹ	[illegible]
	6.2.1 ڈیجیٹل لاجک اور لاجک گیٹس	111
	6.2.2 بنیادی لاجک گیٹس	112
	6.2.3 ٹروتھ ٹیبل	113
	6.2.4 مزید لاجک گیٹس اور انکے ٹروتھ ٹیبل	116

NOT FOR SALE

باب نمبر	عنوان	صفحہ نمبر
	6.2.5 بنیادی گیٹس کی مدد سے NAND، NOR، XOR اور XNOR گیٹ بنانا	118
	6.2.6 بُولین ایکسپریشن کو لاجک سرکٹ میں تبدیل کرنا	121
	6.3 بُولین فنکشن کو کارناف میپ کی مدد سے مختصر کرنا	122
	6.3.1 کارناف میپ	122
	6.3.2 دو ویری ایبل والے بولین فنکشن کا K-Map کے ذریعے اختصار	125
	6.3.3 تین ویری ایبل والے بُولین فنکشن کا K-Map کے ذریعے اختصار	126
	6.3.4 مختصر شدہ فنکشن سے لاجک سرکٹ بنانا	128
	اہم نکات	131
	مشق	131
باب نمبر 7: ورلڈ وائیڈ ویب اور HTML		133
	7.1 ورلڈ وائیڈ ویب کا تعارف	134
	7.1.1 ورلڈ وائیڈ ویب کی اصطلاحات	134
	7.1.2 ویب سائٹس کی مختلف اقسام	141
	7.2 HTML کا تعارف	146
	7.2.1 ہائپر ٹیکسٹ مارک اپ لینگویج	146
	7.2.2 HTML ڈاکومنٹ بنانا اور براؤزر میں دکھانا	146
	7.2.3 HTML ٹیگز کا استعمال	148
	7.2.4 HTML کے ہیڈ اور باڈی ٹیگز	148
	7.3 ٹیکسٹ فارمیٹنگ	149
	7.3.1 ٹیکسٹ فارمیٹنگ کی اساسیات	149
	7.3.2 ٹیکسٹ فارمیٹنگ ٹیگز اور ان کا استعمال	151
	7.4 HTML میں فہرست / لسٹ بنانا	155
	7.4.1 لسٹ کی مختلف اقسام	156
	7.4.2 فہرست / لسٹ بنانا	157
	7.5 تصاویر اور پس منظر	161
	7.5.1 ویب پیج میں تصویر لگانا	161
	7.5.2 ویب پیج میں تصویر کا سائز مقرر کرنا	163

NOT FOR SALE

باب نمبر	عنوان	صفحہ نمبر
	7.5.3 پس منظر اور پیش منظر کا رنگ بدلنا	165
	7.5.4 پس منظر میں تصویر لگانا	166
	7.6 ہائپر لنک	167
	7.6.1 ہائپر لنک کیا ہے؟	167
	7.6.2 اینکر ٹیگ	167
	7.6.3 ویب پیج کی طرف ہائپر لنک بنانا	167
	7.6.4 ویب پیج کے اندر ہائپر لنک بنانا	168
	7.6.5 گرافیکل ہائپر لنک تیار کرنا	171
	7.7 جدول یا ٹیبل بنانا	172
	7.8 فریم بنانا	177
	7.8.1 فریم	177
	7.8.2 فریم سیٹ	177
	7.8.3 فریم سیٹ تیار کرنا	177
	7.8.4 ایک سے زیادہ فریمز کے ساتھ فریم سیٹ بنانا	178
	اہم نکات	180
	مشق	181
	سوالات کے جوابات	182
	فرہنگ	183
	انڈیکس	188
	ضمیمہ نمبر 1: اشکالات کی فہرست	191
	ضمیمہ نمبر 2: جدول کی فہرست	196
	ضمیمہ نمبر 3: نشانات کی فہرست	197
	ضمیمہ نمبر 4: ++ Notepad کی کمانڈز کی فہرست	198
	ضمیمہ نمبر 5: کتابیات	199

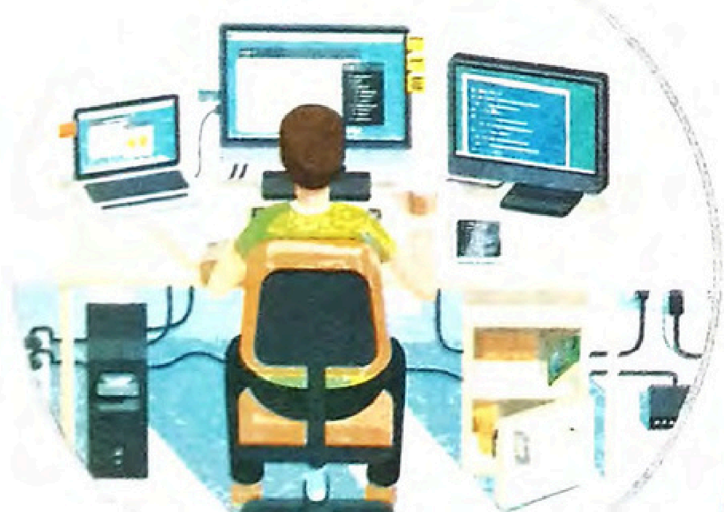
NOT FOR SALE

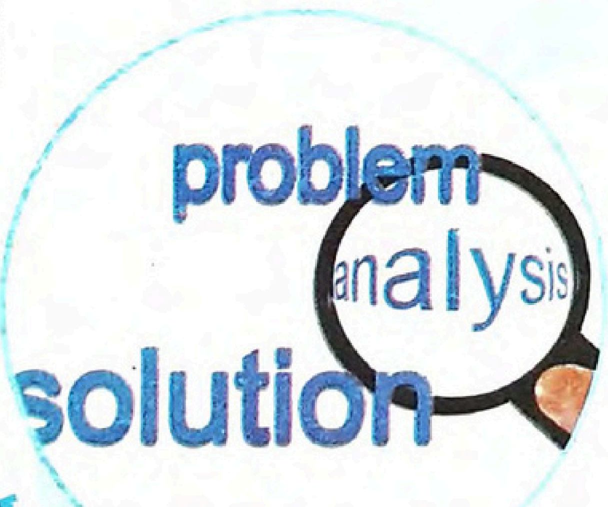
باب نمبر 1

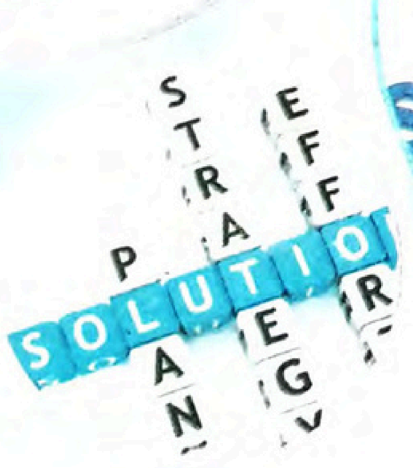
پروگرامنگ کا

اس باب کی تکمیل کے بعد طلبہ اس قابل ہو جائیں گے کہ وہ:

- مسئلے کی تعریف اور اس کا تجزیہ کر سکیں۔
- مسئلہ کے حل کے لیے منصوبہ بندی کر سکیں۔
- مسئلے کے تمام درست حل واضح کر کے ان میں سے بہترین حل کا انتخاب کر سکیں۔
- الگورتھم کی تعریف کر سکیں۔
- مسئلے کے حل میں الگورتھم کا کردار بیان کر سکیں۔
- الگورتھم کی کارکردگی جانچنے کے اصول بیان کر سکیں۔
- مختلف حسابی سوالات کے حل کے لیے الگورتھم لکھ سکیں۔
- فلوچارٹ کی تعریف کر سکیں۔
- فلوچارٹ کی اہمیت تفصیل سے بیان کر سکیں۔
- کسی بھی حل طلب مسئلے کی ضروریات کا تعین کر سکیں۔
- فلوچارٹ کی علامات کے استعمال کو سمجھ سکیں۔
- مختلف حسابی سوالات کے حل کے لیے فلوچارٹ واضح کر سکیں۔







NOT FOR SALE

تعارف (Introduction)

اس باب میں دو اہم نکات زیر بحث رہیں گے۔ سب سے پہلے ہم کمپیوٹر سائنس میں الگورتھم اور ڈیٹا سٹرکچر کے بر محل استعمال کا جائزہ لیں گے۔ خاص طور پر یہ کہ ہمیں ان دونوں موضوعات کو کیوں پڑھنا چاہیے اور ان کو جاننے کے بعد ہم کس طرح ایک اچھے حل کنندہ (Problem Solver) بن سکتے ہیں۔ اس کے بعد ہم فلو چارٹ کے تصورات کا جائزہ لیں گے۔

1.1 مسئلے کو سمجھنا (Understanding a Problem)

کمپیوٹر کے ذریعے کسی بھی مسئلے کو حل کرنے کے لیے ہمیں ایک کمپیوٹر پروگرام لکھنا پڑتا ہے۔ کمپیوٹر پروگرام لکھنے کے لیے ہمیں دو چیزیں معلوم ہونی چاہئیں؛ آپ کمپیوٹر سے کیا کروانا چاہتے ہیں اور وہ کونسے ذرائع ہیں جن کی مدد سے آپ کمپیوٹر سے کام لینا چاہ رہے ہیں۔ پہلا حصہ (جسے آپ الگورتھم بنانا (Algorithm Development) کہہ سکتے ہیں) دوسرے حصے کے مقابلے میں زیادہ مشکل ہے۔ اس مقصد کے لیے بہت سارے آلات جیسے کہ فلو چارٹ اور سوڈوکوڈ (Pseudo Code) موجود ہیں جن سے اس عمل میں مدد لی جاسکتی ہے۔

شکل نمبر 1.1 مسئلے کو سمجھنا

1.1.1 مسئلہ (The Problem)

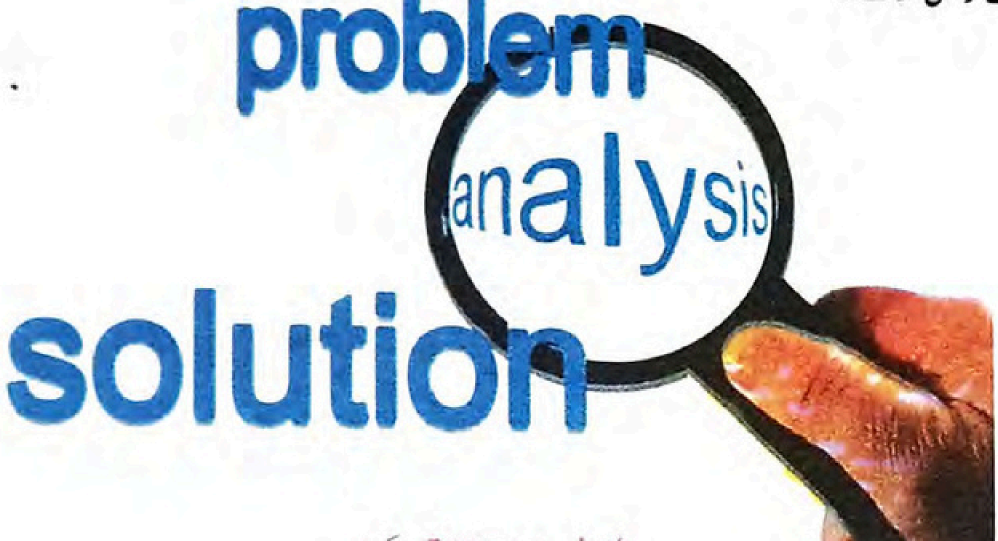
مسئلہ ایک رکاوٹ، مشکل یا چیلنج، یا کوئی بھی ایسی صورت حال ہے جس کو حل کرنا مقصود ہو۔ مسئلہ کی سب سے پہلی پہچان یہ ہے کہ جو ہونا چاہیئے اور جو ہو رہا ہے اس میں فرق آجائے۔ مسئلے کی آسان تعریف کرنے کے لیے اپنے آپ سے چند سوالات پوچھیں۔ مثال کے طور پر اگر آپ ایک ایسا پروگرام لکھنا چاہتے ہیں جس میں آپ پانچ نمبروں کا اوسط (Average) معلوم کرنا چاہتے ہیں تو اپنے آپ سے چند سوالات پوچھیں:

NOT FOR SALE

- اوسط کا اصل مطلب کیا ہے؟
- آپ کیسے اوسط معلوم کر سکتے ہیں؟
- اس طرح سے ہم مسئلے یا مشکل کو درست طور پر بیان کر سکتے ہیں۔

1.1.2 مسئلے کا تجزیہ (Problem Analysis)

کمپیوٹر بذات خود اتنا سمجھدار نہیں ہے کہ وہ کسی بھی مسئلے کا تجزیہ کر سکے اور پھر اس کے لیے حل بھی تجویز کر سکے۔ یہ صرف انسان (یعنی پروگرامر) کا کام ہے کہ وہ کسی بھی مسئلے کا تجزیہ کرے اور اس مسئلے کو حل کرنے کے لیے ہدایات (Code) لکھے اور پھر یہ ہدایات کمپیوٹر کو مہیا کر کے مسئلے کو حل کرے۔



شکل نمبر 1.2 مسئلے کا تجزیہ کرنا

مسئلے کے چناؤ اور وضاحت ہو جانے کے بعد سب سے پہلے جس چیز کی ضرورت ہے وہ یہ ہے کہ مسئلے کی وجوہات کا تعین کیا جائے۔

وجوہات کا تجزیہ کرنے کے لیے:

- مسئلے کے متعلق آپ کا تصور بالکل واضح ہونا چاہیے تا کہ آپ اس کا درست تجزیہ کر سکیں۔
- مسئلے کے متعلق ابھرنے والے تمام سوالات پر ایک دفعہ پھر نظر ثانی کر لیں تا کہ یہ اطمینان ہو جائے کہ آپ کے سامنے درج ذیل نکات واضح ہیں:
  - مسئلہ کس سے متعلق ہے؟
  - مسئلے سے کتنی دفعہ پالا پڑتا ہے؟
  - یہ کب اور کہاں پایا جاتا ہے؟
  - اس سے کون متاثر ہوتا ہے؟
- ممکنہ وجوہات کے بارے میں سوچیں۔
- وجوہات کی گروپ بندی کریں۔
- وجوہات کی تصویروں سے وضاحت کریں۔

جب مسئلے کی وجوہات کا تجزیہ ہو جائے تو پھر ان کے حل کے لیے منصوبہ بندی کریں۔

NOT FOR SALE

1.1.3 مسئلے کے حل کی منصوبہ بندی (Planning a Solution)

تجزیے کے بعد، مسئلے کے حل کی منصوبہ بندی کی جاتی ہے۔ کسی بھی مسئلے کو متبادل طریقوں سے حل کیا جا سکتا ہے۔ تاہم بہترین حل اسے کہا جائے گا جو کم لاگت، کم وقتی، تیز اور کم پیچیدہ ہو۔

شکل نمبر 1.3 مسئلے کے حل کے لیے منصوبہ بندی

مسئلے کے حل کے لیے ذیل میں چند طریقہ کار بیان کیے گئے ہیں:

- **تجرید Abstraction**: کسی بھی عملی صورتحال پر ممکنہ حل کو لاگو کرنے سے پہلے اس کو کسی بھی نمونہ (Model) پر آزمایا جائے۔
- **قیاس Analogy**: اُس حل کو آزمایا جائے جو اس مسئلے سے ملتے جلتے مسئلے کو حل کر چکا ہو۔
- **دماغ لڑنا Brainstorming**: لوگوں کے ایک گروہ میں تمام ممکنہ حل پیش کیے جائیں تا کہ ان پر بحث و مباحثہ کے بعد بہترین حل تیار کیا جا سکے اور اس کا چناؤ کیا جا سکے۔

شکل نمبر 1.4 ممکنہ جوابات (Brainstorming)

NOT FOR SALE

- **تقسیم اور فتح Divide and Conquer**: بڑے اور پیچیدہ مسئلے کو چھوٹے چھوٹے آسان قابل حل مسئلوں میں تقسیم کیا جائے تاکہ حل کرنا آسان ہو جائے۔
- **اختصار Reduction**: مسئلے کو دوسرے ایسے مسائل میں تبدیل کیا جائے جن کا حل پہلے سے موجود ہو۔
- **تحقیق Research**: اپنے تجربات کی روشنی میں، ایک جیسے مسائل کے پہلے سے موجود حل کو اپنے مقصد کے لیے تبدیل کریں۔
- **سعی و خطا Trial and Error**: تمام ممکنہ حل آزماناجب تک درست حل نہ مل جائے۔

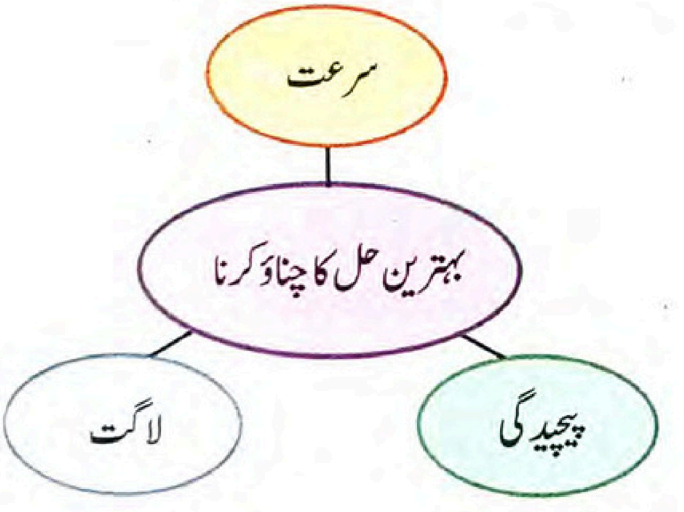
1.1.4 مسئلے کے صریح حل (Candid Solutions of a Problem)

"مسئلے کو حل کرنے کے عمل کو" حل ہی کہا جائے گا۔ مسئلے کا بہترین اور پسندیدہ حل وہ ہو گا جس کی لاگت کم سے کم ہو۔ مسئلے کو حل کرنے کے لیے بہت سے متبادل طریقے ہو سکتے ہیں۔ تاہم مسئلے کا بہترین اور مناسب ترین حل معلوم کرنا ہی مسئلے کا صریح حل قرار دیا جائے گا۔

1.1.5 بہترین حل کا چناؤ کرنا (Selecting the best Solution)

کسی بھی مسئلے کے بہترین حل کے چناؤ کے لیے درج ذیل عناصر بہت اہمیت کے حامل ہیں۔

- سرعت
- لاگت
- پیچیدگی



شکل نمبر 1.5 بہترین حل کا چناؤ کرنا

i. سرعت (Speed)

مسئلے کے لیے تجویز شدہ حل کتنی تیزی سے مسئلے کو حل کر سکتا ہے یہ ایک ایسا پیمانہ ہے جو مؤثر ترین حل کی نشاندہی کرتا ہے۔

ii. لاگت (Cost)

کسی حل کی لاگت سے مراد تجویز شدہ حل کے لیے درکار میموری، ہارڈ ویئر اور سافٹ ویئر ہیں۔ کسی بھی حل کو اس وقت بہترین کہا جائے گا جو کم سے کم لاگت میں مناسب ترین اور مطلوبہ نتائج مہیا کرے۔

NOT FOR SALE

iii. پیچیدگی (Complexity)

کسی بھی حل کے لیے درکار وسائل خاص طور پر، درکار کُل وقت، اس مسئلے کی پیچیدگی کو ظاہر کرتا ہے۔ یعنی کتنے مراحل سے کتنی میموری استعمال کر کے حل تک پہنچنا ہے۔

1.2 الگورتھمز (Algorithms)

1.2.1 الگورتھم کیا ہے؟ (What is an Algorithm)

مناسب اور منطقی اقدامات کی وہ فہرست جو کسی مخصوص مسئلے کے حل کے طور پر لکھی جاتی ہیں انہیں الگورتھم کہا جاتا ہے۔ غیر رسمی طور پر الگورتھم کی تعریف اس طرح بھی کی جاسکتی ہے کہ الگورتھم ایک ایسا مرحلہ وار حسابی عمل ہے جو ایک عدد یا رقم یا اعداد کے مجموعے کو بطور ان پُٹ لیتا ہے اور کوئی بھی رقم یا عدد یا اعداد کے مجموعہ کو نتیجہ کے طور پر آؤٹ پٹ میں ظاہر کرتا ہے۔ یعنی الگورتھم حسابی مرحلوں / اقدامات کا ایک ترتیب وار سلسلہ ہے جو اِن پُٹ کو آؤٹ پُٹ میں تبدیل کرتا ہے۔ پس الگورتھم کو کسی بھی مسئلے کے حل کے طور پر بھی استعمال کیا جاسکتا ہے۔

1.2.2 مسئلے کے حل میں الگورتھم کا کردار (Role of Algorithm in Problem Solving)

کسی بھی مسئلے کو حل کرنے کے لیے الگورتھم بہت اہم کردار ادا کرتا ہے۔ مثال کے طور پر اگر کوئی پیچیدہ مسئلہ درپیش ہو تو اس کے لیے بہت ساری معلومات جمع کرنا پڑیں گی اور بہت سارے ڈیٹا کو پروسیس کرنا پڑے گا۔ لیکن اگر ہم معلومات اکٹھا کرنے اور ڈیٹا پروسیس کرنے کے اس عمل کو چھوٹے چھوٹے ترتیب وار مراحل میں تقسیم کر دیں تو اس سے ہمارا پیچیدہ مسئلہ حل کرنا کافی حد تک آسان ہو جائے گا۔ پیچیدہ مسائل کے حل کے لئے تقسیم اور فتح (Divide and Conquer) ایک بہت ہی مشہور اور کار آمد اصول ہے جس سے مسائل کا حل پیش کرنا بہت آسان ہو جاتا ہے۔ مثال کے طور پر 20 نمبروں کا اوسط معلوم کرنا ہو تو اس مسئلے کو اگر چھوٹے چھوٹے مجرد (Discrete) اقدامات کے مجموعے یعنی الگورتھم میں تقسیم کیا جائے تو اس سے مسئلے کو آسانی سے حل کیا جاسکتا ہے۔

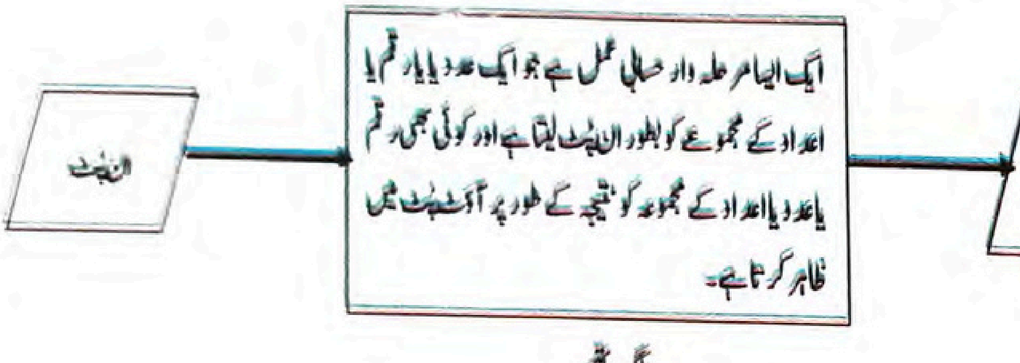
1. کوئی سے 20 اعداد منتخب کریں۔
2. تمام اعداد کو جمع کریں۔
3. مرحلہ نمبر 2 سے حاصل شدہ نتیجہ کو 20 سے یعنی اعداد کی کل تعداد سے تقسیم کریں۔
4. اوسط نکل آئے گا۔

1.2.3 الگورتھم کی کارکردگی کا معیار جانچنے کے پیمانے (Criteria for Measuring Efficiency of an Algorithm)

کمپیوٹر سائنس میں کسی بھی الگورتھم کی خصوصیات کو پرکھ کر ہم اس کی کارکردگی معلوم کرسکتے ہیں۔ الگورتھم کی خصوصیات میں ہم یہ جانچنے کی کوشش کرتے ہیں کہ کوئی بھی الگورتھم کس حد تک وسائل کو استعمال کرتا ہے؟ بہت ہی عام استعمال ہونے والے دو پیمانے درج ذیل ہیں:

- سرعت یا چلنے کے لیے درکار وقت (Speed or Running Time): یہ وہ وقت ہے جو ایک الگورتھم مکمل ہونے کے لیے لیتا ہے۔
- درکار میموری: کسی آپریشن کے دوران الگورتھم کی طرف سے استعمال کی گئی میموری۔

مزید بر آں اسکی کارکردگی کو درج ذیل عوامل کی بنیاد پر ماپا جا سکتا ہے۔



شکل نمبر 1.6 الگورتھم کی کارکردگی

i. درکار ان پُٹ (Inputs needed)

کسی الگورتھم کو ملنے والی ان پٹ کی مقدار سے پتا لگایا جاسکتا ہے کہ اس الگورتھم کو کتنا وقت درکار ہے۔
مثال کے طور پر ایک چھوٹے قالب (Matrix) کے مقابلے میں ایک بڑے قالب کو ضرب دینے کے لیے زیادہ وقت درکار ہو گا۔ اسی طرح کبھی کبھار کوئی الگورتھم ایک جیسی مقدار کی اِن پُٹ کے لئے مختلف وقت صرف کرتا ہے۔

ii. پروسیسنگ کو مکمل کرنا (Processing to be completed)

الگورتھم کو اپنی پروسیسنگ مکمل کرنے کے لیے کتنا وقت درکار ہے اس سے الگورتھم کی کارکردگی کو ماپا جا سکتا ہے۔ پروسیسنگ کی تکمیل کا وقت سائیکل ٹائم، انتظار (Wait) اور پروسیسنگ کا وقت (Processing Time) پر مشتمل ہوتا ہے۔

iii. فیصلہ کرنا (Decision to be taken)

مسئلے کو حل کرنے کے دوران مختلف فیصلے کرنا ایک بہت ہی اہم صلاحیت ہے جس سے الگورتھم کی کارکردگی میں بہتری لائی جاسکتی ہے اور موزوں ترین نتائج حاصل کیئے جاسکتے ہیں۔

iv. حاصل شدہ آؤٹ پُٹ (Outputs to be provided)

اگر ایک الگورتھم اپنی ہر اِن پٹ کے لیے مطلوبہ نتائج مہیا کرے تو اس کا مطلب یہ ہے کہ یہ الگورتھم درست ہے۔ کیونکہ ایک غلط الگورتھم کچھ ان پُٹس پر ختم ہی نہیں ہو گا یا وہ ایسے نتائج فراہم کرے گا جو کہ نا قابل قبول ہوں گے۔ اس لیے الگورتھم والے حل جو قابل قبول نتائج فراہم کریں ان کو ہی مسئلے کا بہترین حل کہا جائے گا۔

NOT FOR SALE

1.2.4 مسائل کے حل کے لیے الگورتھم کا استعمال (Algorithm for Solving Some Problems)

مسئلہ نمبر 1

پانچ نمبروں کا حاصل جمع، حاصل ضرب اور اوسط معلوم کرنا۔

الگورتھم: فرض کیا کہ پانچ اعداد دیئے گئے ہیں n1, n2, n3, n4, n5۔ یہ الگورتھم ان اعداد کا حاصل ضرب PRO، حاصل جمع SUM اور اوسط AVG معلوم کرے گا۔

```
Step 1.   Read n1, n2, n3, n4, n5
Step 2.   Set SUM =n1+n2+n3+n4+n5
Step 3.   Set PRO =n1*n2*n3*n4*n5
Step 4.   Set AVG =SUM/5
Step 5.   Write SUM, PRO, AVG
Step 6.   Exit
```

مسئلہ نمبر 2

دیئے گئے تین غیر مساوی اعداد میں سے سب سے بڑے عدد کو منتخب کرنا۔

الگورتھم: فرض کیا تین اعداد n1, n2, n3 ہیں۔ یہ الگورتھم ان تین اعداد میں سے سب سے بڑا عدد (LARG) معلوم کرے گا۔

```
Step 1.   Read n1, n2, n3
Step 2.   If n1>n2 AND n1>n3 then Set LARG=n1
          Else if n2>n3 then Set LARG=n2
          Else Set LARG=n3
          [End of If structure]
Step 3.   Write LARG
Step 4.   Exit
```

مسئلہ نمبر 3

دی گئی قوت اور کمیت کی بنیاد پر کسی بھی متحرک جسم کا اسراع معلوم کرنا۔

الگورتھم: یہ الگورتھم کسی بھی متحرک چیز کا اسراع ACCEL معلوم کرے گا جبکہ اس چیز کی کمیت MASS اور لگائی گئی قوت FORCE دی گئی ہو۔

```
Step 1.   Read MASS, FORCE
Step 2.   Set ACCEL= FORCE / MASS
Step 3.   Write ACCEL
Step 4.   Exit
```

مسئلہ نمبر 4

کسی کُرے (Sphere) اور مکعب (Cube) کا حجم معلوم کرنا۔

الگورتھم: یہ الگورتھم کسی مکعب کا حجم VC جس کے ایک ضلع کی لمبائی A ہو اور کُرے کا حجم VS جس کا رداس (Radius) R ہو، معلوم کرے گا۔

```
Step 1.    Read A. R.
Step 2.    Set VC= A*A*A
Step 3.    Set VS= 4/3*3.14*R*R*R
Step 4.    Write VC. VS
Step 5.    Exit
```

مسئلہ نمبر 5

مثلث کا رقبہ معلوم کرنا۔

الگورتھم: یہ الگورتھم کسی بھی مثلث، جس کے قاعدہ کی لمبائی B اور اُنچائی H ہو، کا رقبہ AREA معلوم کرتا ہے۔

```
Step 1.    Read B. H
Step 2.    Set AREA= B*H / 2
Step 3.    Write AREA
Step 4.    Exit
```

مسئلہ نمبر 6

متوازی الاضلاع (Parallelogram) کا رقبہ معلوم کرنا۔

الگورتھم: یہ الگورتھم ایسے متوازی الاضلاع کا رقبہ AP معلوم کرے گا جس کے قاعدہ کی لمبائی B اور اُنچائی H ہو۔

```
Step 1.    Read B. H
Step 2.    Set AP= B*H
Step 3.    Write AP
Step 4.    Exit
```

مسئلہ نمبر 7

ذوذنقہ (Trapezium) کا رقبہ معلوم کرنا۔

الگورتھم: یہ الگورتھم کسی ذوذنقہ کا رقبہ ATRAP معلوم کرتا ہے، جس کے متوازی اضلاع کی لمبائی A اور B ہو اونچائی H (اضلاع کے درمیان عمودی فاصلہ) ہو۔

Step 1.   Read A. B

Step 2.   Set ATRAP $= \frac{(A+B)}{2} \times H$

Step 3.   Write ATRAP

Step 4.   Exit

مسئلہ نمبر 8

مضامین میں حاصل شدہ نمبروں کی بنیاد پر گریڈ کا تعین کرنا۔

الگورتھم: یہ الگورتھم مضامین میں حاصل شدہ نمبروں SUBMARKS کی بنیاد پر ان مضامین کے لیے گریڈ GRADE کا تعین کرے گا۔

```
Step 1.    Read SUBMARKS
Step 2.    If  SUBMARKS>=80 Set GRADE= A+
           Else if  SUBMARKS>=70 Set GRADE= A
           Else if  SUBMARKS>=60 Set GRADE= B
           Else if  SUBMARKS>=50 Set GRADE= C
```


NOT FOR SALE

Else if SUBMARKS>=40 Set GRADE= D
Else if SUBMARKS>=33 Set GRADE= E
Else Set GRADE=FAIL
[End of if structure]

Step 3. Write GRADE
Step 4. Exit

مسئلہ نمبر 9

کئی سال بعد متوقع آبادی معلوم کرنا۔

الگورتھم: یہ الگورتھم موجود آبادی P اور نمو کی شرح R رکھنے والے علاقے کی T سال بعد متوقع آبادی PE کو معلوم کرتا ہے۔

Step 1. Read P, R, T
Step 2. Set EP= P*R*T
Step 3. Write EP
Step 4. Exit

مسئلہ نمبر 10

دیئے گئے عدد کی طاقت معلوم کرنا۔

الگورتھم: یہ الگورتھم کسی بھی دیئے گئے عدد NUM کی X طاقت EXP معلوم کرتا ہے۔

Step 1. Read NUM, X
Step 2. Set EXP= NUM ^ X (^ کیرٹ قوت نما کے لیے استعمال ہوتا ہے)
Step 3. Write EXP
Step 4. Exit

مسئلہ نمبر 11

دیئے گئے دو اعداد کے درمیان پے در پے طاق اعداد معلوم کرنا۔

الگورتھم: یہ الگورتھم کسی عدد NUM سے لے کر آخری عدد N تک مسلسل طاق اعداد کو معلوم کرتا ہے۔

Step 1. Read NUM, N
Step 2. Repeat Steps 3 and 4 while NUM<= N
Step 3. Write NUM
Step 4. Set NUM=NUM+2
[End of while loop]
Step 5. Exit

مسئلہ نمبر 12

کسی بھی دیئے ہوئے عدد کے لیے دو اعداد کے درمیان جزو ضربی (Multiples) کی کل تعداد معلوم کرنا۔

الگورتھم: یہ الگورتھم کسی عدد NUM کے لیے دو اعداد N1 اور N2 کے درمیان آنے والے اجزائے ضربی کی کل تعداد معلوم کرتا ہے جبکہ N2 بڑا ہے N1 سے۔

NOT FOR SALE

```
Step 1.   Read NUM, N1, N2
Step 2.   Set M = 0 [Initialize M.]
Step 3.   For A=N1 to N2
Step 4.   Set MUL= NUM % A    (% is used for remainder)
Step 5.   If MUL = 0 Then M=M+1
Step 6.   Next A
Step 7.   Write M
Step 8.   Exit
```

مسئلہ نمبر 13

کسی بھی دیئے گئے عدد کے لیے ضرب کا پہاڑہ بنانا۔

الگورتھم: یہ الگورتھم دیئے گئے عدد NUM کے لیے پہاڑہ بناتا ہے۔

```
Step 1.   Set C=1 [Initialize counter.]
Step 2.   Read NUM
Step 3.   Repeat Steps 4,5 and 6 while C<=10
Step 4.   Set TAB= NUM*C
Step 5.   Write NUM '*' C '=' TAB
Step 6.   C=C+1 [Increment counter variable.]
          [End of while loop]
Step 7.   Exit
```

مسئلہ نمبر 14

سینٹی گریڈ درجہ حرارت کو فارن ہائیٹ درجہ حرارت میں تبدیل کرنا۔

الگورتھم: یہ الگورتھم سینٹی گریڈ درجہ حرارت C کو فارن ہائیٹ F درجہ حرارت میں تبدیل کرتا ہے۔

```
Step 1.   Read C
Step 2.   Set F= 32+C*9/5
Step 3.   Write F
Step 4.   Exit
```

مسئلہ نمبر 15

فارن ہائیٹ درجہ حرارت کو سینٹی گریڈ درجہ حرارت میں تبدیل کرنا۔

الگورتھم: یہ الگورتھم فارن ہائیٹ درجہ حرارت F کو سینٹی گریڈ C درجہ حرارت میں تبدیل کرتا ہے۔

```
Step 1.   Read F
Step 2.   Set C= (F-32)*5/9
Step 3.   Write C
Step 4.   Exit
```

مسئلہ نمبر 16

دو صحیح اعداد کے درمیان تمام جفت اعداد معلوم کرنا (جبکہ n2 بڑا ہے n1 سے)۔

الگورتھم: یہ الگورتھم دو اعداد n1 اور n2 کے درمیان آنے والے تمام جفت اعداد معلوم کرتا ہے جبکہ n1 چھوٹا ہے n2 سے اور n1 ایک جفت عدد ہے۔

NOT FOR SALE

Array

Step 1. Read n1, n2
Step 2. Set EVEN=n1
Step 3. Repeat Steps 4 and 5 while EVEN<= n2
Step 4. Write EVEN
Step 5. Set EVEN=EVEN +2
[End of while loop]
Step 6. Exit

مسئلہ نمبر 17

اعداد کی دو علیحدہ فہرستوں میں موجود نمبروں کو جمع کرنا۔

الگورتھم: یہ الگورتھم دو فہرستوں L1[n] اور L2[n] میں موجود متعلقہ اعداد کو بالترتیب جمع کر کے ایک تیسری فہرست L3[n] میں رکھتا ہے۔

Step 1. Read L1[n], L2[n]
Step 2. Set I=1 [Initialize counter.]
Step 3. Repeat Steps 4 and 5 while I<= n
Step 4. Set L3[I]=L1[I]+L2[I]
Step 5. Set I=I+1 [Increment counter.]
[End of while loop]
Step 6. Exit

مسئلہ نمبر 18

دو اعداد کا عاد اعظم (GCD) معلوم کرنا۔

الگورتھم: یہ الگورتھم دیئے گئے دو اعداد M اور N کا عاد اعظم (GCD) معلوم کرتا ہے۔

Step 1. Read M, N
Step 2. If N=0, Go to Step 5; otherwise, Proceed to Step 3
Step 3. Divide M by N and assign the value of the remainder to R
Step 4. Assign the value of N to M and the value of R to N and go to Step 2
Step 5. Write M
Step 6. Exit

مسئلہ نمبر 19

معلوم کرنا کہ ایک دیا گیا عدد مفرد عدد (Prime Number) ہے یا نہیں۔

الگورتھم: یہ الگورتھم معلوم کرے گا کہ ایک دیا گیا عدد NUM مفرد عدد (Prime Number) ہے یا نہیں ۔

Step 1. Read NUM
Step 2. Set i=2 [Initialize counter]
Step 3. Repeat step 4 and 5 while i<=NUM/2
Step 4. If NUM%i=0 Then Write 'Not a prime number' and go to Step 7
Step 5. Set i=i+1 [Increment counter.]
[End of while loop]
Step 6. If i=NUM/2, Then Write 'Prime number'
Step 7. Exit

NOT FOR SALE

مسئلہ نمبر 20

دو اعداد n1 اور n2 کے درمیان بالترتیب پائے جانے والے مفرد اعداد (Prime Numbers) معلوم کرنا (جبکہ n1 بڑا ہے n2 سے)

الگورتھم: یہ الگورتھم دو اعداد n1 اور n2 کے درمیان بالترتیب پائے جانے والے مفرد اعداد معلوم کرے گا۔ (جبکہ n1 چھوٹا ہے n2 سے)۔

```
Step 1.   Read n1, n2
Step 2.   Set n=n2+1
Step 3.   Repeat Step 4 while n<n1
Step 4.   Repeat Step 5 for i=2 to  n-1.
Step 5.   If n%i=0 Then Set n=n+1 Go to Step 3
          Else
             (a)Write n
             (b) Set n=n+1
Step 6.   Exit
```

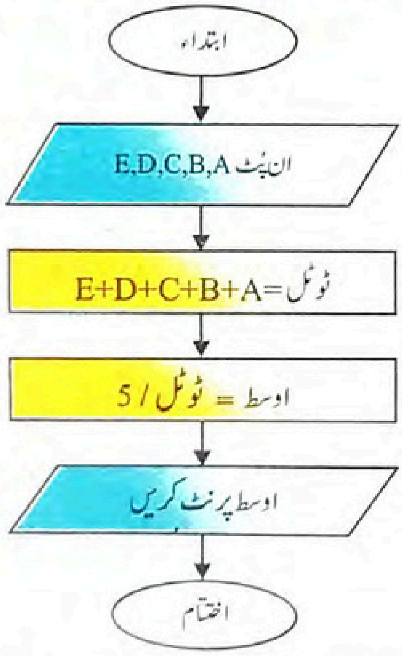
1.3 فلوچارٹ (FLOWCHART)

1.3.1 فلوچارٹ کیا ہے؟ (What is a Flowchart?)

فلوچارٹ دو الفاظ کا مجموعہ ہے جیسے فلو اور چارٹ۔ فلوچارٹ الگورتھم کا تصویری (Diagrammatic) اظہار ہے۔ فلوچارٹ مختلف علامات پر مشتمل ہوتا ہے جو کسی بھی پروگرام کے متعلق معلومات دکھاتا ہے۔ یہ تمام علامات ایک دوسرے سے تیر کے نشان کی مدد سے ملے ہوتے ہیں جو کہ اس عمل کے بہاؤ کی سمت کو ظاہر کرتے ہیں جو آخر میں کسی حل پر منتج ہوتا ہے۔ جیسا کہ شکل نمبر 1.7 میں دکھایا گیا ہے۔

1.3.2 ایک مسئلے کے حل میں فلوچارٹ کی اہمیت

(Importance of a Flowchart for Solving a Problem)



شکل نمبر 1.7 پانچ نمبروں کا اوسط معلوم کرنے کا فلوچارٹ

NOT FOR SALE

کسی مسئلے کو حل کرنے کے لیے فلو چارٹ بہت اہم ہوتے ہیں کیونکہ:

- یہ دیکھنے والے کو پروسیس سمجھنے میں مدد دیتے ہیں۔
- مطلوبہ اِن پُٹ اور آؤٹ پُٹ کو ظاہر کررہے ہوتے ہیں۔
- پروسیس کے مختلف مراحل کے درمیان تعلق کو ظاہر کررہے ہوتے ہیں۔
- فیصلہ سازی میں مدد فراہم کرتے ہیں۔
- پروسیس کے بہاؤ کو ظاہر کررہے ہوتے ہیں۔
- اہم اقدامات کو اُجاگر کرتے ہیں اور غیر اہم مراحل کو ختم کردیتے ہیں۔

1.3.3 فلو چارٹ کی ضروریات (Flowchart Requirements)

فلو چارٹ بنانے والے کو دیئے گئے مسئلے یا الگورتھم میں موجود درج ذیل ضروریات کو ہر حال میں معلوم کرلینا چاہیے۔

- تمام درکار اِن پُٹ
- پروسیسنگ کی تکمیل
- کیے جانے والے فیصلے
- مہیا کردہ آؤٹ پُٹ

تمام درکار ان پُٹ (Inputs needed)

ان پٹ ہر وہ چیز ہے جو پروسیسنگ کے دوران درکار ہوتی ہے۔ فلو چارٹ بنانے والے (Developer) کو پتہ ہونا چاہیے کہ کیا ان پٹ ہے؟ مسئلے کی تعریف (Problem Statement) سے ان پُٹ معلوم کی جاسکتی ہے۔

مثال کے طور پر اگر کسی دائرے کا رقبہ معلوم کرنا مقصود ہو تو اس مسئلے کے لیے درکار ان پُٹ دائرے کا رداس (R) ہے۔

پروسیسنگ کی تکمیل (Processing to be completed)

جیسا کہ فلو چارٹ کسی نہ کسی کلیے یا کلیات کے مجموعے کا حل پیش کرتے ہیں یا کسی متغیر (Variable) کے لیے ابتدائی قیمت متعین (Initialize) کرتے ہیں۔ اس لیے فلو چارٹ بنانے والے کو پتہ ہونا چاہیے کہ اسے مطلوبہ نتائج حاصل کرنے کے لیے کونسا کلیہ Formula یا حسابی عمل درکار ہے۔ مثال کے طور پر کسی بھی دائرے کا رقبہ معلوم کرنے کے لیے درج ذیل کلیہ استعمال ہوگا۔

Area = Pi * R * R

جبکہ Pi= 3.1415

فیصلہ سازی کرنا (Decision to be taken)

فلو چارٹ میں فیصلہ سازی تب کی جاتی ہے جب کوئی کلیہ یا اظہار خیال صحیح (True) یا غلط (False) ہو۔ مثال کے طور پر جب کسی طالب علم کا نتیجہ معلوم کرنا ہو کہ وہ فیل ہے یا پاس تو اس کے لیے طالب علم کے حاصل کردہ فیصد (PM) پر فیصلہ کیا جائے گا۔ جیسا کہ

If PM>=40 Then Result="PASS" otherwise Result ="FAIL"

NOT FOR SALE

فیصلہ سازی کرنا (Decision to be taken)

آؤٹ پُٹ ہی اصلی نتیجہ ہے۔ فلو چارٹ کے آخر میں آؤٹ پٹ ضرور دکھانا چاہیے۔ مثال کے طور پر رقبہ، نتیجہ وغیرہ۔

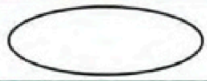
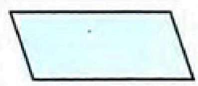
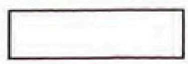
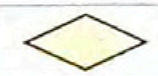
1.3.4 فلو چارٹ کی علامات (Flowchart Symbols)

الگورتھم میں موجود مختلف اقدامات کے لیے فلو چارٹ کے اندر مخصوص شکلیں استعمال ہوتی ہیں ان کو فلو چارٹ کی علامات کہا جاتا ہے۔ لائنوں اور تیر کے نشانات ان اقدامات کی ترتیب اور آپس میں تعلق کو ظاہر کرتے ہیں۔ ذیل میں فلو چارٹ کی چند اہم علامات دی گئی ہیں۔

1. بیضوی **(Oval)**: یہ فلو چارٹ کے آغاز (Start) یا اختتام (End) کو ظاہر کرتا ہے۔
2. تیر کے نشانات **(Arrow Line)**: یہ ایک مرحلے سے دوسرے مرحلے تک کنٹرول کی تبدیلی کو ظاہر کرتے ہیں۔ یہ ایک مرحلے سے دوسرے مرحلے تک پیش رفت کو بھی ظاہر کرتے ہیں۔
3. متوازی الاضلاع **(Parallelogram)**: یہ ان پُٹ اور آؤٹ کو ظاہر کرتے ہیں۔ مثال کے طور پر READ A ، INPUT N ، PRINT N ، DISPLAY B وغیرہ۔
4. مستطیل **(Rectangle)**: یہ پراسیسنگ کے مرحلے کو ظاہر کرتا ہے۔
5. ڈائمنڈ **(Diamond)**: فلو چارٹ میں جب کوئی فیصلہ کرنا ہو یا کسی سوال کا جواب دینا ہو تو ایسی صورتِ حال میں یہ علامت استعمال ہوتی ہے۔ جیسا کہ ہاں یا نہیں (Yes / No) یا صحیح یا غلط (True / False)۔ سوال کے جواب سے پتہ چلتا ہے کہ آگے کس سمت میں جانا ہے۔ مثلاً

IS A>B? , IS X = 10?

فلو چارٹ کی بنیادی علامات اور ان کے افعال

علامات	نام	کام / استعمال
	آغاز / اختتام (Start / End)	بیضوی علامات آغاز یا اختتام کو ظاہر کرتی ہے۔
	تیر کا نشان (Arrows)	ایسی لائن علامات کو جوڑنے کے لیے استعمال ہوتی ہے اور علامات کے درمیان تعلق کو ظاہر کرتی ہے۔
	ان پُٹ / آؤٹ پُٹ (Input / Output)	متوازی الاضلاع کی علامت ان پُٹ یا آؤٹ پُٹ کو ظاہر کرتی ہے۔
	پروسیس (Process)	مستطیل کی علامت پروسیس کو ظاہر کرتی ہے۔
	فیصلہ (Decision)	ڈائمنڈ کی علامت فیصلے کو ظاہر کرتی ہے۔

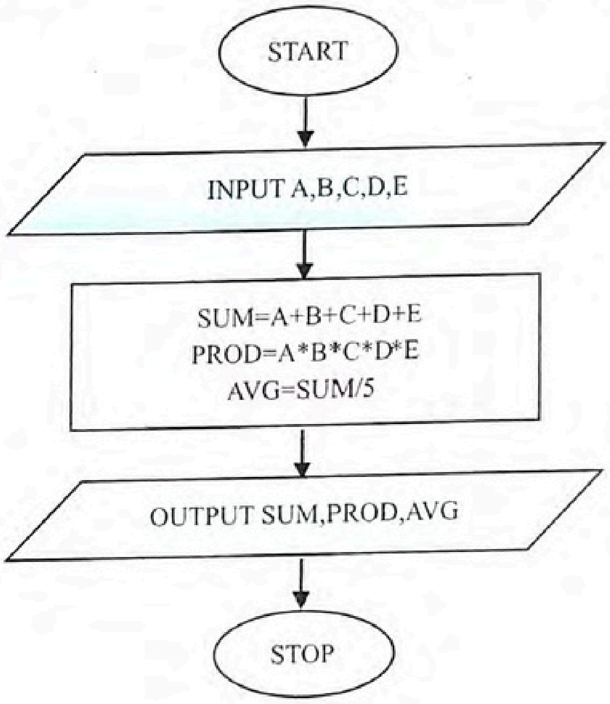
جدول نمبر 1.1 فلو چارٹ کی علامات

NOT FOR SALE

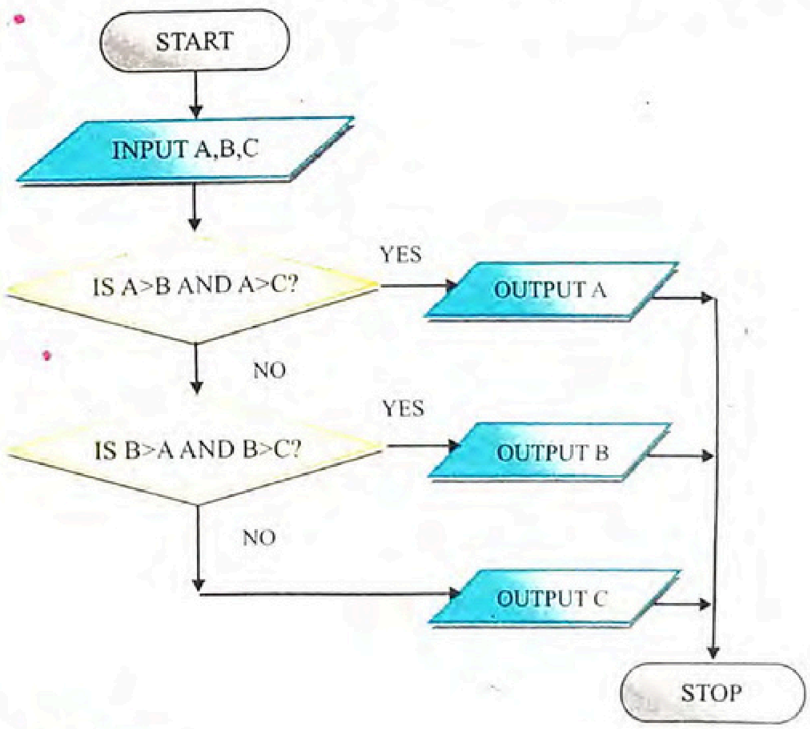
1.3.5 چند مسائل کے حل کے لیے فلو چارٹ (Flowcharts to Solve Some Problems)

اس حصے میں، پہلے حصے میں لکھے گئے چند الگورتھم کے لیے فلو چارٹ بنائے گئے ہیں تاکہ مسائل کے حل کے طریقہ کار کو با آسانی سمجھا جا سکے۔

فلو چارٹ نمبر 1: پانچ اعداد A,B,C,D,E کا حاصل جمع، حاصل ضرب اور اوسط معلوم کرنے کے لیے فلو چارٹ۔

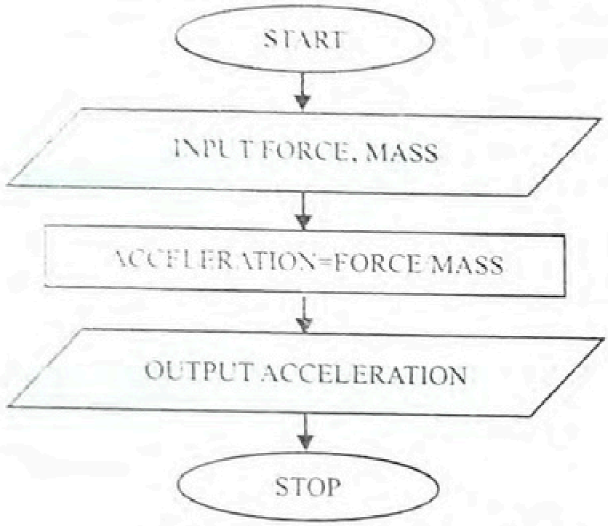


فلو چارٹ نمبر 2: تین غیر مساوی اعداد میں سے سب سے بڑا عدد معلوم کرنا۔

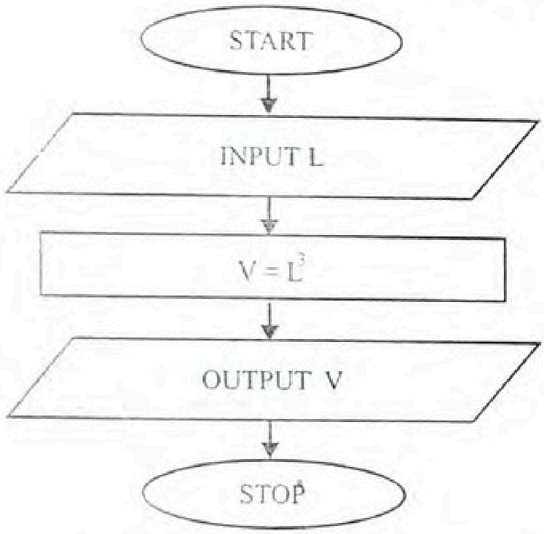


NOT FOR SALE

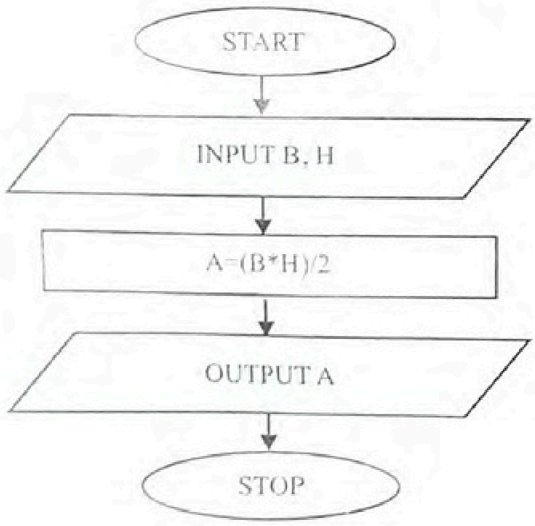
فلو چارٹ نمبر 3: کسی بھی متحرک جسم کا اسراع معلوم کرنا جبکہ اس کی کمیت اور عامل قوت معلوم ہو۔



فلو چارٹ نمبر 4: کسی بھی مکعب کے ایک ضلع کی لمبائی ان پٹ کر کے اس کا حجم معلوم کرنا۔

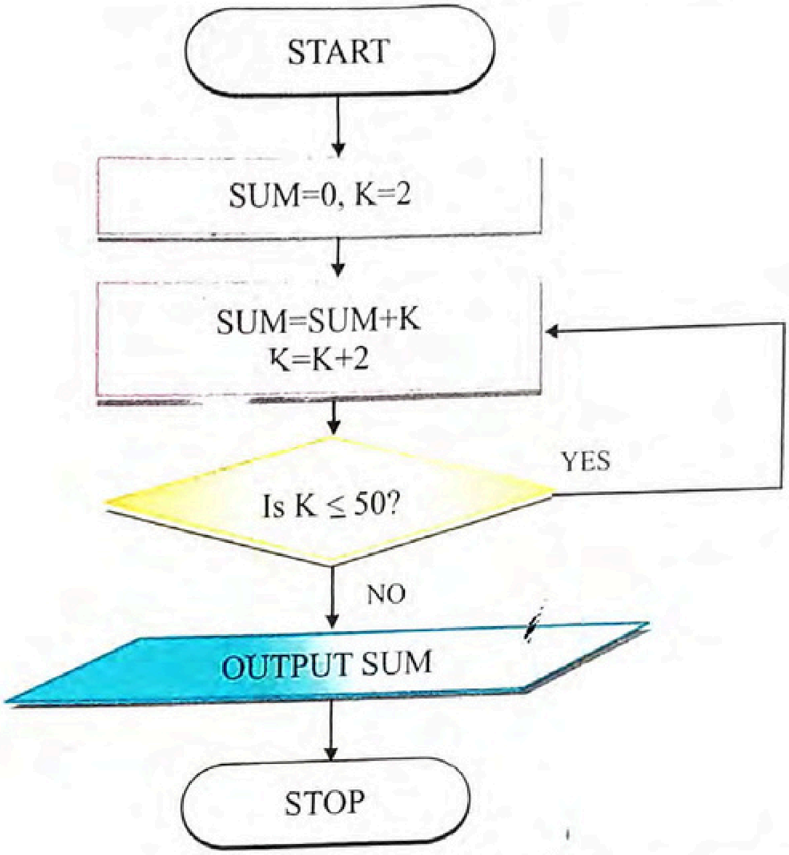


فلو چارٹ نمبر 5: کسی مثلث کا رقبہ معلوم کریں جبکہ اس کا قاعدہ اور اُونچائی کی لمبائی دی گئی ہو۔

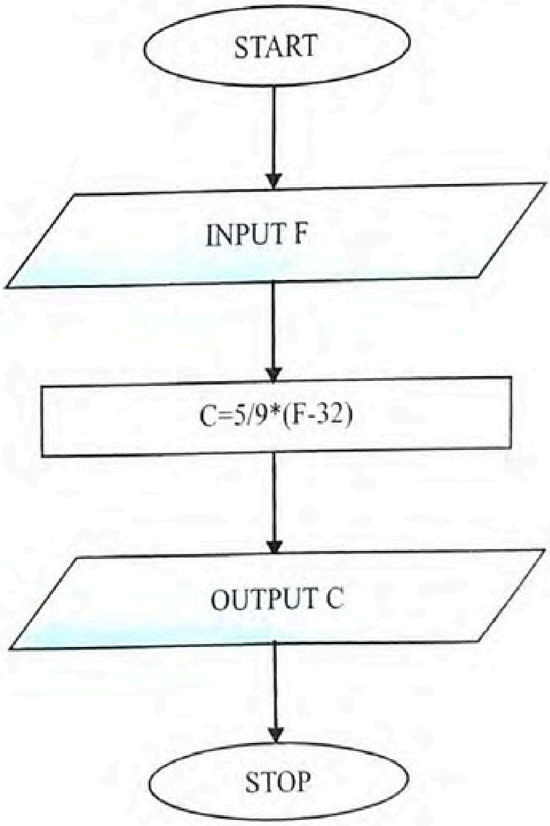


NOT FOR SALE

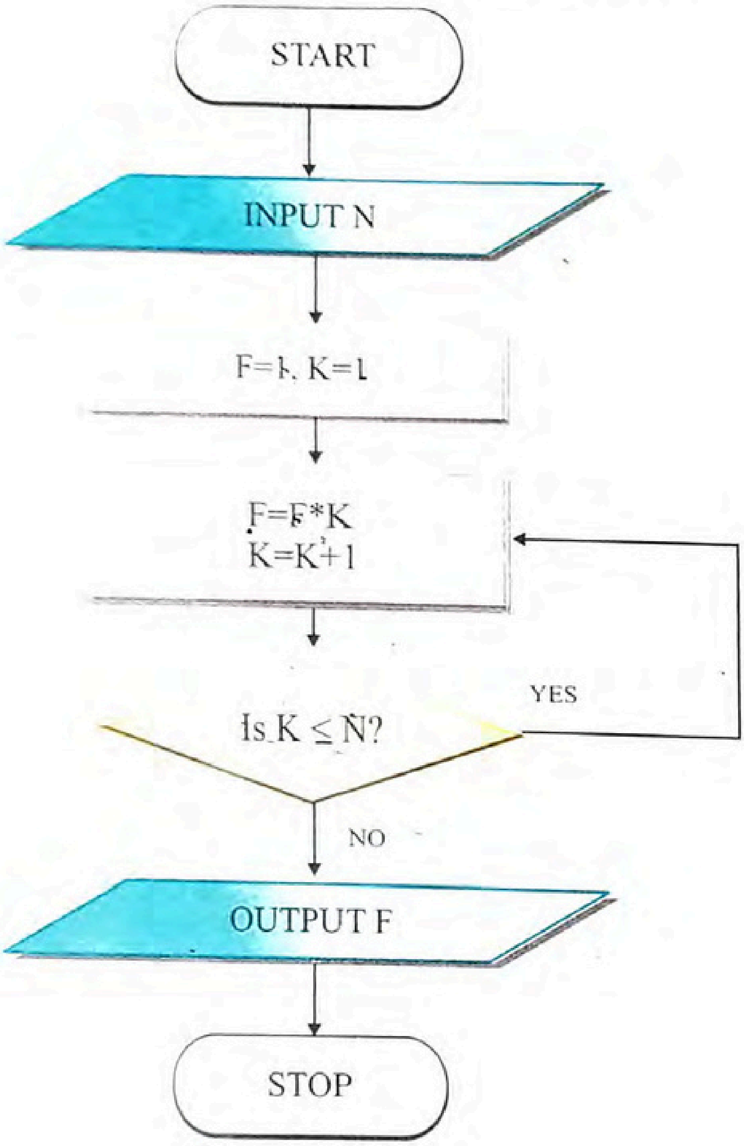
فلوچارٹ نمبر 10: 1 سے لیکر 50 تک جفت اعداد کا مجموعہ معلوم کرنا۔



فلوچارٹ نمبر 11: فارن ہائیٹ درجہ حرارت کو سینٹی گریڈ درجہ حرارت میں تبدیل کرنا۔



فلو چارٹ نمبر 12: فیکٹوریل (Factorial) معلوم کرنا۔



اہم نکات

- مسئلہ ایک رکاوٹ، دشواری یا چیلنج یا ایسی صورتحال ہے جسے دور کرنے یا حل کرنے کی ضرورت ہو۔
- بہترین حل اسے کہا جائے گا جو کم لاگت، کم وقتی، تیز اور کم پیچیدہ ہو۔
- کسی بھی مسئلے کو حل کرنے کے کئی متبادل طریقے ہو سکتے ہیں۔
- مناسب اور منطقی اقدامات کی وہ فہرست جو کسی خاص مسئلے کو حل کرنے کے طور پر لکھی جاتی ہے اسے الگورتھم کہا جاتا ہے۔
- کمپیوٹر سائنس میں کسی بھی الگورتھم کی خصوصیات کو پرکھ کر ہم اس کی کارکردگی معلوم کر سکتے ہیں۔ الگورتھم کی خصوصیات میں ہم یہ جانچنے کی کوشش کرتے ہیں کہ کوئی بھی الگورتھم کس حد تک وسائل کو استعمال کرتا ہے؟

NOT FOR SALE

- مسئلے کو حل کرنے اور فیصلہ سازی کی صلاحیت الگورتھم کی استعداد کار کو بہتر بنانے اور بہترین نتیجہ حاصل کرنے کے لئے اہم ہنر ہے۔
- فلو چارٹ الگورتھم کو تصویری یا علامتی طور پر ظاہر کرتا ہے۔
- الگورتھم میں موجود مختلف اقسام کے اقدامات کے لئے فلو چارٹ کے اندر مختلف علامات استعمال ہوتی ہیں ان علامات کو فلو چارٹ کی علامات کہتے ہیں۔
- بیضوی / اوول شکل فلو چارٹ کے آغاز یا اختتام کو ظاہر کرتی ہے۔
- تیر کے نشانات ایک مرحلے سے دوسرے مرحلے تک کنٹرول کی تبدیلی کو ہر ظاکرتے ہیں۔ یہ ایک مرحلے سے دوسرے مرحلے تک پیش رفت کو بھی ظاہر کرتے ہیں۔
- متوازی الاضلاع، اِن پُٹ / آؤٹ پُٹ کو ظاہر کرنے کے لئے استعمال ہوتا ہے۔
- مستطیل پروسیسنگ کے مرحلے کو ظاہر کرتا ہے۔
- فلو چارٹ میں جب کوئی فیصلہ کرنا ہو یا کسی سوال کا جواب دینا ہو تو ایسی صورت حال میں ڈائمنڈ کی علامت استعمال ہوتی ہے۔ جیسا کہ ہاں یا نہیں یا صحیح یا غلط۔

مشق

صحیح جواب کا انتخاب کریں۔

____________ ایک رکاوٹ، مشکل یا چیلنج، یا کوئی بھی ایسی صورت حال ہے جس سے نکلنا مقصود ہو۔

الگورتھم | مسئلہ | (ج) پیچیدگی | (د) ان میں سے کوئی بھی نہیں

____________ الگورتھم کی ان خصوصیات کو بیان کرتی ہے جن سے ہم اس الگورتھم کے بارے میں یہ جاننے کی کوشش کرتے ہیں کہ الگورتھم کس حد تک وسائل کو استعمال کرتا ہے؟

الگورتھم کی کارکردگی | (ب) بہترین حل
الگورتھم کی پیچیدگی | (د) ان میں سے کوئی بھی نہیں

مناسب اور منطقی اقدامات کی وہ فہرست جو کسی مخصوص مسئلے کے حل کے طور پر لکھی جاتی ہے ____________ کہلاتا ہے۔

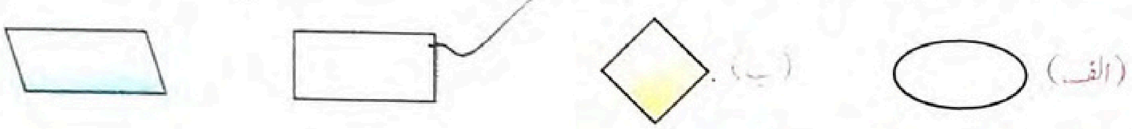
الگورتھم | (ب) فلو چارٹ
الگورتھم کی پیچیدگی | (د) یہ تمام

NOT FOR SALE

iv. الگورتھم حسابی مرحلوں/اقدامات کا ایک ترتیب وار سلسلہ ہے جو ان پٹ کو ____________ میں تبدیل کرتا ہے۔

(الف) ڈیٹا سٹرکچر (ب) الگورتھم (ج) آؤٹ پٹ (د) یہ تمام

v. ان میں سے کونسی علامت پروسیسنگ کو ظاہر کرتی ہے؟



سوال نمبر2۔ درج ذیل کے مختصر جوابات تحریر کریں۔

i. "مسئلے کا تجزیہ" کی تعریف کریں۔

ii. مسئلے کے حل کی منصوبہ بندی کیسے کی جاتی ہے؟

iii. "مسئلے کا صریح حل" کی تعریف کریں۔

iv. مسئلے کے حل کے لیے کوئی سے تین طریقوں کی تعریف کریں۔

v. مسئلے کے بہترین حل کے چناؤ کے لیے کوئی سے تین عناصر کے نام لکھیں۔

سوال نمبر3۔ الگورتھم کی تعریف کریں اور مسئلے کے حل [illegible]

سوال نمبر4۔ الگورتھم کی کارکردگی جانچنے کے لیے درج ذیل پیمانوں کی وضاحت کریں۔

i. درکار ان پٹ ii. پروسیسنگ کو مکمل کرنا

iii. فیصلہ کرنا iv. حاصل شدہ آؤٹ پٹ

سوال نمبر5۔ فلوچارٹ کیا ہے؟ فلوچارٹ کی علامات کو [illegible]

سوال نمبر6۔ کسی بھی دیئے گئے عدد کا فیکٹوریل (Factorial) معلوم کرنے کے لیے الگورتھم لکھیں اور پھر اس کے لیے فلو چارٹ بھی بنائیں۔

سوال نمبر7۔ دو درجی مساوات (Quadratic Equations) [illegible]

سوال نمبر8۔ کسی بھی طالب علم کے حاصل کردہ مضامین کا اوسط معلوم کرنے کے لیے الگورتھم لکھیں اور فلو چارٹ بنائیں۔

NOT FOR SALE

"C" پروگرامنگ

اس باب کی تکمیل کے بعد طلبہ اس قابل ہو جائیں گے کہ وہ:

- پروگرام کی ترکیب (Syntax) اور مفہوم (Semantic) واضح کر سکیں۔
- کمپیوٹر کی زبانوں کی مختلف اقسام کو تفصیل سے بیان کر سکیں جیسے ہائی لیول لینگویج اور لو لیول لینگویج۔
- ہائی لیول لینگویج کی خصوصیات کو تفصیلاً بیان کر سکیں۔
- مختلف طرح کی ہائی لیول لینگویج کو جان سکیں۔
- کمپائلر اور انٹرپریٹر کا فرق واضح کر سکیں۔
- انٹیگریٹڈ ڈویلپمنٹ انوائرمنٹ (IDE) کے تصور کو بیان کر سکیں۔
- C لینگویج کے مختلف اجزاء (ایڈیٹر، کمپائلر، لنکر، لوڈر اور ڈی بگر) کو تفصیلاً بیان کر سکیں۔
- ہیڈر فائلز کی تعریف کر سکیں۔
- ریزروڈ ورڈز (Reserved Words) کی پہچان کر سکیں۔
- C پروگرام کے بنیادی ڈھانچے (Structure) کو بیان کر سکیں۔
- Comments کے گرائمر اور مقصد کو سمجھ سکیں۔
- متغیر (Variable) اور مستقل (Constant) میں فرق کر سکیں۔
- Variable کے نام سے متعلق اصولوں کو تفصیلاً بیان کر سکیں۔
- C پروگرام میں استعمال ہونے والی ڈیٹا ٹائپس (Data Types) کو جان سکیں۔
- ٹائپ کاسٹنگ (Typecasting) کا استعمال کر سکیں۔
- *const* کی تعریف کر سکیں۔
- Variable اور Constant بنانے اور ابتدائی مقدار مقرر کرنے کے عمل کو تفصیلاً بیان کر سکیں۔

NOT FOR SALE

2.1 پروگرامنگ کا ایک تعارف (Introduction to Programming)

کمپیوٹر ایک ایسا آلہ ہے جو دی گئی ہدایات کے مطابق عمل کرتا ہے۔ واضح ہدایات کا ایسا مجموعہ جو کمپیوٹر کو فراہم کیا جائے کمپیوٹر پروگرام کہلاتا ہے۔ کمپیوٹر پروگرامنگ ایک ایسا عمل ہے جس میں ہم کسی بھی مسئلے کو حل کرنے کے لیے ایک کمپیوٹر پروگرام بناتے ہیں۔ دنیا میں کمپیوٹر کے لیے بہت ساری زبانیں استعمال ہوتی ہیں جیسے C ، C C++اور Java وغیرہ۔ ان میں سے ہر ایک کی اپنی کچھ نہ کچھ خصوصیات ہیں۔ کمپیوٹر پروگرام کمپیوٹر میں ہونے والے مختلف عوامل کو کنٹرول کرتا ہے اور یہ کسی نہ کسی پروگرامنگ لینگویج میں لکھا جاتا ہے تاکہ کوئی مخصوص کام سرانجام دیا جاسکے۔

2.1.1 کمپیوٹر پروگرام (Computer Program)

کمپیوٹر پروگرام ہدایات (Instructions) کا ایسا مجموعہ ہے جو کمپیوٹر کو بتا سکے کہ کیا کرنا اور کیسے کرنا ہے۔ کوئی بھی کمپیوٹر، کمپیوٹر پروگرام میں دی گئی ہدایات کے مطابق عمل کرتا ہے۔ کمپیوٹر پروگرام کسی بھی پروگرامنگ لینگویج میں لکھے جاتے ہیں۔ جو فرد کمپیوٹر پروگرام بناتا ہے اسے پروگرامر کہا جاتا ہے۔ پروگرامر پروگرام بناکر کمپیوٹر کو ہدایات دیتا ہے کہ کس طرح ڈیٹا کو پروسیس کرے تاکہ وہ انفارمیشن میں تبدیل ہو جائے۔ پروگرامر مختلف پروگرامنگ لینگویج یا پروگرامنگ آلات کو استعمال کرکے پروگرام بناتا ہے۔ ذیل میں دی گئی دو اصطلاحات کمپیوٹر پروگرام سے متعلق ہیں۔

پروگرام کی ترکیب (Program Syntax)

ترکیب (Syntax) دراصل کسی بھی پروگرامنگ لینگویج کے وہ اصول ہیں جن کے تحت کوئی بھی پروگرام لکھا جاتا ہے۔ یہ وہ طریقہ کار ہے جس سے ہم پروگرام کے اندر صحیح ہدایات (Statements) لکھ سکتے ہیں۔ کسی بھی قدرتی زبان کی گرائمر اور پروگرامنگ لینگویج کی ترکیب دونوں ایک ہی چیز ہیں۔ جیسا کہ C لینگویج میں آؤٹ پُٹ کو دکھانے کے لیے ()printf اسٹیٹمنٹ استعمال ہوتی ہے، جس کی ترکیب یہ ہے :

printf("escape sequence/format specifier/string", variable/expression/constant);

پروگرام کا مفہوم (Program Semantic)

سیمنٹک (Semantic) پروگرامنگ لینگویج کے معانی کا علم ہے جو ایک لفظ یا پورے جملے پر لاگو ہو سکتا ہے۔ یہ اُن اقدامات کی مرحلہ وار ترتیب ہے جو کمپیوٹر پروگرام کو چلاتے ہوئے ملحوظ رکھی جاتی ہے۔ مثال کے طور پر ذیل میں دی گئی اسٹیٹمنٹ C کے پروگرام میں دو ویری ایبل x اور y کا حاصل ضرب product کے ویری ایبل میں سٹور کرے گی۔

*product = x*y;*

2.1.2 پروگرامنگ لینگویجز (Programming Languages)

الفاظ، علامات اور کوڈ کا مجموعہ جن کی مدد سے آپ کوئی بھی پروگرام لکھ سکتے ہیں انہیں پروگرامنگ لینگویج کہا جاتا ہے۔ مختلف

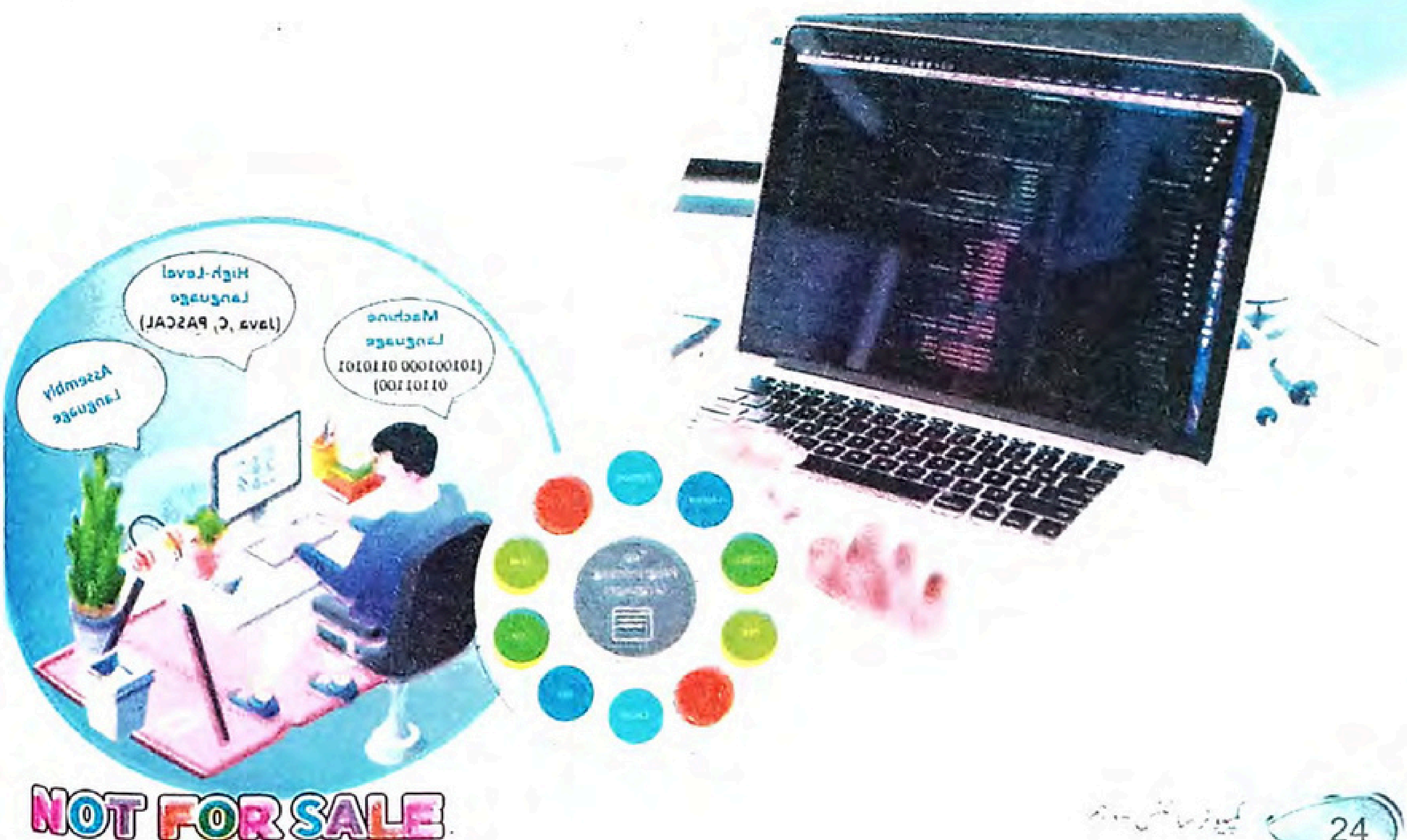
"NOT FOR SALE"

2

"C" پروگرامنگ

اس باب کی تکمیل کے بعد طلبہ اس قابل ہو جائیں گے کہ وہ:

- پروگرام کی ترکیب (Syntax) اور مفہوم (Semantic) واضح کر سکیں۔
- کمپیوٹر کی زبانوں کی مختلف اقسام کو تفصیل سے بیان کر سکیں جیسے ہائی لیول لینگویج اور لو لیول لینگویج۔
- ہائی لیول لینگویج کی خصوصیات کو تفصیلاً بیان کر سکیں۔
- مختلف طرح کی ہائی لیول لینگویج کو جان سکیں۔
- کمپائلر اور انٹرپریٹر کا فرق واضح کر سکیں۔
- انٹیگریٹڈ ڈویلپمنٹ انوائرمنٹ (IDE) کے تصور کو بیان کر سکیں۔
- C لینگویج کے مختلف اجزاء (ایڈیٹر، کمپائلر، لنکر، لوڈر اور ڈی بگر) کو تفصیلاً بیان کر سکیں۔
- ہیڈر فائلز کی تعریف کر سکیں۔
- ریزروڈ ورڈز (Reserved Words) کی پہچان کر سکیں۔
- C پروگرام کے بنیادی ڈھانچے (Structure) کو بیان کر سکیں۔
- Comments کے گرائمر اور مقصد کو سمجھ سکیں۔
- متغیر (Variable) اور مستقل (Constant) میں فرق کر سکیں۔
- Variable کے نام سے متعلق اصولوں کو تفصیلاً بیان کر سکیں۔
- C پروگرام میں استعمال ہونے والی ڈیٹا ٹائپس (Data Types) کو جان سکیں۔
- ٹائپ کاسٹنگ (Typecasting) کا استعمال کر سکیں۔
- *const* کی تعریف کر سکیں۔
- Variable اور Constant بنانے اور ابتدائی مقدار مقرر کرنے کے عمل کو تفصیلاً بیان کر سکیں۔



NOT FOR SALE

2.1 پروگرامنگ کا ایک تعارف (Introduction to Programming)

کمپیوٹر ایک ایسا آلہ ہے جو دی گئی ہدایات کے مطابق عمل کرتا ہے۔ واضح ہدایات کا ایسا مجموعہ جو کمپیوٹر کو فراہم کیا جائے کمپیوٹر پروگرام کہلاتا ہے۔ کمپیوٹر پروگرامنگ ایک ایسا عمل ہے جس میں ہم کسی بھی مسئلے کو حل کرنے کے لیے ایک کمپیوٹر پروگرام بناتے ہیں۔ دنیا میں کمپیوٹر کے لیے بہت ساری زبانیں استعمال ہوتی ہیں جیسے ++C ، C ،C اور Java وغیرہ۔ ان میں سے ہر ایک کی اپنی کچھ نہ کچھ خصوصیات ہیں۔ کمپیوٹر پروگرام کمپیوٹر میں ہونے والے مختلف عوامل کو کنٹرول کرتا ہے اور یہ کسی نہ کسی پروگرامنگ لینگوئج میں لکھا جاتا ہے تاکہ کوئی مخصوص کام سرانجام دیا جاسکے۔

2.1.1 کمپیوٹر پروگرام (Computer Program)

کمپیوٹر پروگرام ہدایات (Instructions) کا ایسا مجموعہ ہے جو کمپیوٹر کو بتاسکے کہ کیا کرنا اور کیسے کرنا ہے۔ کوئی بھی کمپیوٹر، کمپیوٹر پروگرام میں دی گئی ہدایات کے مطابق عمل کرتا ہے۔ کمپیوٹر پروگرام کسی بھی پروگرامنگ لینگوئج میں لکھے جاتے ہیں۔ جو فرد کمپیوٹر پروگرام بناتا ہے اسے پروگرامر کہا جاتا ہے۔ پروگرامر پروگرام بناکر کمپیوٹر کو ہدایات دیتا ہے کہ کس طرح ڈیٹا کو پروسیس کرے تاکہ وہ انفارمیشن میں تبدیل ہو جائے۔ پروگرامر مختلف پروگرامنگ لینگوئج یا پروگرامنگ آلات کو استعمال کرکے پروگرام بناتا ہے۔ ذیل میں دی گئی دو اصطلاحات کمپیوٹر پروگرام سے متعلق ہیں۔

پروگرام کی ترکیب (Program Syntax)

ترکیب (Syntax) دراصل کسی بھی پروگرامنگ لینگوئج کے وہ اصول ہیں جن کے تحت کوئی بھی پروگرام لکھا جاتا ہے۔ یہ وہ طریقہ کار ہے جس سے ہم پروگرام کے اندر صحیح ہدایات (Statements) لکھ سکتے ہیں۔ کسی بھی قدرتی زبان کی گرائمر اور پروگرامنگ لینگوئج کی ترکیب دونوں ایک ہی چیز ہیں۔ جیسا کہ C لینگوئج میں آؤٹ پٹ کو دکھانے کے لیے ()printf اسٹیٹمنٹ استعمال ہوتی ہے، جس کی ترکیب یہ ہے :

printf("escape sequence/format specifier/string", variable/expression/constant);

پروگرام کا مفہوم (Program Semantic)

سیمنٹک (Semantic) پروگرامنگ لینگوئج کے معانی کا علم ہے جو ایک لفظ یا پورے جملے پر لاگو ہو سکتا ہے۔ یہ اُن اقدامات کی مرحلہ وار ترتیب ہے جو کمپیوٹر پروگرام کو چلاتے ہوئے ملحوظ رکھی جاتی ہے۔ مثال کے طور پر ذیل میں دی گئی اسٹیٹمنٹ C کے پروگرام میں دو ویری ایبل x اور y کا حاصل ضرب product کے ویری ایبل میں سٹور کرے گی۔

*product = x*y;*

2.1.2 پروگرامنگ لینگویجز (Programming Languages)

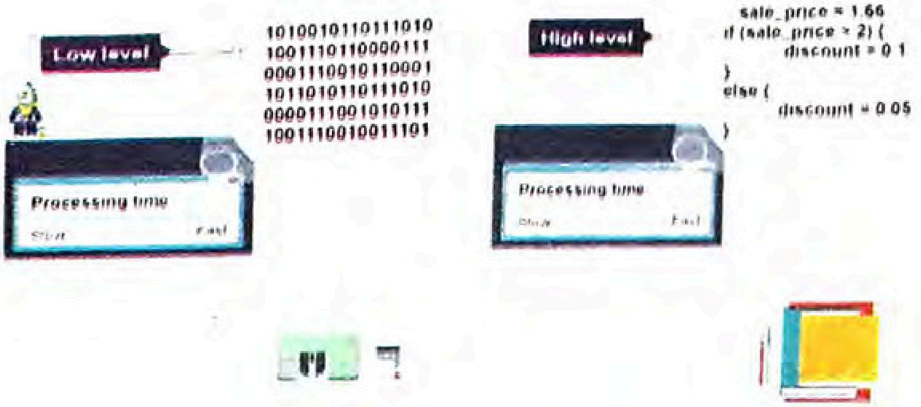
الفاظ، علامات اور کوڈ کا مجموعہ جن کی مدد سے آپ کوئی بھی پروگرام لکھ سکتے ہیں انہیں پروگرامنگ لینگوئج کہا جاتا ہے۔ مختلف

NOT FOR SALE

طرح کے کام کمپیوٹر پر انجام دینے کے لیے مختلف اقسام کی پروگرامنگ لینگویج استعمال ہوتی ہیں۔ یہ لینگویجز کاروباری، سائنسی اور عمومی مسائل کے حل کے لیے استعمال ہوتی ہیں۔

پروگرامنگ لینگویجز کی دو اقسام میں درجہ بندی کی جاسکتی ہے۔

- لولیول لینگویجز (Low Level Languages)
- ہائی لیول لینگویجز (High Level Languages)



شکل نمبر 2.1 لولیول اور ہائی لیول پروگرامنگ لینگویجز

لولیول لینگویجز (Low Level Languages)

کمپیوٹر ہائی لیول لینگویجز یا انگریزی میں دی جانے والی ہدایات کو براہ راست نہیں سمجھ سکتا۔ یہ صرف مشین لینگویج جیسے بائنری لینگویج، میں دی گئی ہدایات کو سمجھ اور چلا سکتا ہے۔ یہ لینگویجز کمپیوٹر ہارڈ ویئر کے بہت قریب ہیں اِس لیے کمپیوٹر اِن لینگویجز کو با آسانی سمجھ سکتا ہے۔ ان لینگویج میں پروگرامنگ کرنے کے لیے کمپیوٹر کے ہارڈ ویئر کی ساخت سے متعلق بہت گہرا مطالعہ ہونا ضروری ہے۔

لولیول لینگویجز کی دو اقسام ہیں۔

1. مشین لینگویج (Machine Language)

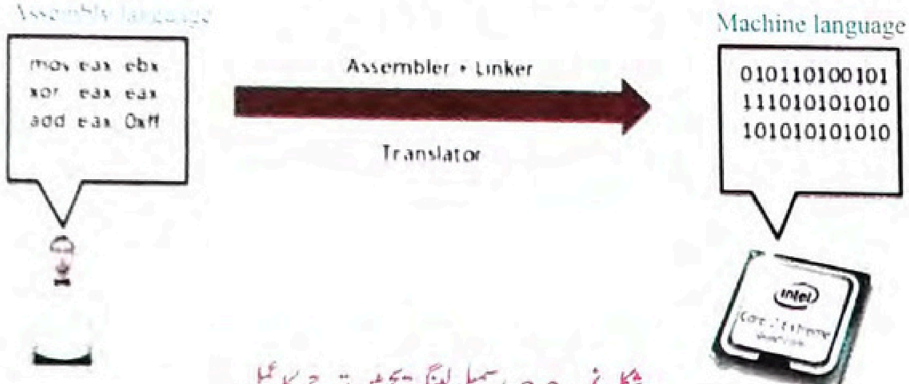
زبان کی وہ قسم جس میں ہم ہدایات کو بائنری شکل میں لکھتے ہیں مشینی زبان کہلاتی ہے۔ یہ کمپیوٹر کی وہ واحد زبان ہے جسے کمپیوٹر براہ راست سمجھ جاتا ہے۔ یہ کمپیوٹر کی بنیادی یا اساسی زبان ہے۔ اس زبان میں لکھے گئے پروگرام کمپیوٹر پر بہت تیزی کے ساتھ چلتے ہیں۔ کمپیوٹر کے اندر مشین لینگویج کو بائنری اعداد (bits) یعنی 0 اور 1 کے تسلسل سے ظاہر کیا جاتا ہے۔ 0 کی علامت چارج کی غیر موجودگی اور 1 چارج کی موجودگی کو ظاہر کرتا ہے۔ انسان کے لیے مشینی زبان سمجھنا زیادہ مشکل ہے۔ اس زبان میں پروگرام لکھنا یا لکھے ہوئے پروگرام میں ترمیم کرنا بہت دیر طلب کام ہے۔ مشینی زبان کو پہلی نسل کی زبان (First Generation Language) بھی کہا جاتا ہے۔

2. اسمبلی لینگویج (Assembly Language)

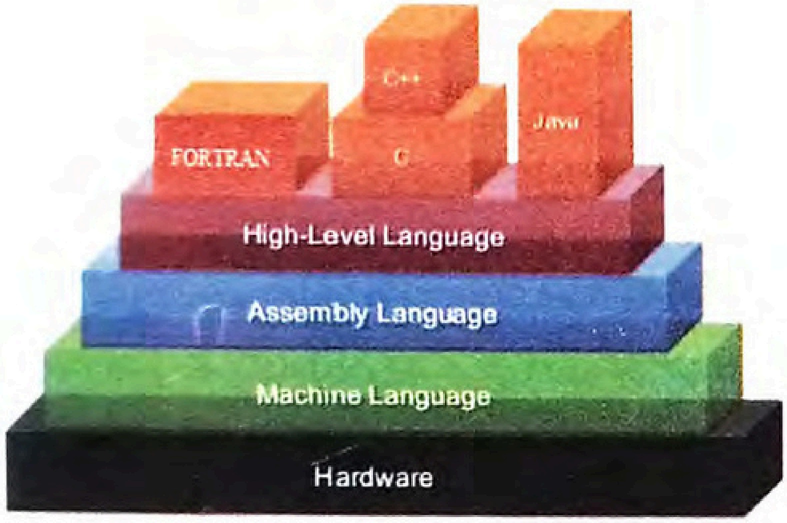
اسمبلی لینگویج بھی لولیول لینگویج ہے۔ یہ مشین لینگویج سے ایک درجہ اوپر ہے۔ اس لینگویج میں بائنری کوڈ کی بجائے علامات کو

NOT FOR SALE

استعمال کیا جاتا ہے۔ ان علامات کو (نمونکس) mnemonics کہا جاتا ہے۔ مثال کے طور پر دو اعداد کی تفریق کے لیے SUB اسٹیٹمنٹ استعمال کی جاتی ہے۔ اسے علاماتی زبان (Symbolic Language) بھی کہا جاتا ہے۔ مشینی زبان کے مقابلے میں اسمبلی لینگویج میں پروگراموں کو لکھنا یا ان میں ترمیم کرنا زیادہ آسان ہے۔ اسمبلی لینگویج میں لکھے کوڈ کو پہلے مشین لینگویج میں ترجمہ کرنے کے لیے اسمبلر کو استعمال کیا جاتا ہے اور پھر اسے کمپیوٹر پر چلایا جاتا ہے۔ اسمبلی لینگویج زیادہ تر سسٹم سافٹ ویئرز بنانے کے لیے استعمال کی جاتی ہے۔ اس کو دوسری نسل کی زبان (Second Generation Language) بھی کہا جاتا ہے۔



شکل نمبر 2.2 اسمبلی لینگویج میں ترجمے کا عمل



شکل نمبر 2.3 مختلف پروگرامنگ لینگویجز

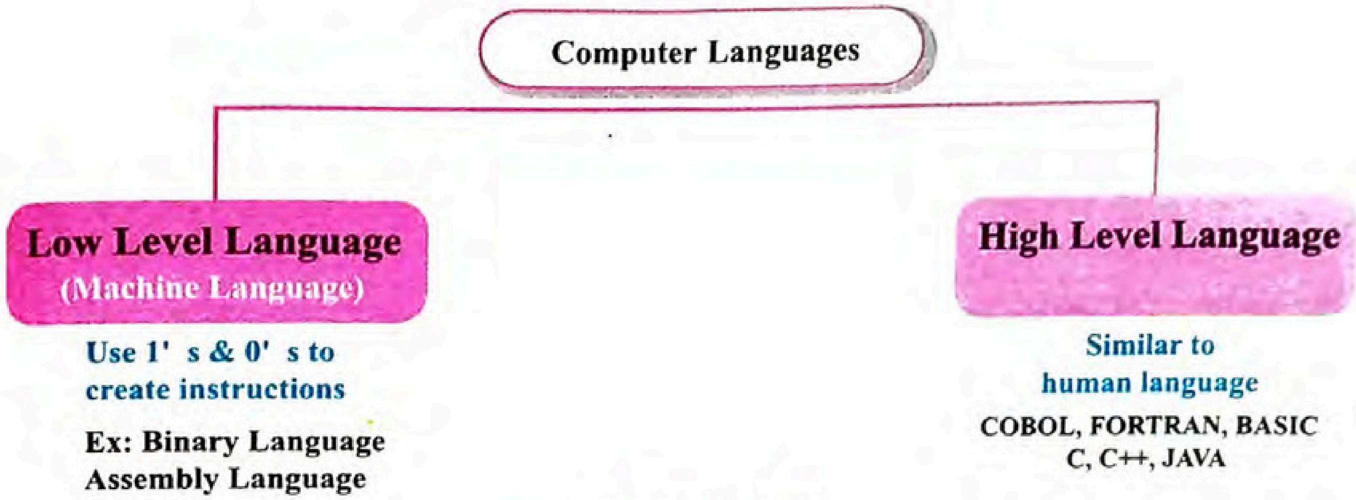
ہائی لیول لینگویجز (High Level Languages)

ہائی لیول لینگویجز انگریزی سے ملتے جلتے الفاظ پر مشتمل ہوتی ہے۔ ہائی لیول لینگویج بنانے کا مقصد یہ تھا کہ لوگ اپنی زبان یعنی انگریزی میں آسانی سے پروگرام لکھ سکیں۔

ہائی لیول لینگویج بنیادی طور پر نمونکس کوڈ کے بجائے انگریزی کے الفاظ اور حسابی علامات اِستعمال کرتی ہے۔ دنیا میں بہت ساری ہائی لیول لینگویجز ہیں جن میں پروگرام لکھنے کے قوانین اور گرائمر ایک دوسرے سے مختلف ہیں۔ ہائی لیول لینگویج میں لکھے گئے پروگرام کو کمپیوٹر پر چلانے سے پہلے مشین کوڈ میں ترجمہ کیا جاتا ہے۔ ہر ہائی لیول لینگویج کا ترجمہ کرنے کے لیے ایک مخصوص مترجم (Translator)

NOT FOR SALE

پروگرام ہوتا ہے۔ہائی لیول لینگوئج کی چند مثالیں C, Java, Visual Basic اور Pascal ہیں۔ہائی لیول لینگوئج کو درج ذیل اقسام میں تقسیم کیا جاسکتا ہے۔



شکل نمبر 2.4 کمپیوٹر لینگویجز

- پروسیجرل لینگوئج (Procedural Language)
- سٹرکچرڈ لینگوئج (Structured Language)
- آبجیکٹ اورینٹڈ پروگرامنگ لینگوئج (Object-Oriented Programming Language)

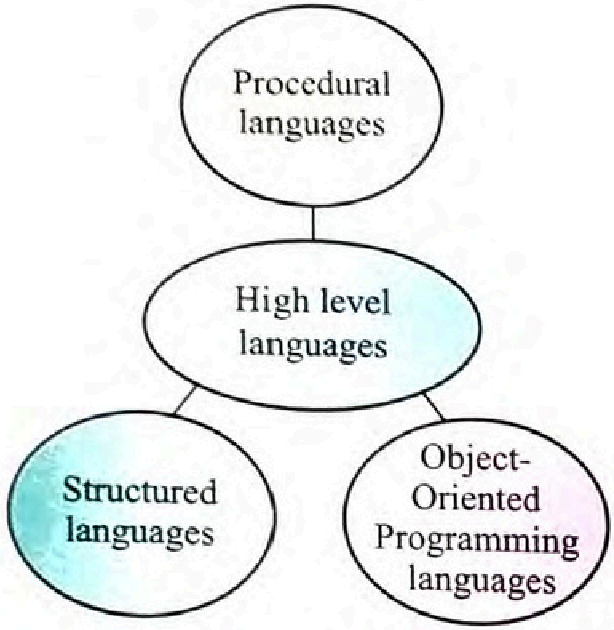
پروسیجرل لینگویجز (Procedural Languages)

پروسیجرل لینگوئج کو تیسری نسل کی زبان (Third Generation Language) بھی کہا جاتا ہے۔اس زبان میں پروگرام کو پروسیجرز یا ماڈیولز کی شکل میں تیار کیا جاتا ہے۔پروسیجر ہدایات کی وہ ترتیب ہے جس کا کوئی نہ کوئی منفرد نام ہو۔پروسیجرز کو چلانے کے لیے اس کے نام سے پکارا جاتا ہے۔پروسیجر ایک دفعہ لکھا جاتا ہے لیکن اسے پروگرام میں جتنی دفعہ درکار ہو استعمال کیا جاسکتا ہے اور اس کا کوڈ بار بار لکھنے کی ضرورت نہیں پڑتی۔

پروسیجرل لینگوئج کی چند مثالیں C، Pascal، FORTRAN اور BASIC ہیں۔

سٹرکچرڈ لینگویجز (Structured languages)

ایسی پروگرامنگ لینگوئج جس میں پروگرام کو چھوٹے چھوٹے حصوں، بلاک یا پروسیجرز میں تقسیم کیا جاسکے اسے سٹرکچرڈ پروگرامنگ لینگوئج کہا جاتا ہے۔اس طرح کے پروگرام کی ترتیب اوپر سے نیچے کی طرف ہوتی ہے۔اس طرح کی زبانوں کی عام مثالیں ALGOL، PL/1، Pascal اور ADA ہیں۔



شکل نمبر 2.5 ہائی لیول لنگویجز

آبجیکٹ اورینٹڈ پروگرامنگ لینگوئج (Object Oriented Programming Language)

آبجیکٹ اورینٹڈ پروگرامنگ میں، پروگرام آبجیکٹس کے ایک سیٹ کا استعمال کر کے تیار کیا جاتا ہے۔ ایک آبجیکٹ پروگرام ایک ایسا جزو ہے جس سے ماڈلز یو کا ایک سیٹ اور ڈیٹا سٹرکچر منسلک ہوتے ہیں۔ یہ پروگرام کو ڈیزائن کرنے کے لئے ایک جدید اور آسان تکنیک ہے۔ ایک بار جب کسی بھی پروگرام کے لئے کوئی آبجیکٹ ڈیزائن ہو جائے تو پھر اسے کسی بھی دوسرے پروگرام میں دوبارہ استعمال کیا جاسکتا ہے۔ کچھ مقبول اور عام استعمال ہونے والی آبجیکٹ اورینٹڈ پروگرامنگ (OOP) لینگویجز کی مثالیں ++ C اور JAVA ہیں۔

2.1.3 ہائی لیول پروگرامنگ لینگویجز کی خصوصیات (Characteristic of High Level Programming Languages)

ہائی لیول پروگرامنگ لینگویجز کی چند خصوصیات درج ذیل ہیں۔

سیکھنے میں آسان (Easy to Learn)

لو لیول پروگرامنگ کے مقابلے میں ہائی لیول پروگرامنگ لینگویجز سیکھنے میں بہت آسان ہوتی ہیں۔ پروگرام کے لیے لکھی گئی اسٹیمنٹس بالکل انگریزی زبان جیسے جملوں کی طرح ہی ہوتی ہیں۔

سمجھنے میں آسان (Easy to Understand)

ہائی لیول پروگرامنگ لینگوئج میں ایک پروگرامر کا لکھا ہوا کوئی بھی پروگرام دوسرا پروگرامر آسانی سے سمجھ سکتا ہے۔ کیونکہ پروگرام میں لکھی گئی ہدایات انگریزی زبان کی طرح ہی ہوتی ہیں۔

NOT FOR SALE

پروگرام لکھنے میں آسانی (Easy to Write Program)

ہائی لیول پروگرامنگ لینگویج میں، ایک نیا پروگرام آسانی سے بہت ہی کم وقت میں لکھا جاسکتا ہے۔

بڑے اور پیچیدہ سافٹ ویئر کچھ دنوں یا مہینوں میں تیار کیے جاسکتے ہیں۔

غلطیوں کا پتہ لگانے اور انہیں دور کرنے میں آسانی (Easy to Detect & Remove Errors)

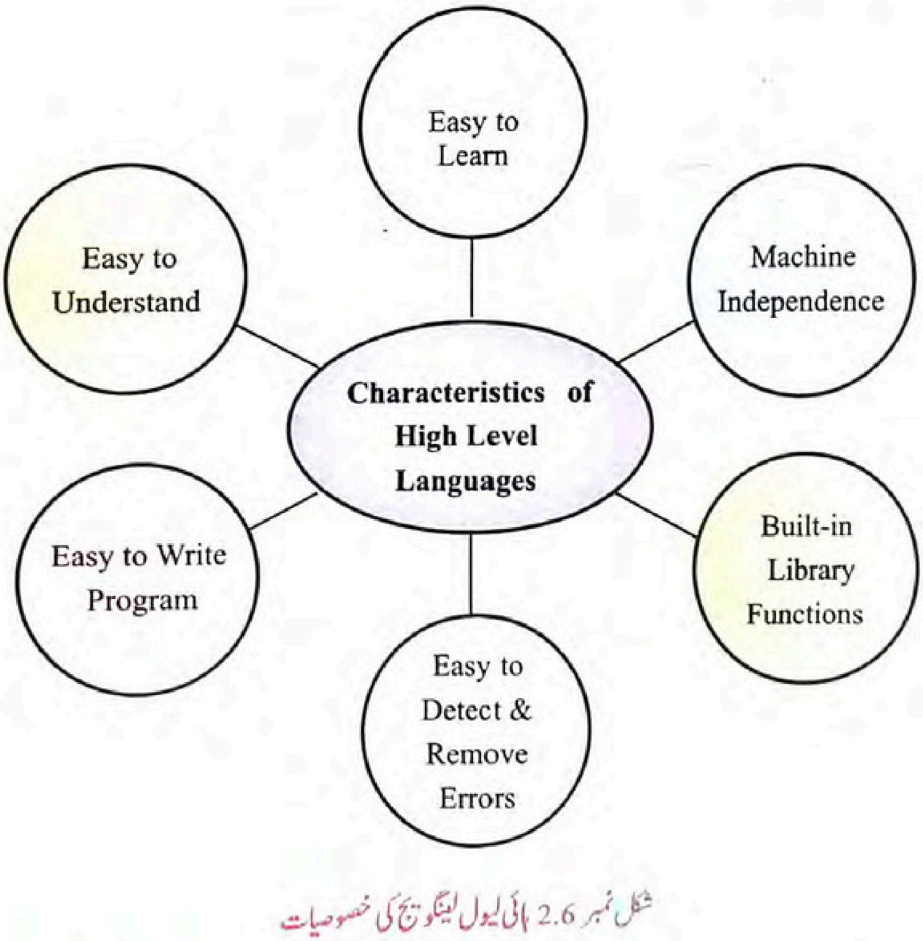
پروگرام میں موجود غلطیوں کو آسانی سے معلوم کیا جاسکتا ہے اور انہیں دور بھی کیا جاسکتا ہے۔ زیادہ تر غلطیاں نئے پروگرام کو کمپائل کرنے کے دوران ظاہر ہو جاتی ہیں۔

بنے بنائے لائبریری فنکشنز (Built-in Library Functions)

ہر ہائی لیول پروگرامنگ لینگویج میں بڑی تعداد میں بلٹ ان فنکشنز یا طریقہ کار مہیا ہوتے ہیں جو نئے پروگراموں کی ڈیزائننگ کے دوران مخصوص کاموں کو انجام دینے کے لئے استعمال ہوتے ہیں۔ اس طرح سے، پروگرامر کا بہت سا وقت بچ جاتا ہے۔

مشین پر غیر انحصاری (Machine Independence)

ہائی لیول لینگویج میں لکھا ہوا پروگرام کسی خاص مشین پر منحصر نہیں ہوتا۔ اس کا مطلب یہ ہے کہ ایک قسم کے کمپیوٹر پر لکھا ہوا ایک پروگرام دوسری قسم کے کمپیوٹر پر بآسانی چلایا جاسکتا ہے۔



شکل نمبر 2.6 ہائی لیول لینگویج کی خصوصیات

2.1.4 چند مقبول ہائی لیول پروگرامنگ لینگویجز (Some Popular High Level Programming Languages)

C لینگویج (C Language)

ڈینس رچی، بانی C لینگویج

C ایک عمومی مقاصد کے لیے استعمال ہونے والی کمپیوٹر پروگرامنگ لینگویج ہے جو 1972ء میں امریکہ کے اے ٹی اینڈ ٹی بیل لیبارٹریز میں ڈینس رچی (Denis Richie) نے تیار کی۔

C ایک سادہ اور سٹرکچرڈ اور ڈیزائنڈ پروسیجرل پروگرامنگ لینگویج ہے۔ یہ مختلف پلیٹ فارم پر آسانی سے منتقل ہو سکتی ہے۔ اور اسی وجہ سے صنعت اور پیشہ ورانہ پروگرامنگ میں وسیع پیمانے پر استعمال ہوتی ہے۔ یونیکس UNIX پر آپریٹنگ سسٹم C میں تیار کیا گیا ہے۔

C ++ لینگویج (C++ Language)

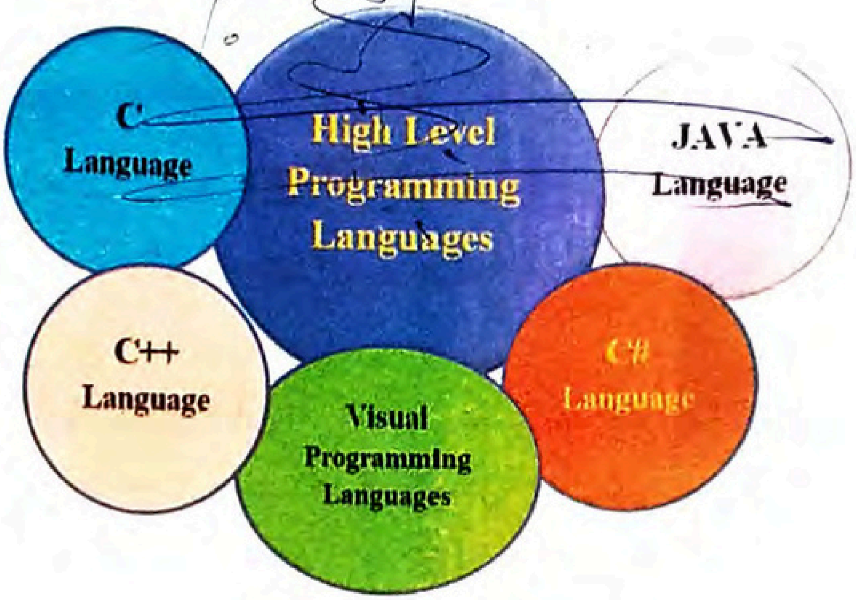
C ++ 1980ء کی دہائی کے شروع میں بجرنے سٹراسٹرپ (Bjarne Stroustrup) نے تیار کی تھی۔ یہ ایک آبجیکٹ اورینٹڈ پروگرامنگ (OOP) لینگویج ہے۔ یہ C لینگویج کی ہی توسیع ہے۔ C ++ میں C لینگویج کے تمام عناصر شامل ہیں اور ساتھ ہی ساتھ آبجیکٹ، کلاسز، ایونٹ اور دیگر آبجیکٹ اورینٹڈ کے تمام تصورات کے ساتھ کام کرنے کی اضافی خصوصیات ہیں۔ زیادہ تر ایپلی کیشن پروگرام، جیسے ورڈ پروسیسرز اور اسپریڈ شیٹ C ++ میں ہی تیار کی گئی ہیں۔

وژول پروگرامنگ لینگویجز (Visual Programming Languages)

وژول پروگرامنگ کی اصطلاح ونڈوز پر مبنی ایپلی کیشنز یا گرافیکل یوزر انٹرفیس (GUI) ایپلی کیشنز بنانے یا تیار کرنے کے لیے استعمال ہوتی ہے۔ وژول پروگرامنگ لینگویجز، جن کو بعض اوقات پانچویں نسل کی پروگرامنگ لینگویج بھی کہا جاتا ہے، آبجیکٹ اورینٹڈ پروگرامنگ لینگویجز ہیں۔ اس پروگرامنگ لینگویج میں وژول یا گرافیکل یوزر انٹرفیس (GUI) کی سہولیات ہونے کی وجہ سے یہ GUI والی ایپلی کیشنز تیار کرنے کے لیے نہایت مفید ہے۔

بہت مقبول اور وسیع پیمانے پر استعمال ہونے والی وژول پروگرامنگ لینگویجز کی مثال وژول بیسک (VB)، وژول C ++ ، C# اور Java ہیں۔

NOT FOR SALE



شکل نمبر 2.7 مقبول ہائی لیول پروگرامنگ لینگویجز

C # لینگویج (C# Language)

#C یا سی شارپ (C-Sharp) ایک آبجیکٹ اوریئنٹڈ پروگرامنگ لینگویج ہے۔ اسے مائیکروسافٹ کارپوریشن نے سن 2000ء میں تیار کیا۔ یہ Visual Basic اور Java کے عناصر اور اجزاء کے ساتھ ++ C پر مبنی ہے۔ اس کا استعمال ایسی ویب ایپلی کیشنز میں کیا جاتا ہے جسے انٹرنیٹ پر استعمال کیا جاسکے۔ #C میں کمپیوٹر گیم، یوٹیلٹی پروگرام، آپریٹنگ سسٹم، کمپائلر، کاروباری ایپلی کیشنز اور ویب ایپلی کیشنز سمیت تمام قسم کے پروگرام تیار کیے جاسکتے ہیں۔

جاوا لینگویج (JAVA Language)

Java ایک ہائی لیول آبجیکٹ اوریئنٹڈ پروگرامنگ لینگویج ہے جو سن مائیکروسسٹمز (Sun Microsystems) نے 1991ء میں تیار کی۔

یہ بھی ++ C کی طرح ہی ہے لیکن یہ خاص طور پر ویب پیج ڈیزائننگ کے لئے استعمال کی جاتی ہے۔

جاوا ایپلیٹس (Applets) کو جاوا لینگویج میں ہی تیار کیا جاتا ہے۔ جاوا ایپلیٹ ایک چھوٹا جاوا پروگرام ہے جو کسی اور پروگرام جیسے ویب براؤزر میں چلایا جاتا ہے۔ جاوا ایپلیٹ کو ویب پیجز کے اندر سمو دیا جاتا ہے تاکہ انٹرنیٹ پر کسی تصویر یا امیج اینیمیشن کو متحرک دکھایا جاسکے اور کسی ڈیٹا بیس تک رسائی حاصل کی جاسکے۔

2.1.5 کمپائلر اور انٹرپریٹر (Compiler and Interpreter)

کمپائلر (Compiler)

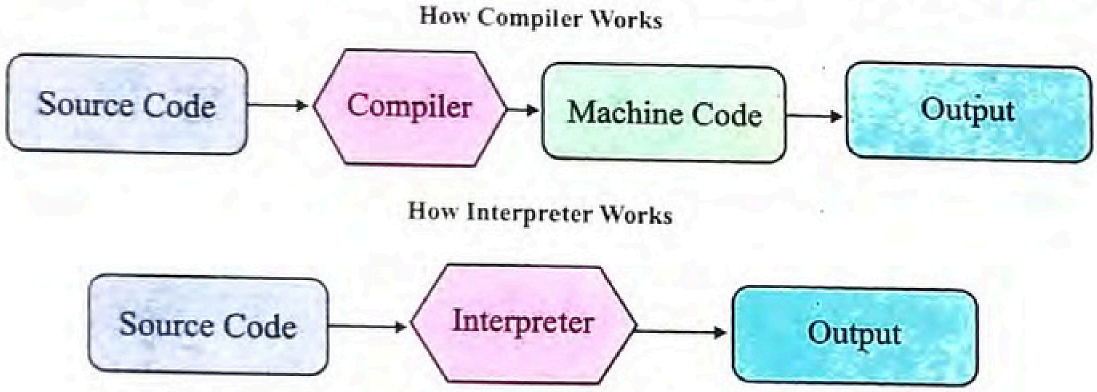
کمپائلر ایک سسٹم سافٹ ویئر ہے جو ہائی لیول پروگرامنگ لینگویج کے سورس کوڈ (Source Code) کو مشین لینگویج آبجیکٹ کوڈ

میں ترجمہ کرتا ہے۔ یہ انگریزی کی طرح ++ C لینگویج میں لکھے ہوئے کوڈ کو مشین لینگویج (بائنری ہندسے 0 اور 1) میں ترجمہ کرتا ہے جسے کمپیوٹر براہ راست سمجھتے ہیں۔ یہ پہلے پورے پروگرام کو مشین لینگویج میں تبدیل کرتا ہے اور پھر اس کے بعد اسے چلاتا ہے۔ پروگرامنگ لینگویجز جیسے ++ C / C، Pascal اور Java کمپائلر کو ہی استعمال کرتی ہیں۔

انٹرپرٹر (Interpreter)

انٹرپرٹر بھی ہائی لیول پروگرامنگ لینگویج کا مترجم ہے جو پروگرام کی اسٹیٹمنٹ کو ایک ایک کرکے ترجمہ کرتا ہے اور ساتھ ہی ساتھ چلاتا بھی ہے۔ یہ پروگرام کی ایک اسٹیٹمنٹ کو لے کر اسے مشین لینگویج میں تبدیل کرتا ہے، اس پر عمل درآمد کرتا ہے، اور پھر اگلی اسٹیٹمنٹ پر چلا جاتا ہے۔

ہر دفعہ جب کوئی اسٹیٹمنٹ پڑھی جائے گی تو اسے چلانے سے پہلے ضرور مشین لینگویج میں ترجمہ کیا جائے گا۔ کمپائلر کمپیوٹر پر کسی پروگرام کو چلانے سے پہلے پورے پروگرام کو ایک ہی بار پڑھتا ہے اور آبجیکٹ پروگرام میں ترجمہ کرتا ہے۔ انٹرپرٹر کا استعمال کرنے والی پروگرامنگ لینگویجز کی مثالوں میں جاوا اسکرپٹ (JAVA Script)، BASIC، Visual Basic اور Perl شامل ہیں۔



شکل نمبر 2.8 کمپائلر اور انٹرپرٹر کیسے کام کرتے ہیں

2.2 پروگرامنگ کا انوائرمنٹ (Programming Environment)

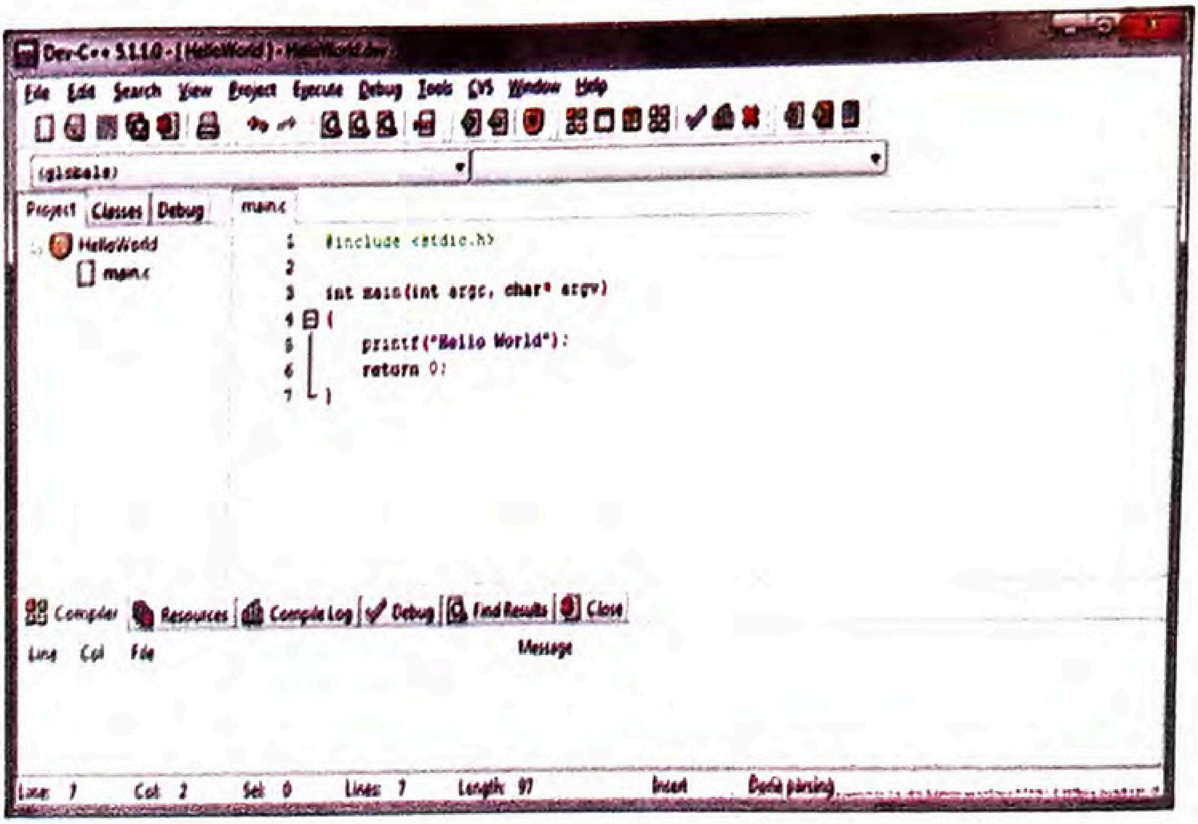
پروگرامنگ انوائرمنٹ پروگرامنگ ٹولز کا ایک مجموعہ ہے جو کمپیوٹر پروگرام بنانے، کمپائل کرنے اور انہیں چلانے کے لیے استعمال کیا جاتا ہے۔ اس میں ایڈیٹر (Editor)، کمپائلر (Compiler)، لنکر (Linker)، ڈی بگر (Debugger) وغیرہ شامل ہیں۔ اسے انٹیگریٹڈ ڈویلپمنٹ انوائرمنٹ (IDE) کا نام بھی دیا جاتا ہے جو کمپیوٹر پروگرام بنانے میں مدد کرتا ہے۔ انٹیگریٹڈ ڈویلپمنٹ انوائرمنٹ در حقیقت ایک ایپلیکیشن سافٹ ویئر ہے جو گرافیکل یوزر انٹرفیس (GUI) مہیا کرتا ہے۔

2.2.1 انٹیگریٹڈ ڈویلپمنٹ انوائرمنٹ (IDE)

انٹیگریٹڈ ڈویلپمنٹ انوائرمنٹ (IDE) پروگراموں کو تیار کرنے، مرتب کرنے اور چلانے کے لئے پروگرامنگ لینگویج

NOT FOR SALE

ڈویلپمنٹ ٹول ہے۔ IDE کمپیوٹر سافٹ ویئر ہے جہاں پروگرام بنانے کے لئے درکار تمام عوامل اور آلات کو یکجا کیا گیا ہے۔ آج کل کے جدید IDEs میں پروگرامرز کے لیے بہت ہی آسان اور خوبصورت GUI موجود ہیں۔C لینگویج کے لیے بھی ایک IDE استعمال ہوتا ہے، جیسا کہ شکل نمبر 2.9 میں دکھایا گیا ہے۔



شکل نمبر 2.9 C لینگویج کا IDE

2.2.2 C پروگرامنگ انوائرمنٹ کے مختلف اجزاء (Modules of the C Programming Environment)

ذیل میں C لینگویج میں پروگرامنگ کے لیے درکار اجزاء بیان کیے گئے ہیں۔

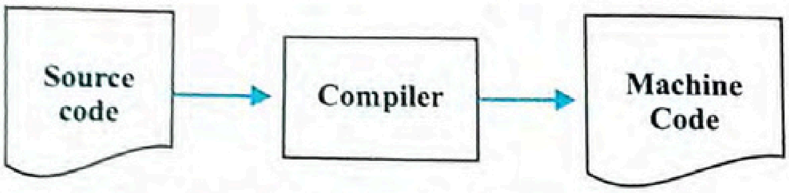
1. ایڈیٹر (Editor)

ایڈیٹر یا ٹیکسٹ ایڈیٹر ایک ایسا پروگرام ہے جو صارفین کو C لینگویج سورس پروگرام بنانے اور اس میں ترمیم کرنے کے قابل بناتا ہے۔C کے پروگرام کی ایکسٹینشن *c.* ہوتی ہے۔ مثلاً student.c ایسا ہی ایک پروگرام ہے۔ تقریباً ہر ایک ایڈیٹر پروگرام کمپائل کرنے کے دوران پروگرام میں موجود تمام غلطیوں کو واضح کرتا ہے تاکہ انہیں آسانی سے دور کیا جاسکے۔

2. کمپائلر (Compiler)

ایک ایسا کمپیوٹر سافٹ ویئر ہے جو C لینگویج پروگرام سورس کوڈ (Source Code) کو مشین کوڈ (آبجیکٹ پروگرام) میں ترجمہ کرتا ہے تاکہ کمپیوٹر اسے سمجھ کر چلا (execute) سکے۔C لینگویج کے کسی بھی کمپائل شدہ پروگرام کی ایکسٹینشن (پہچان) *obj.* ہوتی ہے۔ مثلاً ***student.obj***۔ کمپائل کرنے کے عمل کو ذیل میں شکل نمبر 2.10 میں دکھایا گیا ہے۔

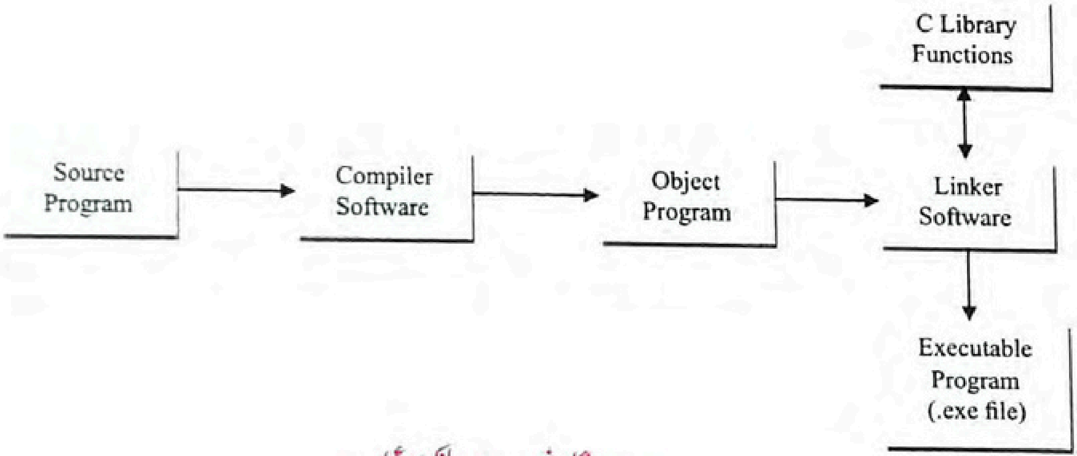
NOT FOR SALE



شکل نمبر 2.10 سورس کوڈ کو کمپائل کر کے مشین کوڈ میں تبدیل کرنے کا عمل

3. لنکر (Linker)

ایک کمپیوٹر پروگرام جو ایک یا ایک سے زیادہ آبجیکٹ کوڈز لیتا ہے اور ان کو ایک ہی قابل عمل پروگرام میں جوڑتا ہے، اسے لنکر کہتے ہیں۔ C لینگویج ایگزیکیوٹیبل پروگرام کی ایکسٹینشن exe. ہے جیسے student.exe لنکر کے پورے عمل کو شکل نمبر 2.11 میں دکھایا گیا ہے۔



شکل نمبر 2.11 لنکر کا عمل

4. لوڈر (Loader)

یہ ایک پروگرام ہے جو C زبان کے پروگراموں کو میموری میں لوڈ کرتا ہے اور پھر ان پر عمل درآمد کرتا ہے۔

5. ڈیبگر (Debugger)

کمپیوٹرز میں ڈیبگنگ، کمپیوٹر پروگرام کوڈ میں موجود غلطیوں (Errors) کو تلاش کرنے اور ٹھیک کرنے کے عمل کو کہتے ہیں۔

C لینگویج ڈیبگر ایک کمپیوٹر پروگرام ہے جو صارفین کو پروگرام چلانے دیتا ہے، متغیر کی اقدار کی جانچ پڑتال یا فنکشن کو بھیجی گئی اقدار کو دیکھنے اور پروگرام میں غلطیوں کو ڈھونڈنے اور دور کرنے میں مدد کرتا ہے۔

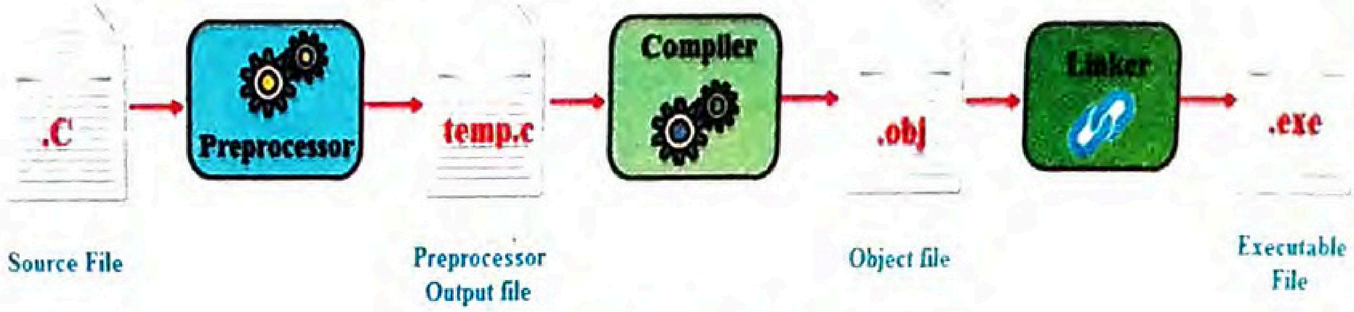
2.3 پروگرامنگ کی اساسیات

C پروگرامنگ کا آغاز (Getting started with C)

C ایک مشہور اور مؤثر پروگرامنگ زبان ہے جو سسٹم اور ایپلی کیشن سافٹ ویئر کے لئے وسیع پیمانے پر استعمال ہوتی ہے۔

NOT FOR SALE

C پروگرام بنانے اور چلانے کے لیے، آپ کو C لینگویج کمپائیلر کی ضرورت ہوتی ہے۔ C لینگویج کے لیے بہت سارے مفت لیکن پیشہ ورانہ کمپائلر دستیاب ہیں۔



شکل نمبر 2.12 کمپائل اور لنک کرنے والا C پروگرام

C کے ہر مکمل پروگرام میں ایک فنکشن ہوتا ہے جس کو main() (مین) فنکشن کہا جاتا ہے۔ فنکشن در حقیقت بہت ساری اسٹیٹمنٹس کا مجموعہ ہوتا ہے جو کوئی نہ کوئی کام انجام دینے کے لئے استعمال ہوتا ہے۔ جب پروگرام چلتا ہے تو سب سے پہلے main فنکشن ہی چلایا جاتا ہے۔ اور فنکشن کے ذریعے ہم دوسرے فنکشن چلا سکتے ہیں، چاہے وہ یوزر کے بنائے ہوئے ہوں یا بلٹ ان (Built-in)۔ C کمپائلر کے ساتھ آنے والے سٹینڈرڈ فنکشنز تک رسائی کے لیے آپ کو include # کی مدد سے ہیڈر (header) فائل شامل کرنے کی ضرورت ہوتی ہے۔

2.3.1 ہیڈر فائل (Header File)

مختلف کاموں کو انجام دینے کے لئے ہیڈر فائلیں سٹینڈرڈ لائبریری فنکشن کا ایک مجموعہ ہیں۔ مختلف مقاصد کے لئے بہت ساری ہیڈر فائلیں ہیں۔ ہر ہیڈر فائل میں مختلف قسم کے بنے بنائے فنکشن ہوتے ہیں۔ ایک ہی پروگرام میں بہت ساری ہیڈر فائلوں کو شامل کیا جاسکتا ہے۔ پروگرام میں ہیڈر فائل کو اپنے اندر موجود کسی بھی فنکشن کے استعمال سے پہلے پہلے، شامل کرنا ضروری ہے۔

ہیڈر فائل کی h. ایکسٹینشن ہے۔ پروگراموں میں ہیڈر فائلوں کو شامل کرنے کے لئے include# پری پروسیسر ڈائریکٹو استعمال کیا جاتا ہے۔ یہ فائلیں C کمپائلر سسٹم میں پہلے سے دستیاب ہوتی ہیں۔

ہیڈر فائلوں کو عام طور پر INCLUDE سب ڈائریکٹری میں محفوظ کیا جاتا ہے۔ ہیڈر فائل کا نام Angle Bracket < > میں لکھا جاتا ہے۔

مثال کے طور پر

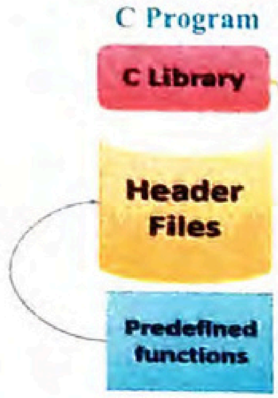
```
# include <stdio.h>
```

ہر ہیڈر فائل میں فنکشن کے ایک خاص گروپ کے لئے معلومات شامل ہوتی ہیں۔ stdio.h ہیڈر فائل کی طرح C میں سٹینڈرڈ اِن پٹ اور آؤٹ پٹ فنکشن کے اعلامیہ پر مشتمل ہے جو اِن پٹ حاصل کرنے اور آؤٹ پٹ کو پرنٹ کرنے کے لئے استعمال ہوتا ہے۔ اسی

NOT FOR SALE

طرح، ہیڈر فائل <math.h> ریاضی کے فنکشن کو استعمال کرنے کے لیے شامل کی جاتی ہیں۔ مثلاً

```
# include <math.h>
```



شکل نمبر 2.13 C زبان میں ہیڈر فائلیں

2.3.2 کلیدی الفاظ (Reserved Words)

وہ الفاظ جو کسی بھی زبان کے ذریعہ خصوصی مقاصد کے لئے استعمال ہوتے ہیں انہیں کلیدی الفاظ یا ریزروڈ ورڈز (Reserved Words) یا (Keywords) کہتے ہیں۔ ہر Keyword کے معنی مستقل ہوتے ہیں اور یوزر کے ذریعہ اسے تبدیل نہیں کیا جاسکتا۔ ہر Keyword کو چھوٹے حروف میں لکھا جاتا ہے جیسے آٹو (auto)، بریک (break) وغیرہ۔ C لینگوئج 32 Keywords کو سپورٹ کرتی ہے۔ ذیل میں دیئے گئے جدول میں چند ایک Keywords لکھے گئے ہیں۔

C لینگوئج میں 32 کلیدی الفاظ

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

2.3.3 C پروگرام کا بنیادی ڈھانچہ

C پروگرام میں تین اہم حصے ہوتے ہیں:

- پری پروسیسر ڈائریکٹیوز (Preprocessor Directives)
- مین فنکشن (The main () function)
- مین فنکشن کی باڈی (Body of main { })

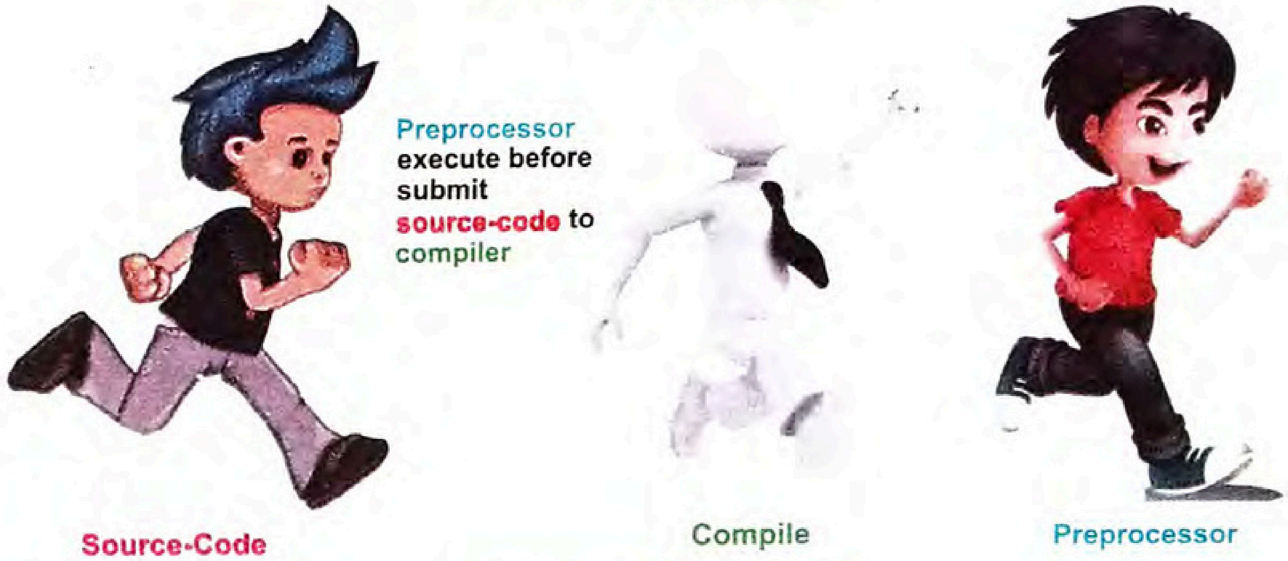
NOT FOR SALE

پری پروسیسر ڈائریکٹیوز (Preprocessor Directives)

اصل پروگرام کے آغاز سے پہلے جو ہدایات کمپائلر کو دی جاتی ہیں ان کو پری پروسیسر ڈائریکٹو کہا جاتا ہے۔ انھیں کمپائلر ڈائریکٹیو بھی کہتے ہیں۔ پری پروسیسر ڈائریکٹیو کمپائلر کی ہدایات پر مشتمل ہوتی ہیں۔ کمپائلیشن کے وقت ان ہدایات سے خصوصی ہدایات یا کوڈ کو پروگرام میں شامل کیا جاتا ہے۔

یہ پری پروسیسر ڈائریکٹیو عام طور پر ہیش سائن # اور مطلوبہ الفاظ، جیسے، include#، define# وغیرہ سے شروع ہوتی ہیں۔

include# پری پروسیسر ڈائریکٹیوز



شکل نمبر 2.14 C لینگویج میں پری پروسیسر کی مثال

include# کمپائلر کو کسی بھی بیرونی (External) ہیڈر فائلوں کو پروسیس کرنے کی اجازت دیتی ہے۔ جب یہ پری پروسیسر C کمپائلر کے ذریعہ انجام پاتا ہے، تو وہ ہیڈر فائل تلاش کرتا ہے جو زاویائی بریکٹ < > میں لکھا ہوا ہے اور اسے سورس فائل کے اندر کاپی کر دیتا ہے۔

سائنٹکس:

```
#include <header-file.h>
```

یا

```
#include "header-file.h"
```

مثالیں:

```
#include<stdio.h>
#include<conio.h>
```

NOT FOR SALE

کچھ اہم ہیڈر فائلیں:

<stdio.h>

stdio.h ہیڈر فائل سٹینڈرڈ اِن پٹ آؤٹ پٹ ہیڈر کے لئے ہے۔ اس میں پروگرام میں سٹینڈرڈ اِن پٹ / آؤٹ پٹ فنکشنز یعنی *scanf* () اور *printf* () شامل ہیں۔

<conio.h>

conio.h ہیڈر فائل کنسول اِن پٹ / آؤٹ پٹ کے لئے ہے۔ C پروگرامنگ میں استعمال ہونے والی یہ کنسول اِن پٹ / آؤٹ پٹ فنکشنز پر مشتمل ہوتی ہے۔ conio.h کے عام طور پر استعمال ہونے والے کچھ فنکشنز ہیں () clrscr ، () getch ، getche () وغیرہ۔ *conio.h* کے فنکشنز کو اسکرین کو صاف کرنے، ٹیکسٹ اور رنگ تبدیل کرنے، ٹیکسٹ کو حرکت دینے، کونسی key دبائی گئی ہے اس کا پتہ لگانے اور اس کے علاوہ بہت سارے کاموں کے لیے استعمال کیا جاتا ہے۔

<math.h>

math.h ہیڈر فائل ریاضی کے عمل کو انجام دینے کے لئے استعمال ہوتے ہیں۔ یوزر پروگراموں میں ریاضی کے فنکشنز استعمال کر سکتے ہیں، اعداد کی مطلق قیمت کا حساب لگانے، لاگرتھم معلوم کرنے اور کسی زاویہ کے sine ، cosine معلوم کرنے کے لئے ٹرگونو میٹرک فنکشن استعمال کرتے ہیں۔

<string.h>

سٹرنگ لائبریری string.h میں ایسے مفید فنکشنز ہوتے ہیں جن کو سٹرنگ کے لیے استعمال کیا جاتا ہے۔ جیسے () strcpy ، () strcat ، () strcmp ، () strlen ، وغیرہ۔

main() فنکشن

یہ فنکشن ہر C پروگرام میں ہمیشہ موجود رہنا چاہئے۔ main فنکشن سے ہی C پروگرام پر عمل درآمد شروع ہوتا ہے۔ جب C کا کوئی بھی پروگرام چلایا جاتا ہے تو کنٹرول سب سے پہلے براہ راست () main فنکشن میں منتقل ہو جاتا ہے۔ اگر پروگرام میں مین فنکشن نہ ہو تو پروگرام کمپائل نہیں ہو گا اور غلطی کا پیغام (Error Message) ظاہر ہو گا۔

main() فنکشن کا سائنٹکس یہ ہے:

```
main( )
    {
        Program statement.........
    }
```


NOT FOR SALE

main( ) فنکشن کی باڈی (Body of main( ) Function)

پروگرام کی اسٹیٹمنٹس main( ) فنکشن کے اندر لکھی جاتی ہیں، main فنکشن کی باڈی { } بریکٹ پر مشتمل ہوتی ہے۔ یہ اسٹیٹمنٹس پروگرام کی باڈی کہلاتی ہیں۔ جب فنکشن چلایا جاتا ہے تو اِس میں موجود اسٹیٹمنٹس اپنا کام انجام دیتی ہیں۔

C اسٹیٹمنٹس (C Statements)

C اسٹیٹمنٹ ایک ایسی عبارت ہے جو حقیقت میں کچھ کر کے دکھاتی ہے۔ اصل میں کمپیوٹر کو کوئی کام کرنے کی ہدایت ہے۔ مثال کے طور پر؛ (*"Hello World"*) ایک C اسٹیٹمنٹ ہے جو اسکرین پر Hello World لکھا ہوا دکھاتی ہے۔ C کے اسٹیٹمنٹ عام طور پر چھوٹے حروف میں لکھے جاتے ہیں۔ C میں ہر اسٹیٹمنٹ سیمی کالن (Semicolon) کے ساتھ ختم ہوتی ہے۔

2.3.4 C میں تبصرے یا کمنٹس کا استعمال (Comments in C)

ایک کمنٹ پروگرام کے سورس کوڈ کی وضاحت ہے۔ یہ ڈویلپر کو کوڈ کی لاجک کی وضاحت کرنے میں مدد کرتا ہے اور پروگرام کو بہتر بناتا ہے۔ اس کا استعمال کمپیوٹر پروگرام / اسٹیٹمنٹ کے مقصد اور عمل کی ایک مختصر وضاحت فراہم کرنے کے لئے کیا جاتا ہے۔ جو پروگرام کو صارفین کے لئے زیادہ قابل فہم بناتا ہے۔ کمنٹس عملدرآمد نہ کرنے والی لائنیں ہیں اور ترجمے کے دوران کمپائلر ان کو نظر انداز کرتا ہے۔ یہ نوٹ کرنا ضروری ہے کہ کمنٹس لائن صرف پروگرام کے کوڈ میں ظاہر ہوتی ہیں اور پروگرام کے آؤٹ پٹ پر ان کا کوئی اثر نہیں ہوتا ہے۔ C پروگرام میں کہیں بھی کمنٹس ظاہر ہو سکتے ہیں اور پروگرامز کمنٹ لائن میں کچھ بھی لکھنے کے لئے آزاد ہے۔

C میں کمنٹس لکھنے کے دو طریقے ہیں۔

1۔ سنگل لائن کمنٹ: سنگل لائن پر کمنٹ // (دو سلیش) استعمال کر کے شامل کیے جاتے ہیں۔ // (دو سلیش) کے دائیں طرف لکھی ہوئی عبارت کمنٹس کے طور پر سمجھی جاتی ہے۔ یہ انداز کسی پروگرام کی ایک اسٹیٹمنٹ پر کمنٹ لکھنے کے لئے مفید ہے۔ مثال کے طور پر:

```
//this is a comment line
// Sorting process
//Function 1 starts here
```

2۔ ملٹی لائن کمنٹس: کمنٹ کے آغاز میں ملٹی لائن کمنٹ کو */ سے کوڈ میں شامل کیا جاتا ہے۔ کثیر لائن تبصرے ختم کرنے کے لئے /* کا استعمال کیا جاتا ہے۔ جب کوئی کمنٹ کئی لائنوں پر مشتمل ہو تو یہ انداز سب سے زیادہ کارآمد ہے۔ مثال کے طور پر:

```
/*
Program-Name:       Examination Record
Programmer-Name:    M. Khalid
Date:               6th May
Purpose:            Program to show Examination Record
*/
```

سلیش-ایسٹیریک */ کمنٹ کمپائلر کو بتاتا ہے کہ اس کے بعد آنے والی ہر چیز کو نظر انداز کر دے جب تک کہ اس کو /* کا نشان دوبارہ نہیں مل جاتا۔

NOT FOR SALE

comment.cpp

```
#include <iostream>
using namespace std;

int main()
{
     // C++ program to shows how to comment in C++
     // This is the single line comment

     /*  This is
         Multiple Line
         Comment
      */

      cout<<"Single And Multiple line comment will not display ";

     return 0;
}
```

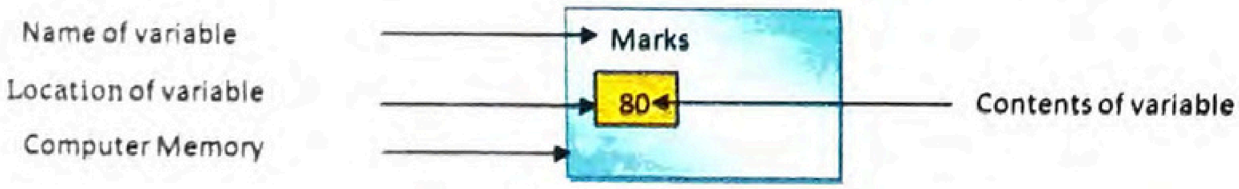
شکل نمبر 2.15 سنگل لائن اور ملٹی لائن کمنٹس

2.4 متغیرات اور مستقلات (Variables and Constants)

2.4.1 متغیر (Variable)

ایک ایسی مقدار جس کی قیمت پروگرام کے عمل کے دوران تبدیل ہو جائے اسے متغیر یا ویری ایبل (Variable) کہا جاتا ہے۔

ایک ویری ایبل کمپیوٹر میموری میں اسٹوریج یا میموری والے مقام کو ظاہر کرتا ہے۔ اسی وجہ سے ویری ایبل میں ذخیرہ شدہ ڈیٹا بھی عارضی ہوتا ہے۔ یہ صرف پروگرام چلنے کے دوران استعمال ہوتا ہے اور اس پر کارروائی کی جاسکتی ہے۔ ویری ایبل میں ذخیرہ شدہ ڈیٹا پروگرام ختم ہونے پر خود بخود ختم ہو جاتا ہے۔



شکل نمبر 2.16 کمپیوٹر میموری میں ویری ایبل

ویری ایبل کی اقسام

ویری ایبل دو قسم کے ہوتے ہیں، عددی اور حرفی (Numeric & Character)۔

عددی (Numeric) ویری ایبل کا استعمال کمپیوٹر پروگراموں میں عددی اقدار کو ظاہر کرنے کے لئے کیا جاتا ہے۔ وہ انٹیجر اور فلوٹنگ پوائنٹ ویلیو کی نمائندگی کرتے ہیں۔ عددی ویری ایبل کی کچھ مثالوں میں marks، sum, n, width وغیرہ ہیں۔

NOT FOR SALE

کریکٹر ویری ایبل کمپیوٹر پروگراموں میں کیریکٹر ویلیو کی نمائندگی کرتے ہیں۔ یہ کسی ایک کیریکٹر یا حروف کے مجموعے کو ظاہر کرتے ہے۔ کیریکٹر ویری ایبل کی کچھ مثالیں Gender،Name ،وغیرہ ہیں۔

2.4.2 مستقل (Constant)

کانسٹنٹ ایک مقدار ہے جو پروگرام کے چلنے کے دوران تبدیل نہیں کی جاسکتی ہے۔

حقیقت میں کانسٹنٹ ایک ایسی قیمت ہوتی ہے جو کسی پروگرام میں براہ راست ٹائپ کی جاتی ہے۔ یہ پروگرام میں اس جگہ ظاہر ہوتی ہے جہاں اس کی ضرورت ہو۔ مثال کے طور پر، مندرجہ ذیل اسٹیٹمنٹ میں ایک سٹرنگ کانسٹنٹ موجود ہے:

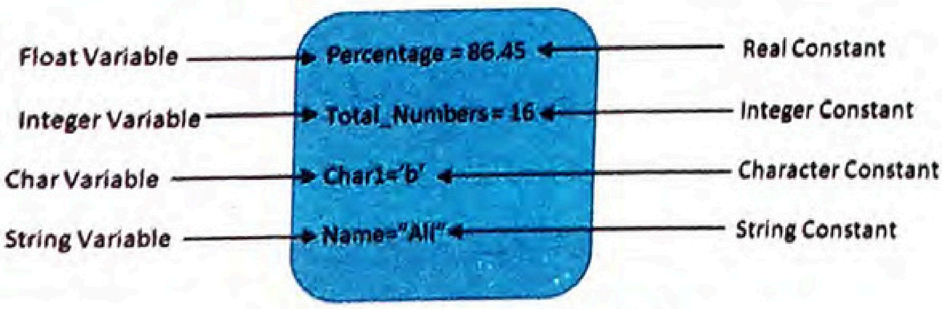
"Pakistan Zinda Bad"

```
printf("Pakistan Zinda Bad");
```

اسی طرح ذیل کی مثال میں بھی:

```
int age= 19;
```

Age ایک عددی ویری ایبل ہے اور 19 وہ کانسٹنٹ قیمت ہے جو Age میں سٹور کی گئی ہے۔



شکل نمبر 2.17 ویری ایبل اور کانسٹنٹ کی اقسام

2.4.3 متغیر کا نام رکھنے کیلئے اُصول (Rules for Writing Variable Names)

C پروگرام میں استعمال ہونے والے کسی بھی متغیر کو درج ذیل اصولوں کے مطابق ہونا چاہئے:

1. انہیں ہمیشہ ایک حرف (Letter) سے شروع ہونا چاہئے، تاہم کچھ سسٹم پہلے کیریکٹر کی حیثیت سے انڈر سکور کی اجازت بھی دیتے ہیں۔
2. ویری ایبل کے نام کی لمبائی C کے کمپائلر پر منحصر ہے۔
3. ویری ایبل کے نام میں خالی جگہ (Blank Space) رکھنے کی اجازت نہیں ہوتی۔
4. ویری ایبل کا نام کوئی ریزرو ورڈ / کی ورڈ نہیں ہونا چاہئے۔
5. اس میں کوئی خاص علامت نہیں ہونا چاہئے جیسے، #،^ وغیرہ۔

NOT FOR SALE

6. ایک ہی پروگرام میں ایک ویری ایبل کا نام جو ایک ڈیٹا ٹائپ کے لئے استعمال ہو چکا ہے وہ کسی دوسری ڈیٹا ٹائپ کے لئے استعمال نہیں کیا جا سکتا۔

C حروف کے اعتبار سے ایک حساس زبان ہے۔ اس میں چھوٹے اور بڑے حروف میں لکھے گئے ایک جیسے نام مختلف ویری ایبل سمجھے جاتے ہیں۔ مثال کے طور پر، ویری ایبل Pay اور pay دو مختلف ویری ایبل ہیں۔

درست اور غلط ویری ایبل ناموں کی کچھ مثالیں درج ذیل ہیں۔

ویری ایبل کا نام	درست / غلط	وجوہات
Num	درست	
Perform	درست	
Switch	غلط	کلیدی لفظ Keyword
Aysha	درست	
int	غلط	کلیدی لفظ Keyword
3pak	غلط	پہلا حرف عدد ہے۔
x-y	غلط	Underscore کے علاوہ خاص علامات کی اجازت نہیں ہے
Roll No	غلط	نام کے درمیان میں خالی جگہ کی اجازت نہیں۔

2.4.4 C میں استعمال ہونے والی ڈیٹا کے اقسام (Datatypes in C)

C کے تمام کمپائلر میں ویری ایبل کو درج ذیل بنیادی ڈیٹا ٹائپ (Datatypes) تفویض کر سکتے ہیں۔

- انٹیجر (int) Integer
- فلوٹنگ پوائنٹ (Float) اور ڈبل پریسین فلوٹنگ پوائنٹ (double)
- کیریکٹر (char) Character

1. انٹیجر (int, long) ٹائپ (Integer Type)

انٹیجر صحیح اعداد ہوتے ہیں ان کی مقداروں کی حدود مشین پر منحصر ہوتی ہیں۔ پروگرامنگ کی ایک اچھی زبان وہ ہوتی ہے جو پروگرامر کو مختلف اقسام اور حدود (range) والے اعداد کے استعمال کے لیے سہولت فراہم کرے تاکہ اعداد کو انکی درکار میموری کے مطابق کمپیوٹر میں سٹور کیا جا سکے۔ C میں انٹیجر اسٹوریج کے لیے 3 مختلف کلاسیں استعمال ہوتی ہیں۔ جیسے *short*، *int* اور *long*۔ یہ تمام ڈیٹا ٹائپ دو طرح کی ہیں *signed* اور *unsigned*۔

NOT FOR SALE

پروگرام میں انٹیجر کی تعریف کرنے کے لیے درج ذیل اسٹیٹمنٹ لکھی جاسکتی ہے:

int variable;

مندرجہ ذیل جدول میں ان تمام ڈیٹا اقسام کے لیے درکار میموری اور قیمتوں کی حدود لکھی گئی ہیں۔

ویری ایبل کی ڈیٹا قسم	میموری میں درکار بائٹس	قیمت کی حد (Range)
int	2	32,768- سے 32,767+
long	4	2,147,483,648- سے 2,147,483,647+
unsigned int	2	0 سے 65,535

2. فلوٹنگ پوائنٹ ڈیٹا ٹائپ (float, double, long double)

جاننا ضروری ہے۔

32 بٹ سسٹم میں، انٹیجر ویری ایبل کا سائز 2 بائٹس سے 4 بائٹس تک بڑھایا گیا ہے۔

نقطہ اعشاریہ والے اعداد اصل میں حقیقی اعداد کے اظہار کے لیے استعمال ہوتے ہیں۔

ان اعداد کی نقطہ اشاریہ کے بعد 7 اعداد تک درست قیمت ظاہر ہوتی ہے۔ فلوٹنگ پوائنٹ نمبرز کو کی ورڈ float کے ذریعے ظاہر کیا جاتا ہے۔ جب فلوٹنگ پوائنٹ نمبر کی درستگی ناکافی ہوتی ہے تو، ہم نمبر کی وضاحت کے لئے double اور long double کا استعمال کرسکتے ہیں۔ پروگرام میں فلوٹنگ پوائنٹ کی تعریف کرنے کے لیے درج ذیل اسٹیٹمنٹ لکھی جاسکتی ہے۔

float variable;

عموماً مندرجہ بالا فلوٹنگ پوائنٹ ڈیٹا ٹائپس کے لئے کمپائلر بائٹس کی تعداد مختص کرتا ہے جو کہ ذیل میں دی گئی ہیں۔

ویری ایبل کی ڈیٹا قسم	میموری میں درکار بائٹس	قیمت کی حد
float	4	3.4×10^{-38} to 3.4×10^{38}
double	8	1.7×10^{-308} to 1.7×10^{308}
long double	10	3.4×10^{-4932} to 1.1×10^{4932}

3. کریکٹر (char) ڈیٹا ٹائپ (Character datatype)

کیریکٹر ویری ایبل میں صرف ایک حرف (Character) ہی سٹور ہو سکتا ہے۔

کیریکٹر ویری ایبل میں ایک حرف کو سٹور کرنے کے لیے C کمپائلر، میموری میں صرف ایک بائٹ مختص کرتا ہے۔

پروگرام میں کیریکٹر ویری ایبل کی تعریف درج ذیل اسٹیٹمنٹ سے کی جاسکتی ہے:

char ch;

NOT FOR SALE

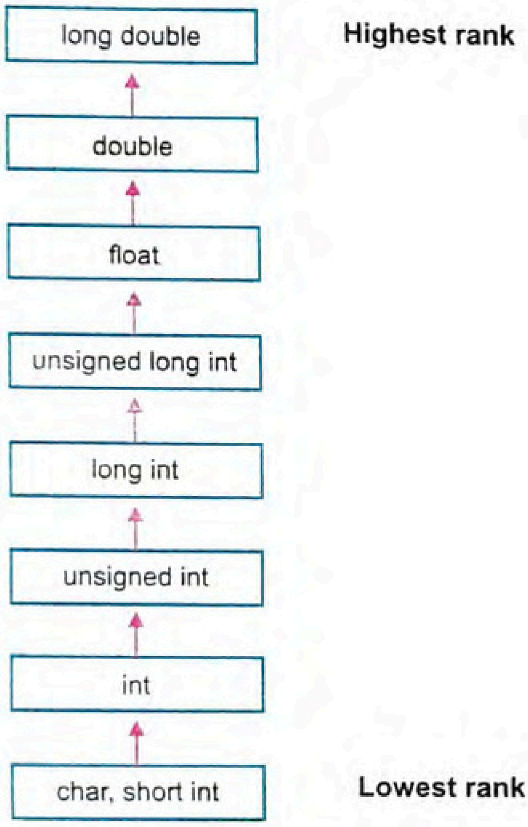
2.4.5 ٹائپ کاسٹنگ (Type Casting)

پروگرام چلنے کے عمل کے دوران کسی ایک ڈیٹا ٹائپ کو کسی دوسری ڈیٹا ٹائپ میں تبدیل کرنے کے عمل کو ٹائپ کاسٹنگ کے نام سے جانا جاتا ہے۔ ٹائپ کاسٹنگ دو طریقوں سے کی جاسکتی ہے۔

- مخفی ٹائپ کاسٹنگ
- ظاہری ٹائپ کاسٹنگ

1. مخفی ٹائپ کاسٹنگ (Implicit Type Casting)

C کمپائلر اس قسم کی ٹائپ کاسٹنگ خود کر لیتا ہے۔ ریاضی کے عمل میں شامل آپرینڈز کو ایک ہی قسم کا ہونا چاہیے۔ اگر پرآینڈز (operands) کی ڈیٹا ٹائپ مختلف ہیں تو، چھوٹی ڈیٹا ٹائپ والی ویلیو کو بڑی ڈیٹا ٹائپ میں تبدیل کیا جاتا ہے۔



شکل نمبر 2.18 ٹائپ کاسٹنگ کی ترتیب

مثال:

فرض کریں x کوئی انٹیجر ہے اور y ایک long ویری ایبل ہے اور اگر درج ذیل کو ہم حل کرنا چاہیں:

x+y

اگرچہ x کی ڈیٹا ٹائپ y کی ڈیٹا ٹائپ سے چھوٹی ہے اس لیے جمع کے عمل کے دوران x کی قیمت کو بھی long ڈیٹا ٹائپ میں تبدیل کر دیا جائے گا۔

NOT FOR SALE

2. ظاہری ٹائپ کاسٹنگ (Explicit Type Casting)

اس طرح کی ٹائپ کاسٹنگ پروگرامر خود کرتا ہے۔ یہ کاسٹ آپریٹر (Cast operator) کا استعمال کر کے کی جاتی ہے۔ کاسٹ آپریٹر کمپیوٹر کو یہ بتاتا ہے کہ ڈیٹا کو ایک قسم سے دوسری قسم میں تبدیل کرنا ہے۔

کاسٹ آپریٹر کا سانٹکس یہ ہے:

```
(type) expression;
```

(type): یہ ڈیٹا کی قسم کو ظاہر کرتا ہے جس میں اوپیرینڈ (operand) کو تبدیل کرنا ہے۔

expression: یہ کانسٹنٹ، ویری ایبل یا expression کو ظاہر کرتا ہے جس کی ڈیٹا ٹائپ کو تبدیل کرنا ہے۔

مثال:

فرض کیا x اور y دو float قسم کے ویری ایبل ہیں۔ x کی قیمت 10.3 ہے اور y کی قیمت 5.2 ہے اور ہم نے ذیل کی عبارت کو حل کرنا ہے:

x%y

مندرجہ بالا عمل پروگرام میں ایک غلطی (error) پیدا کر دے گا کیونکہ x اور y کی ڈیٹا قسم float ہے اور موڈولس (Modulus) پر آپریٹر % کسی float ویری ایبل کے ساتھ کام نہیں کر سکتا۔ یہ صرف انٹیجر پر ہی عمل کرتا ہے۔ اس لیے مندرجہ بالا عبارت کو اس طرح لکھا جاسکتا ہے تاکہ غلطی پیدا نہ ہو۔

(int) x% (int) y

اس عبارت میں x اور y کی قیمتیں پہلے انٹیجر میں تبدیل کر دی جائیں گی اور پھر ان کو حل کیا جائے گا۔ x کی قیمت کو 10 میں اور y کی قیمت کو 5 میں تبدیل کر دیا جائے گا۔ اس عبارت کا نتیجہ 0 (صفر) ہو گا۔

2.4.6 const کا استعمال (Constant Qualifier)

const کو پروگرام کے اندر کانسٹنٹ (Constant) کی تعریف کرنے کے لیے استعمال کیا جاتا ہے۔ کانسٹنٹ کی تعریف کرنے کے لیے اس کا نام اور اس کی ڈیٹا ٹائپ لکھنا لازمی ہے۔ کانسٹنٹ کو ابتدائی قیمت دینا ضروری ہے۔ ابتدائی قیمت پروگرام چلنے کے دوران تبدیل نہیں کی جاسکتی۔

سانٹکس:

const کی مدد سے کانسٹنٹ بنانے کو ذیل میں دکھایا گیا ہے:

const data type identifier = value;

NOT FOR SALE

جبکہ؛

const	ایک کی ورڈ ہے جس سے یہ بتایا جاتا ہے کہ کانسٹنٹ بنایا جارہا ہے۔
data type	بتاتا ہے کہ کانسٹنٹ کی ڈیٹا ٹائپ کونسی ہے۔
identifier	جو کانسٹنٹ بنایا جارہا ہے یہ اس کا نام ہے۔
value	یہ کانسٹنٹ کی ابتدائی قیمت کو ظاہر کرتا ہے۔

مثالیں:

```
const int N = 100;
const float R = 3.5;
```

مذکورہ بالا مثالوں میں ایک انٹیجر کانسٹنٹ N ہے جس کی ابتدائی قیمت 100 اور ایک فلوٹ کانسٹنٹ R ہے جس کی ابتدائی قیمت 3.5 ہے۔ پروگرام کے چلنے کے دوران ان قیمتوں کو تبدیل نہیں کیا جاسکتا۔

یہی کام define# پری پروسیسر کا استعمال کرتے ہوئے بھی کیا جاسکتا ہے:

```
#define N 100
#define R 3.5
```

2.4.7 ویری ایبل بنانا (Declaration) اور ابتدائی قیمت رکھنا (Initialization)

ویری ایبل بنانا (Declartion of Variables)

ویری ایبل کا نام رکھنا اور اس کی ڈیٹا ٹائپ متعین کرنے کو ویری ایبل بنانا یا ویری ایبل ڈکلریشن کہا جاتا ہے۔ تمام ویری ایبل جو پروگرام میں استعمال ہوں گے انہیں ایک اسٹیٹمنٹ کے ذریعے بنایا جاتا ہے جسے ویری ایبل ڈیکلریشن اسٹیٹمنٹ کہا جاتا ہے۔ ویری ایبل بنانے کے لیے ذیل کا سائنٹکس استعمال ہوتا ہے:

```
Datatype List of variables;
```

جبکہ؛

Datatype	ڈیٹا ٹائپ ویری ایبل کی ڈیٹا ٹائپ بتاتا ہے۔ مثال کے طور پر *int*، *float* وغیرہ۔
List of variables	یہ کافی سارے ویری ایبل کی فہرست ہوگی جو ایک دوسرے سے کاما (Comma) کے ذریعے جدا ہونگے۔

مختلف قسم کے ویری ایبل بنانے کے لیے علیحدہ علیحدہ اسٹیٹمنٹ استعمال کی جائیں گی۔ مثال کے طور پر انٹیجر ویری ایبل abc، xyz، d اور s بنانے کے لیے درج ذیل اسٹیٹمنٹ استعمال ہوگی۔

```
int abc, xyz, d, s;
```


NOT FOR SALE

انٹیجر ویری ایبل a اور xy بنانے کے لیے، اور فلوٹ ویری ایبل b بنانے کے لیے، کیریکٹر ویری ایبل ch بنانے کے لیے اور doubleویری ایبل sum بنانے کے لیے ذیل میں دی گئی اسٹیٹمنٹ استعمال ہونگی۔

```
int a, xy;
float b;
char ch;
double sum;
```

ویری ایبل کی ابتدائی قیمت رکھنا (Intialization of Variable)

جب ایک ویری ایبل بنایا جاتا ہے تو میموری میں اس کے لیے جگہ رکھ دی جاتی ہے۔ ویری ایبل کو بناتے وقت کوئی قیمت رکھنے کے عمل کو انیشیلائزیشن (Initialization of Variable) کہا جاتا ہے۔

مثال کے طور پر، انٹیجر ویری ایبل a، bاور c کو 60، 43اور 77 کی قیمتیں اور n کو 3.35 ویلیو تفویض کر کے فلوٹ ٹائپ کے ویری ایبل بنانے کے عمل کو ذیل کی اسٹیٹمنٹ میں دکھایا گیا ہے :

```
int a = 60,  b = 43,  c = 77;
float n=3.35;
```

اہم نکات

- ہدایات (instructions) کا ایک مجموعہ جو ایک کمپیوٹر کو بتاتا ہے کہ کیا کرنا ہے اور کیسے کرنا ہے اسے پروگرام کہتے ہیں۔
- سائنٹکس (Syntax) دراصل کسی بھی پروگرامنگ لینگوئج کے وہ قواعد یا اصول ہیں جن کے تحت کوئی بھی پروگرام لکھا جاتا ہے۔
- سیمینٹک (Semantic) ایک پروگرامنگ لینگوئج کی اسٹیٹمنٹ کے مطلب کی وضاحت کرتا ہے۔
- پروگرام لکھنے کے لئے استعمال ہونے والے الفاظ، علامات اور کوڈز کا ایک مجموعہ پروگرامنگ لینگوئج کہلاتا ہے۔
- لو لیول لینگوئج کمپیوٹر ہارڈویئر کے زیادہ قریب ہیں۔
- کمپیوٹر زبان کی وہ قسم جس میں ہم ہدایات کو بائنری شکل میں لکھتے ہیں اسے مشینی زبان (Machine Language) کہا جاتا ہے۔
- اسمبلی لینگوئج بھی لو لیول لینگوئج ہے۔
- ہائی لیول لینگوئج انگریزی الفاظ پر مبنی زبانیں ہیں۔
- انٹرپرٹر ہائی لیول پروگرامنگ لینگوئج کا مترجم ہوتا ہے۔
- کمپائلر سسٹم سافٹ ویئر ہے جو ہائی لیول لینگوئج کے سورس کوڈ کو مشین لینگوج آبجیکٹ کوڈ میں ترجمہ کرتا ہے۔

NOT FOR SALE

- انٹیگریٹڈ ڈویلپمنٹ انوائرمنٹ (IDE) پروگراموں کو تیار کرنے، مرتب کرنے اور چلانے کے لئے پروگرامنگ لینگویج ڈویلپمنٹ ٹول ہے۔
- مختلف کاموں کو انجام دینے کے لئے ہیڈر فائلیں سٹینڈرڈ لائبریری کے فنکشن کا مجموعہ ہیں۔
- وہ الفاظ جو کسی بھی زبان کے ذریعہ خصوصی مقاصد کے لئے استعمال ہوتے ہیں انہیں Reserved Words کہتے ہیں۔
- پروگرام کے حقیقی آغاز سے پہلے کمپائلر کو جو ہدایات دی جاتی ہیں انھیں پری پروسیسر ڈائرکٹیوز کہتے ہیں۔
- ایک ایسی مقدار جس کی قیمت پروگرام کے عمل کے دوران تبدیل ہوسکتی ہے اسے متغیر / ویری ایبل کہا جاتا ہے۔
- کانسٹنٹ وہ مقدار ہے جسے پروگرام کے چلنے کے دوران تبدیل نہیں کیا جاسکتا۔
- پرواگرم کے چلنے کے دوران کسی ایک ڈیٹا ٹائپ کو دوسرے ڈیٹا ٹائپ میں تبدیل کرنے کے عمل ٹائپ کاسٹنگ کے نام سے جانا جاتا ہے۔

مشق

سوال نمبر 1۔ موزوں جواب کا انتخاب کریں۔

i. ____________ ہدایات (Instructions) کا ایسا مجموعہ ہے جو کمپیوٹر کو بتاسکے کہ کیا کرنا اور کیسے کرنا ہے۔

(الف) کمپائلر (ب) پروگرام (ج) کمنٹس (د) ہارڈویئر

ii. لو لیول لینگویج ____________ کے قریب تر ہیں۔

(الف) انسانی زبان (ب) مشینی زبان (ج) کمپیوٹر ہارڈویئر (د) کمپیوٹر سافٹ ویئر

iii. ذیل میں کونسا C پروگرامنگ کا جزو نہیں ہے۔

(الف) کمپائلر (ب) ایڈیٹر (ج) لنکر (د) کنورٹر

iv. کمپائلر کے آؤٹ پٹ کو ____________ کہا جاتا ہے۔

(الف) پروگرام (ب) سورس کوڈ (ج) لنکڈ کوڈ (د) آبجیکٹ کوڈ

v. کونسی لینگویج میں ہدایات (statements) کو بائنری کوڈ میں لکھا جاتا ہے؟

(الف) مشین لینگویج (ب) اسمبلی لینگویج

(ج) ہائی لیول لینگویج (د) ان میں سے کوئی نہیں

vi. کونسی زبان ہائی لیول لینگویج نہیں ہے؟

(الف) BASIC (ب) اسمبلی لینگویج (ج) Pascal (د) FORTRAN

NOT FOR SALE

vii. ان میں کون سا کمپائلر ڈائریکٹیو ہے۔

(الف) #include<stdio.h> (ب) Using namespace std;

(ج) int main( ) (د) یہ تمام

viii. سورس کوڈ کو آبجیکٹ کوڈ میں تبدیل کرنے کے عمل کو ________ کہا جاتا ہے۔

(الف) Compilation (ب) Executing

(ج) Linking (د) Saving

ix. ذیل میں کس کی قیمت پروگرام چلنے کے دوران تبدیل ہو سکتی ہے؟

(الف) کانسٹنٹ (ب) ویری ایبل (ج) نام (د) علامت

x. پروگرام کا آغاز ________ فنکشن سے ہوتا ہے۔

(الف) const( ) (ب) main( ) (ج) name( ) (د) start( )

سوال نمبر2۔ ذیل میں دیئے گئے مختصر سوالات کے جوابات تحریر کریں۔

i. پروگرام سائنٹکس اور سیمنٹک میں فرق بیان کریں۔

ii. ہائی لیول لینگوئج اور لو لیول لینگوئج میں فرق بیان کریں۔

iii. IDE کیا ہے؟

iv. OOP کیا ہے؟

v. ہائی لیول لینگوئج کی خصوصیات بیان کریں۔

vi. کمپائلر اور انٹرپرٹر میں فرق بیان کریں۔

vii. ہیڈر فائل کیا ہے؟

viii. سورس پروگرام اور آبجیکٹ پروگرام میں کیا فرق ہے؟

ix. کلیدی الفاظ یا ریزروڈ ورڈز کیا ہوتے ہیں؟

x. ویری ایبل کے نام کے لیے قواعد تحریر کریں۔

xi. *const* کس مقصد کے لیے استعمال ہوتا ہے؟

سوال نمبر3۔ پروگرامنگ لینگوئج کیا ہے؟ اس کی مختلف اقسام کے بارے میں تفصیل سے لکھیں۔

سوال نمبر4۔ ہائی لیول لینگوئج کیا ہے؟ کوئی سی پانچ ہائی لیول لینگوئج کے بارے میں تفصیل سے لکھیں۔

سوال نمبر5۔ C پروگرام کا بنیادی ڈھانچہ کیا ہے؟ پری پروسیسر ڈائریکٹیو کی مختلف اقسام کی تفصیل بیان کریں۔

سوال نمبر6۔ کانسٹنٹ اور ویری ایبل کیا ہیں؟ ان میں فرق کو مثالوں سے واضح کریں۔

سوال نمبر7۔ C پروگرامنگ لینگوئج میں استعمال ہونے والی ڈیٹا ٹائپ کو مثالوں سے تفصیلاً بیان کریں۔

سوال نمبر8۔ Type Casting کیا ہے؟ مثالوں سے واضح کریں۔

سوال نمبر9۔ C لینگوئج میں ویری ایبل کیسے بنائے جاتے ہیں اور ان کی ابتدائی قیمت کیسے رکھی جاتی ہے؟ مثالیں دیں۔

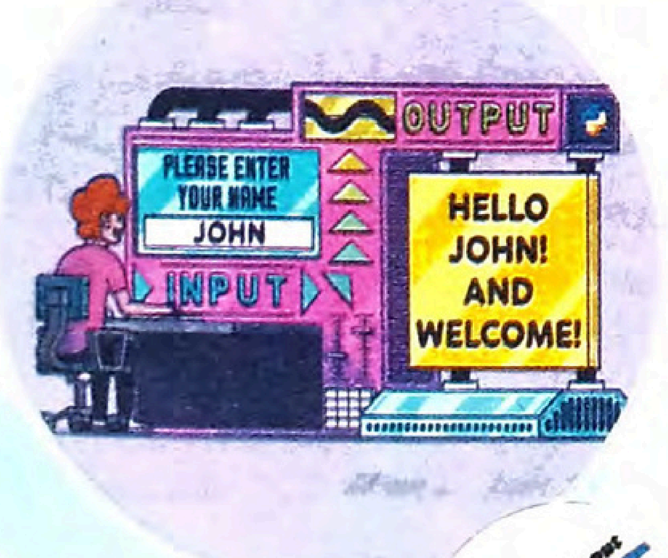
NOT FOR SALE

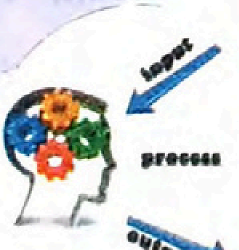
باب نمبر
3

اِن پُٹ اور آؤٹ پُٹ کا عمل

اس باب کی تکمیل کے بعد طلبہ اس قابل ہو جائیں گے کہ وہ:

- آؤٹ پُٹ فنکشنز: ()printf، ()puts اور ()cout کا استعمال کر سکیں۔
- اِن پُٹ فنکشنز: ()gets، ()getchar، ()getch، ()getche، ()scanf اور cin کو استعمال کر سکیں۔
- اسٹیٹمنٹ ٹرمنیٹر یعنی (semicolon) استعمال کر سکیں۔
- فارمیٹ سپیسیفائر کی تعریف کر سکیں۔
- اسکیپ سیکونس (Escape Sequence) کی تعریف کر سکیں اور پروگرام میں ان کے استعمال کو جان سکیں۔
- حسابی عوامل (Arithmetic Operators) کی تعریف کر سکیں اور پروگرام میں ان کے استعمال کو جان سکیں۔
- اسائنمنٹ آپریٹرز (Assignment Operators) کے استعمال کو جان سکیں۔
- نسبتی آپریٹرز (Relational Operators) کی تعریف کر سکیں اور پروگرام میں ان کے استعمال کو جان سکیں۔
- منطقی آپریٹرز (Logical Operators) کی تعریف کر سکیں اور پروگرام میں ان کے استعمال کو جان سکیں۔
- اسائنمنٹ آپریٹرز اور مساوی آپریٹر کے مابین فرق کر سکیں۔
- یونری (Unary) اور بائنری (Binary) آپریٹرز میں فرق بیان کر سکیں۔
- ثلاثی (Ternary) آپریٹر کی تعریف کر سکیں۔
- آپریٹرز کی آپس میں فوقیت کی تعریف کر سکیں اور اسے سمجھ سکیں۔





NOT FOR SALE

3.1 اِن پُٹ اور آؤٹ پُٹ فنکشنز

C پروگرام تین اہم کام انجام دیتا ہے یعنی وہ ڈیٹا اِن پُٹ کے طور پر حاصل کرتا ہے، ڈیٹا پر کارروائی کرتا ہے اور آؤٹ پُٹ تیار کرتا ہے۔ اِن پُٹ سے مراد ڈیٹا کو وصول کرنا ہے جبکہ آؤٹ پُٹ سے مراد ڈیٹا کا اظہار ہے۔ عام طور پر اِن پُٹ کی بورڈ سے وصول کیا جاتا ہے اور آؤٹ پُٹ اسکرین پر ظاہر ہوتا ہے۔ ہر پروگرامنگ لینگویج میں اِن پُٹ / آؤٹ پُٹ آپریشن سب سے اہم کام ہوتا ہے۔ C لینگویج بہت سے اِن پُٹ / آؤٹ پُٹ فنکشن پیش کرتی ہے۔ ڈیٹا حاصل کرنے اور پھر اسے ویری ایبل کو تفویض کرنے کے لئے استعمال ہونے والے فنکشن کو اِن پُٹ فنکشن کہتے ہیں۔ وہ فنکشن جو کمپیوٹر میموری سے ڈیٹا وصول کرنے اور پھر اسے آؤٹ پُٹ ڈیوائسز کو بھیجنے کے لئے استعمال ہوتے ہیں انہیں آؤٹ پُٹ فنکشن کہا جاتا ہے۔

3.1.1 آؤٹ پُٹ فنکشن (Output Functions)

C لینگویج درج ذیل آؤٹ پُٹ فنکشن کا استعمال کرتی ہے، جیسے:

- printf()
- puts()

جبکہ ++C لینگویج آؤٹ پُٹ کے لئے درج ذیل کو استعمال کرتی ہے۔

- cout

(a) printf( ) فنکشن

C میں سب سے عام آؤٹ پُٹ فنکشن میں سے ایک printf( ) فنکشن ہے۔ printf( ) اسکرین پر آؤٹ پُٹ ظاہر کرنے کے لئے استعمال ہوتا ہے۔

یہ پیغامات، ویری ایبل کی قیمت اور ریاضی کی عبارتوں (Arithmetic Expressions) کے جوابات کو پرنٹ کرنے کے لئے استعمال ہوتا ہے۔ printf میں "f" فارمیٹڈ کی مختصر شکل ہے اور اشارہ کرتا ہے کہ اسکرین پر چھپی ہوئی عبارت کو فارمیٹ کیا جائے گا۔ فنکشن printf کو print-eff کے نام سے ظاہر کیا جاتا ہے۔ اس کی وضاحت سٹنڈرڈ ہیڈر فائل stdio.h میں کی گئی ہے۔ لہٰذا printf() کو استعمال کرنے کے لیے ہمیں اپنے پروگرام میں ہیڈر فائل stdio.h شامل کرنا چاہئے۔

printf کا سائنٹکس

```
printf("String");
```

یا

```
printf("format string", var1, var2, var3, ..., varN);
```

format string یہ بتاتا ہے کہ کتنے آرگومنٹس استعمال کئے گئے ہیں اور ان کی اقسام کیا ہیں۔ آرگومنٹ var1، var2، ...، varN وہ ویری ایبل ہیں جن کی format string میں دیئے گئے سپیسیفائر (specifier) کے مطابق تشکیل اور پرنٹ کی جاتی ہے۔ قیمت ڈیٹا ٹائپ کے مطابق استعمال کیا جاتا ہے۔ آرگومنٹس کی تعداد، ترتیب اور ٹائپ کا format specifier سے مماثل ہونا ضروری ہوتا ہے۔

پروگرام 3.1 حروف کے ایک سٹرنگ کو اسکرین پر پرنٹ کرنے کا پروگرام

```
# include <stdio.h>
void main()
{
printf("Welcome to Programming Environment\n");
printf("This is my first program in C language\n");
}
```

آؤٹ پٹ

```
Welcome to Programming Environment
This is my first program in C language
```

پروگرام 3.2 printf( ) کی مدد سے 1 2 3 4 پرنٹ کرنے کا پروگرام

```
# include <stdio.h>
void main ( )
{
int a=1,b=2,c=3,d=4;
printf("%d\t%d\t%d\t%d",a,b,c,d);
}
```

آؤٹ پٹ

```
1    2    3    4
```

(b) puts( ) فنکشن

اس کا استعمال اسکرین پر سٹرنگ کو پرنٹ کرنے کے لئے کیا جاتا ہے۔ اس کی تعریف سٹینڈرڈ ہیڈر فائل stdio.h میں موجود ہوتی ہے۔ اس کا سائنٹکس یہ ہے:

puts(string);

- *string* کوئی سٹرنگ کانسٹنٹ بھی ہو سکتا ہے یا سٹرنگ ویری ایبل بھی۔
- سٹرنگ کانسٹنٹ کی صورت میں، سٹرنگ کو دوہرے کاموں (Double Quotes) میں بند کیا جائے گا۔
- سٹرنگ ویری ایبل کی صورت میں یہ دوہرے کاموں (Double Quotes) کے بغیر لکھا جائے گا۔

پروگرام 3.3 puts( ) فنکشن کی مدد سے سٹرنگ کو اسکرین پر دکھانا۔

```
#include<string.h>
void main()
{
puts("My name is Khalid ") ;

}
```

آؤٹ پٹ

```
My name is Khalid
```


NOT FOR SALE

(c) cout آبجیکٹ

C++ پروگرام میں آؤٹ پُٹ حاصل کرنے کے لیے cout کا استعمال کیا جاتا ہے۔ اسے سی آؤٹ پڑھا جاتا ہے اور اس کا مطلب ہے کنسول آؤٹ پُٹ۔ اس کو >>(Insertion Operator) کے ساتھ استعمال کیا جاتا ہے، جسے ڈبل اینگل بریکٹ بھی کہا جاتا ہے جیسا کہ شکل نمبر 1.3 میں دکھایا گیا ہے۔ واضح رہے کہ >>cout کو استعمال کرنے کے لیے ہمیں پروگرام میں ہیڈر فائل iostream کو ;using namespace std کے ساتھ لازمی طور پر کوڈ کی پہلی لائن کے طور پر شامل کرنا پڑتا ہے۔

اس کا سائنٹکس یہ ہے:

```
cout<<Variable/Constant/Expression;
```

جبکہ

Variable: کوئی بھی ویری ایبل ہو سکتا ہے جس کی قیمت اسکرین پر دکھانا مقصود ہو۔

Constant: مستقل یا کانسٹنٹ کی قیمت بھی اسکرین پر پرنٹ کی جا سکتی ہے۔

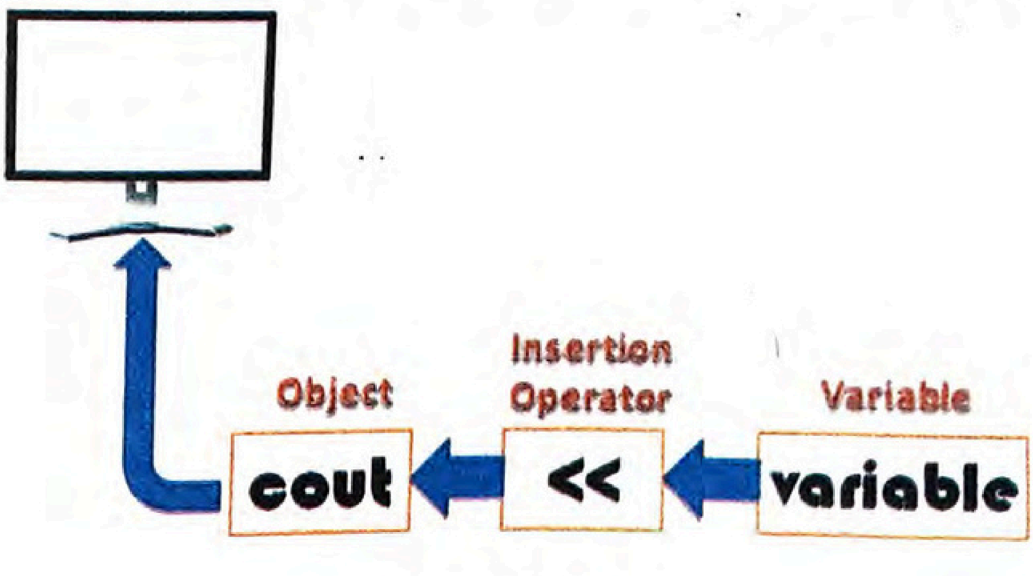
Expression: ریاضی کی کسی عبارت سے حاصل شدہ نتیجہ کو بھی اسکرین پر پرنٹ کیا جا سکتا ہے۔

پروگرام میں ڈبل اینگل بریکٹ کے بعد لکھا ہوا پیغام بطور آؤٹ پُٹ کے ظاہر ہو گا۔

مثال کے طور پر:

```
cout<<"I love my country";
```

مذکورہ بالا لائن اسکرین پر I love my country دکھائے گی۔ سٹرنگ کانسٹنٹ ہمیشہ دوہرے کاموں میں بند ہوتا ہے۔



شکل نمبر 3.1 cout آبجیکٹ

پروگرام 3.4 دواعداد کے حاصل جمع اور حاصل ضرب کو دکھانے کا پروگرام۔

```
using namespace std;
#include<iostream>
int main()
{
    int a=15, b=22, c;
    cout<<"\nThis program will add and multiply two numbers"<<endl;
    c=a+b;
    cout<<"\nThe sum of "<<a<<" and "<<b<<" is "<<c<<endl;
    cout<<"\n The product of "<<a<<" and "<<b<<" is "<<a*b<<endl;
    return 0;
}
```

آؤٹ پٹ

```
This program will add and multiply two numbers
The sum of 15 and 22 is 37
The product of 15 and 22 is 330
```

یہاں ہم نے endl (اینڈ ایل کے طور پر بولا جاتا ہے) کا استعمال کیا ہے جو بالکل اسکیپ سیکونس n\ کی طرح کام کرتا ہے۔ یہ لائن کو ختم کر دیتا ہے اور کرسر کو اگلی لائن کے آغاز میں لے جاتا ہے۔

3.1.2 اِن پُٹ فنکشنز (Input Functions)

C پروگرامنگ لینگوئج اِن پٹ آپریشنز انجام دینے کے لئے بلٹ ان فنکشنز مہیا کرتی ہے۔ اِن پُٹ آپریشنز پروگرام چلاتے وقت کی بورڈ سے یوزر سے ویلیو (اِن پُٹ) لینے کے لئے استعمال ہوتے ہیں۔ C لینگوئج مندرجہ ذیل بنے بنائے اِن پُٹ فنکشن فراہم کرتی ہے۔

- scanf()
- getch()
- getche()
- getchar()
- gets()

جبکہ C ++ لینگوئج اِن پُٹ کے لئے درج ذیل کو استعمال کرتی ہے:

- cin

(a) scanf() فنکشن

scanf() فنکشن کا استعمال کی بورڈ سے مختلف ڈیٹا اقسام کے متعدد ڈیٹا ویلیوز کو پڑھنے کے لئے کیا جاتا ہے۔ scanf() فنکشن کی وجہ سے پروگرام آگے نہیں بڑھے گا اور نتائج نہیں دکھائے گا جب تک آپ ویری ایبل کے لیے قیمتیں ان پٹ نہیں کرتے۔ scanf() فنکشن کی عمومی شکل درج ذیل ہے۔

scanf ("format specifiers", &var1, &var2, &var3,......... ;).

جبکہ،

"format specifier" فارمیٹ تفصیلات کے ذریعہ ویری ایبل کی شکل متعین کرتا ہے۔ یہ دہرے کاموں میں لکھا جاتا ہے۔

&var1, &var2, & var3,...... ویری ایبل کی فہرست ہے۔ جس میں ایک ویری ایبل کو دوسرے ویری ایبل سے کاما کی مدد سے علیحدہ

NOT FOR SALE

کیا گیا ہے۔ یہ وہ ویری ایبل ہیں جنہیں کوئی قیمت دینی ہے۔ & کی علامت کو scanf کے ساتھ استعمال کیا جاتا ہے جو میموری ایڈریس کو ظاہر کرتی ہے۔

مثال کے طور پر، ڈیٹا ٹائپ انٹیجر کے دو عددی ویری ایبل، n1 اور n2 میں قیمتیں درج کرنے کے لئے scanf ( ) فنکشن ایسے لکھا جائے گا:

scanf("%d,%d", &a, &b);

%d کا فارمیٹ سپیسیفائیر انٹیجر ویری ایبل کے لیے استعمال ہوتا ہے۔

پروگرام 3.5 طالب علم کے رول نمبر، حاصل کردہ نمبر، کل نمبر اِن پُٹ کروا کے فیصد معلوم کرنے کے لیے پروگرام۔

```
#include<stdio.h>
void main()
{
int rn,tm,om;
float pm;
printf("\nEnter Roll number = ");
scanf("%d",&rn);
printf("\nEnter Total marks = ");
scanf("%d",&tm);
printf("\nEnter Obtained marks = ");
scanf("%d",&om);
pm=om/tm*100;
printf("\nPercentage marks = %f",pm);
}
```

آؤٹ پُٹ

```
Enter Roll number = 10
Enter Total marks = 800
Enter Obtained marks = 650
Percentage marks = 81.25
```

(b) getche () ،getch () اور getchar () فنکشنز

getch() ، getche() اور getchar() اِن پُٹ فنکشنز ہیں جو کی بورڈ سے ایک حرف (Character) کی اِن پُٹ کے لیے استعمال ہوتے ہیں۔ ان فنکشنز میں کسی بھی آرگومنٹ کی ضرورت نہیں ہے۔ ان فنکشنز کے ساتھ صرف خالی بریکٹ ہی استعمال کی جاتی ہے۔

getch( )

getch () کا فنکشن کی بورڈ سے ایک کریکٹر اِن پُٹ کرنے کے لیے استعمال ہوتا ہے لیکن اس فنکشن سے جو کریکٹر اِن پُٹ کیا جاتا ہے وہ سکرین پر دکھائے بغیر اِن پُٹ ہوتا ہے اور ویری ایبل میں اسٹور ہو جاتا ہے۔

NOT FOR SALE

پروگرام 3.6

```
#include<stdio.h>
void main()
{
    char ch;
    printf("Enter Char:");
    cch=getch();
    printf("\n");
    putchar(ch);
}
```

آؤٹ پٹ

```
Enter Char:
j
```

getche()

getche ( ) فنکشن بھی کی بورڈ سے ایک کیریکٹر پڑھتا ہے لیکن echo کے ساتھ۔ یعنی جو حرف اِن پُٹ کیا جاتا ہے وہ اسکرین پر نظر بھی آتا ہے۔

پروگرام 3.7

```
#include<stdio.h>
void main()
{
    char ch;
    printf("Enter Character:");
    ch=getche();
    printf("\n");
    putchar(ch);
}
```

آؤٹ پٹ

```
Enter Char: k
k
```

getchar()

getchar () فنکشن کی بورڈ سے ایک حرف کی اِن پُٹ لیتا ہے لیکن اِن پُٹ کرنے کے بعد Enter key ضرور دبانی پڑتی ہے۔

پروگرام 3.8

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char ch;
    printf( " Enter Char: ");
```

آؤٹ پٹ

```
Enter Char: h
h
```


NOT FOR SALE

```
        ch= getchar();
        printf("\n");
        putchar(ch);
}
```

() gets فنکشن

() gets فنکشن اس قابل ہے کہ یہ کریکٹر سٹرنگ کے درمیان خالی جگہ (White Spaces) کو بھی بطور سٹرنگ لیتا ہے۔ gets فنکشن اسٹرنگ کی ایک لائن یا حرفوں کی ترتیب کو پڑھتا ہے جب تک کہ یوزر Enter key نہ دبائے۔ اس کی عمومی شکل درج ذیل ہے۔

سائنٹکس:

***gets*(Array_Name);**

پروگرام 3.9

```
#include<stdio.h>
void main()
{
        char name[20];          // Here [20] specifies the size of the string
        printf("Enter your Name: ");
        gets(name);
        puts(name);
}
```

آؤٹ پُٹ

```
Enter your Name: Muhammad Khalid
Muhammad Khalid
```

()gets فنکشن خالی جگہ (White space) کے حروف کے ساتھ ایک سٹرنگ لیتا ہے۔ اس صورت میں، پروگرام چلاتے وقت کی بورڈ سے اسٹرنگ اِن پُٹ کی جائے گی اور جب یوزر Enter بٹن دبائے گا تب اِن پُٹ مکمل ہو گی۔ ()scanf اور ()gets کے فنکشنز میں یہ فرق ہے کہ ()scanf تمام حروف کو اس وقت تک پڑھتا ہے جب تک کہ space نہیں آ جاتا یعنی ٹیب یا نئی لائن کا سامنا نہ کرنا پڑے، جب کہ ()gets خالی جگہوں اور ٹیب سمیت سب کچھ پڑھے گا جب تک کہ یوزر Enter key نہیں دباتا۔

پروگرام 3.10 کسی شخص کا نام اور عمر (سالوں میں) حاصل کرنے کے لئے ایک پروگرام لکھیں۔ مہینوں میں عمر کا حساب لگائیں اور مہینوں میں اس شخص کی عمر پرنٹ کریں۔

```
#include<stdio.h>
void main()
{
        int age,age_mon;
        char name[40];
        printf("Enter your Name: ");
        gets(name);
        printf("\nEnter your Age: ");
```

آؤٹ پُٹ

```
Enter your Name: Muhammad Anwar
Enter your Age: 16
Age in months = 12
```

NOT FOR SALE

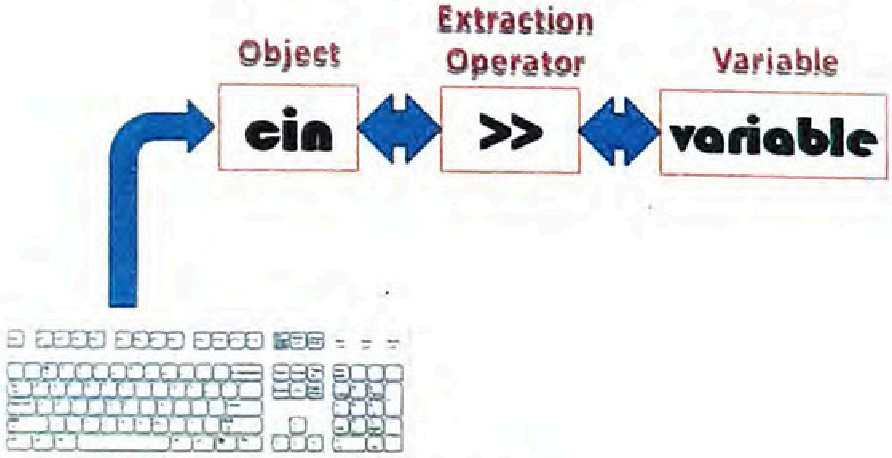
```
scanf("%d",&age);
age_mon=age*12;
printf("\nAge in months = %d ",age_mon);
}
```

(c) cin

کسی C ++ پروگرام میں کی بورڈ سے رن ٹائم پر اِن پٹ لینے کے لیے cin کنسول اِن پٹ کا استعمال کیا جا سکتا ہے۔cin کو سی ان پڑھا جاتا ہے۔<< cin آپریٹر شامل کر کے سٹینڈرڈ اِن پٹ کو حاصل کرتا ہے۔ آپریٹر کے بعد ویری ایبل ہونا ضروری ہے جو اِن پٹ ڈیٹا کو سٹور کرنے کے لئے استعمال ہوتا ہے جیسا کہ شکل نمبر 2.3 میں دکھایا گیا ہے۔ یہ پروگرام کو اس وقت تک روکے رکھتا ہے جب تک کی بورڈ سے اِن پٹ درج نہیں ہو جاتی۔cin کو عددی قیمت اور کیریکٹر اِن پٹ دونوں کے لیے استعمال کیا جا سکتا ہے۔cin کو پروگرام میں استعمال کرنے کے لیے ہمیں پروگرام میں ہیڈر فائل iostream کو ;using namespace std کے ساتھ لازمی طور پر کوڈ کی پہلی لائن کے طور پر شامل کرنا پڑتا ہے۔

اس کا سا ئنٹیکس درج ذیل ہے۔

cin>>var1>> var2>>.......;



شکل نمبر 3.2 cin کا استعمال

مثال کے طور پر:

cin>>k;

یہ کی بورڈ سے ایک قیمت حاصل کرے گا اور اسے ویری ایبل k میں محفوظ کرے گا۔

cin>>a>>b>>c;

یہ کی بورڈ سے تین ویلیوز حاصل کرے گا اور اسے ویری ایبل b،a اور c میں محفوظ کرے گا۔

NOT FOR SALE

ایک اہم بات یہ نوٹ کی جائے کہ، یہ لازمی ہے کہ کی بورڈ سے درج کی گئی قیمت ویری ایبل کی ڈیٹا ٹائپ سے مماثل ہونی چاہیئے۔ scanf() فنکشن کی طرح، cin میں ویری ایبل کے نام کے ساتھ & آپریٹر کی ضرورت نہیں ہوتی ہے۔

پروگرام 3.11

فارن ہائیٹ درجہ حرارت کو سینٹی گریڈ درجہ حرارت میں تبدیل کرنے کے لئے ایک پروگرام لکھیں۔ فارن ہائیٹ درجہ حرارت رن ٹائم پر کی بورڈ سے اِن پُٹ ہونا چاہیئے۔

```
using namespace std;
#include<iostream>
int main( )
{
  float ftemp, ctemp;
  cout<<"\n Please input a value for Fahrenheit temperature: ";
  cin>>ftemp;
  ctemp = (ftemp-32) * 5 / 9;
  cout<<"\nCelsius equivalent of Fahrenheit :"<< ftemp<<" is "<<ctemp;
  return 0;
}
```

آؤٹ پٹ

```
Please input a value for Fahrenheit temperature: 98.6
Celsius equivalent of Fahrenheit : 98.6 is 37
```

3.1.3 اسٹیٹمنٹ ٹرمینیٹر (Statement Terminator)

C کے ہر پروگرام میں ایک اسٹیٹمنٹ کو دوسری اسٹیٹمنٹ سے جدا کرنے کے لیے اسٹیٹمنٹ کے اختتام پر سیمی کالن (;) کا استعمال کیا جاتا ہے۔ یہ سیمی کالن کمپائلر کو بتاتا ہے کہ ایک اسٹیٹمنٹ اس جگہ ختم ہو گئی ہے۔ اگر ایک اسٹیٹمنٹ کو سیمی کالن سے ختم نہیں کیا جاتا تو، C کا کمپائلر کمپائل کرنے کے عمل کے دوران Error Message دے گا اور پروگرام کمپائل نہیں کیا جائے گا۔

مثال کے طور پر،

```
printf("Hello, World! \n"); // Semicolon is used to show end of statement
```

3.1.4 فارمیٹ سپیسیفائرز (Format Specifiers)

فارمیٹ سپیسیفائر اِن پُٹ اور آؤٹ پُٹ مقاصد کے لئے C میں استعمال ہوتا ہے۔ اس کے استعمال کرنے سے کمپائلر یہ سمجھ سکتا ہے کہ scanf() فنکشن اور printf() فنکشن میں ویری ایبل کس قسم کا ڈیٹا لیتا یا پرنٹ کرتا ہے۔

ذیل میں فارمیٹ سپیسیفائرز کی ایک فہرست دی گئی ہے۔

Format specifier	Purpose
%d or %i	Signed integer
%ld	long integer
%f	floating-point (decimal notation)
%g	floating-point (exponential notation)
%e	floating-point (Exponential notation)
%c	single character
%s	String

یہ بنیادی فارمیٹ سپیسیفائر ہیں۔ ہم فارمیٹ سپیسیفائر کے ساتھ کچھ اور حصے بھی شامل کر سکتے ہیں۔

ایک مائنس علامت (-) بائیں سیدھ (left alignment) کو بتاتا ہے۔

% کے بعد کا نمبر فیلڈ کی کم سے کم چوڑائی متعین کرتا ہے۔ اگر سٹرنگ چوڑائی سے کم ہے تو، یہ خالی جگہوں سے پُر ہو گا۔

ایک پیریڈ (. فیلڈ) کی چوڑائی کو پریسین سے الگ کرنے کے لئے استعمال کیا جاتا ہے۔

انٹیجر فارمیٹ سپیسیفائر (%d, %ld and %i)

d% کا فارمیٹ سپیسیفائر انٹیجر اور ld% کا فارمیٹ سپیسیفائر لانگ انٹیجر کو پرنٹ کرنے کے لیے استعمال کیا جاتا ہے۔

پروگرام 3.12 انٹیجر فارمیٹ سپیسیفائر کے استعمال کو ظاہر کرتا ہے۔

```
#include<stdio.h>
void main()
{
	int a, b;
	a = 14;
	b = a / 2;
	printf("a=%d\n",a);		// Here %d will be replaced with the value of a
	printf("b=%d\n",b);		// Here %d will be replaced with the value of b
}
```

آؤٹ پٹ

```
a=14
b=7
```

فلوٹنگ پوائنٹ فارمیٹ سپیسیفائر (%f, %e and %g)

f% کا فارمیٹ سپیسیفائر فلوٹنگ پوائنٹ اعداد کو نقطہ اعشاریہ کے ساتھ دکھاتا ہے۔ فارمیٹ سپیسیفائر e% فلوٹنگ پوائنٹ نمبر کو ایکسپونینشل نوٹیشن (Exponential Notation) میں اِن پٹ کرنے اور دکھانے کے لیے استعمال کیا جاتا ہے۔

g% کا فارمیٹ سپیسیفائر بھی e% کی طرح ہی کام کرتا ہے۔

NOT FOR SALE

پروگرام 3.13 میں فلوٹ فارمیٹ سپیسیفائر کا استعمال دکھایا گیا ہے۔

```
#include<stdio.h>
void main()
#include<conio.h>
void main()
{
int a;
float b;
a = 15;
b = a / 2;
printf("a=%d\n",a);          // Here %d will be replaced with the value of a
printf("b=%f\n",b);          // Here %f will be replaced with the value of b
printf("b=%e\n",b);          // Here %e will be replaced with the value of b //in exponential
}
```

آؤٹ پُٹ

```
a=15
b=7,500000
b=7.5e+0
```

کریکٹر فارمیٹ سپیسیفائر %c (Character Format Specifier)

کریکٹر فارمیٹ سپیسیفائر ایک وقت میں صرف ایک کریکٹر کو اِن پُٹ کرنے کے لیے استعمال کیا جاتا ہے۔ پروگرام 3.14 میں %c کے استعمال کو دکھایا گیا ہے۔

پروگرام 3.14

```
#include<stdio.h>
void main()
{
    char ch='A';
    printf("Character=%c\n",ch);     // Here %c will be replaced with the value of ch
}
```

آؤٹ پُٹ

```
Character=A
```

3.1.5 اسکیپ سیکونسز (Escape Sequences)

اسکیپ سیکونسز ایسے خاص حروف ہیں جو پرنٹ تو نہیں ہوتے لیکن آؤٹ پُٹ کی شکل میں ترمیم کرنے کے لئے استعمال ہوتے ہیں۔ یہ حرفوں کا مجموعہ ہیں جو بیک سلیش \ کے ساتھ شروع ہوتے ہیں، اس کے بعد حرف یا خاص علامت ہوتی ہے۔ آؤٹ پُٹ کو فارمیٹ کرنے کے لئے اسکیپ سیکونسز کا استعمال کیا جاتا ہے۔ یہ اکیلے یا سٹرنگ کانسٹنٹ کے درمیان استعمال ہوتے ہیں۔ انہیں ہمیشہ دوہرے کاموں میں بند کیا جاتا ہے۔ آؤٹ پُٹ سٹرنگ میں کہیں بھی اسکیپ سیکونسز استعمال کی جاسکتی ہے۔

NOT FOR SALE

عام طور پر استعمال ہونے والے اسکیپ سیکونسز یہ ہیں:

اسکیپ سیکونسز	مقصد	مثال	آؤٹ پُٹ
\n	نئی لائن (پرنٹنگ کنٹرول کو اگلی لائن کے آغاز پر منتقل کرتا ہے)	printf("Hello\nWorld");	Hello World
\t	افقی ٹیب (کرسر کو ایک ٹیب آگے بڑھاتی ہے۔)	printf("Hello\tWorld");	Hello World
\a	بیل (الرٹ) کمپیوٹر کے اندرونی سپیکر میں بیپ کی آواز پیدا کرتی ہے۔	printf("Hello \a how are you?");	Beep after Hello and before how are you?
\b	Backspace (پرنٹ کے بعد کرسر اپنی جگہ واپس منتقل کرنے کے لیے استعمال کیا جاتا ہے)	printf("Hello\bWorld ");	"Hello" is printed but \b deletes 'o'. Then "World" is printed. So "HellWorld" is displayed on the screen.
\f	فارم فیڈ (خالی فارم فیڈ کرنے کے بعد آؤٹ پٹ کو پرنٹر پر پرنٹ کرنے کے لیے استعمال کیا جاتا ہے)	printf("Hello\fWorld");	HelloWorld After printing "Hello", the computer will include a blank paper and then print the remaining output. It is used during printing.
\'	سنگل کوٹ (سنگل کوٹ ' کو پرنٹ کرنے کے لیے استعمال کیا جاتا ہے)	printf("\'Hello World\'");	'Hello World'
\"	دوہرا کوٹ (دوہرے کوٹ " کو پرنٹ کرنے کے لیے استعمال کیا جاتا ہے۔)	printf("\"Hello World\"");	"Hello World"
\\	Backslash (بیک سلیش پرنٹ کرنے کے لیے استعمال کیا جاتا ہے)	printf(\\Hello World\\);	\Hello World\

3.2 آپریٹرز (Operators)

آپریٹرز خصوصی علامتیں ہیں جو کمپائلر کے ذریعے ڈیٹا پر مختلف قسم کے آپریشن دیتی ہیں۔ C لینگویج میں استعمال ہونے والے بہت سے آپریٹرز ہیں۔ ان میں حسابی(Arithmetic) آپریٹرز، اسائنمنٹ (Assignment) آپریٹرز، ریلیشنل (Relational) آپریٹرز اور لاجیکل (Logical) آپریٹرز شامل ہیں۔

NOT FOR SALE

3.2.1 حسابی آپریٹرز (Arithmetic Operators)

حسابی آپریٹرز وہ علامات ہیں جو ڈیٹا پر ریاضی کے عوامل کے لئے استعمال ہوتی ہیں۔ C لینگوئج مختلف آپریٹرز کو مختلف حسابی عمل انجام دینے کے لئے معاونت کرتی ہے جیسے جمع، تفریق، تقسیم، ضرب وغیرہ۔ C میں تمام حسابی آپریٹرز کی فہرست مندرجہ ذیل ہے۔

آپریشن	علامت	تفصیل
جمع	+	دو قیمتوں کو جمع کرتا ہے۔
تفریق	–	ایک قیمت کو دوسری سے تفریق کرتا ہے۔
ضرب	*	دو قیمتوں کو ضرب دیتا ہے۔
تقسیم	\	ایک قیمت کو دوسری قیمت سے تقسیم کرتا ہے۔
ماڈولس	%	دو انٹیجر کی تقسیم سے حاصل شدہ باقی ماندہ قیمت معلوم کرتا ہے۔

تمام حسابی آپریٹرز صرف عددی قیمتوں کے ساتھ ہی عمل کر سکتے ہیں۔ فرض کریں ہمارے پاس A اور B دو ویری ایبلز ہیں جہاں A = 8 اور B = 3 ہے۔ A اور B پر مندرجہ ذیل حسابی عوامل استعمال ہو سکتے ہیں۔

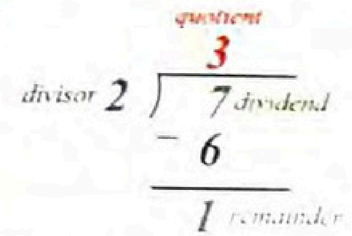
آپریٹر	مثال (int A=8, B=3)	نتیجہ
+	A+B	11
–	A–B	5
*	A*B	24
\	A/B	2 (Integer Division)
%	A%B	0

ماڈولس آپریٹرز کے حوالے سے چند اہم نکات درج ذیل ہیں۔

- C میں ماڈولس آپریٹر فیصد علامت % آپریٹر کے ذریعہ ظاہر کیا جاتا ہے۔
- ماڈولس آپریٹر کو ریمینڈر آپریٹر بھی کہا جاتا ہے۔
- ماڈولس آپریٹر صرف دو آپرینڈز کے درمیان کام کرتا ہے۔
- ماڈولس آپریٹر صرف انٹیجر پر ہی کام کرتا ہے۔
- اگر ماڈولس آپریٹر میں 0 کو کسی عدد سے تقسیم کیا جائے تو اس کا حاصل ہمیشہ 0 ہو گا۔ مثال کے طور پر 0%7 کا حاصل 0 ہو گا۔
- ایکسپریشن 5%7 میں، 5 عدد 7 سے تقسیم نہیں ہو سکتا، اس کا نتیجہ 5 ہو گا۔

NOT FOR SALE

تقسیم کے آپریٹر کا کام



"The remainder is 1 when 7 is divided by 2"

شکل نمبر 3.3 تقسیم کا عمل

ڈویژن آپریٹر کا عمل دوسرے آپریٹرز سے مختلف ہے۔ ڈویژن آپریشن کا نتیجہ ہمیشہ ایک عدد ہوتا ہے اگر شمار کنندہ (Dividend) اور مخرج (Divisor) انٹیجر ہوں۔ اس صورت میں کسری (Fractional) حصہ ختم (Truncate) کر دیا جاتا ہے۔ مثال کے طور پر، 7/2 کا نتیجہ 3 ہے، جیسا کہ شکل نمبر 3.3 میں دکھایا گیا ہے۔

3.2.2 اسائنمنٹ آپریٹر = (Assignment Operator)

اسائنمنٹ آپریٹر کو ویری ایبل (میموری لوکیشن) کو قیمت تفویض کرنے کے لئے استعمال کیا جاتا ہے۔ C میں ایک ہی اسائنمنٹ آپریٹر ہے۔ C میں یہ = علامت اپنے دائیں جانب موجود عبارت (expression) کو حل کر کے اس کی قیمت بائیں جانب موجود ویری ایبل کو تفویض کرتا ہے۔ یہ ویری ایبل کو عددی یا حرف کی قیمت تفویض کرنے کے لئے استعمال کیا جاتا ہے۔ تاہم، اس آپریٹر کے ساتھ اسٹرنگ ڈیٹا کسی ویری ایبل کو تفویض نہیں کیا جاسکتا ہے۔ اسائنمنٹ سٹیمنٹ کا فارمیٹ درج ذیل ہے۔

Variable = Character_constant/Variable/Expression;

مثال کے طور پر نیچے اسائنمنٹ ٹیبل پر غور کریں۔

Operation	Description
x = 10	Assigns 10 to variable x
y = (5 + 4) / 2	Evaluates expression (5+4)/2 and assign result to y
z = x + y	Evaluates x + y and assign result to z
5 = x	Error, you cannot re-assign a value to a constant
10 = 10	Error, you cannot re-assign a value to a constant

اسائنمنٹ آپریٹر کے دائیں جانب (RHS) کانسٹنٹ، ایکسپریشن یا ویری ایبل ہونا ضروری ہے جبکہ بائیں ہاتھ کی طرف (LHS) ایک ویری ایبل (درست میموری لوکیشن) ہونا چاہئے۔

3.2.3 کمپاؤنڈ اسائنمنٹ آپریٹرز (Compound Assignment Operator)

C کمپاؤنڈ اسائنمنٹ آپریٹرز فراہم کرتی ہے جو اسائنمنٹ آپریٹر اور حسابی آپریٹرز کا مجموعہ ہے۔ کمپاؤنڈ اسائنمنٹ آپریٹرز حسابی عوامل کو بہت آسانی سے حل کرنے کی سہولت فراہم کرتے ہیں۔

کمپاؤنڈ اسائنمنٹ آپریٹر کا عمومی سائنٹکس درج ذیل ہے۔

NOT FOR SALE

سائنٹکس:

variable op= expression;

فرض کریں ہمارے پاس A اور B کے دو ویری ایبلز ہیں تو 8 = A اور 3 = B ہے۔ A اور B پر کمپاؤنڈ اسائنمنٹ آپریٹرز استعمال کیے جاسکتے ہیں۔

آپریٹر	مثال (int A=8, B=3)	نتیجہ
+=	A+=B or A=A+B	11
−=	A−=3 or A=A−3	5
*=	A*=7 or A=A*7	56
/=	A/=B or A=A/B	2 (Integer Division)
%=	A%=5 or A=A%5	3
a=b	Value of b will be assigned to a	

پروگرام 11 ایسا پروگرام جو انٹیجر پر تمام کمپاؤنڈ آپریٹرز کو استعمال کرتا ہے۔

```
#include <stdio.h>
void main()
{
    int a;
    a=10;
    printf("value of a :%d\n",a);
    a+=5;
    printf("value of a after a+=5:%d\n",a);
    a-=5;
    printf("value of a after a-=5:%d\n",a);
    a*=2;
    printf("value of a after a*=2:%d\n",a);
    a/=2;
    printf("value of a after a/=2:%d\n",a);
    a%=2;
    printf("value of a after a%=2:%d",a);
}
```

آؤٹ پُٹ

```
Value of a. 10
Value of a after a+=5: 15
Value of a after a– =5: 10
Value of a after a*=2: 20
Value of a after a/=2: 10
Value of a after a%=2: 0
```

3.2.4 اضافہ / انکریمنٹ آپریٹر ++ (Increment Operator)

انکریمنٹ آپریٹر دوہری جمع کی علامت ++ سے ظاہر کیا جاتا ہے۔ اس کا استعمال عددی ویری ایبل کی قیمت میں 1 کا اضافہ کرنے کے لئے کیا جاتا ہے۔ انکریمنٹ آپریٹر صرف ویری ایبل کے ساتھ استعمال کیا جاسکتا ہے۔ اسے کانسٹنٹ یا ایکسپریشن کے ساتھ استعمال

NOT FOR SALE

نہیں کیا جاسکتا۔

مثال کے طور پر اگر کسی ویری ایبل x کی قیمت میں 1 کا اضافہ کرنا ہو تو اس کو ایسے لکھیں گے:

$$x = x + 1;$$

انکریمنٹ آپریٹر ++ استعمال کر کے یہی کام ایسے کرے گا:

$$x++;$$

یہ نوٹ کرنا ضروری ہے کہ x++ یا x = x + 1 یا x + = 1 ایک ہی کام کرتے ہیں۔

انکریمنٹ آپریٹر ویری ایبل سے پہلے یا بعد میں لکھا جاسکتا ہے۔ اگر یہ ویری ایبل سے پہلے لکھا گیا ہے تو، اسے پریفکس انکریمنٹ آپریٹر کہا جاتا ہے۔ اگر یہ ویری ایبل کے بعد لکھا گیا ہے، تو یہ پوسٹ فکس انکریمنٹ آپریٹر کے نام سے جانا جاتا ہے۔

پری فکس انکریمنٹ آپریٹر (Prefix Increment Operator)

پریفکس انکریمنٹ آپریٹر ویری ایبل کی موجودہ قیمت میں 1 کا اضافہ کر دیتا ہے اور پھر اس کی قیمت کو عبارت میں استعمال کیا جاتا ہے۔

مثال کے طور پر،

پروگرام 3.16

```
# include <stdio.h>
void main ( )
{
        int a,b,c,sum;
        a=1;
        b=1;
        c=3;
        \sum= a+ b+ (++c);
        printf("Sum  =  %d:\n",sum);
        printf("c=     %d",c);
}
```

آؤٹ پٹ

```
sum=6
c=4
```

مندرجہ بالا پروگرام میں c کی قیمت میں 1 کا اضافہ ہو گا اس سے پہلے کہ یہ عبارت میں استعمال ہو۔ اس لیے پروگرام کے اختتام پر sum کی قیمت 6 ہو گی اور c کی قیمت 4 ہو گی۔

پوسٹ فکس انکریمنٹ آپریٹر (Postfix Increment Operator)

جب انکریمنٹ آپریٹر کسی بھی عبارت میں پوسٹ فکس انداز میں استعمال کیا جاتا ہے، تو اس میں ویری ایبل کی قیمت میں، عبارت میں استعمال ہونے کے بعد، 1 کا اضافہ کرے گا۔

NOT FOR SALE

مثال کے طور پر، اگر مذکورہ بالا مثال میں، انکریمنٹ آپریٹر پوسٹ فکس انداز میں استعمال ہوتا ہے تو نتیجہ مختلف ہو گا۔ اسٹیٹمنٹ کچھ ایسے ہو گی:

Sum = a + b + c ++;

ایسی صورت میں، c کی موجودہ قیمت کو عبارت میں استعمال کرنے کے بعد اس میں 1 کا اضافہ کیا جائے گا۔ اس طرح عمل درآمد کے بعد، Sum 5 کے برابر ہو گی اور c کی قیمت 4 ہو گی۔

3.2.5 ڈیکریمنٹ آپریٹر -- (Decrement Operator)

ڈیکریمنٹ آپریٹر کو دوہری مائنس -- علامت سے ظاہر کیا جاتا ہے۔ یہ ایک عددی ویری ایبل کی قیمت سے 1 گھٹانے کے لئے استعمال ہوتا ہے۔ ڈیکریمنٹ آپریٹر صرف ویری ایبل کے ساتھ استعمال کیا جاسکتا ہے۔ اسے کانسٹنٹ یا ایکسپریشن کے ساتھ استعمال نہیں کیا جاسکتا۔

مثال کے طور پر، ویری ایبل x کی قیمت سے 1 کم کرنے کے لئے، ہم درج ذیل اسٹیٹمنٹ کو ایسے لکھیں گے:

--x; یا x--;

x-- یا x=x-1 یا x-=1 ایک ہی طرح کام کرتے ہیں۔

پریفکس ڈیکریمنٹ آپریٹر (Prefix Decrement Operator)

پریفکس ڈیکریمنٹ آپریٹر ویری ایبل کی موجودہ قیمت سے 1 کم کر دیتا ہے۔ پھر اس کی قیمت کو عبارت میں استعمال کیا جاتا ہے۔

مثال کے طور پر:

پروگرام 3.17

```
#include <stdio.h>
void main ()
{
	int a, b, c, sum;
	a=1;
	b=1;
	c=3;
	sum=a  + b + (--c);
	printf("sum=  %d\n",sum);
	printf("c =  %d",c);
}
```

آؤٹ پٹ

```
sum=4
c=2
```

مندرجہ بالا پروگرام میں c کی قیمت میں 1 کی کمی کر دی جائے گی اور اس کے بعد اسے عبارت میں استعمال کیا جائے گا۔ اس طرح اسٹیٹمنٹ کے چلنے کے بعد sum کی قیمت 4 اور c کی قیمت 2 ہو گی۔

پوسٹ فکس ڈیکریمنٹ آپریٹرز (Postfix Decrement Operator)

جب کسی عبارت میں ڈیکریمنٹ آپریٹر پوسٹ فکس انداز میں استعمال کیا جائے تو یہ ویری ایبل کی قیمت کو پہلے عبارت میں استعمال کرے گا اور پھر اس کے بعد اس کی قیمت میں 1 کو گھٹائے گا۔

مثال کے طور پر، اگر مذکورہ بالا مثال میں، ڈیکریمنٹ آپریٹر پوسٹ فکس انداز میں استعمال ہوتا ہے تو نتیجہ مختلف ہوگا۔ اس صورت میں اسٹیٹمنٹ کچھ ایسے لکھی جائے گی: sum = a+ b + c--;

اس صورت میں، c کی موجودہ قیمت کو عبارت میں استعمال کرنے کے بعد c کی قیمت سے 1 گھٹایا جائے گا۔ اس طرح عمل درآمد کے بعد، sum کی قیمت 5 کے برابر ہوگی اور c کی قیمت 2 ہوگی۔

3.2.6 نسبتی آپریٹرز (Relational Operators)

نسبتی یا ریلیشنل آپریٹرز کو پروگراموں میں کسی بھی قسم کی شرط (Condition) کی وضاحت کرنے کے لئے استعمال کیا جاتا ہے۔ ایک ریلیشنل آپریٹر دو قیمتوں کا موازنہ کرتا ہے۔ اس کا نتیجہ درست (True) یا غلط (False) کی صورت میں ہوتا ہے۔ ریلیشنل آپریٹرز کو بعض اوقات موازنہ (comparison) آپریٹرز کہا جاتا ہے کیونکہ وہ ٹیسٹ کنڈیشن میں استعمال ہوتے ہیں جن کا نتیجہ ہاں یا ناں میں ہوتا ہے۔ ++C / C درج ذیل ریلیشنل آپریٹرز فراہم کرتا ہے:

Operator	Symbol	Form	Operation
Greater than	>	x > y	true if x is greater than y, false otherwise
Less than	<	x < y	true if x is less than y, false otherwise
Greater than or equals	>=	x >= y	true if x is greater than or equal to y, false otherwise
Less than or equals	<=	x <= y	true if x is less than or equal to y, false otherwise
Equality	==	x == y	true if x equals y, false otherwise
Inequality	!=	x != y	true if x does not equal y, false otherwise

ریلیشنل آپریٹرز کا استعمال (Use of Relational Operators)

اگر x = 10، y = 20 اور z = 5 ہوں تو ریلیشنل آپریٹرز کا آؤٹ پٹ ذیل کے جدول میں دکھایا گیا ہے:

Form	Output Returned
x > y	False
y < z	False
x == z	False
x <= y	True
z >= y	False
y != x	True

NOT FOR SALE

3.2.7 لاجیکل یا منطقی آپریٹر (Logical Operators)

ایک سے زیادہ کنڈیشن ایکسپریشن کا جائزہ لینے کے لئے لاجیکل آپریٹرز استعمال کیے جاتے ہیں۔C میں تین لاجیکل آپریٹرز ہیں:

- AND &&
- OR ||
- NOT !

(a) اینڈ آپریٹر && (AND Operator)

AND آپریٹر کے لئے استعمال ہونے والی علامت && ہے۔ یہ دو شرائط کا موازنہ کرنے کے لئے استعمال ہوتا ہے۔ اگر دونوں شرائط کا نتیجہ True ہے تو مجموعی نتیجہ True آئے گا اور اگر شرائط میں سے کوئی بھی غلط (False) ہے تو مجموعی نتیجہ False رہے گا۔

Condition 1	Operator	Condition 2	Result
False	&&	False	False
False	&&	True	False
True	&&	False	False
True	&&	True	True

مثال: فرض کریں کہ ہمارے پاس دو ویری ایبل A = 100 اور B = 50 ہیں۔

اس (A> 10) && (B> 10) کا نتیجہ درست (True) ہے، کیونکہ دونوں ہی شرائط درست (True) ہیں۔

اس (A> 50) && (B> 50) کا نتیجہ غلط (False) ہے، کیونکہ پہلی شرط کا نتیجہ درست (True) ہے لیکن دوسری شرط کا نتیجہ غلط (False) ہے۔

(b) OR آپریٹر || (OR Operator)

OR آپریٹر کے لئے استعمال ہونے والی علامت || ہے یہ دو شرائط کا موازنہ کرنے کے لئے استعمال ہوتا ہے۔ اگر کوئی ایک شرط بھی درست (True) ہو تو اس کا نتیجہ True ہو گا۔ اگر دونوں شرائط کا نتیجہ غلط (False) ہے تو اس کا نتیجہ غلط (False) ہو گا۔

Condition 1	Operator	Condition 2	Result
False	||	False	False
False	||	True	True
True	||	False	True
True	||	True	True

مثال: فرض کریں کہ ہمارے پاس دو ویری ایبل A = 100 اور B = 50 ہیں۔

اس (A> 50) || (B> 50) کا نتیجہ درست (True) ہو گا کیونکہ ایک شرط کا نتیجہ درست (True) ہے۔ کیونکہ ایک شرط صحیح ہے۔

NOT FOR SALE

(c) NOT آپریٹر ! (NOT Operator)

NOT آپریٹر کے لئے ! علامت استعمال ہوتی ہے۔ یہ کسی شرط کے نتیجے کو اُلٹنے کے لئے استعمال ہوتی ہے۔ اگر شرط کا نتیجہ غلط (False) ہے تو اس کا نتیجہ صحیح (True) ہوگا اور اگر شرط کا نتیجہ صحیح (True) ہے تو اس کا نتیجہ غلط (False) ہوگا۔

Operator	Condition	Result
!	True	False
!	False	True

مثال: فرض کریں کہ ہمارے پاس دو ویری ایبل A = 100 اور B = 50 ہیں۔

حالت !(A == B) صحیح ہے، کیونکہ A B کے برابر نہیں ہے۔

3.2.8 اسائنمنٹ آپریٹر (=) اور مساوی آپریٹر (==) میں فرق

اسائنمنٹ آپریٹر = ایک ویری ایبل کی قیمت مقرر کرنے کے لئے استعمال کیا جاتا ہے جبکہ مساوی آپریٹر == دو ایک جیسی قسم کی قیمتوں کا موازنہ کرنے کے لئے استعمال ہوتا ہے۔

مثال کے طور پر:

```
x=11;
y=a;
c='s';
```

مذکورہ بالا اسٹیٹمنٹ میں، ویری ایبل x کی قیمت 11 مقرر کی گئی ہے، y کو ویری ایبل a کی قیمت تفویض کی گئی ہے اور c کو 's' کیریکٹر تفویض کیا گیا ہے۔

ریلیشنل آپریٹر == کے استعمال سے شرائط بنائی جاتی ہیں تاکہ کمپیوٹر کوئی بھی فیصلہ کر سکے۔

مثال کے طور پر:

```
b == 15;
x == y;
```

پہلی ایکسپریشن میں، اگر B کی قیمت 15 کے برابر ہے تو شرط کا نتیجہ صحیح ہے ورنہ یہ غلط ہے۔ دوسری حالت میں، مساوی آپریٹر کو جانچ پڑتال کے لئے استعمال کیا جاتا ہے کہ آیا x کی ویلیو y کی ویلیو کے برابر ہے، اگر ایسا ہے تو شرط کا نتیجہ صحیح ہے ورنہ غلط ہے۔

3.2.9 یونری (Unary) آپریٹر اور بائنری (Binary) آپریٹر میں فرق

(a) یونری آپریٹرز (Unary Operators)

آپریٹر کی ایک قسم جو سنگل او پیرینڈ (رقم) کے ساتھ کام کرتی ہے، یونیری آپریٹر کہلاتا ہے۔

مندرجہ ذیل آپریٹرز یونری آپریٹر ہیں:

-, ++,--

NOT FOR SALE

مندرجہ بالا آپریٹرز ایک اوپرینڈ (رقم) کے ساتھ استعمال ہوتے ہیں۔

```
-a;
n++;
--x;
```

(b) بائنری آپریٹر (Binary Operators)

آپریٹر کی ایک قسم جس کو اپنا عمل مکمل کرنے کے لیے دو رقموں کی ضرورت ہو اسے بائنری (Binary) آپریٹر کہا جاتا ہے۔

بائنری آپریٹرز یہ ہیں:

+,-,*, /, %

کچھ مثالیں یہ ہیں:

```
x+y;
a-b;
y/z;
x*t;
```

3.2.10 مشروط/کنڈیشنل آپریٹر ؟ (Conditional Operators)

کنڈیشنل آپریٹر ایک ایسا سٹرکچر ہے جو فیصلے کرنے کے لیے استعمال ہوتا ہے۔ یہ بالکل سادہ سے *if-else* کی جگہ استعمال کیا جاتا ہے۔ اس کو ٹرنری آپریٹر بھی کہا جاتا ہے کیونکہ یہ تین آپریٹڈز کے بغیر اپنا عمل مکمل نہیں کر سکتا۔

سائنٹکس:

(Condition)?true-case statement: false-case statement;

Conditions	یہ کوئی بھی ریلیشنل یا لاجیکل عبارت ہوگی۔
true-case	یہ اس وقت چلے گا جب شرط کا نتیجہ True ہوگا۔
false-case	یہ اس وقت چلے گا جب شرط کا نتیجہ False ہوگا۔

مثال:

فرض کریں ہمارے پاس ایک ویری ایبل a = 5 ہے تو درج ذیل اسٹیٹمنٹ

```
x= (a>50)? 1:0;
```

اگر شرط a> 50 درست ہے تو x کو 1 ویلیو ملے گی، اگر شرط غلط ہے تو x کو 0 ویلیو ملے گی۔ جیسا کہ a چھوٹا ہے 50 سے لہٰذا x = 0۔

3.2.11 آپریٹرز اور ان کی فوقیت (Operators and their Precedence)

آپریٹرز کی فوقیت اصل میں وہ قواعد ہیں جو اس چیز کا تعین کرتے ہیں کہ کسی بھی عبارت میں کون سا عمل پہلے انجام پائے گا۔
4 + 2 * 3 جیسے ایکسپریشن کو حل کرنے کے لیے ہمیں یہ پتہ ہونا چاہیئے کہ آپریٹرز کیا کرتے ہیں، اور ان کو لاگو کرنے کے لئے صحیح ترتیب کیا ہے۔ جس ترتیب میں آپریٹرز کو کسی عبارت میں استعمال کیا جاتا ہے اس کو آپریٹر ترجیح (Operator Precedence) کہا جاتا ہے۔ عام حسابی ترجیحی قواعد (جس میں ضرب کا عمل جمع سے پہلے کیا جاتا ہے) کا استعمال کرتے ہوئے، ہم جانتے ہیں کہ دی گئی عبارت 4+(2*3) کا نتیجہ 10 آئے گا۔

NOT FOR SALE

C میں، تمام آپریٹرز کو ایک خاص سطح کی ترجیح دی جاتی ہے۔ جن کی ترجیح زیادہ ہے ان کو پہلے حل کیا جاتا ہے۔ یہ نوٹ کرنا ضروری ہے کہ ضرب اور تقسیم کی ترجیح جمع اور منفی سے زیادہ ہے۔ کمپائلر ان ترجیحات کا استعمال کر کے اس بات کا تعین کرتے ہیں کہ عبارت کو کیسے حل کرنا ہے۔

3*2+4 کچھ (3*2)+4 کی صورت میں حل کی جائے گی کیونکہ ضرب کے عمل کو جمع پر فوقیت حاصل ہے۔

آپریٹرز کو ذیل میں زیادہ ترجیح سے لے کر کم ترجیح تک درج کیا گیا ہے۔

Precedence	Operators	Type
1	!,-,++,--	Logical NOT, Unary Operators
2	*, / , %	Multiplication, Division & Modulus
3	+, -	Addition & Subtraction
4	<, >, <=, >=	Relational operators
5	= =, !=	Relational operators
6	&&	Logical AND
7	||	Logical OR
8	=	Assignment operator

پروگرام 3.18 C لینگویج میں استعمال ہونے والے تمام آپریٹرز کی فوقیت واضح کرنے والا پروگرام۔

```
#include<stdio.h>
void main()
{
    int a=20;
    int b=10;
    int c=15;
    int d=5;
    int e;
    e=(a+b)*c/d;    //(30*15)/5
    printf("Value of (a+b)*c/d is : %d\n", e);
    e=((a+b)*c)/d;  //(30*15)/5
    printf("Value of ((a+b)*c)/d is  : %d\n", e);
    e=(a+b)*(c/d);  //(30)*(15/5)
    printf("Value of (a+b)*(c/d) is  : %d\n", e);
    e=a+(b*c)/d;    // 20+(150/5)
    printf("Value of a+(b*c)/d is  : %d\n", e);
}
```

آؤٹ پٹ

```
Value of (a + b) * c / d is : 90
Value of ((a + b) * c) / d is  : 90
Value of (a + b) * (c / d) is  : 90
Value of a + (b * c) / d is  : 50
```


NOT FOR SALE

اہم نکات

- پروگرامنگ میں، اِن پُٹ وہ ڈیٹا ہوتا ہے جو کمپیوٹر کو پروسیسنگ کے لئے دیا جاتا ہے اور آؤٹ پُٹ وہ ہوتا ہے جو ڈیٹا پر کارروائی کے بعد کمپیوٹر تیار کرتا ہے۔
- اسائنمنٹ / اِن پُٹ اسٹیٹمنٹ کے ذریعہ ڈیٹا ویری ایبل کو تفویض کیا جا سکتا ہے۔
- فارمیٹ سپیسیفائر ان پُٹ اور آؤٹ پُٹ فنکشن میں استعمال ہوتے ہیں۔
- ایسکیپ سیکوینس سے مراد حرفوں کا مجموعہ ہوتا ہے جس کا آغاز بیک سلیش / سے ہوتا ہے۔ اس کے بعد صرف خصوصی علامت ہوتی ہے۔
- scanf( ) فنکشن پروگرام کے نفاذ کے دوران ویری ایبل کے لیے values کو ان پُٹ کرنے کے لئے استعمال ہوتا ہے۔
- پروگرام پر عمل درآمد کے دوران کی بورڈ سے کسی ایک کیریکٹر کو پڑھنے کے لئے getche () ، getch() ، اور getchar() فنکشن استعمال ہوتے ہیں۔
- سیمی کالن ؛ ہر اسٹیٹمنٹ کے آخر میں استعمال ہوتا ہے۔
- آپریٹر علامتیں ہیں جو ڈیٹا پر مختلف کارروائیوں کے لئے استعمال کی جاتی ہیں۔
- C لینگوئج کمپاؤنڈ اسائنمنٹ آپریٹرز فراہم کرتی ہے جو اسائنمنٹ آپریٹر اور حسابی آپریٹرز کا مجموعہ ہے۔
- انکریمنٹ آپریٹر کی نمائندگی ڈبل پلس ++ کے ذریعہ کی جاتی ہے۔
- ڈیکریمنٹ آپریٹر کی نمائندگی ڈبل مائنس -- کے ذریعہ کی جاتی ہے۔
- ریلیشنل آپریٹرز کو پروگراموں میں شرائط لکھنے کے لئے استعمال کیا جاتا ہے۔
- آپریٹر کی ایک قسم جو سنگل اوپیرینڈ کے ساتھ کام کرتی ہے، یونیری آپریٹر کہلاتا ہے۔
- آپریٹر کی ایک قسم جو دو رقموں کے ساتھ کام کرتی ہے اسے بائنری آپریٹر کہا جاتا ہے۔
- آپریٹر کی ایک قسم جو تین رقموں کے ساتھ کام کرتی ہے اسے ٹرنری آپریٹر کہا جاتا ہے۔

مشق

سوال نمبر 1۔ موزوں جواب کا انتخاب کریں۔

i. ایسے فنکشنز جو ڈیٹا لیتے اور پھر انہیں کسی ویری ایبل میں اسٹور کرتے ہیں ــــــــــــــــ کہلاتے ہیں۔

(الف) آؤٹ پُٹ فنکشنز (ب) سٹرنگ فنکشنز (ج) عددی فنکشنز (د) اِن پُٹ فنکشنز

ii. ان میں کون سا فنکشن آؤٹ پُٹ کو سکرین پر دکھانے کے لیے استعمال ہوتا ہے؟

(الف) printf() (ب) scanf() (ج) gets() (د) getchar()

NOT FOR SALE

iii. کون سا فنکشن پروگرام کے چلنے کے دوران کسی ویری ایبل کی قیمت رکھنے کے لیے استعمال ہوتا ہے؟

(الف) printf() فنکشن　(ب) cout فنکشن

(ج) stdio.h فنکشن　(د) scanf() فنکشن

iv. C لینگویج میں ہر اسٹیٹمنٹ کا اختتام ______________ سے کیا جاتا ہے۔

(الف) فل سٹاپ　(ب) ڈبل کوٹ　(ج) سیمی کالن　(د) کامہ

v. ذیل میں کون سا فارمیٹ سپیسیفائر انٹیجر کے لیے استعمال ہوتا ہے؟

(الف) %s　(ب) %c　(ج) %d　(د) %f

vi. آؤٹ پٹ پر نئی لائن ڈالنے کے لیے کون سا اسکیپ سیکونس استعمال کیا جاتا ہے؟

(الف) \new　(ب) \t　(ج) \n　(د) \line

vii. ان میں کون سا آپریٹر ماڈولس آپریٹر کہلاتا ہے؟

(الف) +　(ب) %　(ج) /　(د) *

viii. کون سا آپریٹر ویری ایبل کی قیمت میں ایک عدد کا اضافہ کر دیتا ہے؟

(الف) '+' آپریٹر　(ب) اسائنمنٹ آپریٹر

(ج) ڈیکریمنٹ آپریٹر　(د) انکریمنٹ آپریٹر

ix. کون سا آپریٹر ویری ایبل کی قیمت میں عبارت میں استعمال ہونے کے بعد، 1 کا اضافہ کرے گا؟

(الف) پوسٹ فکس آپریٹر　(ب) پری فکس آپریٹر

(ج) ماڈولس آپریٹر　(د) بائنری آپریٹر

x. کون سا آپریٹر ہے جس کا نتیجہ True ہو گا اگر اس کی دونوں شرائط کا نتیجہ True ہو؟

(الف) AND آپریٹر　(ب) OR آپریٹر

(ج) NOT آپریٹر　(د) اوپر کے تمام

سوال نمبر 2۔ ذیل میں دیئے گئے مختصر سوالات کے جوابات تحریر کریں۔

i. فارمیٹ سپیسیفائر کیا ہے؟ مثالیں دیں۔

ii. اسکیپ سیکونس کیوں استعمال کیے جاتے ہیں؟ مثالیں دیں۔

iii. gets() فنکشن کس مقصد کے لیے استعمال ہوتا ہے؟

NOT FOR SALE

iv. ()getch اور ()getche فنکشنز میں فرق واضح کریں۔

v. ذیل میں دی گئی عبارتوں کو حل کریں۔

(الف) 9-5*(6+2) (ب) 60/10*24+3 (ج) 001%50-100%3

vi. اسائنمنٹ آپریٹر اور کمپاؤنڈ اسائنمنٹ آپریٹر میں فرق بیان کریں۔

سوال نمبر 4۔ آؤٹ پُٹ فنکشن کیا ہے؟ آؤٹ پُٹ فنکشن کی مختلف اقسام کو تفصیل سے بیان کریں۔

سوال نمبر 6۔ اِن پُٹ فنکشن میں استعمال ہونے والے فارمیٹ سپیسیفائر اور اسکیپ سیکونس کی تفصیل سے وضاحت کریں۔

سوال نمبر 8۔ ایک پروگرام لکھیں جو مثلث کا قاعدہ اور اونچائی اِن پُٹ کرے اور پھر مثلث کا رقبہ معلوم کر کے پرنٹ کرے۔

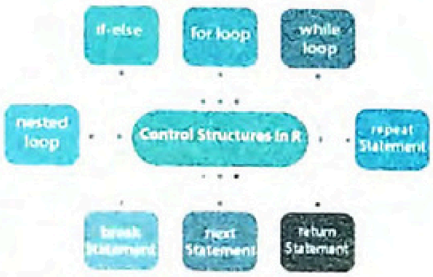
سوال نمبر 10۔ ایک پروگرام لکھیں جو طالب علم کا نام اور ایڈریس اِن پُٹ کرے اور اسے اسکرین پر پرنٹ کرے۔ اس مقصد کے لیے ()gets اور ()puts کے فنکشن استعمال کریں۔

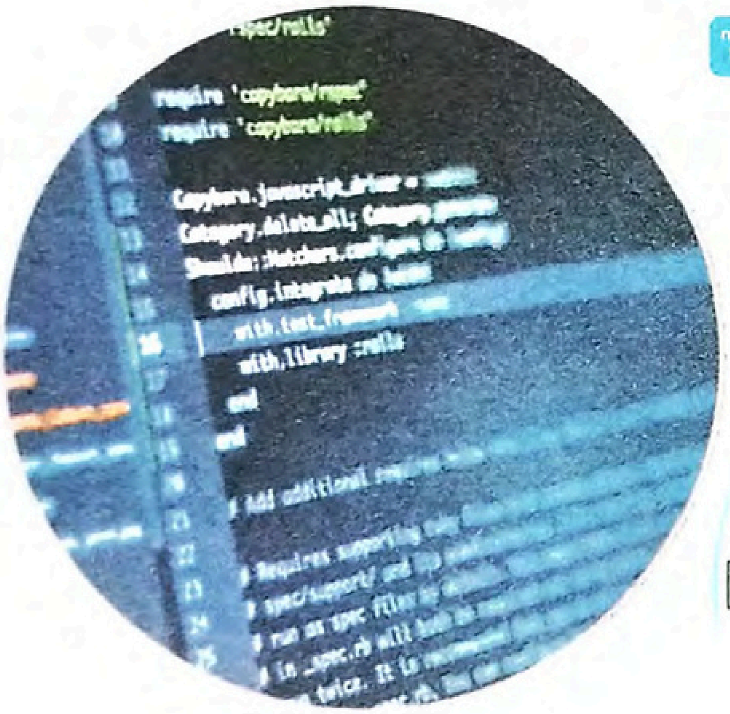
NOT FOR SALE

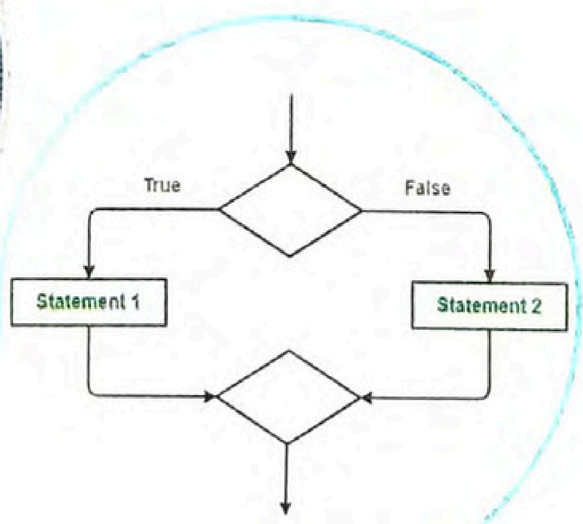
باب نمبر 4

اس باب کی تکمیل کے بعد طلبہ اس قابل ہو جائیں گے کہ وہ:

- کنٹرول اسٹیٹمنٹ کی تعریف کر سکیں۔
- کنڈیشنل اسٹیٹمنٹ کی تعریف کر سکیں۔
- if اسٹیٹمنٹ اور اس کے استعمال کو جان سکیں۔
- if-else اسٹیٹمنٹ کے سٹرکچر اور استعمال کو جان سکیں۔
- switch اسٹیٹمنٹ کے سٹرکچر، اس کے استعمال اور اس میں case، default اور break کے کردار کو سمجھ سکیں۔
- نیسٹڈ سیلیکشن سٹرکچر کو استعمال کر سکیں۔
- تمام سیلیکشن سٹرکچرز کے درمیان فرق کر سکیں۔
- باب 1 میں لکھے گئے فلوچارٹس کے لیے کوڈ لکھ سکیں۔







NOT FOR SALE

4.1 کنٹرول سٹرکچر (Control Structure)

کنٹرول سٹرکچر در حقیقت سٹرکچرڈ پروگرامنگ لینگویج کے بنیادی اجزاء ہیں۔ کنٹرول سٹرکچرز پروگرام کے چلنے کی عمومی ترتیب کو بدلنے کے لیے استعمال ہوتے ہیں۔ ہمیں پروگرام کے چلنے کی ترتیب کو بدلنے کی ضرورت کیوں ہوتی ہے؟ اس کی وجہ "فیصلہ سازی (Decision Making)" ہے۔

حقیقی زندگی میں ایسے حالات آتے ہیں جب ہمیں کچھ فیصلے کرنے کی ضرورت ہوتی ہے اور ان فیصلوں کی بنیاد پر، ہم فیصلہ کرتے ہیں کہ ہمیں آگے کیا کرنا چاہئے۔ مثال کے طور پر، ہمیں مختلف اختیارات دیئے جاسکتے ہیں جیسے "میڈیکل" یا "انجینئرنگ" کرنا۔ ہم کچھ شرائط (جیسے اپنی ذاتی دلچسپی، مواقع کی گنجائش وغیرہ) کا تجزیہ کرکے فیصلہ کرتے ہیں جو فیصلہ کرتے ہیں اس سے ہم اپنی زندگی کی سمت کے بہاؤ کو بدل دیتے ہیں۔ C پروگرامنگ میں بھی اسی طرح کے حالات پیدا ہوتے ہیں جہاں ہمیں کسی فیصلے کی بنیاد پر پروگرام کو ایک یا دوسرے راستے پر چلانے کی ضرورت ہوتی ہے۔

4.1.1 کنٹرول اسٹیٹمنٹس (Control Statements)

پروگراموں میں کچھ ایسے سیکشنز یا ہدایات کے بلاکس ہوتے ہیں جو صرف ضرورت پڑنے پر ہی کام میں لائے جاتے ہیں۔ ضرورت پڑنے پر پروگرام کا کوئی خاص حصہ ہی چلایا جاتا ہے۔ جبکہ کوڈ کا ایک حصہ مصروف ہو تو دوسرے حصے غیر فعال ہوتے ہیں۔ کنٹرول اسٹیٹمنٹ کی مدد سے پروگرامر اس بات کی نشاندہی کرتے ہیں کہ کوڈ کے کون سے سیکشن مخصوص اوقات میں استعمال ہوں۔ تاہم، کسی مسئلے کی ضرورت کو سامنے رکھتے ہوئے پروگرام کے کسی خاص حصے کو چلانے یا نہ چلانے کا فیصلہ کیا جاتا ہے۔ کنٹرول اسٹیٹمنٹ سورس کوڈ میں شامل عناصر ہوتے ہیں جو پروگرام پر عمل درآمد کے بہاؤ کو کنٹرول کرتے ہیں۔ ایسی اسٹیٹمنٹ جو پروگرام چلنے کے دوران اس کی اسٹیٹمنٹ کی ترتیب کا تعین کرتی ہیں کنٹرول اسٹیٹمنٹس (Control Statements) کہلاتی ہیں۔

کنٹرول اسٹیٹمنٹس (Control Statements) کی دو اہم اقسام ہیں۔

1۔ کنڈیشنل / سلیکشن اسٹیٹمنٹس (Conditional / Selection Statements)

2۔ تکراری یا لوپنگ اسٹیٹمنٹس (Repetition/Looping Statements) (ان پر باب 5 میں تفصیلی بحث کی جائے گی)

4.1.2 کنڈیشنل / سلیکشن اسٹیٹمنٹس (Conditional / Selection Statements)

مفید پروگرام لکھنے کے لیے، ہمیں شرائط کی جانچ پڑتال اور اس کے مطابق پروگرام کے طرز عمل کو تبدیل کرنے کی صلاحیت کی ضرورت ہوتی ہے۔ سلیکشن اسٹیٹمنٹ، جن کو بعض اوقات کنڈیشنل اسٹیٹمنٹ بھی کہا جاتا ہے، ہمیں یہ صلاحیت فراہم کرتی ہیں۔ بعض

NOT FOR SALE

او قات پروگرام کو کسی خاص حالت یار یلیشنل ایکسپریشن کی بنیاد پر چلانے کی ضرورت ہوتی ہے۔ C لینگویج سلیکشن کنٹرول سٹرکچر کو نافذ کرنے کے لئے درج ذیل اسٹیٹمنٹ فراہم کرتا ہے۔

- *if* اسٹیٹمنٹ
- *if-else* اسٹیٹمنٹ
- *nested if* اسٹیٹمنٹ
- *switch* اسٹیٹمنٹ

4.1.3 if اسٹیٹمنٹ

if اسٹیٹمنٹ ایک فیصلہ ساز اسٹیٹمنٹ ہے۔ یہ سلیکشن سٹرکچر کی سب سے سادہ شکل ہے۔ یہ کسی دی گئی شرط کی بنیاد پر ایک اسٹیٹمنٹ یا اسٹیٹمنٹس کے مجموعہ کو چلاتی ہے یا نظر انداز کر دیتی ہے۔ اس کو کبھی کبھار بائنری سلیکشن بھی کہا جاتا ہے کیونکہ عملدرآمد کی دو ممکنہ راہیں ہیں۔

شرط (Relational Expression) کی شکل میں لکھی جاتی ہے۔ اگر شرط کا نتیجہ درست (True) ہے تو، اسٹیٹمنٹ یا اسٹیٹمنٹس کا مجموعہ جو کہ if اسٹیٹمنٹ کے بعد آتا ہے اسے چلا دیا جاتا ہے ورنہ نظر انداز کر دیا جاتا ہے۔ if اسٹیٹمنٹ کا سائنٹکس درج ذیل ہے:

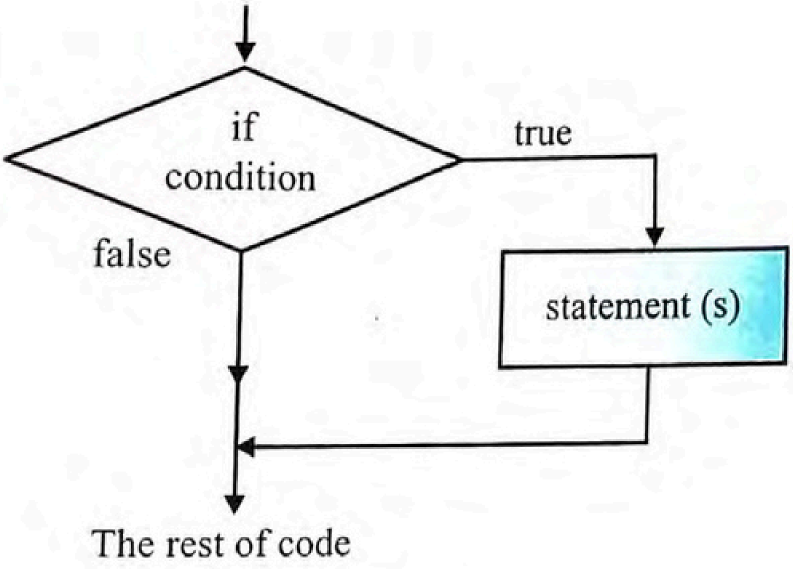
```
if (condition)
    statement;
```

مذکورہ سائنٹکس ایک اکیلی اسٹیٹمنٹ کے لیے استعمال ہوتا ہے۔ اسٹیٹمنٹس کے مجموعہ کو بھی مشروط کیا جا سکتا ہے۔ ایسی صورت میں اسٹیٹمنٹس کے مجموعہ کو قوسین { } کے درمیان لکھا جاتا ہے۔ اسٹیٹمنٹس کے اس مجموعہ کو کمپاؤنڈ اسٹیٹمنٹس (Compound Statements) یا اسٹیٹمنٹس کا بلاک بھی کہا جاتا ہے۔ کمپاؤنڈ اسٹیٹمنٹس کے لیے if اسٹیٹمنٹ کا سائنٹکس درج ذیل ہے:

```
if (condition)
{
statement 1;
statement 2;
.
.
Statement n;
}
```

ذیل میں دیا گیا فلو چارٹ if اسٹیٹمنٹ کے عمل کو ظاہر کرتا ہے۔

NOT FOR SALE



درج ذیل پروگرام if اسٹیٹمنٹ کے استعمال کو واضح کرتا ہے۔

پروگرام 4.1 ایک پروگرام جو اس وقت ان پٹ کیے ہوئے نمبر کو پرنٹ کرے گا جب وہ 50 سے بڑا ہو۔

```
#include<stdio.h>
void main()
{
    int x;
    printf("\n Enter an integer = ");
    scanf("%d",&x);
    if(x >50)
    printf("%d", x);
}
```

```
Enter an integer = 75
75
```

پروگرام 4.2 درج ذیل پروگرام ایک اکیلی اسٹیٹمنٹ کو چلائے گا اگر if اسٹیٹمنٹ میں دی گئی شرط کا نتیجہ True آتا ہے۔

```
#include<stdio.h>
void main()
{
    int a, b;
    a=1000;
    b=500;
    if(a>b) {
        printf("Value of a is greater than b\n"); }
        printf("Thank You");
    }
}
```

```
Value of a is greater than b
Thank You
```

پروگرام 4.3 ایک پروگرام جو دو نمبر ان پٹ کرتا ہے اور ایک پیغام پرنٹ کرتا ہے کہ یہ دونوں مساوی ہیں یا نہیں۔

```
#include<stdio.h>
void main()
{
    int a, b;
    printf("\nEnter first number = ");
    scanf("%d",&a);
    printf("\nEnter second number = ");
    scanf("%d",&b);
    if(a==b)
    printf("Both numbers are equal");
```

آؤٹ پٹ

```
Enter first number = 45
Enter second number = 45
Both numbers are equal
```

4.1.4 if-else سٹیٹمنٹ (if-else Statement)

if-else اسٹیٹمنٹ بھی فیصلے کرنے کے لیے استعمال ہوتی ہے۔ اگر شرط کا نتیجہ True ہو تو یہ اسٹیٹمنٹ کے ایک بلاک کو چلاتی ہے اور اگر شرط کا نتیجہ False ہو تو دوسرے بلاک کو چلائے گی۔ کسی بھی صورتحال میں صرف ایک ہی بلاک چلے گا دوسرا بلاک نظر انداز کر دیا جائے گا۔ اسے دوطرفہ سلیکشن اسٹیٹمنٹ بھی کہا جاتا ہے۔

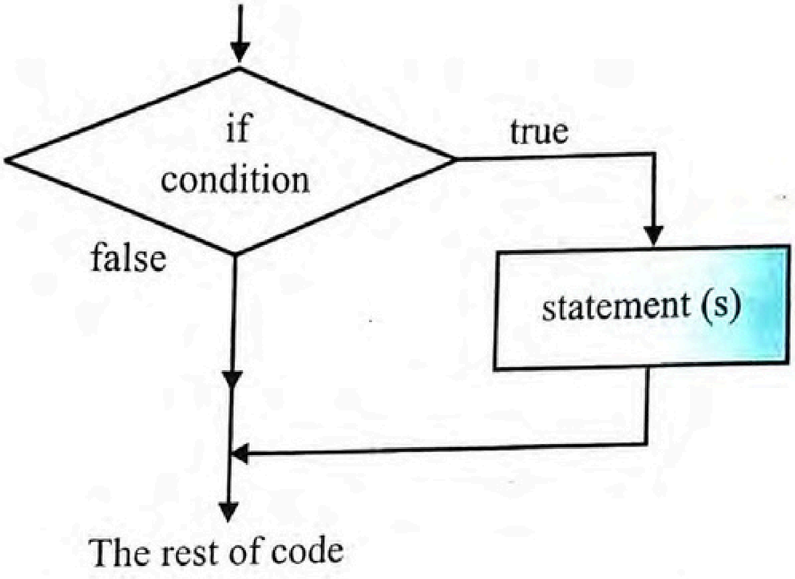
سائنٹکس:

```
Syntax:

 if(condition)
            statement;
else
            statement;
```

دو یا دو سے زیادہ اسٹیٹمنٹس کو کوسین { } میں لکھا جاتا ہے۔ کمپاؤنڈ اسٹیٹمنٹ کے لیے if-else کا سائنٹکس درج ذیل ہے:

```
 if(condition)
 {
     statement 1;
     statement 2;
     .
     .
     statement n;
 }
else
 {
     statement 1;
```



درج ذیل پروگرام if اسٹیٹمنٹ کے استعمال کو واضح کرتا ہے۔

پروگرام **4.1** ایک پروگرام جو اس وقت ان پٹ کیے ہوئے نمبر کو پرنٹ کرے گا جب وہ 50 سے بڑا ہو۔

```
#include<stdio.h>
void main()
{
    int x;
    printf("\n Enter an integer = ");
    scanf("%d",&x);
    if(x >50)
    printf("%d", x);
}
```

```
Enter an integer = 75
75
```

پروگرام **4.2** درج ذیل پروگرام ایک اکیلی اسٹیٹمنٹ کو چلائے گا اگر if اسٹیٹمنٹ میں دی گئی شرط کا نتیجہ True آتا ہے۔

```
#include<stdio.h>
void main()
{
     int a, b;
     a=1000;
     b=500;
     if(a>b) {
          printf("Value of a is greater than b\n"); }
          printf("Thank You");
     }
}
```

```
Value of a is greater than b
Thank You
```


NOT FOR SALE

پروگرام 4.3 ایک پروگرام جو دو نمبر اِن پُٹ کرتا ہے اور ایک پیغام پرنٹ کرتا ہے کہ یہ دونوں مساوی ہیں یا نہیں۔

```
#include<stdio.h>
void main( )
{
    int a, b;
    printf("\nEnter first number = ");
    scanf("%d",&a);
    printf("\nEnter second number = ");
    scanf("%d",&b);
    if(a==b)
    printf("Both numbers are equal");
}
```

آؤٹ پٹ

```
Enter first number = 45
Enter second number = 45
Both numbers are equal
```

4.1.4 if-else سٹیٹمنٹ (if-else Statement)

if-else اسٹیٹمنٹ بھی فیصلے کرنے کے لیے استعمال ہوتی ہے۔ اگر شرط کا نتیجہ True ہو تو یہ اسٹیٹمنٹ کے ایک بلاک کو چلاتی ہے اور اگر شرط کا نتیجہ False ہو تو دوسرے بلاک کو چلائے گی۔ کسی بھی صورتحال میں صرف ایک ہی بلاک چلے گا دوسرا بلاک نظر انداز کر دیا جائے گا۔ اسے دو طرفہ سلیکشن اسٹیٹمنٹ بھی کہا جاتا ہے۔

سائنٹکس:

```
Syntax:

if(condition)
        statement;
else
        statement;
```

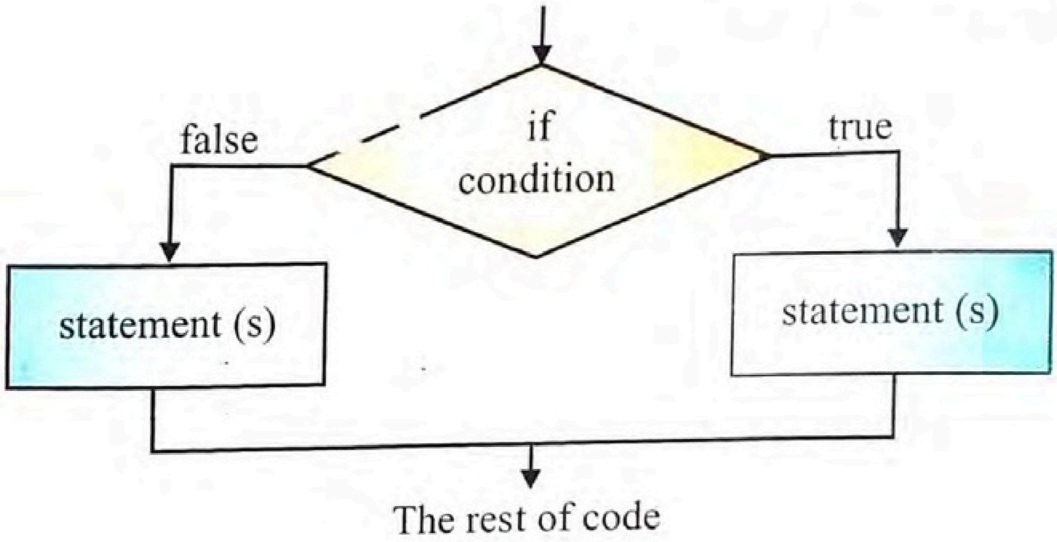
دو یا دو سے زیادہ اسٹیٹمنٹس کو کوسین { } میں لکھا جاتا ہے۔ کمپاؤنڈ اسٹیٹمنٹ کے لیے if-else کا سائنٹکس درج ذیل ہے:

```
if(condition)
{
    statement 1;
    statement 2;
    .
    .
    statement n;
}
else
{
    statement 1;
```


NOT FOR SALE

```
        statement 2;
        .
        .
        statement n;
    }
```

درج ذیل فلو چارٹ if-else کے عمل کو ظاہر کرتا ہے۔



شکل نمبر 4.2 if-else کا سٹرکچر

درج ذیل پروگرام میں if-else اسٹیٹمنٹ کے استعمال کو ظاہر کیا گیا ہے۔

پروگرام 4.4 یہ پروگرام ایک عدد اِن پٹ کرے گا اور یہ معلوم کرے گا کہ یہ عدد جفت (Even) ہے یا طاق (Odd)۔

```
#include<stdio.h>
void main()
{
    int n;
    printf("Enter a number = ");
    scanf("%d",&n);
    if(n%2==0)        //% operator returns remainder after division
        printf("%d is even",n);
    else
        printf("%d is odd",n);
}
```

آؤٹ پٹ

```
Enter a number = 10
10 is even
```

```
Enter a number = 17
17 is odd
```


NOT FOR SALE

پروگرام 4.5 یہ پروگرام دو اعداد میں سے بڑے عدد کو پرنٹ کرے گا۔

```
#include<stdio.h>
void main ()
{
    int x, y;
    printf("Enter two numbers:");
    scanf("%d %d",&x,&y);
    if( x > y )
    printf("\n%d is larger number than %d",x,y);
    else
    printf("\n%d is larger number than %d",y,x);
```

آؤٹ پٹ

```
Enter two numbers = 15
20
20 is larger number than 15
```

پروگرام 4.6 یہ پروگرام تین اعداد میں سے سب سے چھوٹے عدد کو پرنٹ کرے گا۔

```
#include <stdio.h>
void main()
{
    int x, y, z, s;
    printf("Enter three numbers");
    scanf("%d %d %d",&x,&y,&z);
  if(x < y)
    s = x;
  else
    s = y;
  if(s < z)
    printf("The smallest number is :%d",s);
  else
    printf("The smallest number is :%d",z);
}
```

آؤٹ پٹ

```
Enter three numbers = 10
6
12
The smallest number is: 6
```

4.1.5 switch اسٹیٹمنٹ (The switch Statement)

C میں switch اسٹیٹمنٹ ویری ایبل کی قیمت کی جانچ کرتا ہے اور اس کا ایک سے زیادہ case کے ساتھ موازنہ کرتا ہے۔ ایک بار جب case میچ ہو جاتا ہے تو، اس خاص case سے وابستہ اسٹیٹمنٹ کا ایک بلاک چلایا جاتا ہے۔

switch کے ایک بلاک میں ہر case کا ایک مختلف نام / نمبر ہوتا ہے جسے آئیڈینٹیفائر (Identifier) کہا جاتا ہے۔ یوزر کی فراہم کردہ قیمت کا موازنہ switch بلاک کے اندر موجود تمام cases سے کیا جاتا ہے۔ اگر کوئی case میچ نہیں ملا تو default اسٹیٹمنٹ پر عمل درآمد ہو جاتا ہے، اور کنٹرول switch بلاک سے باہر نکل جاتا ہے۔

NOT FOR SALE

یہ if-else کے متبادل کے طور پر استعمال ہوتا ہے۔ switch اسٹیٹمنٹ کثیر الانتخاب (Multiple Choice Statement) اسٹیٹمنٹ ہے۔

switch اسٹیٹمنٹ ایک عبارت کو حل کرتا ہے اور اس سے ایک قیمت (Returned Value) اسے ملتی ہے۔ سوئچ اسٹیٹمنٹ کو عبارت سے صرف ایک ہی ویلیو ملتی ہے جس کی بنیاد پر صرف ایک ہی case چلایا جاتا ہے۔ اسٹیٹمنٹ میں سے صرف ایک صورت عبارت کے ذریعہ واپس کی گئی قیمت پر منحصر ہوتی ہے۔ عبارت سے ملنے والی ویلیو کا موازنہ case میں موجود کانسٹنٹ ویلیو سے کیا جاتا ہے۔ اگر ملنے والی ویلیو case میں موجود ویلیو کے برابر ہو تو۔ case میں موجود تمام اسٹیٹمنٹس چلادی جائیں گی۔ switch کا سائنٹکس درج ذیل ہے:

سائنٹکس:

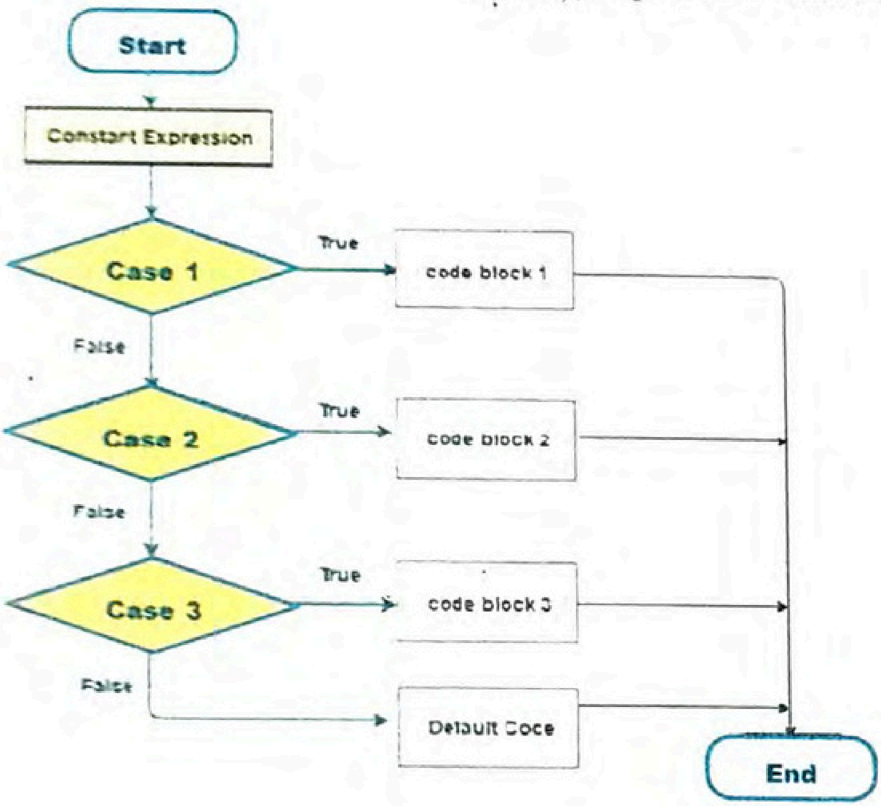
```
switch(expression)
{
        case const-1:
                statements;
                break;
        case const-2:
                statements;
                break;
        .
        .
        case const-n:
                statements;
                break;
        default:
        statements;
}
```

وضاحت:

- درج بالا مذکورہ سائنٹکس میں switch اسٹیٹمنٹ کے سامنے ایک عبارت (Expression) لکھی گئی ہے، پہلے یہ عبارت حل کی جائے گی اس سے جو بھی نتیجہ آئے گا اس قیمت کو ہر case کے ساتھ دی گئی کانسٹنٹ قیمت کے ساتھ موازنہ کیا جائے گا۔ اگر یہ قیمت case والے کانسٹنٹ سے ملتی ہے تو پھر اس case میں دی گئی تمام اسٹیٹمنٹس چلادی جائیں گی۔
- const-1 اور const-2، وغیرہ یہ ہمیشہ عددی کانسٹنٹ یا کریکٹر / حرفی کانسٹنٹ ہونگے۔
- ہر case کے اختتام پر break اسٹیٹمنٹ دی گئی ہے جو switch کی باڈی سے باہر نکلنے کے لئے استعمال ہوتی ہے۔ یہ ہر case کے بلاک کے اختتام پر استعمال ہوگی۔ اگر break کا استعمال نہ کیا جائے تو تمام کے تمام case جو مشابہ case (Matching) کے بعد آئیں گے وہ بھی چلادیئے جائیں گے۔
- switch باڈی میں default بھی استعمال ہوتا ہے۔ اگر کوئی مشابہ case نہ ملے تو default کے تحت جو اسٹیٹمنٹس آئیں گی وہ چلادی جائیں گی، لیکن اس کا استعمال اختیاری (Optional) ہے۔ اگر یہ موجود نہ ہو تو کنٹرول switch کی باڈی سے باہر نکل کر

switch سٹرکچر کے خاتمے کے بعد میں آنے والی پہلی اسٹیٹمنٹ پر پہنچ جائے گا۔

درج ذیل فلوچارٹ switch اسٹیٹمنٹ کے فعل کو ظاہر کرتا ہے۔



درج ذیل پروگرام switch-case کے استعمال کو ظاہر کرتا ہے۔

پروگرام 4.7 یہ پروگرام بتائے گا کہ جو کریکٹر ان پٹ ہوا ہے وہ Vowel ہے یا نہیں۔

```
#include<stdio.h>
void main( )
{
    char ch;
    printf("Enter a character   ");
    scanf("%c",&ch);
    switch (ch)
    {
        case 'a':    printf("vowel a");
        break ;
        case 'e':    printf("vowel e");
        break ;
        case 'o':    printf("vowel o");
        break ;
```

```
Enter a character i
vowel i
```


NOT FOR SALE

```
        case 'i':     printf("vowel i");
        break ;
        case 'u':     printf("vowel u");
        break;
        default :     printf("not a vowel");
    }
}
```

4.1.6 نیسٹڈ سلیکشن سٹرکچر کا استعمال (Using nested selection structures)

نیسٹڈ سلیکشن سٹرکچر کا استعمال اس وقت ہوتا ہے جب کسی کام کو انجام دینے سے پہلے ایک سے زیادہ فیصلے کرنے کی ضرورت پیش آئے۔ نیسٹنگ ایک پروگرامنگ سرگرمی ہے، جس میں ایک پروگرام بلاک اسی طرح کے دوسرے پروگرام بلاک کے اندر رکھا جاتا ہے۔ نیسٹنگ پراسس زیادہ تر سلیکشن کنٹرول سٹرکچر میں استعمال ہوتے ہیں۔

سانٹکس:

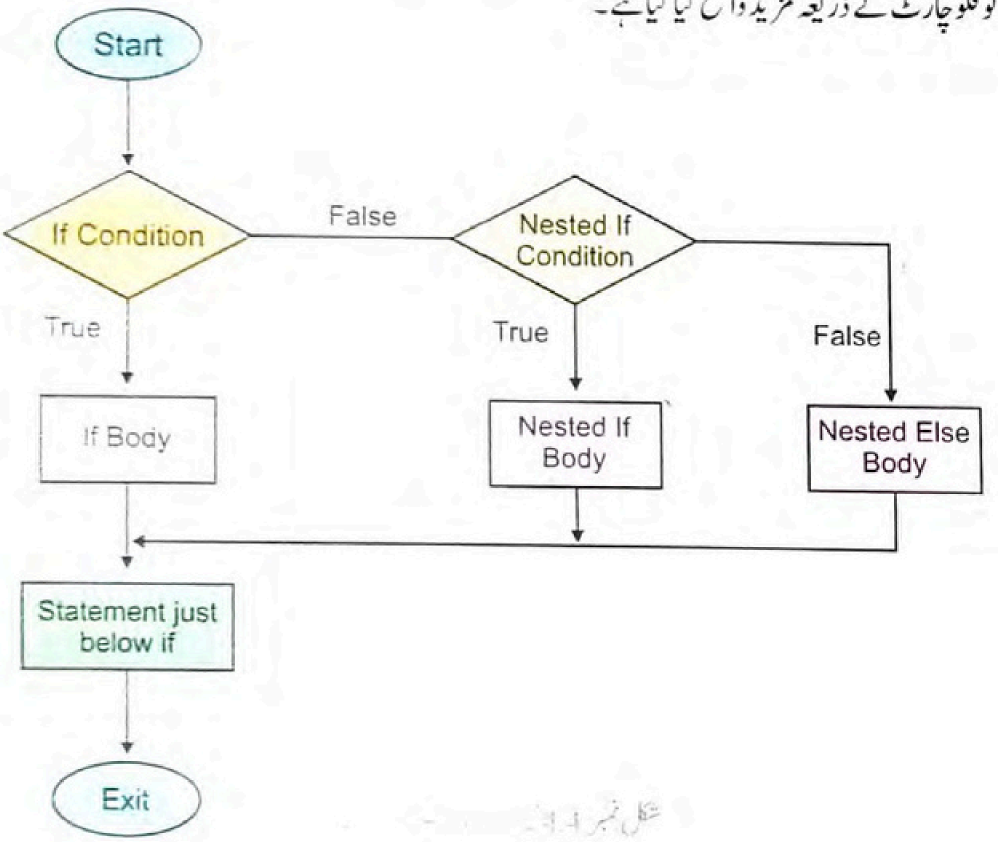
```
if(Condition1)
{
   if(Condition2)
   {
      Statement1;
   }
   else
   {
      Statement2;
   }
}
else
{
   if(Condition3)
   {
      Statement3;
   }
   else
   {
      Statement4;
   }
}
```

نیسٹڈ if-else کیسے کام کرتا ہے:

نیسٹڈ if-else کی اس مثال میں، یہ پہلے Condition 1 کو ٹیسٹ کرتا ہے، اگر یہ Condition درست ہے تو یہ Condition 2 کو ٹیسٹ کرتا ہے، اگر Condition 2 کا نتیجہ درست ہے تو Statement 1 چل جائے گی۔ دوسری صورت

NOT FOR SALE

میں Statement 2 پر عملدرآمد کیا جائے گا۔ اسی طرح، اگر Condition 1 کا نتیجہ غلط ہے تو یہ Condition 3 کی جانچ کرتا ہے۔ اگر Condition 3 کا نتیجہ صحیح ہے تو Statement 3 پر عمل درآمد ہوگا ورنہ Statement 4 پر عمل درآمد ہوگا۔ اس مثال میں پیش کردہ نیسٹنگ کو فلو چارٹ کے ذریعہ مزید واضح کیا گیا ہے۔



شکل نمبر 4.4

پروگرام 4.8 کی بورڈ کی مدد سے ان پٹ ہونے والے عدد کو معلوم کرنا کہ یہ مثبت عدد ہے؟ جفت ہے یا طاق ہے؟

```
using namespace std;
#include<iostream>
int main( )
{
    int num;
    cout<<"\nEnter a number in between -10 and 10:\t";
    cin>>num;
    if(num>0)
    {
        cout<<num<<" is a positive number"<<endl;
        if(num % 2 == 0)
            cout<<num<<" is also an even number"<<endl;
        else
            cout<<num<<" is also an odd number"<<endl;
    }
```

آؤٹ پٹ

```
Enter a number between -10 and 10: 7
7 is a positive number
7 is also an odd number
```


NOT FOR SALE

```
        else
                cout<<num<<"is a negative number"<<endl;
        return 0;
}
```

مذکورہ پروگرام میں، جو عدد ان پٹ ہوگا اسے پہلے چیک کیا جائے گا کہ مثبت ہے یا منفی اور اگر عدد مثبت ہوا تو پہلے *if* کی باڈی کو چلایا جائے گا تاکہ یہ کنڈیشن ٹیسٹ کی جا سکے کہ نمبر جفت ہے یا طاق۔ یہ ضروری ہے کہ پہلی *if* اسٹیٹمنٹ کی باڈی کو نیسٹڈ *if* میں موجود قوسین { } کے اندر بند کر دیا جائے تاکہ اندر والی *if* اسٹیٹمنٹ کو باڈی کا حصہ بنایا جا سکے۔

4.1.7 سلیکشن سٹرکچر کے درمیان فرق (Difference among all selection structure)

if اسٹیٹمنٹ: اس وقت استعمال کیا جاتا ہے جب شرط کے صحیح (True) ہونے پر اسٹیٹمنٹس کے ایک بلاک کو چلانا ہو۔ اگر شرط کا نتیجہ غلط (False) ہو تو اسٹیٹمنٹ کے بلاک کو نظر انداز کر دیا جائے گا اور کنٹرول، if کے بلاک کے بعد آنے والی پہلی اسٹیٹمنٹ پر چلا جائے گا اور پروگرام وہاں سے آگے چلنا شروع ہو جائے گا۔

if-else اسٹیٹمنٹ: یہ ایسی صورت حال میں استعمال کی جاتی ہے جب *if* اسٹیٹمنٹ میں آنے والی شرط کا نتیجہ True ہو تو اسٹیٹمنٹ کا ایک بلاک چلتا ہے لیکن اگر شرط کا نتیجہ False ہو تو *if* کے بعد آنے والے بلاک کو نظر انداز کر دیا جاتا ہے اور **else** کے بعد آنے والے بلاک کو چلایا جاتا ہے۔ یہ اسٹیٹمنٹ کے دو میں سے کسی ایک بلاک کا انتخاب کر کے اسے چلاتا ہے۔

switch-case اسٹیٹمنٹ: یہ کثیر الانتخاب اسٹیٹمنٹ ہے۔ یہ کسی ویری ایبل یا کسی عبارت کے نتیجے کی بنیاد پر اسٹیٹمنٹ کے ایک بلاک کو منتخب کرنے کے لئے استعمال کیا جاتا ہے۔ یہ کسی ویری ایبل کی قیمت یا کسی عبارت کے نتیجے کا موازنہ ہر case میں موجود قیمت کے ساتھ کرتا ہے۔ اور جوڑ مل جانے کی صورت میں اِس case کے بعد آنے والی اسٹیٹمنٹ کے بلاک کو چلا دیتا ہے۔ لیکن اگر اسے کوئی بھی جوڑ نہ ملے تو default (اگر موجود ہے تو) کے بعد آنے والی اسٹیٹمنٹ کے بلاک کو چلایا جاتا ہے ورنہ کنٹرول کو **switch** کے بلاک کے بعد آنے والی پہلی اسٹیٹمنٹ پر بھیج دیا جاتا ہے۔

4.1.8 یونٹ 1 کے فلو چارٹس کے لئے کوڈز

پروگرام **4.9** پروگرام میں ان پٹ کیے گئے عدد base کی طاقت **exponent** معلوم کرنا۔

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    float base, exponent, result;
    cout << "Enter base and exponent respectively:  ";
    cin >> base >> exponent;
```

آؤٹ پٹ

```
Enter base and exponent respectively:  2.3
    4.5
  2.3^4.5 = 42.44
```

NOT FOR SALE

```
        result = pow(base, exponent);
        cout << base << "^" << exponent << " = " << result;
        return 0;
}
```

پروگرام 4.10 جب اُنچائی اور قاعدہ کی لمبائی دی جائے تو مثلث کا رقبہ تلاش کرنا۔

```
* C++ program to find the area of a triangle if base and height are given */
#include <iostream>
using namespace std;
int main()
{
    float b, h, a;
      b = base
      h = height
      a = area
    cout << "Enter the base & height of the triangle::\n";
    cin >> b >> h;
      Calculate area of the triangle
    a = (float)((b * h)  2);
      Output
    cout << " nArea of the triangle = " << a << " sq. units";
    return 0;
}
```

آؤٹ پُٹ

```
Enter the base & height of the triangle::
     5
     7
Area of the triangle = 17.5 sq. units
```

پروگرام 4.11 مکعب کے ضلع کی لمبائی اِن پُٹ کر کے اس کا حجم معلوم کر کے پرنٹ کرنا۔

```
#include <iostream>
using namespace std;
  int main()
  {
        int sid1;
        float volcu;
                cout << " n n Calculate the volume of a cube :\n";
                cout << "---------------------------------------\n";
        cout<<" Input the side of a cube : ";
        cin>>sid1;
        volcu=(sid1*sid1*sid1);
        cout<<" The volume of a cube is : "<< volcu << endl;
        cout << endl;
        return 0;
  }
```

آؤٹ پُٹ

```
Calculate the volume of a cube:
---------------------------------------
Input the side of a cube: 5
The volume of a cube is: 125
```


NOT FOR SALE

پروگرام 4.12 ایک پروگرام لکھیں جس میں نیسٹڈ if-else کے ذریعے سب سے بڑا عدد معلوم کرنے کا کہا ہو۔

```
#include <iostream>
using namespace std;
int main()
{
  float n1, n2, n3;
  cout << "Enter three numbers: ";
  cin >> n1 >> n2 >> n3;
  if(n1 >= n2) {
     if(n1 >= n3)
        cout << "Largest number: " << n1;
     else
        cout << "Largest number: " << n3;
  }
  else {
   if(n2 >= n3)
   cout << "Largest number: " << n2;
   else
   cout << "Largest number: " << n3;
              }
  return 0;
}
```

آؤٹ پٹ

```
Enter three numbers: 2.3
8.3
-4.2
Largest number: 8.3
```

اہم نکات

- کنٹرول سٹرکچر پروگرام کے چلنے کی عمومی ترتیب کو بدلنے کے لئے استعمال ہوتے ہیں۔
- ایسی اسٹیٹمنٹ جو پروگرام چلنے کے دوران اس کی اسٹیٹمنٹ کی ترتیب کو متاثر کرتی ہیں، کنٹرول اسٹیٹمنٹس (Control Statement) کہلاتی ہیں۔
- *if* اسٹیٹمنٹ ایک فیصلہ ساز اسٹیٹمنٹ ہے۔ یہ سیلیکشن سٹرکچر کی آسان ترین شکل ہے۔
- *if-else* اسٹیٹمنٹ بھی فیصلے کرنے کے لیے استعمال ہوتی ہے۔ یہ اسٹیٹمنٹ کے اندر ایک بلاک کو چلاتی ہے اگر شرط کا نتیجہ True ہو اور دوسرے بلاک کو چلاتی ہے جب شرط کا نتیجہ False ہو۔
- *switch* اسٹیٹمنٹ کثیر الانتخاب اسٹیٹمنٹ ہے۔
- نیسٹڈ سیلیکشن سٹرکچر میں، ایک سیلیکشن سٹرکچر، دوسرے سیلیکشن سٹرکچر کے اندر بند ہو سکتا ہے۔

مشق

سوال نمبر 1۔ موزوں جواب کا انتخاب کریں۔

i. کونسی پروگرامنگ اسٹیٹمنٹ فیصلہ کرنے کے لیے استعمال ہوتی ہے؟

(الف) if (ب) break (ج) اسائنمنٹ (د) printf( )

ii. if اسٹیٹمنٹ کی شرط کو ____________ میں بند کیا جاتا ہے۔

(الف) { } (ب) ( ) (ج) [ ] (د) < >

iii. ذیل میں سے کون سا if اسٹیٹمنٹ کا حصہ نہیں ہے؟

(الف) ایسی شرط جس کا نتیجہ کیریکٹر کی صورت میں نکلے۔ (ب) ایسی شرط جس کا نتیجہ True یا False کی صورت میں نکلے۔

(ج) ایک True بلاک (د) ایک False بلاک

iv. case کا اختتام ____________ پر ہوتا ہے۔

(الف) } (ب) { (ج) break (د) default

v. ذیل میں کوڈ کا آؤٹ پٹ کیا آئے گا؟

```
if((1==1)||(2==3))
  printf("Good");
else
  printf("Bad");
```

(الف) Good (ب) Bad (ج) GoodBad (د) No output

سوال نمبر 2۔ C کا ایک پروگرام لکھیں جس میں تین عدد ان پٹ کریں۔ پھر if-else سٹرکچر کی مدد سے ان میں سے سب سے بڑے عدد کو معلوم کریں اور سکرین پر پرنٹ کریں۔

سوال نمبر 3۔ C کا ایک پروگرام لکھیں جو ایک اکیلا کریکٹر ان پٹ کرے اور پھر سکرین پر کوئی پیغام پرنٹ کرے۔ "It is a vowel" یا "It is consonant" یاد رہے *if-else* کا استعمال کرنا ہے۔

سوال نمبر 4۔ سوال نمبر 3 میں دیئے گئے پروگرام کو switch-case کی مدد سے بنائیں۔

سوال نمبر 5۔ درج ذیل پروگرام کا آؤٹ پٹ لکھیں۔

```
#include<stdio.h>
void main( )
{
    int a, b, c;
    a=b=10;
    c=13;
    if((a==b)&&(a>c))
```


NOT FOR SALE

```
    printf("condition is true  \n");
    else
    {
    printf("a is equal to b  \n ");
    printf("a is not greater than c \n");
    }
}
```

سوال نمبر6۔ درج ذیل پروگرام میں موجود غلطیاں معلوم کریں۔

```
#include<stdio.h>
void main( )
{
        int a;
        print("Enter any number =  ");
        scanf("%f",&a);
        if(a>10)
        printf("value is greater than 10");
        else if
        printf("value is less than 10  ")
}
```

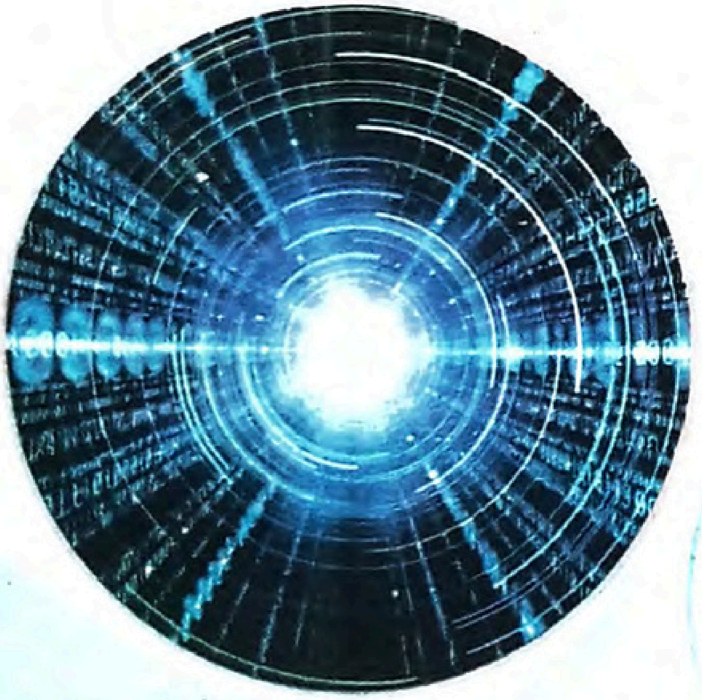
NOT FOR SALE

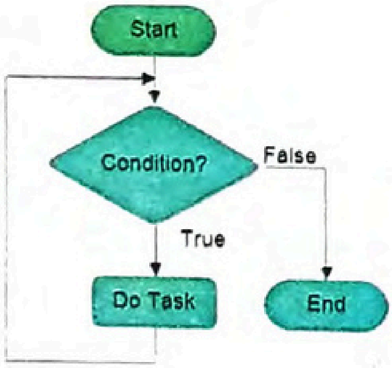
باب نمبر
5

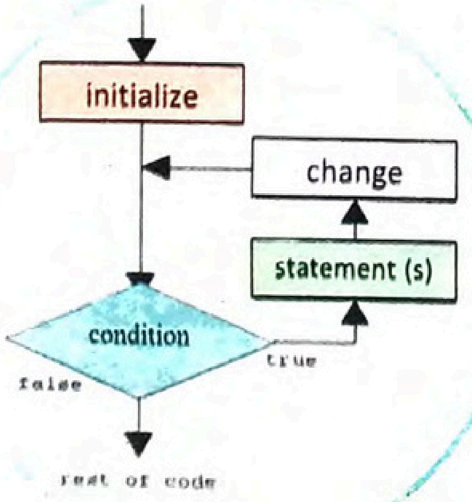
لوپ سٹرکچر

اس باب کی تکمیل کے بعد طلبہ اس قابل ہو جائیں گے کہ وہ:

- لوپ (Loop) کی ساخت تفصیل سے بیان کر سکیں۔
- *for* لوپ کی ساخت جان کر اس کا استعمال کر سکیں۔
- ***while*** لوپ کی ساخت جان کر اس کا استعمال کر سکیں۔
- *do-while* لوپ کی ساخت جان کر اس کا استعمال کر سکیں۔
- break اور continue سٹیٹمنٹس کے استعمال کو تفصیل سے بیان کر سکیں۔
- لوپ کی تمام اقسام میں فرق بیان کر سکیں۔
- نیسڈڈ (Nested) لوپ کے تصور کی وضاحت کر سکیں۔
- باب نمبر 1 میں زیر بحث فلوچارٹس کے لئے کوڈ لکھ سکیں۔







NOT FOR SALE

5.1 تعارف (Introduction)

لوپ کمپیوٹر پروگرامنگ کا ایک سب سے طاقتور تصور ہے جو پروگرامرز کو اپنے پروگرام تخلیق کرنے میں مدد فراہم کرتا ہے جو مؤثر ہوں اور کاروباری ضروریات کو پورا کریں۔ لوپ ہدایات کا ایک ایسا تسلسل ہے جو کہ کسی خاص حالت (Condition) تک پہنچنے تک مستقل طور پر دہرایا جاتا ہے۔ مثال کے طور پر، اگر کوئی پروگرامر پے رول پروسیسنگ کے لئے ایسا پروگرام بناتا ہے جس میں کسی کمپنی میں تنخواہوں اور 1000 ملازمین کے الاؤنس کے حساب کی ضرورت ہوتی ہے۔ کمپیوٹر ایک لوپ کا استعمال کرتے ہوئے، تمام ملازمین کی تنخواہوں اور الاؤنس کی گنتی کرنا شروع کر دیتا ہے، جب تک کہ یہ عمل تمام ملازمین کے لئے پورا نہ کر دے۔ اسی طرح، کسی کمپنی میں موجود تمام ملازمین کی تنخواہوں کا پرنٹ آؤٹ حاصل کرنے کے لیے ایک لوپ کی ضرورت ہوتی ہے، تاکہ اس کے ذریعے تمام ملازمین کے لئے تنخواہوں کی سلپ تیار کی جاسکے اور اسے پرنٹ کیا جاسکے۔ اگر ایسی سرگرمی خود کی جائے تو، اس میں کئی گھنٹے اور دن لگیں گے۔ دوسری طرف لوپ کی مدد سے، وقت اور تھکا دینے والے کاموں کو تیزی سے انجام دیا جاتا ہے۔ کام کو درست انداز میں انجام دینے میں بھی وہ بہت مددگار ثابت ہوتے ہیں۔ C لینگویج میں مختلف قسم کے لوپ اسٹرکچر استعمال ہوتے ہیں۔

5.1.1 لوپ اسٹرکچر (Loop Structure)

ترتیب وار کنٹرول اسٹرکچر میں تمام اسٹیٹمنٹس کو ایک کے بعد دوسری، کی ترتیب میں چلایا جاتا ہے۔ جبکہ دوسری طرف ،کنڈیشنل اسٹرکچر میں کسی بھی شرط کی بنیاد پر کسی اسٹیٹمنٹ کو چلایا جائے گا اور کسی کو نظر انداز کر دیا جائے گا۔ لیکن بعض حالتوں میں، ایک اسٹیٹمنٹ کو یا اسٹیٹمنٹس کے ایک بلاک کو ایک سے زیادہ مرتبہ چلانے کی ضرورت پڑتی ہے۔ اس مقصد کے لئے لوپ اسٹرکچر کا استعمال کیا جاتا ہے۔ ایسا کنٹرول اسٹرکچر جو ایک اسٹیٹمنٹ یا ایک سے زیادہ اسٹیٹمنٹس کو ایک سے زیادہ مرتبہ چلاتا ہے لوپنگ اسٹرکچر (Looping Structure) کہلاتا ہے۔ اس کو تکراری (Repetitive) اسٹرکچر بھی کہا جاتا ہے۔ C میں مختلف قسم کے لوپ دستیاب ہیں۔ لوپس بنیادی طور پر دو مقاصد کے لئے استعمال ہوتے ہیں۔

- کئی اسٹیٹمنٹس کو مقررہ تعداد تک بار بار چلانے کے لئے۔ مثال کے طور پر، ایک یوزر کمپیوٹر سکرین پر اپنا نام 20 بار پرنٹ کرنا چاہتا ہے۔
- کسی شرط کے True یا False نتیجہ کی بنیاد پر کسی اسٹیٹمنٹ یا ایک سے زیادہ اسٹیٹمنٹس کو چلانے کے لیے۔ مثال کے طور پر، ایک یوزر قدرتی اعداد کا ایک مجموعہ اس وقت تک پرنٹ کر سکتا ہے جب تک کہ ایک ویری ایبل X کی قیمت 50 تک نہیں پہنچ جاتی۔

ہمارے پاس C میں تین طرح کے لوپ ہیں۔ ان کا کام تقریباً ایک جیسا ہی ہے، تاہم، یہ مختلف حالات میں استعمال ہو رہے ہیں۔ ضرورت کے مطابق آپ کسی لوپ کا انتخاب کر سکتے ہیں:

- *for* لوپ
- *while* لوپ
- *do-while* لوپ

NOT FOR SALE

5.1.2 for لوپ

for لوپ کی مدد سے ایک یا ایک سے زیادہ اسٹیٹمنٹس کو جتنی دفعہ (Times) چاہیں چلا سکتے ہیں۔ اس لوپ کو پروگرامر بہت زیادہ استعمال کرتے ہیں۔ اس لوپ کا سائنٹکس اور استعمال بہت آسان ہے۔

اس کا سائنٹکس درج ذیل ہے:

```
for (initialization:condition;increment/decrement)
{
        Body of the loop
}
```

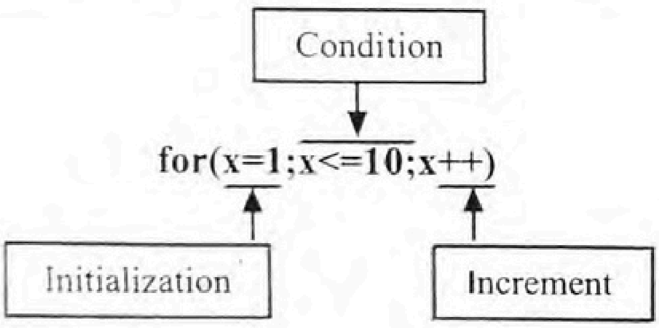
Initialization: اس سے مراد وہ عبارت (Expression) ہے جس سے کاؤنٹر ویری ایبل کی ابتدائی قیمت رکھی جاتی ہے۔ اس کا مطلب ہے کہ یہ حصہ لوپ میں صرف ایک بار چلتا ہے۔

condition: یہ کوئی بھی ریلیشنل ایکسپریشن ہو سکتا ہے۔ لوپ کے اندر تمام اسٹیٹمنٹس اس وقت چلائی جائیں گی جب تک اس condition کا نتیجہ True رہے، جیسے ہی اس condition کا نتیجہ False ہوا تو اسٹیٹمنٹ پھر نہیں چلائی جائیں گی اور کنٹرول لوپ کے بعد پروگرام میں اگلی اسٹیٹمنٹ پر منتقل ہو جائے گا جیسا کہ شکل نمبر 5.1 میں دکھایا گیا ہے۔

Increment/Decrement: لوپ کا یہ حصہ لوپ کے ہر تکرار (Iteration) کے بعد کاؤنٹر ویری ایبل کی قیمت میں تبدیلی کرتا ہے۔

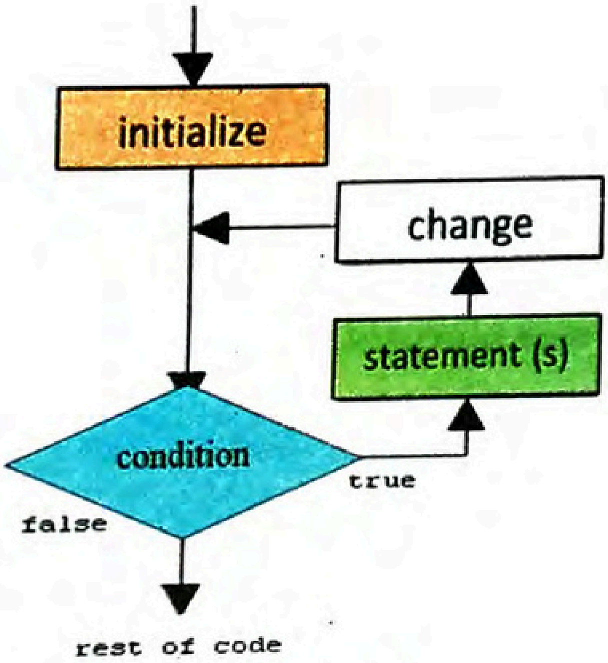
Body of loop: یہ ایک اسٹیٹمنٹ یا اسٹیٹمنٹس کا ایک گروپ ہے جسے بار بار چلایا جاتا ہے یہ قوسین { } کے اندر بند ہوتا ہے۔

مثال کے طور پر:



تکرار کی تعداد لوپ کے اجزاء Initialization، Condition اور Increment/Decrement پر منحصر ہے۔ Initialization والا حصہ ایک دفعہ ہی چلتا ہے جب کنٹرول لوپ میں داخل ہوتا ہے۔ Initialization کے بعد Condition کو چیک کیا جاتا ہے۔ اگر اس کا نتیجہ True ہو تو، کنٹرول لوپ کی باڈی میں داخل ہو کر تمام اسٹیٹمنٹس پر عمل درآمد کرتا ہے۔ پھر Increment/Decrement حصہ پر عمل درآمد کیا جاتا ہے جو کاؤنٹر ویری ایبل کی قیمت کو تبدیل کرتا ہے۔ کنٹرول ایک بار پھر Condition والے حصے پر چلا جائے گا۔ یہ عمل بدستور اس وقت تک جاری رہے گا جب تک condition کا نتیجہ True رہتا ہے جیسے ہی یہ False ہوگا لوپ کو منقطع (Terminate) کر دیا جائے گا۔

NOT FOR SALE



شکل نمبر 5.1 for لوپ کی ترکیب

درج ذیل پروگرام for لوپ کے عمل کو ظاہر کرتا ہے۔

پروگرام 5.1 یہ پروگرام for لوپ کی مدد سے 1 سے لے کر 10 تک تمام اعداد کو پرنٹ کرے گا۔

```
#include <stdio.h>
void main()
{
    int x;
    for (x=1;x<=10;x++)
    printf("%d\t",x);
}
```

آؤٹ پُٹ

```
1    2    3    4    5    6    7    8    9    10
```

پروگرام 5.2 یہ پروگرام 10 سے لیکر 1 تک تمام اعداد کو (الٹی ترتیب میں) پرنٹ کرے گا۔

```
#include <stdio.h>
void main()
{
    int num;
    for (num=10;num>=1;--num)
    printf("%d\t",num);
}
```

آؤٹ پُٹ

```
10    9    8    7    6    5    4    3    2    1
```

پروگرام 5.3 یہ پروگرام کوئی بھی عدد ان پٹ کروائے گا اور پھر اس عدد کا 10 تک پہاڑہ اسکرین پر پرنٹ کرے گا۔

```
# include <stdio.h>
void main()
{
        int tab,n,x;
        printf("\nEnter number for table: ");
        scanf("%d",&n);
        for(x=1;x<=10;x++)
        printf("\n%d X %d = %d",n,x,n*x);
}
```

آؤٹ پٹ

```
Enter number for table: 5
5 × 1 = 5
5 × 2 = 10
5 × 3 = 15
5 × 4 = 20
5 × 5 = 25
5 × 6 = 30
5 × 7 = 35
5 × 8 = 40
5 × 9 = 45
5 × 10 = 50
```

پروگرام 5.4 یہ پروگرام 1 سے لیکر 20 تک تمام طاق اعداد کا حاصل جمع معلوم کر کے اسکرین پر پرنٹ کرے گا۔

```
# include <stdio.h>
void main()
{
      int n, sum;
      sum = 0;
      for (n=1;n<=20;n+=2)
      {
      sum = sum+n;
      }
      printf("\nSUM = %d",sum);
}
```

آؤٹ پٹ

```
SUM = 100
```

5.1.3 while لوپ

while لوپ میں کنڈیشن کو پہلے جانچا جاتا ہے اور اگر اس کا نتیجہ True ہو تو while لوپ میں موجود اسٹیٹمنٹ چلا دی جاتی ہیں۔ یہ عمل اس وقت تک جاری رہتا ہے جب تک کنڈیشن کا نتیجہ False نہ ہو جائے۔ جب کنڈیشن کا نتیجہ False ہو جائے تو کنٹرول لوپ کے باہر آنے والی پہلی اسٹیٹمنٹ پر منتقل ہو جاتا ہے۔ اس لوپ کا استعمال ان صورتوں میں زیادہ موزوں رہتا ہے جب پہلے سے معلوم نہ ہو کہ لوپ نے مطلوبہ نتیجے کے لیے کتنی مرتبہ چلنا ہے۔ while لوپ کا سائنٹکس درج ذیل ہے۔

سائنٹکس:

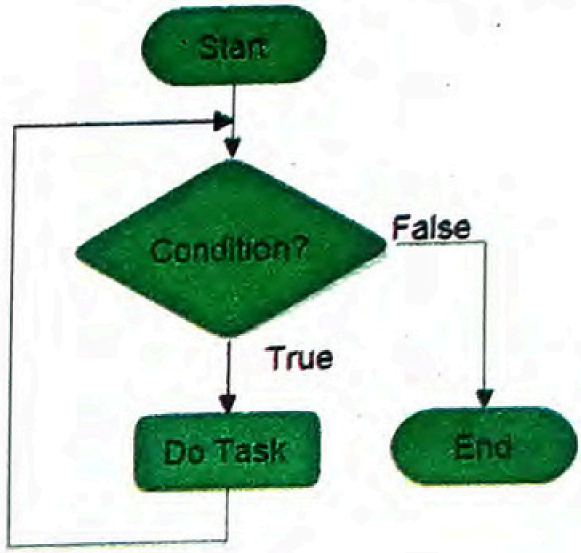
```
while(condition)
{
Body of the loop
}
```


NOT FOR SALE

Condition: یہ ایک ریلیشنل ایکسپریشن ہوتا ہے۔ لوپ کی باڈی صرف اس صورت میں چلے گی جب condition کا نتیجہ True ہو گا اور اگر نتیجہ False ہوا تو پھر لوپ کی باڈی قطعاً نہیں چلے گی۔

Body of Loop: یہ قوسین { } کے درمیان اسٹیٹمنٹ یا اسٹیٹمنٹس کا ایک گروپ ہے جسے بار بار چلانا ہے۔

نوٹ: لوپ کا استعمال کرتے وقت اس بات کا خیال رکھنا ضروری ہے کہ لوپ ویری ایبل کو لوپ کے اندر ہر تکرار (Iteration) پر انکریمنٹ یا ڈیکریمنٹ کر دیا جائے۔ تاکہ لوپ کی کنڈیشن کسی بھی وقت False ہو کر لوپ کو ختم کر سکے ورنہ لوپ غیر معینہ مدت تک چلتی رہے گی۔ لوپ کا کام کرنا شکل نمبر 5.2 میں دکھایا گیا ہے۔



شکل نمبر 5.2 while لوپ کا کام کرنے کا انداز

درج ذیل پروگرام while لوپ کے عمل کو ظاہر کرتا ہے۔

پروگرام 5.4 یہ پروگرام while لوپ کی مدد سے "Pakistan" پانچ دفعہ سکرین پر پرنٹ کرے گا۔

```
# include <stdio.h>
void main()
{
    int n;
    n=1;
    while(n<=5)
    {
    printf("Pakistan\n");
    n++;
    }
}
```

آؤٹ پُٹ

```
Pakistan
Pakistan
Pakistan
Pakistan
Pakistan
```

NOT FOR SALE

پروگرام 5.6 یہ پروگرام while لوپ کی مدد سے 1 سے لے کر 19 تک طاق اعداد کو پرنٹ کرے گا۔

```
# include <stdio.h>
void main()
{
      int n=1; s=0;
      while(s<=100)
      {
      s=s+n;
      printf("%d\n",n);
      n+=2;
      }
}
```

آؤٹ پُٹ

```
1
3
5
7
9
11
13
15
17
19
```

پروگرام 5.7 یہ پروگرام while لوپ کی مدد سے ذیل میں دیئے گئے عددی سلسلے (Series) کا نتیجہ نکال کر اسے سکرین پر پرنٹ کرے گا۔

2+4+6+...+20

```
# include <stdio.h>
void main()
{
      int c, sum=0;
      c=2;
      while(c<=20)
      {
      sum=sum+c;
      c=c+2;
      }
      printf("Sum = %d",sum);
}
```

آؤٹ پُٹ

```
SUM = 110
```

5.1.4 do-while لوپ

do-while لوپ اس وقت تک اسٹیٹمنٹ کو چلاتی رہتی ہے جب تک شرط کا نتیجہ True رہتا ہے۔ اس لوپ میں Condition، لوپ کی باڈی کے بعد لکھی جاتی ہے۔ یہ ایسی صورت حال میں استعمال ہوتی ہے جب لوپ کی باڈی کو بہر صورت کم از کم ایک بار چلانا مقصود ہو چاہے شرط کا نتیجہ پہلی بار غلط (False) ہو جائے۔ do-while لوپ کو شکل نمبر 5.3 میں دکھایا گیا ہے۔

نوٹ: do-while لوپ کا استعمال کرتے وقت اس بات کا خیال رکھنا چاہیئے کہ پہلے، لوپ کے اندر موجود اسٹیٹمنٹ چلائی جاتی ہیں اور پھر شرط کا جائزہ لیا جاتا ہے، اگر نتیجہ درست ہو جائے تو کنٹرول دوبارہ "do" پر چلا جاتا ہے تاکہ اسے بار بار چلایا جا سکے۔ ایسا بار بار ہوتا ہے جب تک کہ شرط کا نتیجہ غلط (False) نہ ہو جائے۔ ایک بار شرط false ہو جائے تب کنٹرول پروگرام میں اگلی اسٹیٹمنٹ پر منتقل ہو جاتا ہے۔

NOT FOR SALE

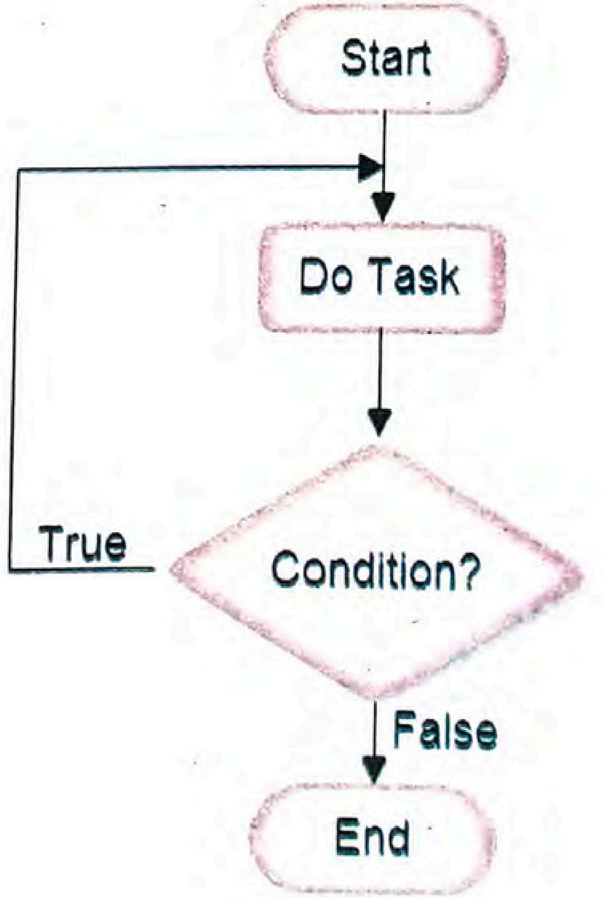
سائنٹکس :

```
do
{
Body of the loop
}
while(condition);
```

do: یہ وہ کلیدی لفظ ہے جو لوپ کے آغاز کی طرف اشارہ کرتا ہے۔

Condition: یہ ریلیشنل ایکسپریشن ہے۔ اگر شرط کا نتیجہ درست ہو تو باڈی کے اندر اسٹیٹمنٹس پر عمل درآمد کیا جاتا رہے گا۔ اگر شرط کا نتیجہ False ہو جائے، تو باڈی کے اندر اسٹیٹمنٹس صرف ایک بار چلیں گی کیونکہ اس میں شرط آخر میں چیک کی جاتی ہے۔ لوپ کو سیمی کالن کے ساتھ ختم ہونا ضروری ہے۔

Body of the loop: یہ قوسین { } کے درمیان اسٹیٹمنٹ یا اسٹیٹمنٹ کا ایک گروپ ہے جس کو بار بار چلانا ہے۔



پروگرام 5.8 یہ پروگرام do-while لوپ کی مدد سے گنتی کے پہلے دس اعداد کو پرنٹ کرے گا۔

```
# include <stdio.h>
void main()
{
    int n;
    n=1;
    do
    {
    printf("%d\n",n);
    n++;
    }
    while(n<=10);
}
```

آؤٹ پُٹ

```
1
2
3
4
5
6
7
8
9
10
```

پروگرام 5.9 یہ پروگرام do-while لوپ کی مدد سے یہ معلوم کرے گا کہ اِن پُٹ کیا گیا عدد طاق ہے یا جفت۔

```
# include <conio.h>
# include <stdio.h>
void main()
{
    int num;
    char choice;
    do
    {
        printf("\nEnter a number to find even or odd");
        scanf("%d", &num);
        if((num%2)==0)
            printf("\nNumber is even");
        else
            printf("\nNumber is odd");
    printf("\nDo you want to check another");
    choice = getche();
    }
    while(choice == 'y');
    printf("\nThank You");
}
```

آؤٹ پُٹ

```
Enter a number to find
even or odd
102
Number is even
```


NOT FOR SALE

5.1.5 break اسٹیٹمنٹ

بریک اسٹیٹمنٹ جب لوپ کی باڈی کے اندر استعمال ہو تو یہ لوپ کو منقطع کر دے گی اور لوپ مطلوبہ تکرار (Iteration) مکمل کئے بغیر ہی ختم ہو جائے گی۔

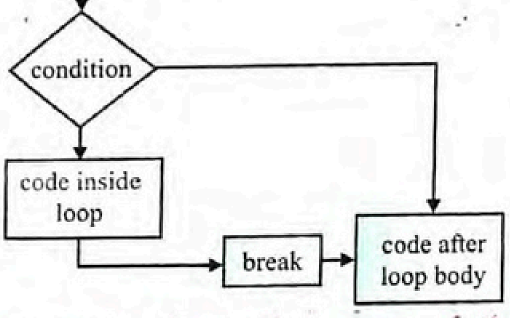
سائنٹکس:

break;

breakاسٹیٹمنٹ کو ان دو صورتوں میں استعمال کیا جاتا ہے۔

فوری طور پر لوپ سے باہر آنے کے لئے breakاسٹیٹمنٹ کا استعمال کریں۔ جب بھی کسی لوپ کے اندر breakاسٹیٹمنٹ کا سامنا کرنا پڑتا ہے تو، کنٹرول اسے ختم کر کے براہ راست لوپ سے نکل آتا ہے۔ لوپ کے اندر اسے کنڈیشنل اسٹیٹمنٹ کے ساتھ استعمال کیا جاتا ہے، تاکہ یہ صرف کسی خاص condition میں ہی چل سکے۔

یہ switch-case سوئچ کیس کنٹرول اسٹرکچر میں case block کے اندر استعمال ہوتا ہے۔ عام طور پر switch-case میں تمام case کے بعد break اسٹیٹمنٹس لکھی جاتی ہے تاکہ اس کے بعد کے case پر عملدرآمد سے بچا جا سکے۔ جب بھی اس کا سامنا switch-case بلاک میں ہوتا ہے، کنٹرول switch-case باڈی سے باہر نکل آتا ہے۔ breakاسٹیٹمنٹ کا کام شکل نمبر 5.4 میں دکھایا گیا ہے۔



شکل نمبر 5.4 breakاسٹیٹمنٹ کا کام کرنے کا طریقہ کار

پروگرام 5.10 یہ پروگرام forلوپ کو استعمال کر کے یوزر سے کی بورڈ کی مدد سے ایک عدد ان پٹ کرے گا اگر نمبر 0 سے بڑا ہوا تو اسے پرنٹ کر کے اگلا عدد اِن پُٹ کر دیا جائے گا۔ اگر یوزر نے 0 یا اس سے چھوٹا یعنی منفی عدد اِن پُٹ کیا تو لوپ منقطع ہو جائے گا۔ یاد رہے کہ یہاں break اسٹیٹمنٹ استعمال ہوئی ہے۔

```
# include <stdio.h>
void main()
{
        int x, num;
        for(x=1;x<=5;x++)
        {
        printf("\nEnter a number: ");
        scanf("%d", &num);
        if(num<=0)
        break;
        printf("\nYou entered  %d\n ",num);
        }
        printf("\n OK");
}
```

آؤٹ پُٹ

```
Enter a number: 3
You entered 3
Enter a number: 0
OK
```

NOT FOR SALE

مذکورہ بالا پروگرام میں لوپ کے اندر break اسٹیٹمنٹ استعمال کی گئی ہے۔ کاؤنٹر ویری ایبل x بتاتا ہے کہ لوپ کو پانچ بار چلنا چاہئے۔ لیکن یہ صرف دو بار ہی چلے گی۔ پہلی بار میں، سکرین پر 3 پرنٹ ہو گا۔

دوسری بار میں ( )printf اسٹیٹمنٹ سوال پرنٹ کرے گی اور یوزر نے 0 اِن پٹ کیا ہے تو اس سے کنٹرول break اسٹیٹمنٹ پر چلا جائے گا۔ break اسٹیٹمنٹ چلنے سے کنٹرول لوپ سے باہر آنے والی پہلی اسٹیٹمنٹ پر منتقل ہو جائے گا۔ اس لیے پیغام (Message) صرف ایک بار ہی پرنٹ ہو گا۔

5.1.6 continue اسٹیٹمنٹ

continue اسٹیٹمنٹ کنٹرول کو لوپ کے آغاز پر واپس منتقل کر دیتی ہے۔ یہ لوپ کی باڈی کے اندر استعمال ہوتی ہے۔ جب بھی، ایک لوپ کے اندر continue اسٹیٹمنٹ کا سامنا کرنا پڑتا ہے تو، کنٹرول اگلے تکرار کے لئے لوپ کے آغاز پر براہ راست منتقل ہو جاتا ہے، موجودہ iteration کے لیے لوپ کی باڈی کے اندر اسٹیٹمنٹ کی تعمیل کو چھوڑ دیتا ہے۔ continue اسٹیٹمنٹ کو شکل نمبر 5.5 میں دکھایا گیا ہے۔

سائنٹکس:

continue;

continue اسٹیٹمنٹ break اسٹیٹمنٹ کے بالکل مخالف ہے۔ اسے بائی پاس اسٹیٹمنٹ بھی کہا جاتا ہے۔

پروگرام 5.11 ایک پروگرام لکھیں اور اس میں while لوپ کے اندر continue اسٹیٹمنٹ کے استعمال کو سمجھیں۔

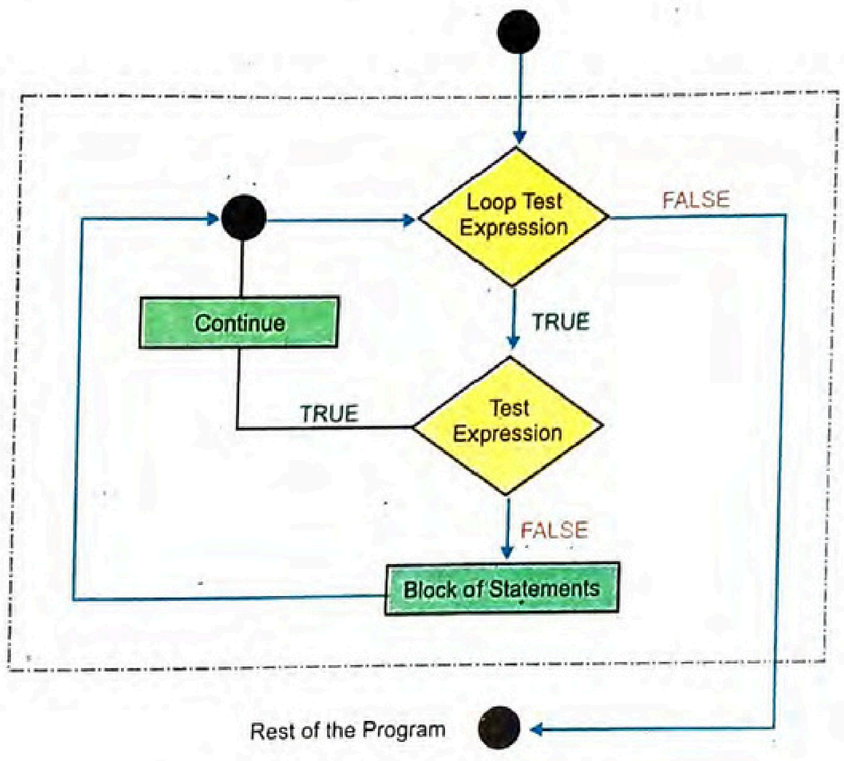
```
#include <stdio.h>
void main()
{
    int c;
    c=1;
    while(c<=5)
    {
    printf("Pakistan\n");
    c++;
    continue;
    printf("Islamabad");
    printf("Peshawar");
    }
}
```

آؤٹ پٹ

```
Pakistan
Pakistan
Pakistan
Pakistan
Pakistan
```

مذکورہ پروگرام میں، لوپ کی پہلی 3 اسٹیٹمنٹ 5 مرتبہ چلائی جائیں گی۔ جبکہ continue اسٹیٹمنٹ کے بعد کی دو اسٹیٹمنٹ نہیں چلیں گی۔ کیونکہ ہر دفعہ جب continue اسٹیٹمنٹ چلتی ہے تو یہ کنٹرول کو لوپ کی باڈی کے بالکل آغاز پر واپس بھیج دیتی ہے۔

NOT FOR SALE



شکل نمبر 5.5 continue اسٹیٹمنٹ کا طریقہ کار

5.1.7 مختلف لوپ سٹرکچرز میں فرق

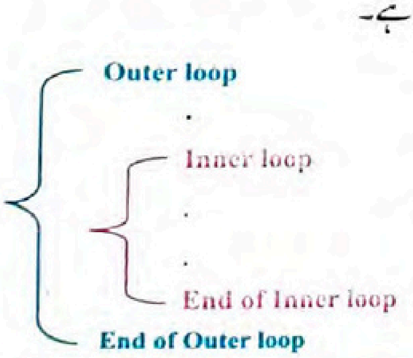
ذیل میں for، while اور do-while کے فرق کو بیان کیا گیا ہے۔

for لوپ: اس کے استعمال کو ترجیح دی جاتی ہے جب پروگرامر پہلے سے جانتا ہو کہ لوپ کی باڈی پر کتنی بار عملدرآمد کیا جائے گا۔

while لوپ: یہ ایسے حالات میں استعمال کرنے کے لئے موزوں ہے جہاں پروگرامر کو پتہ نہیں ہوتا ہے کہ لوپ کی باڈی کو کتنی مرتبہ چلنا ہے۔ اِس لوپ میں کنڈیشن لوپ کی باڈی سے پہلے آتی ہے۔ پہلے، کنڈیشن کی جانچ کی جاتی ہے اور پھر مطلوبہ نتیجہ حاصل کرنے کے لئے لوپ کی باڈی کے اندر اسٹیٹمنٹ کو چلایا جاتا ہے۔

do-while لوپ: یہ ایسے حالات میں استعمال کرنے کے لئے موزوں ہے جہاں پروگرامر چاہتا ہے کہ لوپ کو کم سے کم ایک بار ضرور چلایا جائے۔ do-while لوپ میں، لوپ کی باڈی ٹیسٹ کی condition سے پہلے آتی ہے۔ لوپ کی باڈی کو چلایا جاتا ہے اور پھر اس کی شرط کی جانچ کی جاتی ہے۔ اس لوپ میں، شرط سیمی کالن یعنی اسٹیٹمنٹ ٹرمینیٹر کے ساتھ ختم کر دی جاتی ہے۔

5.1.8 نیسٹڈ لوپ

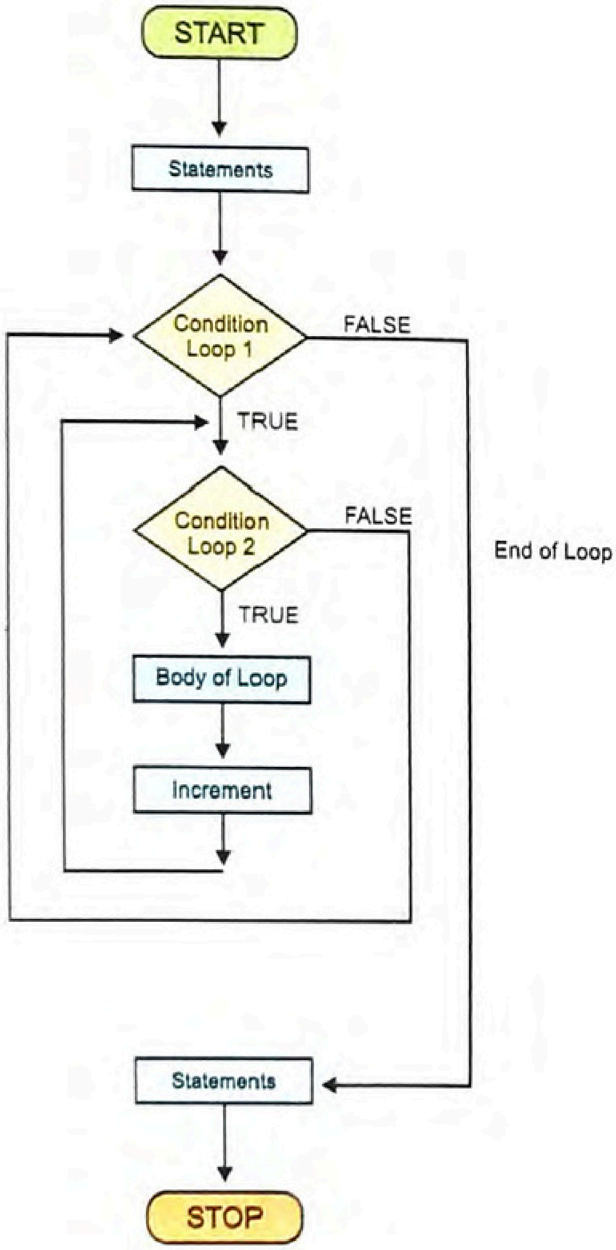


ایک لوپ کے اندر دوسری لوپ کو نیسٹڈ لوپ کہتے ہیں۔ بیرونی لوپ کے اندر اندرونی لوپ مکمل طور پر بند ہوتی ہے۔ ذیل میں دی گئی شکل نیسٹڈ لوپ کے استعمال کی وضاحت کرتی ہے۔

NOT FOR SALE

نیسٹڈ لوپ کا سائنٹکس ذیل میں واضح کیا گیا ہے۔

```
for (initialize counter; test condition; ++ or --)
  {
        for (initialize counter; test condition; ++ or --)
        {
        .        // inner for loop
        }
        .        // outer for loop
  }
```



شکل نمبر 5.6 نیسٹڈ لوپ

NOT FOR SALE

پروگرام 5.12 یہ پروگرام نیسٹڈ لوپ کی مدد سے آؤٹ پٹ دکھائے گا۔

```
#include <stdio.h>
void main()
{
        int i,j;
        for(i=1;i<=5;i++)
        {
        for(j=1;j<=i;j++)
        printf("%d\t",j);
        }
        printf("\n");
        }
}
```

آؤٹ پٹ

```
1
1  2
1  2  3
1  2  3  4
1  2  3  4  5
```

5.1.9 باب 1 کے فلوچارٹس کے لئے کوڈز

پروگرام 5.13 یہ پروگرام ایک عدد کا Factorial معلوم کرے گا۔

```
#include <iostream>
using namespace std;
int main() {
  int i, n, factorial = 1;
  cout<<"Enter a positive integer: ";
  cin>>n;
  for (i = 1; i <= n; ++i) {
     factorial *= i;   // factorial = factorial * i;
  }
  cout<< "Factorial of "<<n<<" = "<<factorial;
              return 0;
}
```

آؤٹ پٹ

```
Enter a positive integer: 5
Factorial of 5 = 120
```

پروگرام 5.14 پروگرام 1 سے 100 تک طاق اعداد کو پرنٹ کرے گا۔

```
#include<iostream>
using namespace std;
int main()
{
      int i=1;
      cout<<"All Odd numbers from 1 to 100 "<<endl<<endl;
      while(i<=100)
      {
       if(i%2==1)
```

NOT FOR SALE

```
    {
        cout<<i<<"\t";
      }
    i++;
    }
    getch();
    return 0;
}
```

آؤٹ پٹ

```
All odd numbers from 1 to 100
1 3 5 7 9 11 13 15 17 19 21 23 25 27 2931 33 35 37 39 41 43 45 47 49 51 53
55 57 5961 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99
```

پروگرام 5.15 یہ پروگرام 1 سے 50 کے درمیان جفت اعداد کا حاصل جمع معلوم کرے گا۔

```
#include <iostream>
using namespace std;
int main() {
  int i, n, sum = 0;
    Take input from user.
  cout << "Print sum of even numbers till : ";
  cin >> n;
  for(i = 1; i <= n; i++) {
      Check for even or not.
    if((i % 2) == 0) {
       sum += i;
    }
  }
  cout << endl << "Sum of even numbers from 1 to 50 " << n << " is : " << sum;
  return 0;
}
```

آؤٹ پٹ

```
Print sum of even numbers till: 50
Sum of even numbers from 1 to 50 is: 1275
```


NOT FOR SALE

اہم نکات

- لوپ ہدایات کا ایک سلسلہ ہے جو کسی خاص حالت تک پہنچنے تک بار بار چلایا جاتا ہے۔
- for لوپ ایک مخصوص تعداد میں ایک یا زیادہ اسٹیٹمنٹ کو چلاتی ہے۔
- while لوپ ایک یا زیادہ اسٹیٹمنٹ پر اس وقت تک عمل درآمد کرتا رہتا ہے جب تک condition کا نتیجہ True رہتا ہے۔
- do-while لوپ اس وقت تک اسٹیٹمنٹ کو چلاتی رہتی ہے جب تک شرط کا نتیجہ True رہتا ہے۔
- break اسٹیٹمنٹ جب لوپ کی باڈی کے اندر استعمال ہو تو کنٹرول کو لوپ سے باہر بھیج دیتا ہے۔
- Continue اسٹیٹمنٹ کنٹرول کو لوپ کے آغاز پر واپس منتقل کرتا ہے۔
- ایک لوپ کے اندر دوسرے لوپ کو نیسٹڈ لوپ کہتے ہیں۔

مشق

سوال نمبر 1۔ موزوں جواب کا انتخاب کریں۔

i. C میں کتنی قسم کے لوپ سٹرکچر ہیں؟

(الف) 4 (ب) 3 (ج) 2 (د) 6

ii. کون سی لوپ ہمیشہ کم از کم ایک مرتبہ ضرور چلتی ہے؟

(الف) do-while (ب) for (ج) while (د) nested

iii. کس لوپ میں شرط ہمیشہ لوپ کی باڈی سے پہلے آتی ہے؟

(الف) while loop (ب) do-while loop
(ج) for loop (د) (الف) اور (ج) دونوں

iv. کس لوپ میں شرط ہمیشہ لوپ کی باڈی کے بعد آتی ہے۔

(الف) for (ب) while (ج) do-while (د) None

v. ذیل میں دی گئی کون سی اسٹیٹمنٹ لوپ کی باڈی سے باہر نکلنے کے لیے استعمال ہوتی ہے؟

(الف) break (ب) terminate (ج) exit (د) (الف) اور (ب) دونوں

vi. اگر ایک لوپ کو دوسری لوپ کے اندر بند کر دیا جائے تو اسے کیا کہا جائے گا؟

(الف) counter loop (ب) complex loop
(ج) outer loop (د) nested loop

NOT FOR SALE

vii. کونسی اسٹیٹمنٹ کنٹرول کو لوپ کی باڈی کے بالکل آغاز میں لے جاتی ہے؟

(الف) continue (ب) break (ج) while (د) for

viii. جب پہلے سے یہ پتہ ہو کہ لوپ نے کتنی دفعہ چلنا ہے تو ایسی صورت میں کونسی لوپ استعمال کرنا چاہیئے؟

(الف) do-while (ب) for (ج) while (د) nested

سوال نمبر۔2 لوپنگ سٹرکچر کی مختلف اقسام پر مثالوں کی مدد سے بحث کریں۔

سوال نمبر۔3 do-while اور while لوپ میں مثالوں کی مدد سے تفصیل سے فرق بیان کریں۔

سوال نمبر۔4 C میں ایک پروگرام لکھیں جو 1 اور 100 کے درمیان آنے والے تمام جفت اعداد کا حاصل ضرب معلوم کرے۔ یاد رہے کہ while لوپ کا استعمال کریں۔

سوال نمبر5۔ for لوپ استعمال کرتے ہوئے ایک پروگرام لکھیں جو A سے لیکر Z تک تمام حروف تہجی پرنٹ کرے۔

سوال نمبر6۔ نیسٹڈ لوپ کی مدد سے ایک پروگرام لکھیں جو ذیل میں دکھائی گئی آوٹ پٹ کو سکرین پر پرنٹ کرے۔

```
            a
          a a
        a a a
      a a a a
    a a a a a
```

سوال نمبر7۔ for لوپ کو استعمال کر کے C میں ایک پروگرام لکھیں جو 1 سے لیکر 50 تک تمام طاق اعداد کا حاصل جمع معلوم کرکے سکرین پر پرنٹ کر سکے۔

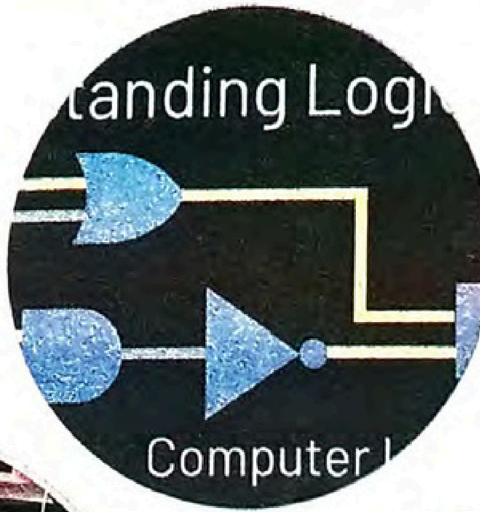
NOT FOR SALE

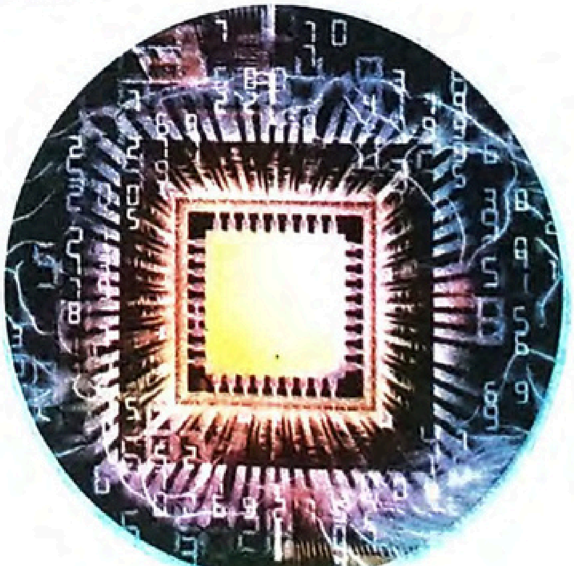
باب نمبر 6

کمپیوٹر لاجک اور لاجک گیٹس

اس باب کی تکمیل کے بعد طلبہ اس قابل ہو جائیں گے کہ وہ:

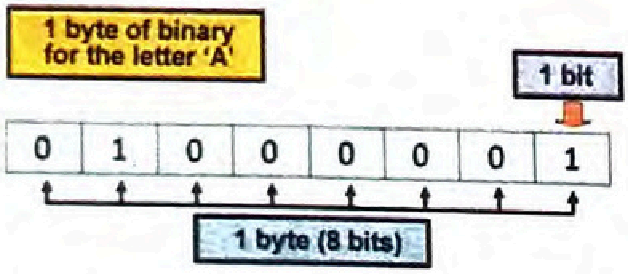
- لوپ (Loop) کی ساخت تفصیل سے بیان کر سکیں۔
- *for* لوپ کی ساخت جان کر اس کا استعمال کر سکیں۔
- *while* لوپ کی ساخت جان کر اس کا استعمال کر سکیں۔
- *do-while* لوپ کی ساخت جان کر اس کا استعمال کر سکیں
- break اور continue سٹیٹمنٹس کے استعمال کو تفصیل سے بیان کر سکیں۔
- لوپ کی تمام اقسام میں فرق بیان کر سکیں۔
- نیسٹڈ (Nested) لوپ کے تصور کی وضاحت کر سکیں۔
- باب نمبر 1 میں زیر بحث فلو چارٹس کے لئے کوڈ لکھ سکیں۔





6.1 کمپیوٹر میں ڈیٹا کا اظہار (Data Representation in a Computer)

ڈیٹا درحقیقت خام اعداد و شمار کا نام ہے۔ ڈیٹا کو حقائق، تصورات یا ہدایات کی نمائندگی کے طور پر بیان کیا جا سکتا ہے جو باقاعدہ انسانی یا الیکٹرانک مشین کے ذریعہ مواصلات، تشریح یا پروسیسنگ کے لئے موزوں ہونا چاہئے۔ ڈیجیٹل کمپیوٹرز میں، یوزر ان پٹ کو پہلے الیکٹریکل پلس (Electrical Pulse) کے طور پر تبدیل اور منتقل کیا جاتا ہے جس کی نمائندگی دو منفرد حالتیں آن (On) اور آف (Off) کر سکتی ہیں۔ "آن حالت" کو "1" اور آف حالت کو "0" سے ظاہر کیا جاتا ہے۔ آن اور آف کے تسلسل سے بجلی کے سگنل بنتے ہیں جن کو کمپیوٹر سمجھ سکتا ہے۔ بائنری نمبر سسٹم میں ایک ہندسے کو بٹ (Bit) اور آٹھ بٹس کے امتزاج کو بائٹ (Byte) کہتے ہیں۔



بائٹ کمپیوٹر میموری کا بنیادی یونٹ ہے جو ڈیٹا کی نمائندگی کے لئے استعمال ہوتا ہے۔ ایک بائٹ ڈیٹا کے ایک کیریکٹر کو ظاہر کرتا ہے۔ یہ ایک حرف تہجی، کوئی عدد یا کوئی مخصوص علامت ہو سکتی ہے۔ ڈیٹا انہی حروف تہجی، اعداد اور علامات کا مجموعہ ہے جنہیں کسی خاص مقصد کے تحت اکٹھا کیا جاتا ہے۔

مثال کے طور پر، جب حرف A کمپیوٹر میں اِن پُٹ کیا جاتا ہے تو اسے پہلے بائنری کوڈ یعنی 01000001 میں تبدیل کیا جاتا ہے۔ اسی طرح باقی تمام حروف بھی کمپیوٹر کو پروسیسنگ کے لئے فراہم کیے جاتے ہیں۔

6.2 لاجک گیٹس (Logic Gates)

لاجک گیٹس ڈیجیٹل کمپیوٹر کی ساخت کی بنیادی اکائی ہیں۔ لاجک گیٹس وولٹیج کی دو حدوں کے درمیان چلتے ہیں اور ڈیجیٹل سگنلوں جو بائنری ڈجٹس 0 اور 1 سے ظاہر کیے جاتے ہیں، لاجک گیٹس / لاجیکل پروسیسنگ اور ایک بائنری آؤٹ پُٹ 0 یا 1 حاصل کرنے کے لئے استعمال کیے جاتے ہیں۔ آسان الفاظ میں، لاجک گیٹس ڈیجیٹل سسٹم میں الیکٹرانک سرکٹس ہیں۔

6.2.1 ڈیجیٹل لاجک اور لاجک گیٹس (Digital Logic and Logic Gates)

ڈیجیٹل لاجک ڈیجیٹل سرکٹس بنانے کے لئے استعمال ہوتے ہیں جن میں بہت سارے لاجک گیٹس شامل ہوتے ہیں۔

لاجک گیٹ ایک ایسا الیکٹرانک آلہ ہے جو ڈیجیٹل سرکٹس کے لئے بلڈنگ بلاک کا کام کرتا ہے۔ لاجک گیٹس بنیادی لاجیکل افعال سرانجام دیتے ہیں جو ڈیجیٹل سرکٹس کی بنیاد ہیں۔ آج کل زیادہ تر الیکٹرانک آلات کے اندر لاجک گیٹس ہی کا استعمال کیا جاتا ہیں۔ مثال کے

NOT FOR SALE

طور پر، لاجک گیٹس اسمارٹ فونز، ٹیبلٹس یا میموری آلات میں موجود ٹیکنالوجیز میں استعمال کیے جاتے ہیں۔

ایک سرکٹ میں، لاجک گیٹس اپنی اِن پُٹ سے منتقل ہونے والے ڈیجیٹل سگنلوں کی بنیاد پر فیصلے کرتے ہیں۔ زیادہ تر لاجک کی دو یا دو سے زیادہ اِن پُٹ ہوتی ہیں اور ایک اکیلی آؤٹ پُٹ فراہم کرتا ہے جو Low یا High ہوتی ہے۔ لاجک گیٹس بنیادی طور پر الجبرا پر مبنی ہوتے ہیں۔ ہر ٹرمینل دو بائنری حالتوں True یا False میں سے ایک پر مشتمل ہوتا ہے۔ False کو 0 سے اور True کو ظاہر کیا جاتا ہے۔ لاجک گیٹ کو لائٹ سوئچ سے تشبیہ دی جا سکتی ہے، جس میں ایک حالت آف (0) اور دوسری آن (1) ہوتی ہے طور پر انٹیگریٹڈ سرکٹس (CI) میں لاجک گیٹس استعمال ہوتے ہیں۔

6.2.2 بنیادی لاجک گیٹس (Basic Logic Gates)

ہر سنگل لاجک گیٹس کے بنیادی عوامل کو عام حسابی عمل جمع (Addition) اور ضرب (Multiplication) کی طرح آ سے سمجھا جا سکتا ہے۔ لاجک گیٹ کی بنیاد پر، مختلف لاجیکل آپریشنز کی آؤٹ پُٹ مختلف ہوتی ہیں۔ ہر لاجک گیٹ ایک ٹروتھ ٹیبل کی پیـ کرتا ہے جو اِن پُٹ اور متعلقہ حاصل شدہ آؤٹ پُٹ کے ممکنہ امتزاج فراہم کرتا ہے۔

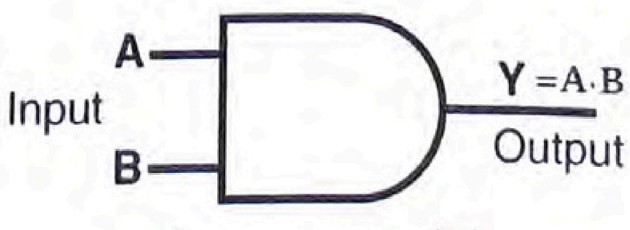
ڈیجیٹل سسٹمز میں آپریشن انجام دینے میں کئی بنیادی لاجک گیٹس استعمال کیے جاتے ہیں۔ AND، OR اور NOT تین بنیـ گیٹس ہیں کیونکہ یہی تین گیٹس مختلف قسم کے دوسرے پیچیدہ گیٹس بنانے کے لئے استعمال ہوتے ہیں۔

AND گیٹ

AND گیٹ ایک لاجیکل گیٹ ہے جس میں دو یا دو سے زیادہ اِن پُٹ اور ایک اکیلی آؤٹ پُٹ ہوتی ہے۔ AND گیٹ لاجـ ضرب کے اصولوں پر کام کرتا ہے۔ AND گیٹ کی آؤٹ پُٹ صرف اُس وقت High یعنی 1 ہو گی جب اسکی تمام اِن پُٹس gh ہونگی۔ (یعنی اگر ایک اِن پُٹ بھی Low یا 0 ہو گی تو آؤٹ پُٹ Low ہو گی)۔ AND گیٹ میں کسی بھی طرح کی اِن پُٹ ہو سکتی ہیں، 2 پُٹس اور 3 اِن پُٹس والے گیٹ سب سے عام ہیں۔

اگر A اور B دو اِن پُٹ ہیں، تو آؤٹ پُٹ Y کو ریاضی کے مطابق ہم ایسے ظاہر کر سکتے ہیں Y = A.B یہاں پر ڈاٹ (.) ND کے عمل کو ظاہر کر رہا ہے۔ AND گیٹ کو درج ذیل علامت سے ظاہر کیا جاتا ہے۔

علامت:



شکل نمبر 6.1 دو اِن پُٹ والا AND گیٹ

جبکہ A، B اور Y بائنری ویری ایبلز ہیں۔

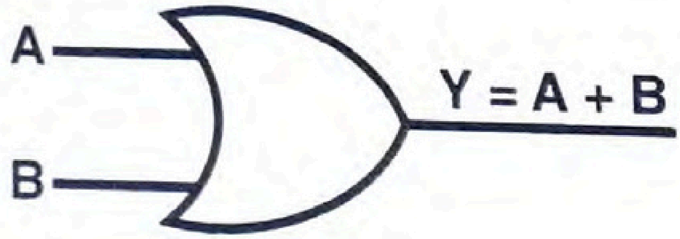
NOT FOR SALE

OR گیٹ

OR گیٹ ایک لاجک گیٹ ہے جو لاجیکل جمع کے لیے استعمال ہوتا ہے۔ لاجیکل OR آپریشن کی آؤٹ پُٹ صرف اُس صورت میں High(1) ہو گی جب اس کی کوئی ایک اِن پُٹ High ہو (یعنی اگر ایک ہی اِن پُٹ High ہو تو آؤٹ پُٹ High ہو گی) اور اگر کوئی ایک اِن پُٹ بھی High نہیں ہے تو، آؤٹ پُٹ Low(0) کے نتائج بر آمد ہوں گے۔

اگر A اور B دو اِن پُٹ ہیں، تو آؤٹ پُٹ Y کو ریاضی کے مطابق ہم ایسے ظاہر کر سکتے ہیں $Y = A + B$ یہاں پلس سائن (+) OR عمل کی نشاندہی کرتا ہے۔ OR گیٹ کی نمائندگی مندرجہ ذیل علامت سے ہوتی ہے۔

علامت:



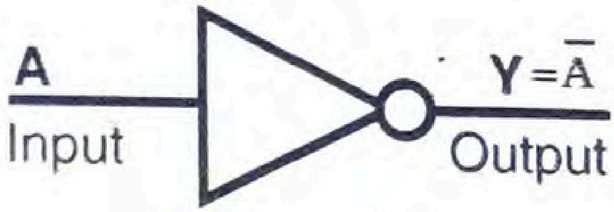
شکل نمبر 6.2 دو اِن پُٹ والا OR گیٹ

NOT گیٹ

NOT گیٹ کا استعمال کسی بھی اِن پُٹ ویلیو کو معکوس (Reverse) کرنے کے لئے استعمال کیا جاتا ہے۔ یہ ایک ایسا گیٹ ہے جس کی آؤٹ پُٹ اِن پُٹ کے اُلٹ ہوتی ہے۔ NOT گیٹ ایک اِن پُٹ لیتا ہے اور ایک ہی آؤٹ پُٹ دیتا ہے۔ اگر NOT گیٹ میں اِن پُٹ '0' ہے تو آؤٹ پُٹ '1' ہو گا۔ اگر اِن پُٹ '1' ہے تو آؤٹ پُٹ '0' ہو گا۔

NOT گیٹ چونکہ اِن پُٹ کو معکوس کر دیتا ہے اس لیے اسے انورٹر (Inverter) بھی کہا جاتا ہے۔

A اِن پُٹ ہے اور Y، NOT گیٹ کا آؤٹ پُٹ ہے، NOT گیٹ کا بلاک ڈایاگرام یا منطق ڈایاگرام درج ذیل ہے:



شکل نمبر 6.3 NOT گیٹ

6.2.3 ٹروتھ ٹیبل (Truth Table)

یہ ایک ٹیبل ہے جس میں اِن پُٹ ویلیوز اور ان کے مطابق آؤٹ پُٹ ویلیوز کے تمام ممکنہ امتزاج کو لکھا ہوتا ہے۔ اگر اِن پُٹ ویریابلز کی تعداد n ہو تو اِن پُٹ کے کل 2 ممکنہ امتزاج ہوں گے۔ ٹروتھ ٹیبل کا استعمال بولین فنکشن یا ایکسپریشن یا لاجک گیٹس کے تمام ممکنہ امتزاج کی نمائندگی کرنے کے لئے کیا جاتا ہے۔ بولین فنکشن / ایکسپریشن بولین ویری ایبل، کانسٹنٹ اور بولین آپریٹرز کا مجموعہ ہے۔

NOT FOR SALE

ایک ٹروتھ ٹیبل میں درج ذیل خصوصیات ہونی چاہئیں۔

- ٹروتھ ٹیبل قطاروں اور کالموں پر مشتمل ہوتا ہے۔
- یہ ڈیجیٹل لاجک سرکٹ کی اِن پُٹ اور آؤٹ پُٹ کے درمیان ربط یا تعلق کو ظاہر کرتا ہے۔
- یہ LOW کے لئے 0 اور HIGH کے لئے 1 کا استعمال کرتے ہوئے اِن پُٹ کے ہر ممکنہ امتزاج کو ظاہر کرتا ہے۔
- تمام ممکنہ اِن پُٹس کو بائیں کالم اور آؤٹ پُٹ کو سب سے دائیں کالم میں دکھایا جاتا ہے۔
- اِن پُٹ والے کالم کو بائنری نمبر کی ترتیب میں اس طرح لکھا جاتا ہے تاکہ اِن پُٹ کے نمبر کو ظاہر کیا جا سکے جس میں اِن پُٹ کی تعداد کے برابر متعدد بٹس ہوتے ہیں۔ مثال کے طور پر، 2 اِن پُٹ کے لئے، ممکنہ امتزاج $2^2 = 4$ ہوں گے۔ 3 اِن پُٹ کے لیے ممکنہ امتزاج $2^3 = 8$ ہوں گے، وغیرہ۔

AND، OR اور NOT کے لیے ٹروتھ ٹیبل

2 اِن پُٹ والے AND گیٹ کا ٹروتھ ٹیبل

فرض کیا X اور Y دو اِن پُٹ ہیں اور F ایک AND گیٹ سے آؤٹ پُٹ ہے، اس طرح AND گیٹ کا ٹروتھ ٹیبل درج ذیل ہے۔

اِن پُٹ		آؤٹ پُٹ
X	Y	F=X.Y
0	0	0
0	1	0
1	0	0
1	1	1

جدول نمبر 6.1 دو اِن پُٹ والے AND گیٹ کا ٹروتھ ٹیبل

3 اِن پُٹ والے AND گیٹ کا ٹروتھ ٹیبل

فرض کیا X، Y اور Z تین اِن پُٹ ہیں اور F ایک AND گیٹ سے آؤٹ پُٹ ہے، اس طرح AND گیٹ کا ٹروتھ ٹیبل درج ذیل ہے۔

اِن پُٹ			آؤٹ پُٹ
X	Y	Z	F=X.Y.Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

جدول نمبر 6.2 تین اِن پُٹ والے AND گیٹ کا ٹروتھ ٹیبل

NOT FOR SALE

2 اِن پُٹ والے OR گیٹ کا ٹروتھ ٹیبل

فرض کیا X اور Y دو اِن پُٹ ہیں اور F ایک OR گیٹ سے آؤٹ پٹ ہے، ٹروتھ ٹیبل درج ذیل ہے۔

اِن پُٹ		آؤٹ پُٹ
X	Y	F=X+Y
0	0	0
0	1	1
1	0	1
1	1	1

جدول نمبر 6.3 دو اِن پٹ والے OR گیٹ کا ٹروتھ ٹیبل

3 اِن پُٹ والے OR گیٹ کا ٹروتھ ٹیبل

فرض کیا X، Y اور Z تین اِن پُٹ ہیں اور F ایک OR گیٹ سے آؤٹ پُٹ ہے، پھر OR گیٹ کا ٹروتھ ٹیبل درج ذیل ہے۔

3 اِن پُٹ والے OR گیٹ کا ٹروتھ ٹیبل

اِن پُٹ			آؤٹ پُٹ
X	Y	Z	F=X+Y+Z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

جدول نمبر 6.4 تین اِن پٹ والے OR گیٹ کا ٹروتھ ٹیبل

NOT گیٹ کا ٹروتھ ٹیبل

اِن پُٹ	آؤٹ پُٹ
X	$F=\overline{X}$
0	1
1	0

جدول نمبر 6.5 NOT گیٹ کا ٹروتھ ٹیبل

NOT FOR SALE

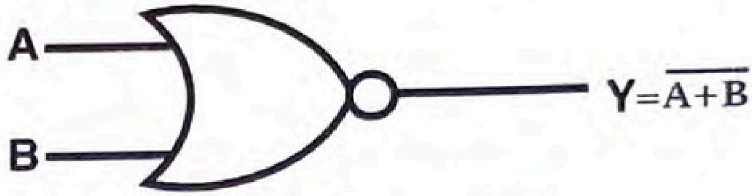
6.2.4 مزید لاجک گیٹس اور ان کے ٹروتھ ٹیبل

NOR گیٹ

ایک NOR گیٹ ("NOT OR" گیٹ") ایک لاجک گیٹ ہے جس کی تمام اِن پُٹس اگر 0 ہوں اور اس کی آؤٹ پُٹ (1) آئے تو اسے NOR گیٹ کہتے ہیں۔ لہٰذا NOR گیٹ ایک OR گیٹ کا اُلٹ ہوتا ہے، اور اس کا سرکٹ ایک OR گیٹ کو ایک NOT گیٹ سے مربوط کر کے تیار کیا جاتا ہے۔ بالکل ایک OR گیٹ کی طرح، کسی NOR گیٹ میں بہت سی اِن پُٹ ہو سکتی ہے لیکن صرف ایک آؤٹ پُٹ تیار ہوتی ہے۔

اگر A اور B دو اِن پُٹ ہیں، تو آؤٹ پُٹ Y کو ایسے لکھا جا سکتا ہے۔ $Y = \overline{A + B}$

یہاں یہ علامت ــــــ کمپلیمنٹ یا معکوس کو ظاہر کر رہی ہے۔ NOR گیٹ کی شکل اور ٹروتھ ٹیبل ذیل میں دکھایا گیا ہے۔



شکل نمبر 6.4 دو اِن پُٹ والا NOR گیٹ

NOR گیٹ کا ٹروتھ ٹیبل

اِن پُٹ		آؤٹ پُٹ
A	B	$Y = \overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

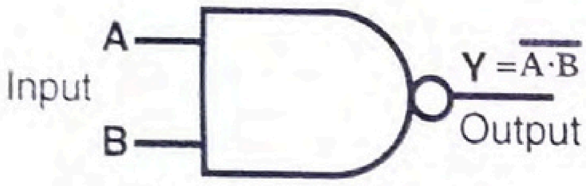
جدول نمبر 6.6 دو اِن پُٹ والے NOR گیٹ کا ٹروتھ ٹیبل

NAND گیٹ

NAND گیٹ ("AND اور NOT گیٹ") ایک منطقی گیٹ ہے جو LOW آؤٹ پُٹ (0) دیتا ہے صرف اس صورت میں جب اس کے سارے اِن پُٹ True ہوں، اور دوسری صورت میں High آؤٹ پُٹ (1)۔ لہٰذا NAND گیٹ ایک AND گیٹ کا معکوس ہے، اور اس کا سرکٹ ایک AND گیٹ کو ایک NOT گیٹ سے جوڑ کر تیار کیا گیا ہے۔ بالکل ایک AND گیٹ کی طرح، ایک NAND گیٹ میں ایک سے زیادہ اِن پُٹ ہو سکتی ہے لیکن آؤٹ پُٹ صرف ایک ہی ہو گی۔ NAND گیٹ لاجک NAND آپریشن انجام دیتا ہے۔ NAND گیٹ کو یونیورسل گیٹ (NOR گیٹ کی طرح) کے نام سے جانا جاتا ہے، جس کا مطلب ہے کہ یہ ایک قسم کا لاجیکل گیٹ ہے جو کسی بھی دوسرے گیٹ کو استعمال کیئے بغیر کسی بھی بولین فنکشن کو نافذ کر سکتا ہے۔

NOT FOR SALE

اگر B،A ایک NAND گیٹ کے دو اِن پُٹ ہیں اور Y اس کا آؤٹ پُٹ ہے تو، NAND گیٹ کو بلاک ڈایاگرام اور ٹروتھ ٹیبل ذیل کی شکل میں دکھایا گیا ہے۔



شکل نمبر 6.5 دو اِن پُٹ والا NAND گیٹ

NAND گیٹ کا ٹروتھ ٹیبل

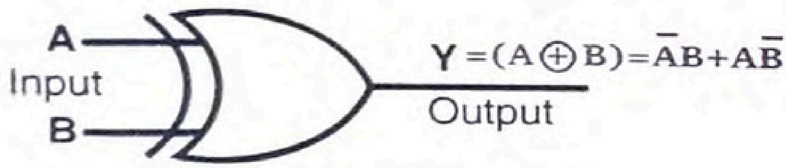
اِن پُٹ		آؤٹ پُٹ
A	B	Y= $\overline{A+B}$
0	0	1
0	1	1
1	0	1
1	1	0

جدول نمبر 6.7 دو اِن پُٹ والے NAND گیٹ کا ٹروتھ ٹیبل

XOR گیٹ (Exclusive-OR Gate)

وہ گیٹ جس کی آؤٹ پُٹ '0' ہوتی ہے جب ساری اِن پُٹ 0 یا 1 کی ہوتی ہے تو اسے XOR گیٹ کہا جاتا ہے۔ لہٰذا XOR گیٹ ایک جیسی اِن پُٹ کی بنیاد پر '0' آؤٹ پُٹ دیتا ہے، اور مختلف اِن پُٹ کی بنیاد پر '1' آؤٹ پُٹ دیتا ہے۔

اگر A اور B، XOR گیٹ کے دو اِن پُٹ ہیں اور Y آؤٹ پُٹ ہے تو، XOR گیٹ کی بلاک ڈایاگرام اور ٹروتھ ٹیبل نیچے دی گئی شکل میں دکھایا گیا ہے۔



شکل نمبر 6.6 دو اِن پُٹ والا XOR گیٹ

XOR گیٹ کا ٹروتھ ٹیبل

اِن پُٹ		آؤٹ پُٹ
A	B	Y=$(A \oplus B) = \overline{A}B + \overline{A}B$
0	0	0
0	1	1
1	0	1
1	1	0

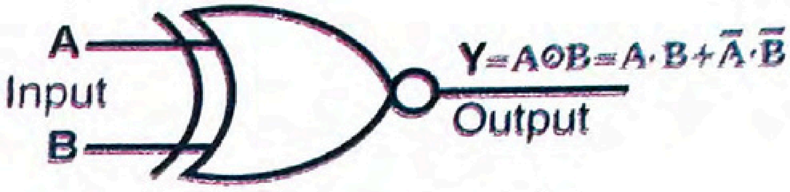
جدول نمبر 6.8 دو اِن پُٹ والے XOR گیٹ کا ٹروتھ ٹیبل

NOT FOR SALE

XNOR گیٹ (Exclusive-NOR Gate)

وہ گیٹ جس کی تمام اِن پُٹ اگر 0 یا 1 ہوں تو آؤٹ پُٹ '1' ہو گی، ایسے گیٹ کو XNOR گیٹ کہا جاتا ہے۔ لہٰذا XNOR گیٹ میں ایک جیسی اِن پُٹ کے لئے آؤٹ پُٹ '1' ہو گی اور مختلف اِن پُٹس کے لیے '0' آؤٹ پُٹ ہو گی۔

اگر A اور B ایک XNOR گیٹ کے دو ان پُٹ ہیں اور Y اس کا آؤٹ پُٹ ہے تو، اس کی بلاک ڈایاگرام اور XNOR گیٹ کا ٹروتھ ٹیبل ذیل میں دکھایا گیا ہے۔



شکل نمبر 6.7 دو اِن پُٹ والا XNOR گیٹ

XNOR گیٹ کا ٹروتھ ٹیبل

اِن پُٹ		آؤٹ پُٹ
A	B	$Y=A.B+\overline{A}'\overline{B}$ or $Y=A\odot B$
0	0	1
0	1	0
1	0	0
1	1	1

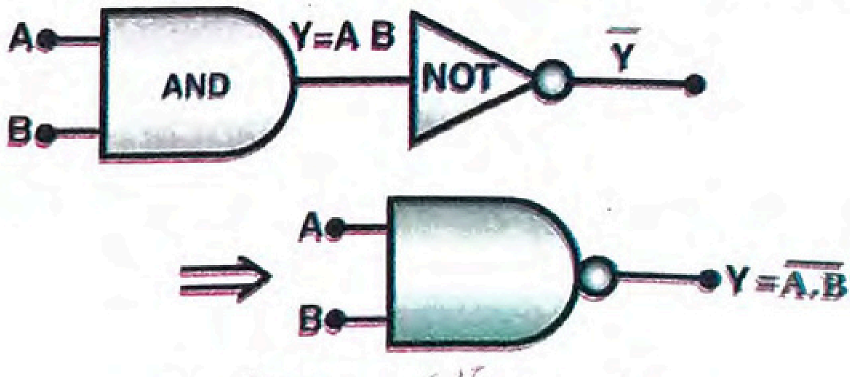
جدول نمبر 6.9 دو اِن پُٹ والے XNOR گیٹ کا ٹروتھ ٹیبل

6.2.5 بنیادی گیٹس کی مدد سے NAND، NOR، XOR اور XNOR گیٹ بنانا

بنیادی گیٹس کو استعمال کر کے NAND، NOR، XOR اور XNOR گیٹس بنائے جا سکتے ہیں۔

بنیادی گیٹس کا استعمال کرتے ہوئے NAND اور NOR گیٹس بنانا

AND گیٹ اور NOT گیٹ استعمال کر کے NAND گیٹ آسانی سے بنایا جا سکتا ہے جیسا کہ ذیل میں دکھایا گیا ہے۔



شکل نمبر 6.8 بنیادی گیٹس کی مدد سے NAND گیٹ بنانا

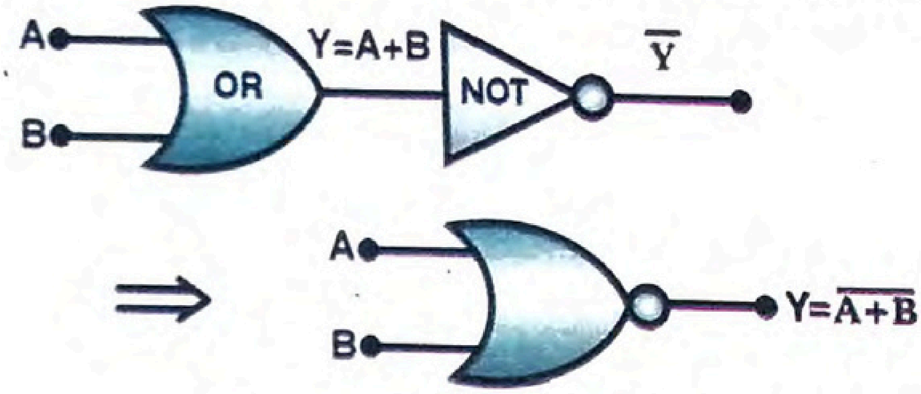
NOT FOR SALE

NAND گیٹ کا ٹروتھ ٹیبل

A	B	Y=A·B	$\overline{Y}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

جدول نمبر 6.10 AND اور NOT گیٹ کو استعمال کر کے NAND گیٹ کا ٹروتھ ٹیبل

اسی طرح NOR گیٹ بھی OR گیٹ اور NOT گیٹ کا استعمال کر کے بنایا جا سکتا ہے جیسا کہ ذیل میں دکھایا گیا ہے۔



شکل نمبر 6.9 OR گیٹ اور NOT گیٹ کا استعمال کر کے NOR گیٹ بنانا

ٹروتھ ٹیبل

A	B	Y=A+B	$\overline{Y}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

جدول نمبر 6.11 OR گیٹ اور NOT گیٹ کا استعمال کر کے NOR گیٹ بنانا

بنیادی گیٹس کا استعمال کرتے ہوئے XOR اور XNOR گیٹس بنانا

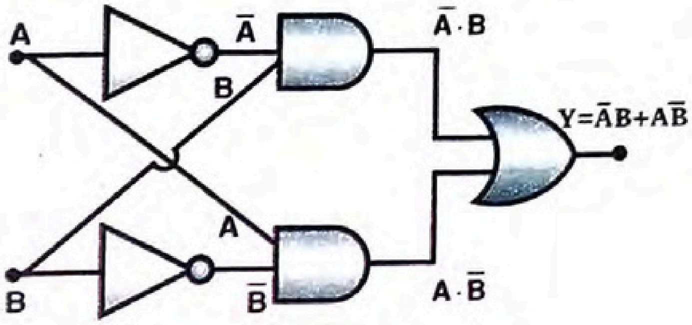
XOR گیٹ درج ذیل فنکشن کو ظاہر کرتا ہے:

$$Y=\overline{A}B+A\overline{B}$$

اس فنکشن میں دو اصطلاحیں استعمال ہوئی ہیں۔ $\overline{A}B$ اور $A\overline{B}$ جن کے لیے دو AND گیٹس درکار ہیں اور ان گیٹس کا آؤٹ پٹ

NOT FOR SALE

ایک OR گیٹ کو بطور اِن پُٹ دیا جائے گا تا کہ XOR گیٹ کو بنایا جا سکے جیسا کہ شکل نمبر 6.10 میں دکھایا گیا ہے۔ A اور B اِن پُٹ تبدیل کرنے کے لئے سرکٹری میں دو NOT گیٹس کا استعمال بھی کیا جاتا ہے۔



شکل نمبر 6.10 XOR گیٹ بنانا

ٹروتھ ٹیبل

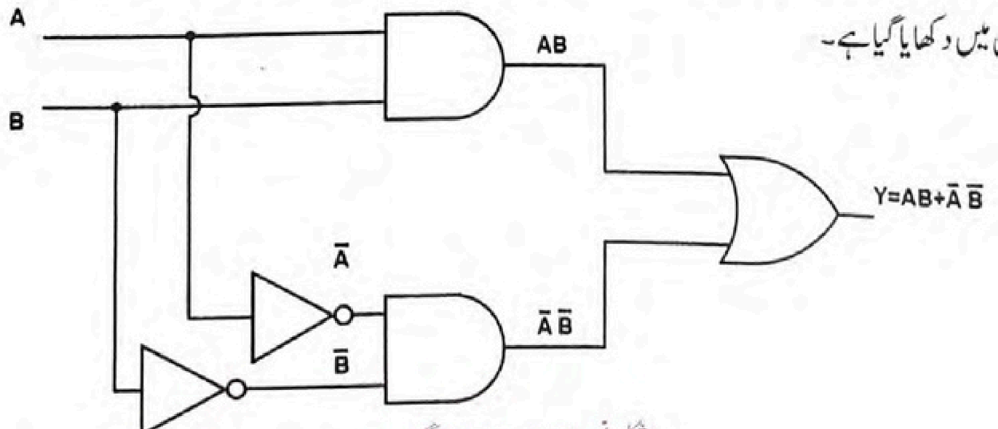
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

جدول نمبر 6.12 XOR گیٹ کا ٹروتھ ٹیبل

XNOR گیٹ درج ذیل فنکشن کو ظاہر کرتا ہے:

$$Y=AB+\bar{A}\bar{B}$$

یہ فنکشن دو اصطلاحات پر مشتمل ہے۔ اس گیٹ کا لاجک سرکٹ دو AND گیٹس پر مشتمل ہے تا کہ **AB** اور $\bar{A}\bar{B}$ بنائے جا سکیں اور دو NOT گیٹس بھی ہیں تا کہ **A** کو $\bar{A}$ اور **B** کو $\bar{B}$ میں تبدیل کیا جا سکے۔ اِس کے ساتھ ساتھ ایک OR گیٹ بھی ہے جیسا کہ درج ذیل شکل میں دکھایا گیا ہے۔



شکل نمبر 6.11 XNOR گیٹ بنانا

NOT FOR SALE

6.2.6 بولین ایکسپریشن کو لاجک سرکٹ میں تبدیل کرنا

ایک بولین ویری ایبل 0 یا 1 کی ویلیو لے سکتا ہے۔ ویری ایبل کی کسی بھی ویلیو کے لئے، بولین فنکشن 0 یا 1 ہو سکتا ہے۔

مندرجہ ذیل بولین فنکشن پر غور کریں۔

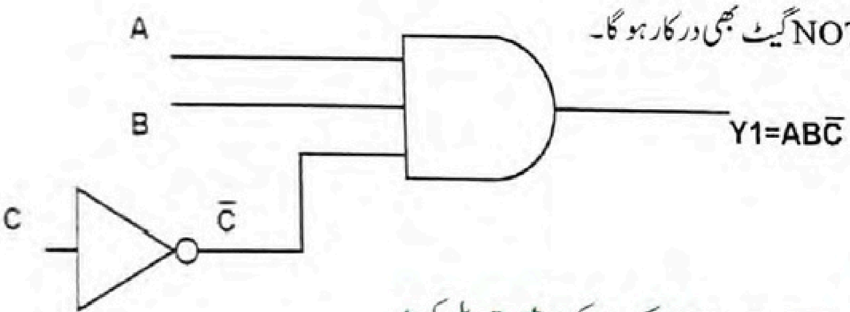
$$F = X + Y$$

$F = 0$ ہو گا، اگر $X = 0$ اور $Y = 0$ ہے۔ X اور Y کے دیگر تمام امتزاج کے لئے، فنکشن 1 کے برابر ہو گا۔

کوئی بھی بولین فنکشن الجبرا کی عبارت سے ایک لاجک سرکٹ میں تبدیل کیا جا سکتا ہے یہ لاجک سرکٹ AND، OR اور NOT گیٹس پر مشتمل ہے۔

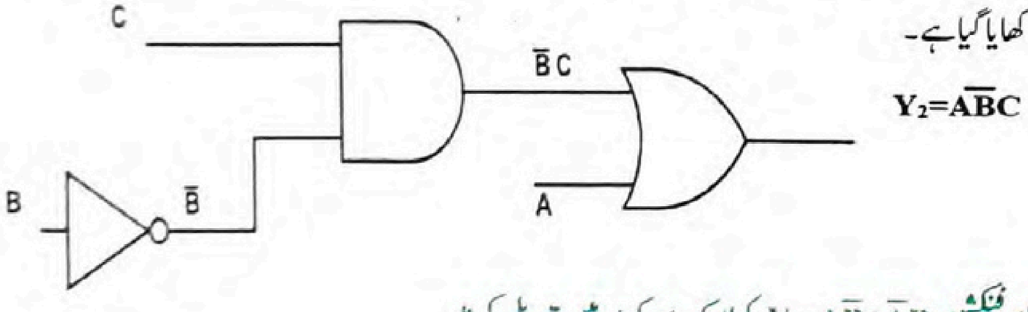
مثال نمبر 1: بولین فنکشن $Y_1=ABC$ کو لاجک سرکٹ گیٹ میں تبدیل کرنا

سنگل AND گیٹ کی اصطلاح کیلئے ضروری ہے $Y_1=ABC$ ۔ AND گیٹ کو ان پٹ کرنے سے پہلے C کو $\overline{C}$ میں تبدیل کرنے کے لیے ایک NOT گیٹ بھی درکار ہو گا۔



مثال 2: بولین فنکشن $Y_2=A+BC$ لاجک سرکٹ میں تبدیل کرنا

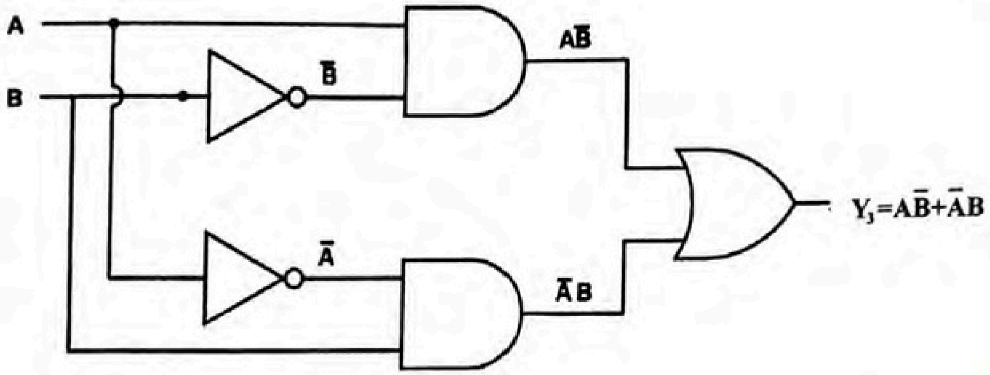
اس فنکشن کا لاجک سرکٹ بنانے کے لئے، $\overline{B}C$ اصطلاح کے لئے ایک AND گیٹ کی ضرورت ہوتی ہے، ایک NOT گیٹ چاہیے تاکہ B کو $\overline{B}$ میں تبدیل کیا جا سکے۔ A اور $\overline{B}C$ پر عملدرآمد کروانے کے لئے OR گیٹ کی ضرورت ہے۔ اس فنکشن کا لاجک سرکٹ نیچے دکھایا گیا ہے۔



مثال 3: بولین فنکشن $Y_3=A\overline{B}+\overline{A}B$ کو لاجک سرکٹ میں تبدیل کرنا

اس فنکشن کے لاجک سرکٹ میں دو AND گیٹس جو کہ $A\overline{B}$ اور $\overline{A}B$ بنانے کے لیے، دو NOT گیٹس جو A کو $\overline{A}$ اور B کو $\overline{B}$ میں تبدیل کرنے اور ایک OR گیٹ جو کہ دونوں AND گیٹس کی آؤٹ پٹ پر OR آپریشن کے لیے استعمال ہونگے۔ جیسا کہ ذیل میں دکھایا گیا ہے۔

NOT FOR SALE



6.3 بولین فنکشن کو کارناف میپ کی مدد سے مختصر کرنا

بولین فنکشن کا اختصار (Simplification of Boolean Function)

بولین فنکشن میں موجود اصطلاحات کی تعداد کو کم سے کم یا ختم کرنے کے عمل کو، تاکہ آخری آؤٹ پُٹ پر کوئی فرق نہ پڑے، اختصار (Simplification) کہا جاتا ہے۔ اس مقصد کے لئے مختلف طریقوں کا استعمال کیا جاتا ہے۔ ٹیلی کمیونیکیشن کے انجینئر، مارس کارناف (Mauric Karnaugh) نے 1953ء میں Bell Labs میں Karnaugh Map تیار کیا۔ K-map بولین فنکشن کو آسان بنانے کا بہترین طریقہ ہے۔ جب مختصر بولین فنکشن ایک لاجک سرکٹ میں تبدیل ہو جاتا ہے تو اس کے لئے لاجک گیٹس کم تعداد میں درکار ہوتے ہیں اور اس وجہ سے اس کی لاگت بھی کم ہو جاتی ہے۔

6.3.1 کارناف میپ (Karnaugh Map)

کارناف میپ یا K-map بولین ایکسپریشن کا ایک تصویری خاکہ ہے۔ Karnaugh Map کو 2D ٹروتھ ٹیبل بھی کہا جاتا ہے کیونکہ ہر K-map، 1D ٹروتھ ٹیبل میں موجود قیمتوں کو ظاہر کرنے کا ایک مختلف فارمیٹ ہے۔

K-map بنیادی طور پر مربع نما خانوں پر مشتمل ہوتا ہے جنہیں سیلز (Cells) کہا جاتا ہے۔ بولین فنکشن کے تمام ممکنہ امتزاج سیلوں کے اندر انکی متعین جگہوں پر لکھے ہوتے ہیں۔ دو ویری ایبلز والا K-map $4 = 2^2$ سیلوں پر مشتمل ہوتا ہے، 4 ویری ایبل والا K-map $16 = 2^4$ سیلوں پر مشتمل ہوتا ہے، اور اسی طرح آگے بھی۔

کارناف میپ کی مدد سے ہم ان تمام ویری ایبلز کو گروپ کی شکل دے سکتے ہیں جن میں کوئی نہ کوئی قدر مشترک ہو اس طرح ہر غیر ضروری ویری ایبل کو ختم کر سکتے ہیں۔ کارناف میپ کو ہم ٹروتھ ٹیبل کی مخصوص شکل بھی کہہ سکتے ہیں۔ درج ذیل شکل دو ویری ایبل والے ٹروتھ ٹیبل اور کارناف میپ کے درمیان مطابقت کا ظاہر کرتی ہے۔

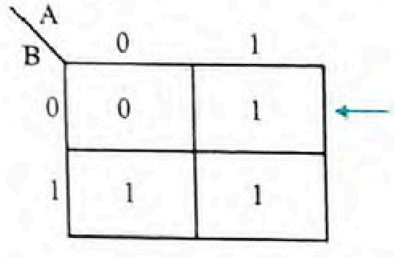
مربعوں کے درمیان موجودہ قیمتیں ٹروتھ ٹیبل کے آؤٹ پُٹ کالم سے لی گئی ہیں۔ اس لیے ٹروتھ ٹیبل کی ہر قطار یعنی Row کے لیے K-map میں ایک مربع مختص ہے۔ کارناف میپ کے کناروں پر دونوں اِن پُٹ ویری ایبل کی قیمتیں درج ہیں۔ A کی قیمتیں بالائی (TOP) کی

NOT FOR SALE

طرف اور B کی قیمتیں بائیں (Left) طرف ہیں۔ زیر نظر شکل اس کی تفصیل بیان کر رہی ہے:

A	B	F
0	0	0
0	1	1
1	0	1 ←
1	1	1

Truth Table.



F.

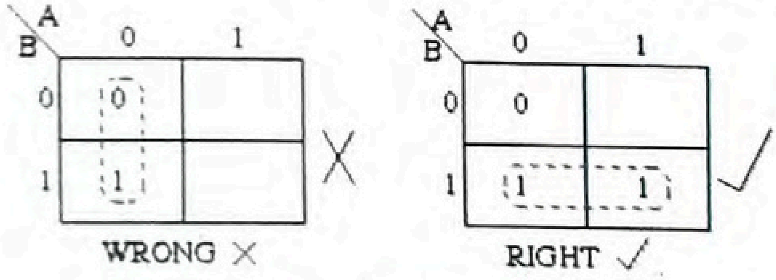
شکل نمبر 6.12 K-map اور ٹروتھ ٹیبل

مثال کے طور پر، map کے اوپر دائیں بائیں کونے پر مربع میں A = 1 اور B = 0 ہے۔ یہ مربع ٹروتھ ٹیبل کی اس قطار سے مطابقت رکھتا ہے جہاں A = 1، B = 0 اور F = 1 ہے۔ یہ بات قابل غور ہے کہ F والے کالم میں درج قیمتیں کسی خاص فنکشن کی نمائندگی کرتی ہے جس کے لیے کارناف میپ بنایا گیا ہے۔

کارناف میپ-اختصار کے قواعد

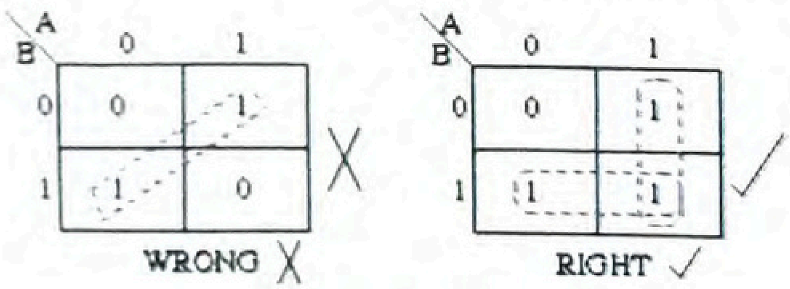
کارناف میپ اختصار کے لیے متصل سیلوں جن میں 1 درج ہو کو درج ذیل قواعد وضوابط کے تحت ایک گروپ میں لے آتا ہے۔ اس طرح ان گروپوں کی مدد سے عبارت مختصر ہو جاتی ہے۔

- گروپ میں کوئی ایسا سیل نہیں آسکتا جس میں زیرو درج ہو۔



شکل نمبر 6.13 1 کے گروپ والا میپ

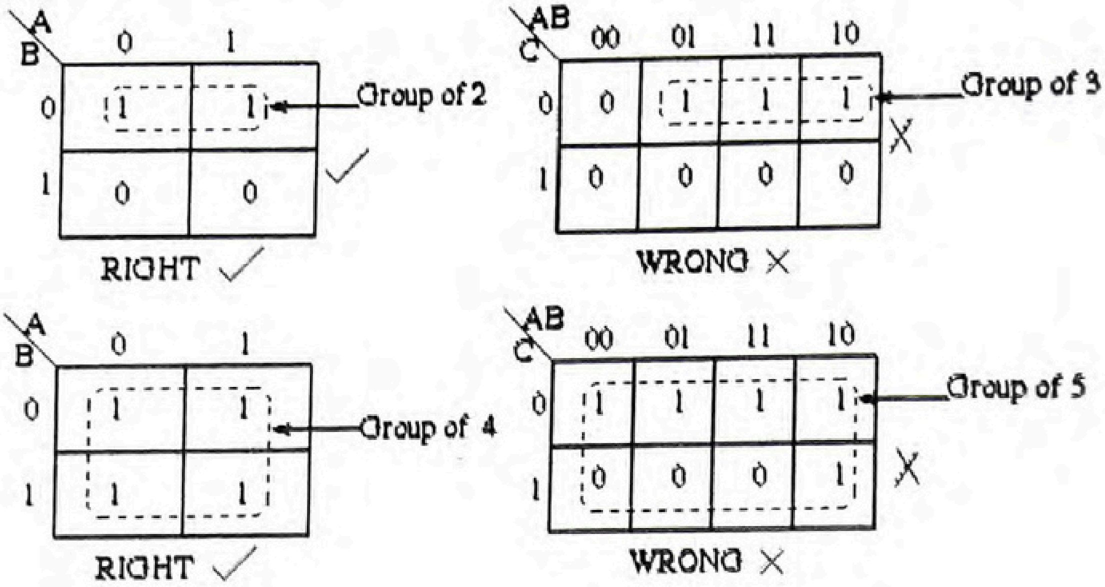
- گروپ افقی یا عمودی ہو سکتے ہیں، لیکن ترچھے نہیں ہو سکتے۔



شکل نمبر 6.14 گروپ میں ترچھے سیل شامل نہیں

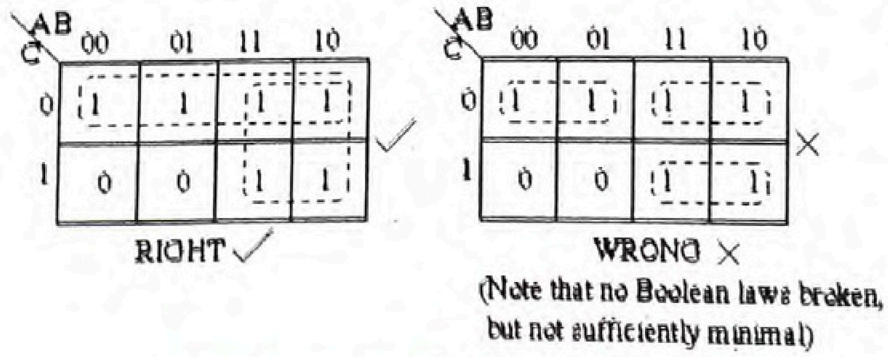
NOT FOR SALE

- ضروری ہے کہ گروپ میں 1، 2، 4، 8 سیل یا عمومی طور پر 2^n سیل ہوں۔



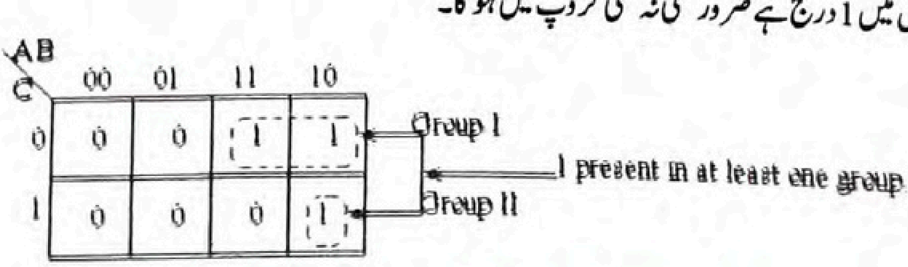
شکل نمبر 6.15 دکھایا گیا ہے کہ گروپ میں 1، 2، 4، 8 سیل یا عمومی طور پر 2^n سیل ہوں گے

- ہر گروپ جہاں تک ممکن ہو بڑا ہونا چاہیے۔



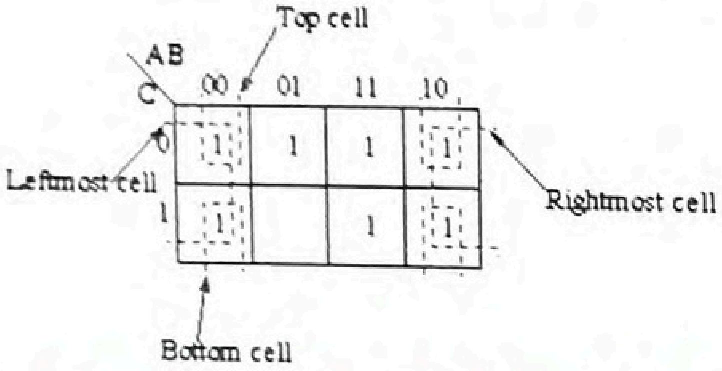
شکل نمبر 6.16 گروپ جہاں تک ممکن ہو بڑے بڑے ہونے چاہئیں

- ہر سیل جس میں 1 درج ہے ضرور کسی نہ کسی گروپ میں ہو گا۔



شکل نمبر 6.17 ہر سیل جس میں 1 درج ہے ضرور کسی نہ کسی گروپ میں ہو گا

- گروپ ٹیبل کے اطراف میں بھی پھیلا ہو سکتا ہے۔ ایک قطار میں بالکل بائیں طرف والا سیل بالکل دائیں طرف والے اور بالکل زیریں سیل بالکل بالائی سیل کے ساتھ گروپ میں لایا جاسکتا ہے۔



شکل نمبر 6.18 K-map کے اطراف میں موجود سیل آپس میں گروپ بناسکتے ہیں

- جہاں تک ممکن ہو گروپ کی تعداد کم سے کم ہونی چاہیے بشرطیکہ پہلے بیان کردہ قواعد میں سے کسی کے خلاف نہ ہو۔

6.3.2 دو ویری ایبلز والے بولین فنکشنز کا K-Map کے ذریعے اختصار

مثال نمبر 1:

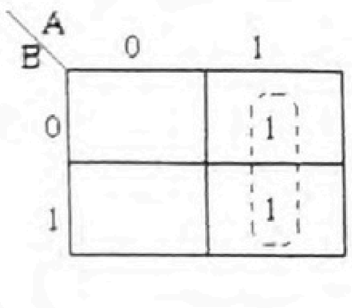
درج ذیل میپ کو دیکھیں، جو فنکشن اس میں پلاٹ کیا گیا ہے وہ یہ ہے: $Z = F(A,B) = A\overline{B} + AB$

درج بالا بولین فنکشن کے لیے ٹروتھ ٹیبل

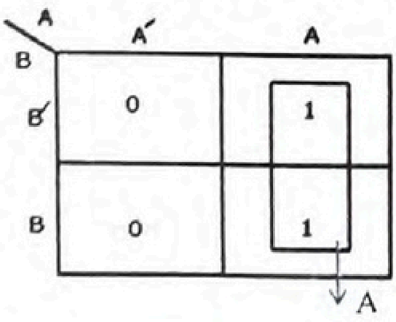
A	B	$Z = F(A,B) = A\overline{B} + AB$
0	0	0
0	1	0
1	0	1
1	1	1

جدول نمبر 6.13 $Z = F(A,B) = A\overline{B} + AB$ کے لیے ٹروتھ ٹیبل

اب ٹروتھ ٹیبل کی تمام قیمتوں کو درج ذیل K-Map میں درج کریں۔



OR



NOT FOR SALE

غور کریں اِن پُٹ ویری ایبل کی قیمتیں ہی قطار اور کالم بنا رہی ہیں۔

یہاں صرف ایک ہی گروپ ہے، جو دو ملحقہ سیلوں کا احاطہ کرتا ہے۔

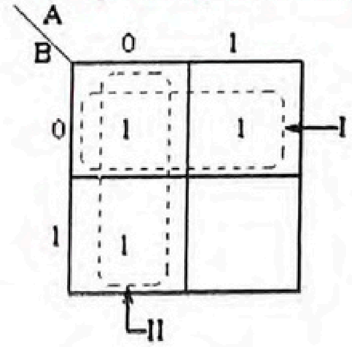
$$Z=F(A,B)=A$$

مثال نمبر 2:

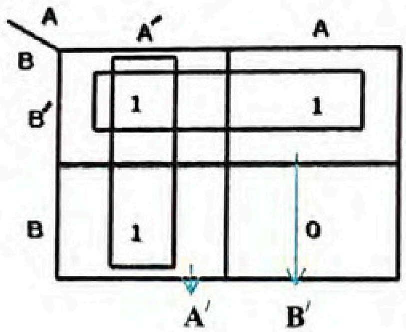
فرض کیا کہ بولین ایکسپریشن ہے $Z=F(A,B)=\bar{A}B+A\bar{B}+A\bar{B}$

A	B	$Z=F(A,B)=\bar{A}\bar{B}+A\bar{B}+A\bar{B}$
0	0	1
0	1	1
1	0	1
1	1	0

جدول نمبر 6.14 $Z=F(A,B)=\bar{A}\bar{B}+A\bar{B}+A\bar{B}$ کے لیے ٹروتھ ٹیبل



OR



- یہاں دو ملحقہ سیلوں پر مشتمل دو گروپ ہیں۔
- $\bar{A}$ کالم وار مشترک ویری ایبل ہے گروپ II میں۔ جبکہ $\bar{B}$ قطار وار مشترک ویری ایبل ہے گروپ 1 میں۔
- دونوں ٹرمز جمع کرکے آخری نتیجہ حاصل کیا جاسکتا ہے۔
- آخری مختصر شدہ ایکسپریشن کی شکل یہ ہوگی:

$$Z=F(A,B)=\bar{A}+\bar{B}$$

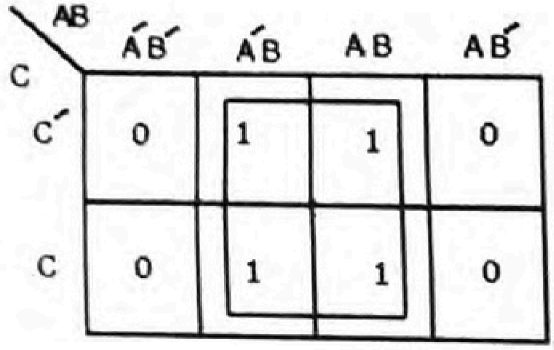
6.3.3 تین ویری ایبلز والے بولین فنکشنز کا K-Map کے ذریعے اختصار

مثال نمبر 1:

$$Z=f(A,B,C)=A.B.C+A.B.C+A.B.C+A.B.C$$

درج بالا ایکسپریشن کے لیے K-Map

NOT FOR SALE

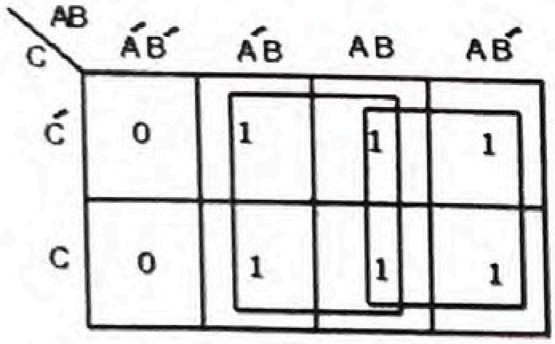


- یہاں پر ایک ہی گروپ ہے جو چار سیلوں پر مشتمل ہے۔
- یہاں قطار وار کوئی ویری ایبل مشترک نہیں ہے لیکن B کالم وار مشترک ویری ایبل ہے۔
- تو مختصر شدہ ایکسپریشن یہ ہو گا:

$$Z= F(A,B,C) = B$$

مثال نمبر2:

درج بالا ایکسپریشن $Z=f(A, B, C) = A.B.C+ \bar{A}.B.C+ A.\bar{B}.\bar{C}+\bar{A}.B.\bar{C}+A.B.\bar{C}+A.\bar{B}.C$ کے لیے K-Map



- یہاں پر دو گروپ ہیں اور ہر گروپ میں چار چار ملحقہ سیل موجود ہیں۔
- گروپ 1 میں A کالم وار مشترک ویری ایبل ہے جبکہ گروپ II میں B قطار وار مشترک ویری ایبل ہے۔
- ان دونوں ویری ایبلز کو جمع کر کے مختصر شدہ ایکسپریشن حاصل کیا جاسکتا ہے۔
- آخر میں مختصر شدہ ایکسپریشن درج ذیل ہوگا:

$$Z= f(A,B,C) = A + B$$

مثال نمبر3:

$$F(A,B,C)= \bar{A}.BC+ \bar{A}.B\,\bar{C}+A\,\bar{B}\,\bar{C}+A\bar{B}C$$

درج بالا ایکسپریشن کے لیے K-Map

NOT FOR SALE

A \ BC	00	01	11	10
0	0	0	1	1
1	1	1	0	0

K-map for F (A, B, C)

A \ BC	00	01	11	10
0	0	0	[1	1]
1	[1	1]	0	0

K-map for F (A, B, C)

یہاں پر دو گروپ کچھ اس طرح ہوں گے ۔

$$\overline{A}.B\overline{A}.B \text{ and } A\overline{B}A\overline{B}$$

آخر میں مختصر شدہ ایکسپریشن درج ذیل ہوگا:

$$F(A,B,C)= \overline{A}.B+A\,\overline{B}=A\oplus B$$

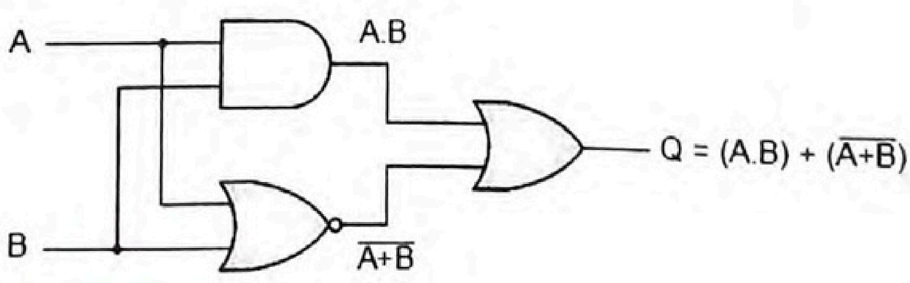
$$F(A,B,C)= \overline{A}.B+A\,\overline{B}=A\oplus B$$

6.3.4 مختصر شدہ درج ذیل فنکشنز سے لاجک سرکٹ بنانا

مثال نمبر 1:

$$Q= f(AB) +(\overline{A + B})$$

سرکٹ تین گیٹس پر مشتمل ہے۔AND گیٹ، NOR گیٹ اور OR گیٹ۔AND گیٹ کے لیے ایکسپریشن A.B ہے جبکہ NOR گیٹ کا ایکسپریشن $\overline{A+B}$ ہے۔ یہ دونوں ایکسپریشنز مل کر OR گیٹ کے لیے علیحدہ سے اِن پُٹ دیتے ہیں، ($\overline{A+B}$) اور (A.B)۔



NOT FOR SALE

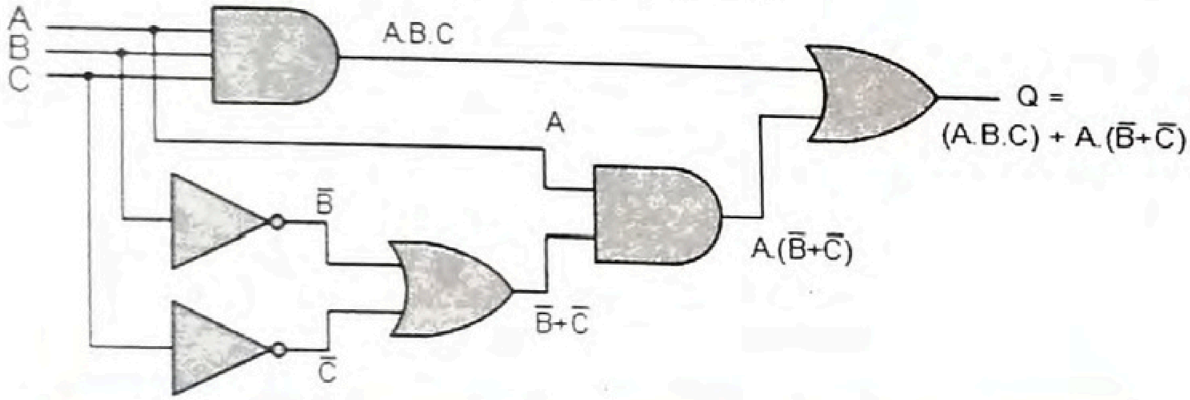
ان گیٹس کا آؤٹ پٹ اس قسم کا ہو گا

$$Q=(A.B)+(\overline{A+B})$$

Inputs		Intermediates		Output
A	B	A.B	$(\overline{A+B})$	Q
0	0	0	1	1
1	0	0	0	0
0	1	0	0	0
1	1	1	0	1

مثال نمبر2:

$$Q=(A.B.C)+A.(\overline{B}+\overline{C})$$



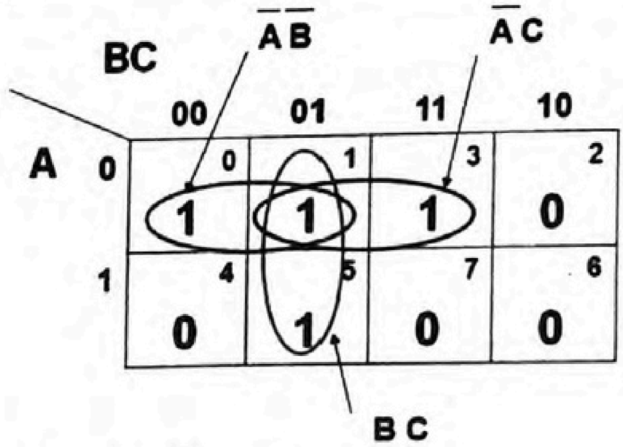
Inputs			Intermediates					Output
C	B	A	A.B.C	$\overline{B}$	$\overline{C}$	$\overline{B}+\overline{C}$	$A.(\overline{B}+\overline{C})$	Q
0	0	0	0	1	1	1	0	0
0	0	1	0	1	1	1	1	1
0	1	0	0	0	1	1	0	0
0	1	1	0	0	1	1	1	1
1	0	0	0	1	0	1	0	0
1	0	1	0	1	0	1	1	1
1	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	1

NOT FOR SALE

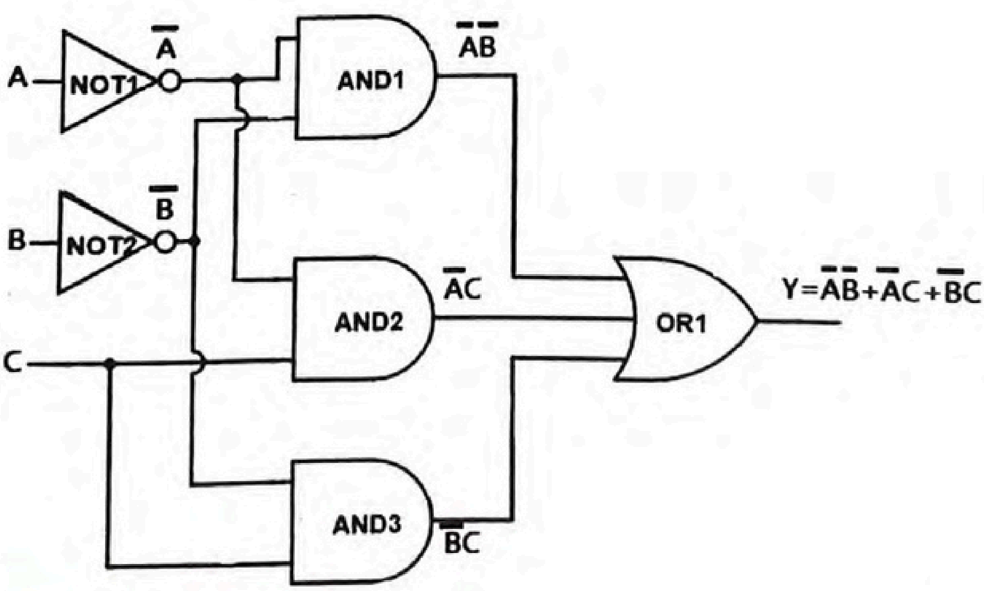
درج ذیل بولین ایکسپریشن کا K-Map کی مدد سے اختصار اور مختصر شدہ فنکشن کی لاجک ڈائیگرام۔

$$Y=\overline{A}\overline{B}\overline{C}+\overline{A}\overline{B}C+\overline{A}BC+A\overline{B}C$$

C	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0



$$Y=\overline{A}\overline{B}+\overline{A}C+\overline{B}C$$



NOT FOR SALE

اہم نکات

- ڈیٹا خام حقائق اور اعداد و شمار ہیں جن پر ڈیجیٹل کمپیوٹر کے ذریعہ کارروائی کی جاتی ہے۔
- لاجک گیٹ ایک الیکٹرانک سرکٹ ہے جو ایک یا زیادہ اِن پُٹ سگنل لیتا ہے اور ایک ہی آؤٹ پُٹ سگنل تیار کرتا ہے۔
- AND گیٹ لاجیکل ضرب کرتا ہیں۔
- OR گیٹ لاجیکل جمع کے لیے استعمال ہوتا ہے۔
- NOT گیٹ اِن پُٹ ویلیو کو تبدیل کرنے کے لئے استعمال ہوتا ہے۔
- ٹروتھ ٹیبل ایک ٹیبل ہے جس میں اِن پُٹ ویلیوز اور ان کے مطابق آؤٹ پُٹ ویلیوز کے تمام ممکنہ امتزاج ہوتے ہیں۔
- جس گیٹ کی آؤٹ پُٹ 1 ہوتی ہے جب تمام اِن پُٹ 0 ہوتے ہیں تو اسے NOR گیٹ کہتے ہیں۔
- جس گیٹ کی آؤٹ پُٹ 0 ہوتی ہے جب تمام ان پُٹ 1 ہوتے ہیں تو اسے NAND گیٹ کہتے ہیں۔
- وہ گیٹ جس کی آؤٹ پُٹ 0 ہوتی ہے جب تمام اِن پُٹس میں 0 یا 1 ہوتے ہیں تو اسے XOR گیٹ کہا جاتا ہے۔
- وہ گیٹ جس کی آؤٹ پُٹ 1 ہوتی ہے جب تمام اِن پُٹ 0 یا 1 ہوتے ہیں تو اسے XNOR گیٹ کہا جاتا ہے۔
- بولین فنکشن / ایکسپریشن بولین ویری ایبل، کانسٹنٹ اور آپریٹر کا مجموعہ ہے۔
- ا بولین فنکشن کا اختصار حتمی آؤٹ پُٹ کو متاثر کیے بغیر بولین فنکشن میں ٹرم یا ویری ایبل کی تعداد کو کم کرنے کا عمل ہے۔
- K-Map ایک جیسے ایکسپریشن کو ایک ساتھ جوڑنے کا ایک خاکہ فراہم کرتا ہے اور اس طرح، ناپسندیدہ ویری ایبل کو ختم کرتا ہے۔

مشق

سوال نمبر 1۔ موزوں جواب کا انتخاب کریں۔

i. جب ڈیٹا کمپیوٹر میں داخل ہوتا ہے تو اُسے ________ میں تبدیل کیا جاتا ہے۔

(الف) بائنری نمبر سسٹم　(ب) اعشاری نمبر سسٹم
(ج) ہیگزاڈیسیمل نمبر سسٹم　(د) آکٹل نمبر سسٹم

ii. ایک الیکٹرانک سرکٹ جو ایک یا ایک سے زیادہ سگنل لیتا ہے اور ایک ہی آؤٹ پُٹ دیتا ہے ________ کہلاتا ہے۔

(الف) شکل　(ب) ٹیبل　(ج) ڈیجیٹل لاجک گیٹ　(د) ان میں سے کوئی نہیں

NOT FOR SALE

iii. کس گیٹ کی آؤٹ پُٹ صرف اس وقت 1 ہو گی جب اس کی تمام اِن پُٹس 1 ہوں؟

(الف) AND گیٹ (ب) OR گیٹ (ج) NOT گیٹ (د) NAND گیٹ

iv. کونسا گیٹ لاجیکل معکوس یا کمپلیمنٹ بناتا ہے؟

(الف) AND گیٹ (ب) OR گیٹ (ج) NOT گیٹ (د) XOR گیٹ

v. وہ گیٹ جس کی آؤٹ پُٹ صرف اس وقت 1 ہو گی جب اس کی تمام اِن پُٹس 0 ہوں ____________ کہلاتا ہے۔

(الف) AND گیٹ (ب) OR گیٹ (ج) NOT گیٹ (د) XOR گیٹ

vi. K-Map ____________ کے لیے استعمال ہوتا ہے۔

(الف) بولین ایکسپریشن کو مختصر کرنے (ب) بولین ایکسپریشن کو شکل کی مدد سے دکھانے

(ج) دو ویری ایبلز کا آپس میں موازنہ کرنے (د) ڈیجیٹل سرکٹ کو ڈیزائن کرنے

سوال نمبر 2۔ کمپیوٹر میں ڈیٹا کس شکل میں ظاہر کیا جاتا ہے؟ مختصراً بیان کریں۔

سوال نمبر 3۔ تین بنیادی گیٹس کونسے ہیں؟ ان کی سرکٹ ڈائیگرام اور ٹروتھ ٹیبل بنائیں۔

سوال نمبر 4۔ درج ذیل بولین ایکسپریشن کے لیے لاجک گیٹ ڈائیگرام بنائیں۔

$$f(A, B, C) = ABC + \overline{A}\,\overline{B} + \overline{C} + A\overline{B}$$

سوال نمبر 5۔ درج ذیل بولین ایکسپریشن کو K-Map کی مدد سے مختصر کریں۔ پھر مختصر شدہ فنکشن کی لاجک ڈائیگرام بھی بنائیں۔

i. $F=\overline{X}Z + \overline{X}\,\overline{Z}$

ii. $F=\overline{A}BC + A\overline{B}C+AB\overline{C}+\overline{ABC}$

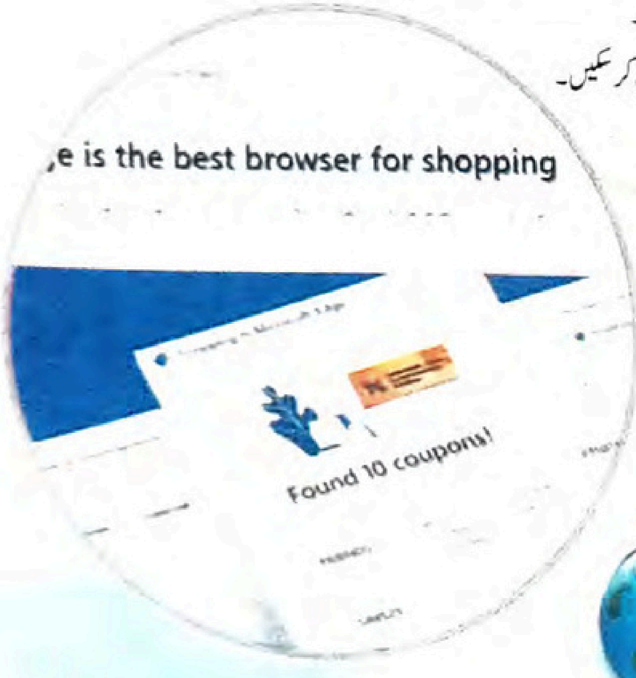
NOT FOR SALE

باب نمبر 7

ورلڈ وائیڈ ویب اور HTML

اس باب کی تکمیل کے بعد طلبہ اس قابل ہو جائیں گے کہ وہ:

- ویب سائٹ سے متعلق اصطلاحات کی تعریف کر سکیں (WWW، ویب پیج، ویب سائٹ، ویب براؤزر، ویب سرور، URL، سرچ انجن، ہوم پیج، ویب ہوسٹنگ)
- مختلف قسم کی ویب سائٹس کی وضاحت کر سکیں۔
- ہائپر ٹیکسٹ مارک اپ لینگوئج (HTML) کی تعریف کر سکیں۔
- ایک HTML فائل بنائیں اور محفوظ کر سکیں۔
- بنا ہوا ویب پیج براؤزر میں دکھا سکیں۔
- HTML عناصر کو مارک اپ کرنے کے لئے استعمال ہونے والے ٹیگز کی شناخت کر سکیں۔
- HTML، ہیڈ سیکشن اور HTML فائل کے باڈی سیکشن کی شناخت کر سکیں۔
- پیج ٹائٹل لگانے کے مختلف مراحل بیان کر سکیں۔ پیراگراف بنا سکیں، لائن بریک لگا سکیں، سپیس یعنی وقفوں کا اضافہ کر سکیں اور ہیڈنگ سب ہیڈنگ لگا سکیں۔
- ٹیکسٹ فارمیٹنگ کے ٹیگز کو پہچان سکیں اور انکے استعمال کو جان سکیں۔
- لسٹ کی مختلف اقسام جیسے آرڈرڈ، اَن آرڈرڈ، ڈیفینیشن اور نیسٹڈ لسٹ بنا سکیں اور ان میں فرق کر سکیں۔
- تصویر کا اضافہ کر سکیں اور اس پر بارڈر لگا سکیں۔
- کسی تصویر کی چوڑائی اور اونچائی کی مختص کر سکیں۔
- کسی ویب پیج کے پس منظر / پیش منظر کے رنگ میں تبدیلی کر سکیں۔
- ویب پیج کے پس منظر میں کوئی تصویر لگا سکیں۔
- اینکر کی تعریف کر سکیں اور اسے بنا سکیں۔
- ہائپر لنک کی تعریف کر سکیں اور اسے بنا سکیں۔
- گرافیکل ہائپر لنک بنا سکیں۔
- ٹیبل تیار کر سکیں اور ویب پیج میں ٹیبل کی مختلف خصوصیات رکھ سکیں۔
- فریم کی تعریف کر سکیں اور فریم اور فریم سیٹ کی مختلف اقسام میں فرق کر سکیں۔
- فریم سیٹ بنا سکیں۔



NOT FOR SALE

7.1 ورلڈ وائڈ ویب کا تعارف

ورلڈ وائڈ ویب، www، w3 یا ویب انٹرنیٹ کے لئے ایک گرافیکل انٹرفیس ہے جو 6 اگست 1991ء کو ٹم برنرز لی (Tim Berners-Lee) کے ذریعہ عوام کے سامنے سب سے پہلے متعارف کرایا گیا تھا۔ کچھ دن بعد 23 اگست 1991ء کو، یہ عوام کے لئے دستیاب تھا۔

ورلڈ وائڈ ویب (www) انٹرنیٹ پر ایک انفارمیشن سسٹم ہے جو ہائپر ٹیکسٹ لنکس کے ذریعہ ڈاکومنٹس کو دوسرے ڈاکومنٹس سے منسلک کرنے کی اجازت دیتا ہے، جس سے یوزر کو ایک ڈاکومنٹ سے دوسرے ڈاکومنٹ میں منتقل ہو کر معلومات تلاش کرنے میں آسانی ہوتی ہے۔

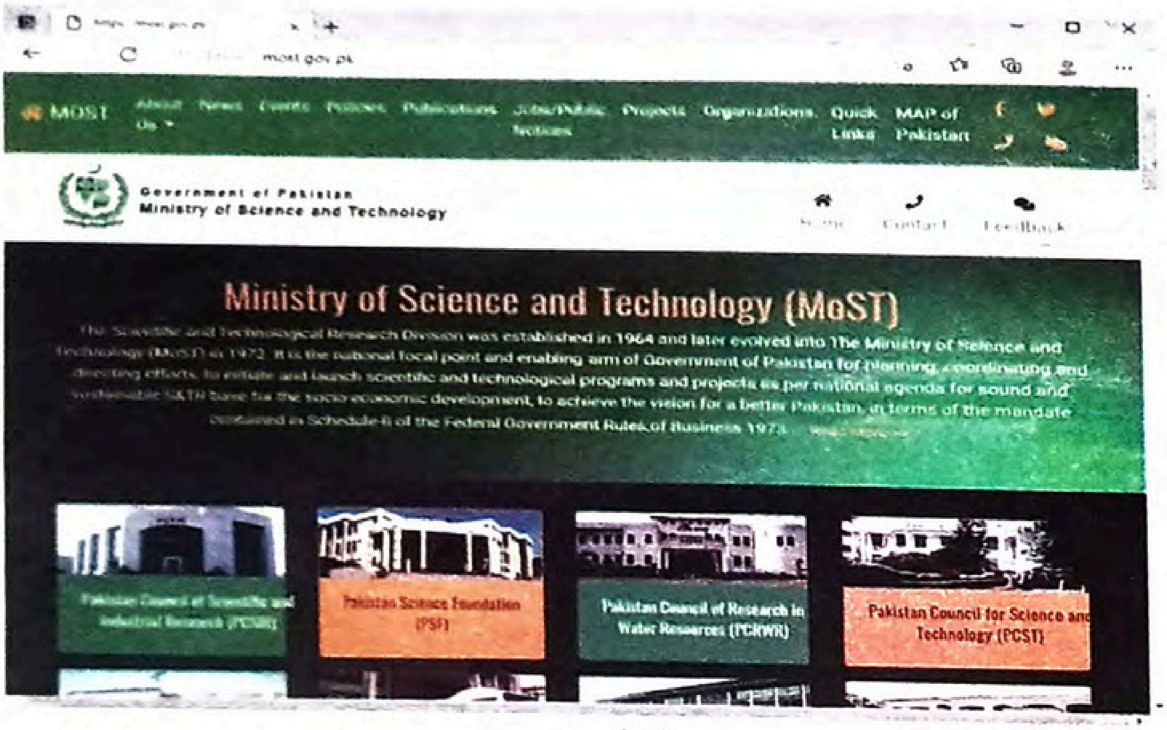
www انٹرنیٹ سے مختلف ہے حالانکہ آج کے زیادہ تر لوگ یہ سمجھتے ہیں کہ www انٹرنیٹ ہی ہے۔ انٹرنیٹ دراصل آپ کے کمپیوٹر کو دوسرے کمپیوٹرز کے ساتھ جوڑتا ہے۔ جبکہ www سے آپ اپنے براؤزر کو استعمال میں لاتے ہوئے مختلف ویب سائٹس دیکھتے ہیں جیسے (google.com)۔

7.1.1 ورلڈ وائڈ ویب کی اصطلاحات

ویب پیج

ایک ویب پیج HTML (ہائپر ٹیکسٹ مارک اپ لینگوئج) میں لکھا ہوا ایک ڈاکومنٹ ہے جو انٹرنیٹ براؤزر میں دیکھا جاتا ہے۔ ہر ویب پیج کی شناخت ایک منفرد URL (یونیفارم ریسورس لوکیٹر) یا ویب ایڈریس سے ہوتی ہے۔ ایک ویب پیج میں ٹیکسٹ، گرافکس، آڈیو، ویڈیو اور ہائپر لنکس سمیت بڑی معلومات شامل ہو سکتی ہے۔ یہ ہائپر لنک دوسرے ویب پیجز کا لنک ہے۔

شکل نمبر 7.1 میں ایک ویب پیج دکھایا گیا ہے۔



شکل نمبر 7.1 ویب پیج

ویب سائٹ

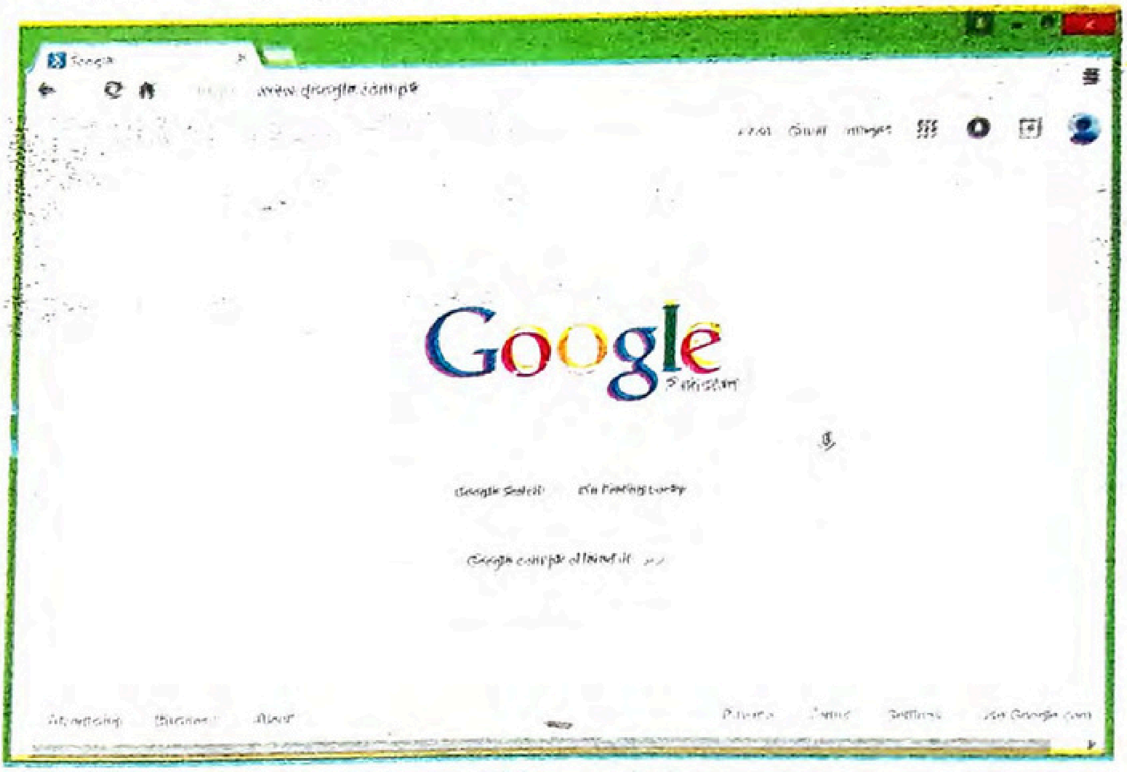
ویب سائٹ ویب پیجز کا ایک مجموعہ ہے جس میں متعلقہ معلومات ہوتی ہیں۔ ویب سائٹ کی مختلف قسمیں ہیں جو مختلف قسم کے مندرجات جیسے خبریں، معلومات، تعلیم اور تفریح فراہم کرتی ہیں۔ پوری دنیا میں قابل رسائی ہونے کے لیے ایک ویب سائٹ کو ویب سرور پر اسٹور کرنا ضروری ہے۔ ہر ویب سائٹ کا ایک منفرد ایڈریس ہوتا ہے جسے URL کے نام سے جانا جاتا ہے۔ مثال کے طور پر ایک ویب سائٹ /http://www.icp.edu.pk کو شکل نمبر 7.2 میں دکھایا گیا ہے۔

شکل نمبر 7.2 اسلامیہ کالج کی ویب سائٹ

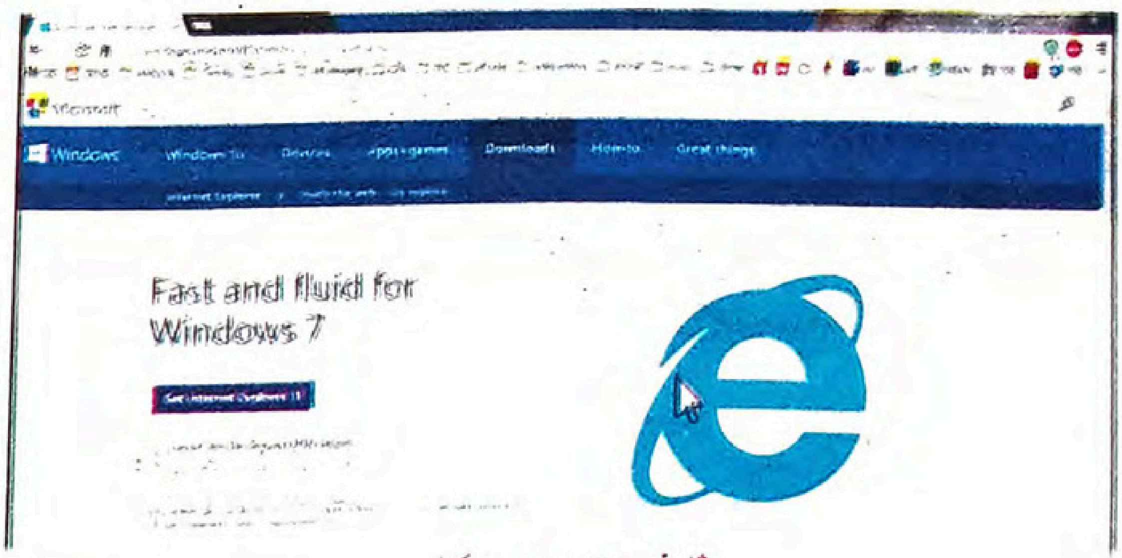
NOT FOR SALE

ویب براؤزر

ویب براؤزر ایک قسم کا سافٹ ویئر ہے جو آپ کو ویب سائٹ پر معلومات حاصل کرنے کی سہولت دیتا ہے۔ ویب پر معلومات URLs کے ذریعہ حاصل کی جاتی ہیں۔ ویب براؤزر یوزرز کو ایک ہی وقت میں متعدد ویب سائٹس کھولنے کی اجازت دیتے ہیں، مختلف براؤزر ونڈوز میں یا ایک ہی ونڈو کے مختلف ٹیبز میں۔ بہت سارے ویب براؤزر موجود ہیں، لیکن چند مشہور یہ ہیں: Microsoft Edge، Mozilla Firefox، Google Chrome، Internet Explorer جو بالترتیب شکل نمبر 7.3(a)، 7.3(c)، 7.3 اور 7.3(d) میں دکھائے گئے ہیں۔ ان تمام براؤزرز میں گرافیکل انٹرفیس ہے جو سیکھنے اور استعمال میں بہت آسان

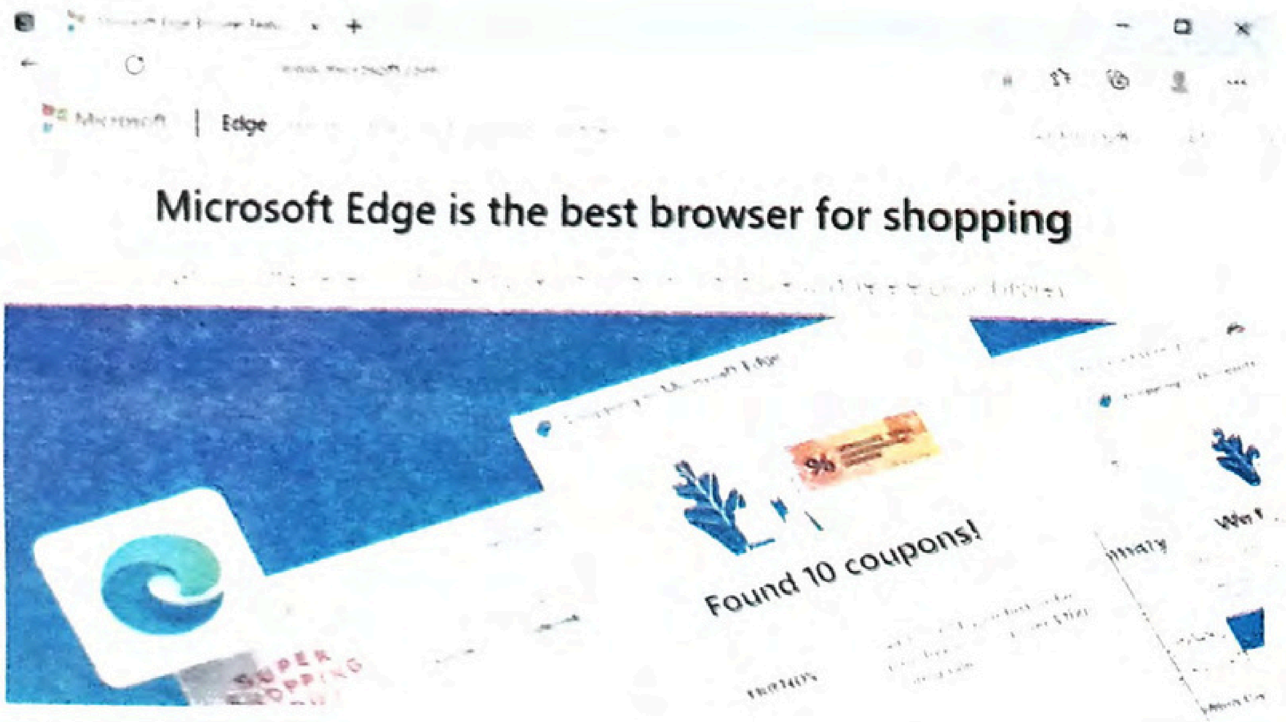


شکل نمبر 7.3(a) گوگل کروم

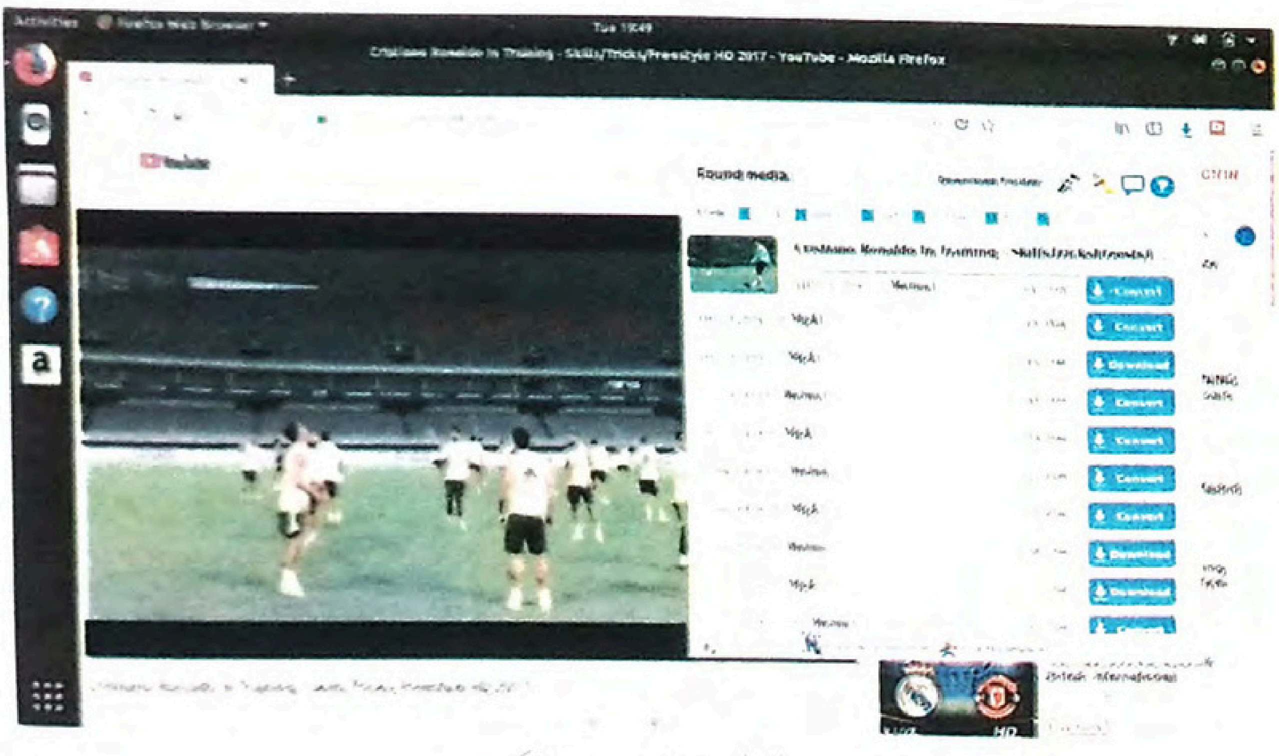


شکل نمبر 7.3(b) انٹرنیٹ ایکسپلورر

NOT FOR SALE



شکل نمبر (c)7.3 مائیکروسافٹ ایج براؤزر



شکل نمبر (d)7.3 موزیلا فائر فاکس

ویب سرور

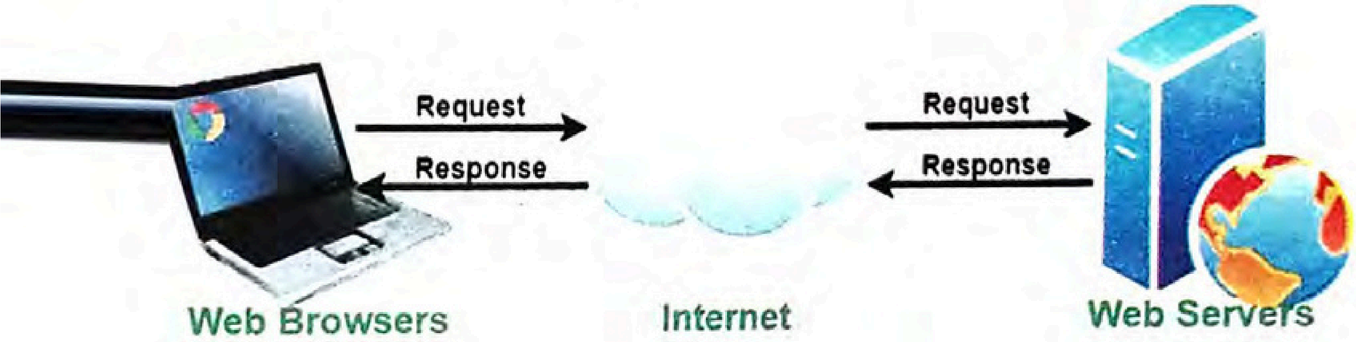
ویب سرور ایک ایسا کمپیوٹر ہے جو انٹرنیٹ پر ایک سے زیادہ یوزر کے لئے معلومات کو ذخیرہ کرنے کے لئے استعمال ہوتا ہے۔ ویب سرور کا بنیادی مقصد یوزر کے لیے ویب پیجز کو محفوظ کرنا، ان پر کارروائی اور ان کی فراہمی ہے۔ یہ دو طرفہ کمیونیکیشن ہائپر ٹیکسٹ ٹرانسفر

NOT FOR SALE

پروٹوکول (HTTP) کا استعمال کر کے کی جاتی ہے۔ ایک ویب سرور عام کمپیوٹر سے کہیں زیادہ تیز اور طاقت ور ہوتا ہے۔ ہر ویب ایک منفرد IP ایڈریس اور ممکنہ طور پر ڈومین کا نام ہوتا ہے۔ مثال کے طور پر، اگر یوزر ویب براؤزر میں https://www.urdupoint.com/news.html میں درج کرتا ہے تو، یہ ویب سرور کو درخواست بھیجتا ہے جس کا ڈو urdupoint.com ہے۔ سرور پھر news.html کے نام سے پیج نکالتا ہے اور اسے یوزر کے براؤزر پر بھیجتا ہے۔

کسی بھی کمپیوٹر کو سرور سافٹ ویئر انسٹال کر کے اور مشین کو انٹرنیٹ سے جوڑ کر ویب سرور میں تبدیل کیا جاسکتا ہے۔

شکل نمبر 7.4 انٹرنیٹ پر ویب سرور سے رابطہ کرنے والے براؤزر کی تصویر کشی کرتا ہے۔



شکل نمبر 7.4 ویب براؤزر اور ویب سرور

یونیفارم ریسورس لوکیٹر (URL)

یو آر ایل (یونیفارم ریسورس لوکیٹر) ایک ایسی فائل کا منفرد ایڈریس ہے جو انٹرنیٹ پر قابل رسائی ہے۔ کسی ویب سائٹ تک جانے کا ایک عام طریقہ یہ ہے کہ ویب براؤزر کی ایڈریس لائن میں اس کے ہوم پیج فائل کا URL درج کریں۔ تاہم، اس ویب سائٹ میں موجود کسی بھی فائل کو URL کے ساتھ بھی لکھا جاسکتا ہے۔

یو آر ایل کی درج ذیل شکل ہے۔

protocol://hostname/other_information

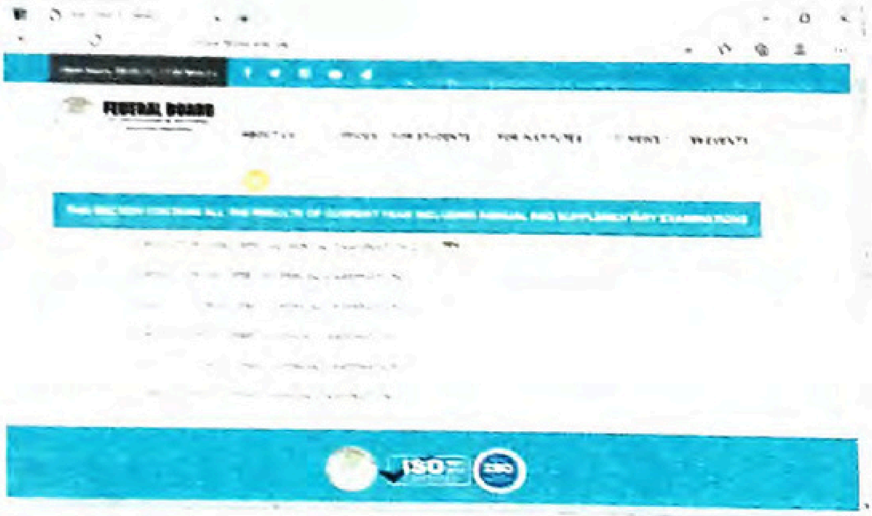
مثال کے طور پر، فیڈرل ایجوکیشن بورڈ کے ہوم پیج کا URL یہ ہے:

https://www.fbise.edu.pk/

(HTTP : ہائپر ٹیکسٹ ٹرانسفر پروٹوکول) بتاتا ہے کہ لنک سے معلومات کو کس طرح منتقل کیا جائے گا۔ ڈومین کا نام www.fbise، وہ کمپیوٹر ہے جس پر وسائل موجود ہیں۔ مخصوص فائلوں یا ذیلی ڈائریکٹریوں کے لنکس Domain name کے بعد مزید واضح ہو سکتے ہیں۔ ڈائریکٹری کے نام ایک فارورڈ سلیش کے ذریعہ الگ کر دیئے جاتے ہیں۔

URL https://www.fbise.edu.pk/result-main.php نتائج کے پیج کو دکھائے گا جیسا کہ شکل نمبر 7.5 میں دکھایا گیا ہے۔

NOT FOR SALE



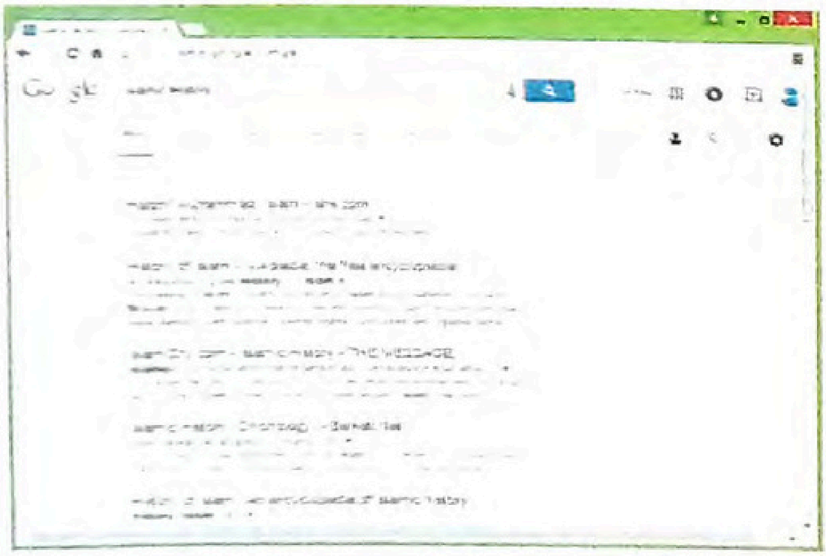
شکل نمبر 7.5 فیڈرل بورڈ کا ویب ایڈریس

سرچ انجن

سرچ انجن ایک ایسی ویب سائٹ ہے جو لوگوں کو کسی خاص موضوع پر مطلوبہ ویب سائٹ تلاش کرنے کی سہولت دیتی ہے۔ ویب پر لاکھوں ویب پیجز میں یوزر جو کچھ تلاش کرنا چاہتا ہے اُس تک ہو بہو پہنچنا اتنا آسان نہیں ہے۔ سرچ انجن کو ویب پر مخصوص معلومات تلاش کرنے میں یوزر کی مدد کے لئے بنایا گیا ہے۔

یوزرز سرچ انجن میں ایک یا ایک سے زیادہ مطلوبہ الفاظ (Keywords) میں ٹائپ کرتے ہیں۔ سرچ انجن نتائج کی ایک فہرست فراہم کرتا ہے جو یوزر کی تلاش کے مطابق ہوتی ہے۔ ہر ویب سائٹ کے لئے دکھائی جانے والی معلومات میں ایک عنوان اور ایک مختصر تفصیل شامل ہے۔ اگر یوزر کسی بھی سائٹ کو کھولنا چاہتا ہے تو وہ صرف لنک پر کلک کرتا ہے۔

آج کل، انٹرنیٹ پر بہت سارے سرچ انجن دستیاب ہیں، ہر ایک کی اپنی صلاحیتیں اور خصوصیات ہیں۔ مقبول سرچ انجن گوگل اور یاہو ہیں۔ گوگل سرچ انجن کا استعمال کرتے ہوئے معلومات کا حصول شکل نمبر 7.6 میں دکھایا گیا ہے۔



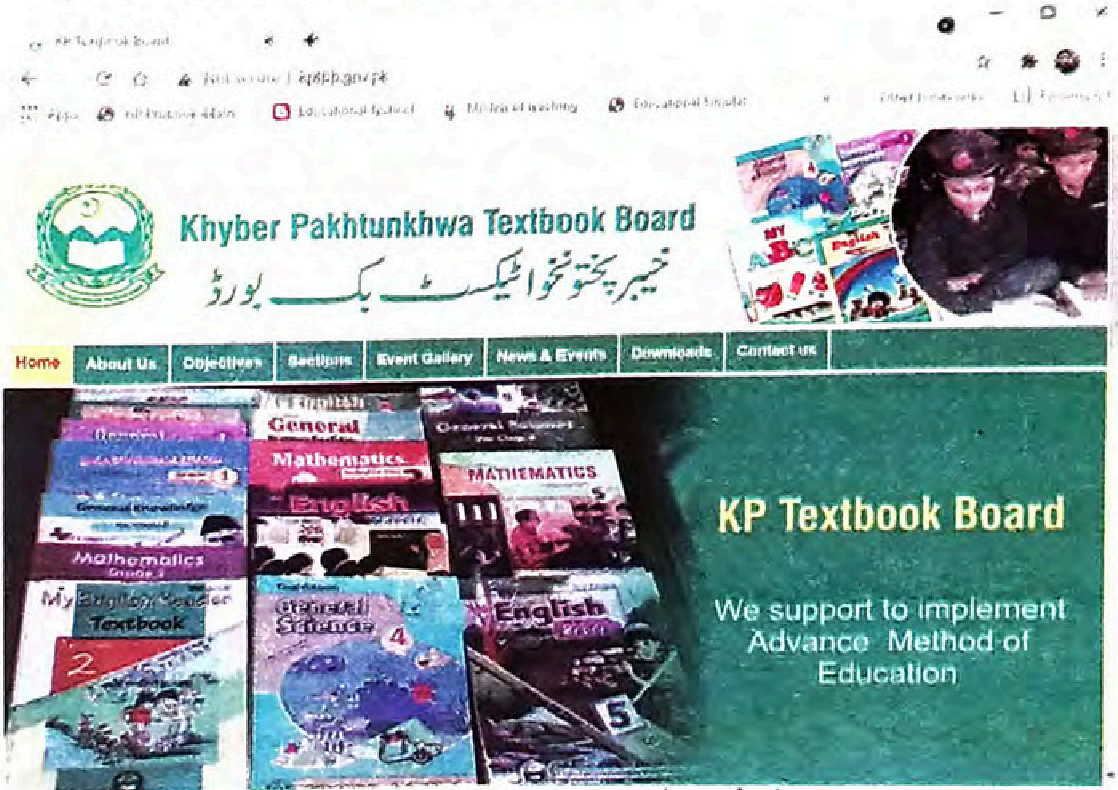
شکل نمبر 7.6: گوگل سرچ انجن کا استعمال کرتے ہوئے معلومات کی تلاش

NOT FOR SALE

ہوم پیج

کسی بھی ویب سائٹ کے فرنٹ پیج یا انڈیکس پیج کو ہوم پیج کہا جاتا ہے۔ یہ کسی ویب سائٹ کا مرکزی یا انتہائی پہلا ویب پیج ہے یوزر کو اس تک رسائی حاصل کرنے پر براؤزر میں کھلتا ہے۔ یہ ویب سائٹ کے دوسرے پیجز سے منسلک ہوتا ہے۔ یہ وہ صفحہ ہے جو ک ویب سائٹ کے وزٹ کرنے پر خود بخود کھلتا ہے۔

خیبر پختونخوا کی درسی کتب کے بورڈ کی ویب سائٹ کا ہوم پیج /http://kptbb.gov.pk شکل نمبر 7.7 میں دکھایا گیا ہے۔



شکل نمبر 7.7 /www.kptbb.gov.pk کا ہوم پیج

ویب ہوسٹنگ

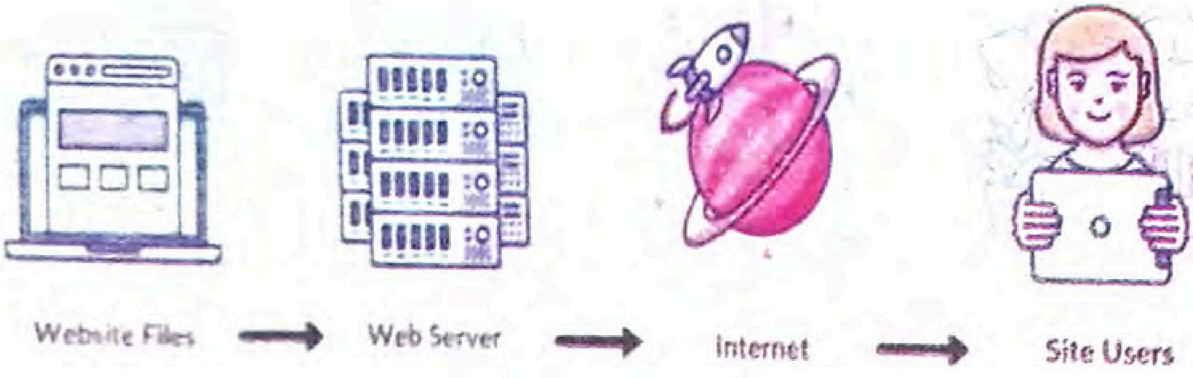
ویب ہوسٹنگ ایک آن لائن سہولت ہے، جو ویب ہوسٹنگ کمپنیوں کے ذریعہ فراہم کی جاتی ہے جو آپ کی ویب سائٹ یا ویب ایپلیکیشن کو ویب سرورز پر شائع کرنے اور کمپیوٹر یوزر کے لئے دستیاب ہونے کے قابل بناتی ہے۔ یہ کمپنیاں ہر وقت کمپیوٹر پر چلنے والی ویب سائٹیں انٹرنیٹ سے منسلک رکھتی ہیں۔

کمپیوٹر استعمال کرنے والے پہلے اپنے کمپیوٹر پر ویب پیج کو ڈیزائن کرتے ہیں اور اس کی جانچ کرتے ہیں تاکہ وہ یقینی بنائیں کہ وہ صحیح طریقے سے کام کر رہے ہیں۔ جب ویب پیج تیار ہو جاتے ہیں تو وہ دوسروں کو دستیاب ہونے کے لیے ویب سرور پر اپ لوڈ کیئے جاتے ہیں۔

ویب ہوسٹنگ کمپنیاں ویب سائٹ کے انتظام کے لیے 24 گھنٹے کی معاونت اور آن لائن کنٹرول فراہم کرتی ہیں۔ کسی ویب سائٹ کو سرور پر درکار جگہ ہوسٹنگ کی قسم پر منحصر ہے۔ ہوسٹنگ کی اہم اقسام Shared، Dedicated، VPS (ورچوئل پرائیویٹ سرور) اور reseller ہیں۔ ان سہولیات کی سرور پر استعمال ہونے والی ٹیکنالوجی، فراہم کردہ انتظامی سہولیات اور اضافی خدمات کی بنیاد پر درجہ بندی کی گئی ہے۔

NOT FOR SALE

ویب سائٹس کے مندرجات وہ تصاویر، آڈیوز اور ویڈیوز ہیں جو انٹرنیٹ سے جڑے سرور پر محفوظ ہیں۔ جب یوزر آپ کی ویب سائٹ دیکھنا چاہتے ہیں تو، وہ آپ کی ویب سائٹ کا پتہ اپنے براؤزر میں ٹائپ کریں گے اور پھر ان کا کمپیوٹر آپ کے سرور سے جڑ جائے گا۔ اس کے بعد آپ کے ویب پیجز انٹرنیٹ براؤزر کے ذریعہ یوزر تک پہنچائے جائیں گے جیسا کہ شکل نمبر 7.8(a) میں واضح کیا گیا ہے۔



شکل نمبر (a) 7.8 ویب ہوسٹنگ کا عمل

7.1.2 ویب سائٹس کی اقسام

انٹرنیٹ پر مختلف قسم کی ویب سائٹس موجود ہیں، ہر ایک مخصوص قسم کی خدمت مہیا کرتی ہے۔ ویب سائٹوں کی سب سے مشہور قسمیں یہ ہیں:

- ویب پورٹلز
- نیوز ویب سائیٹ
- معلوماتی ویب سائیٹ
- تعلیمی ویب سائیٹ
- ذاتی ویب سائیٹ
- کاروباری ویب سائیٹ
- تفریحی ویب سائیٹ

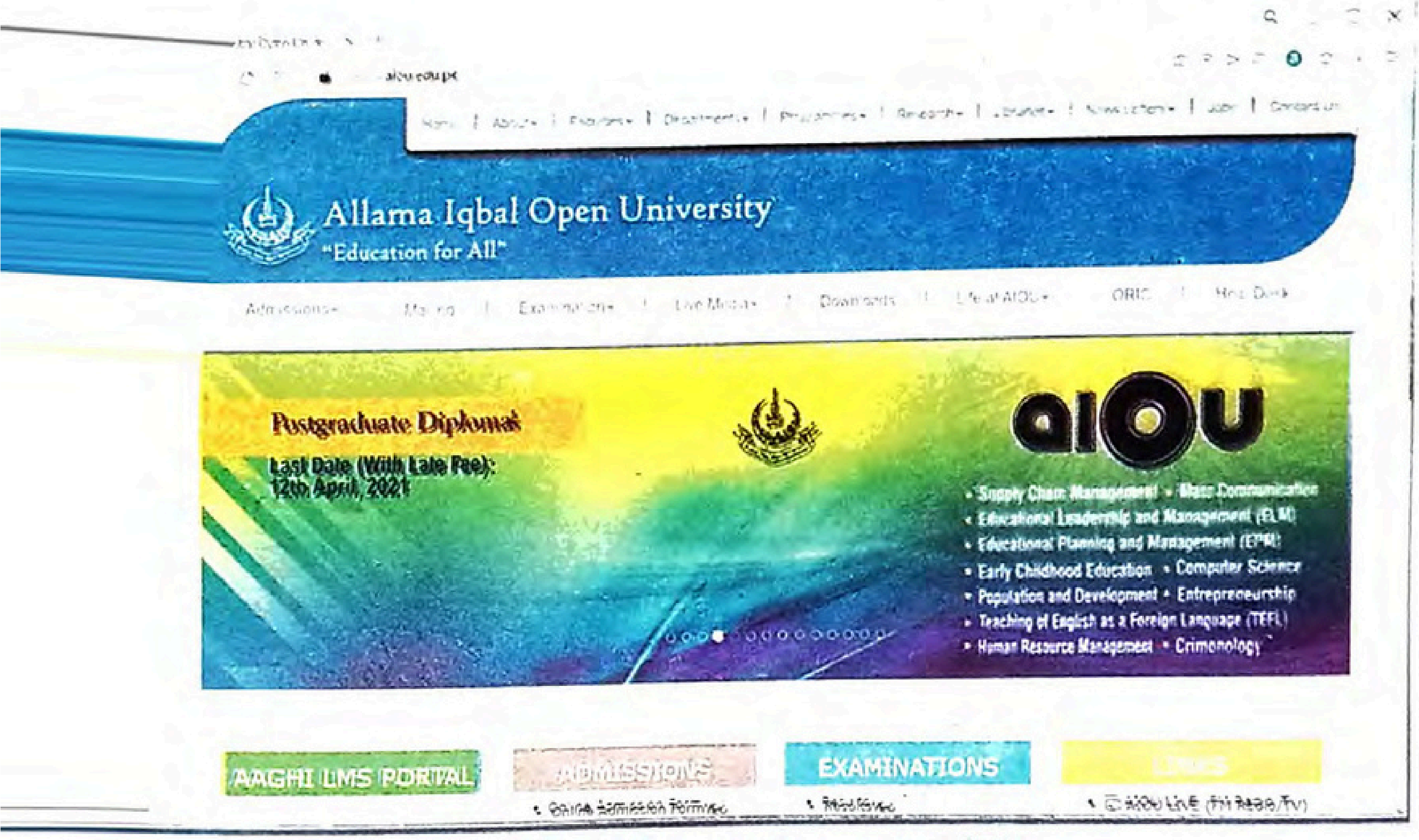
ویب پورٹلز

ایک ویب پورٹل ایک جامع ویب سائٹ ہے جو متعدد خدمات پیش کرتی ہے۔ آسان الفاظ میں، یہ ایک مینجمنٹ سسٹم کی طرح ہے جہاں کمپنیاں یا تنظیمیں معلومات یا ڈیٹا تیار کرتی ہیں اور ان کا تبادلہ کر سکتی ہیں جسے بعد میں دوبارہ استعمال کیا جا سکتا ہے۔ ویب پورٹلز مختلف قسم کی خدمات فراہم کرتے ہیں۔ ان میں آن لائن شاپنگ مالز، خبریں، اسٹاک کی قیمتیں، ای میل اور سرچ انجن وغیرہ شامل ہیں۔ ویب سائٹس کے برعکس، ویب پورٹلز انٹرنیٹ پر نجی مقامات ہیں جن تک ان کے منفرد URLs کے ذریعے ہی رسائی حاصل کی جا سکتی ہے، اور یوزرز کو اپنے لاگ ان IDs اور پاس ورڈ درج کرنے کی ضرورت ہوتی ہے۔

ویب پورٹلز یوزر پر مبنی ہوتے ہیں، اور ہر یوزر کا اپنا پاس ورڈ ہوتا ہے جو ان پورٹلز تک رسائی کے لیے استعمال کیا جا سکتا ہے۔ ان پورٹلز میں متحرک (Dynamic) مواد ہوتا ہے، یعنی جب بھی یوزر ان پورٹلز پر جاتا ہے ہر بار اس مواد میں تبدیلی آجاتی ہے۔

NOT FOR SALE

ویب پورٹلز ویب پر معلومات تک رسائی کا ذریعہ ہیں۔ مثال کے طور پر، کسی یونیورسٹی کا ویب پورٹل یونیورسٹی کے پس فیس کے ڈھانچے، سہولیات، نصاب، محکموں، پروگراموں وغیرہ کے بارے میں معلومات فراہم کرتا ہے۔ علامہ اقبال اوپن ویب پورٹل کو شکل نمبر 7.8(b) میں دکھایا گیا ہے۔



شکل نمبر 7.8(b) علامہ اقبال اوپن یونیورسٹی کا ویب پورٹل

نیوز ویب سائیٹ

آن لائن نیوز سائیٹ دنیا بھر میں تازہ ترین حالات حاضرہ پر مبنی معلومات کے ذریعے لوگوں کو تعلیم اور آگاہی فراہم کرنے میں اہم کردار ادا کرتی ہیں۔ لوگوں کے پاس کل کی چھپی ہوئی اخبار پڑھنے کے لیے زیادہ وقت نہیں ہے، لہذا وہ ہمیشہ تازہ ترین خبریں حاصل کرنے کے لئے نیوز سائیٹ کا رخ کرتے ہیں۔ یہ معلومات تک رسائی کا آسان اور قابل اعتماد ذریعہ ہے جو آپ کو کسی بھی خبر پر اپنی رائے اور خیالات کا اظہار کرنے کا اختیار بھی فراہم کرتا ہے۔ کچھ نیوز ویب سائیٹ ہمارے آس پاس ہونے والی کسی تازہ ترین بریکنگ نیوز کی ویڈیوز اور تصاویر بھی فراہم کرتی ہیں۔ نیوز سائٹ کا سب سے اہم حصہ اس کے آرکائیو سیکشن میں موجود تمام خبروں اور مضامین پر مشتمل ہے تاکہ آپ کسی بھی وقت ماضی کی کوئی بھی خبر تلاش کر سکیں۔ ایک آن لائن نیوز سائیٹ آپ کو دنیا کے مختلف امور پر تازہ ترین اور مستند ذرائع سے معلومات فراہم کرتی ہے۔

نیوز ویب سائٹ کی کچھ مثالیں: www.mashriqtv.pk/E-Paper، www.tribune.com.pk، www.thenews.com.pk، اور www.nation.com.pk۔ روزنامہ مشرق نیوز کی ویب سائٹ کو شکل نمبر 7.9 میں دکھایا گیا ہے۔

NOT FOR SALE

شکل نمبر 7.9 روزنامہ مشرق کی ویب سائٹ

معلوماتی ویب سائٹ

معلوماتی ویب سائٹ وہ سائیٹ ہیں جو ٹیکسٹ، گرافکس، آواز اور ویڈیوز کی شکل میں کسی بھی عنوان کے بارے میں تفصیلی معلومات فراہم کرنے کے لئے بنائی جاتی ہیں۔ مثال کے طور پر، www.encyclopedia.com معلوماتی ویب سائیٹ ہے جو لوگوں اور انسانیت، جانوروں، کاروبار، کمپیوٹر، ادب، صحت، تعلیم وغیرہ جیسے موضوعات کے بارے میں معلومات فراہم کرتی ہے۔ ایک اور مثال، www.wikipedia.org ایک معلوماتی ویب سائیٹ ہے جو لاکھوں مضامین پر معلومات فراہم کرتی ہے۔ اس میں یوزر موضوعات میں ترمیم اور انہیں شئیر بھی کر سکتے ہیں۔ اس ویب سائیٹ کو شکل نمبر 7.10 میں دکھایا گیا ہے۔

شکل نمبر 7.10 معلوماتی ویب سائٹ

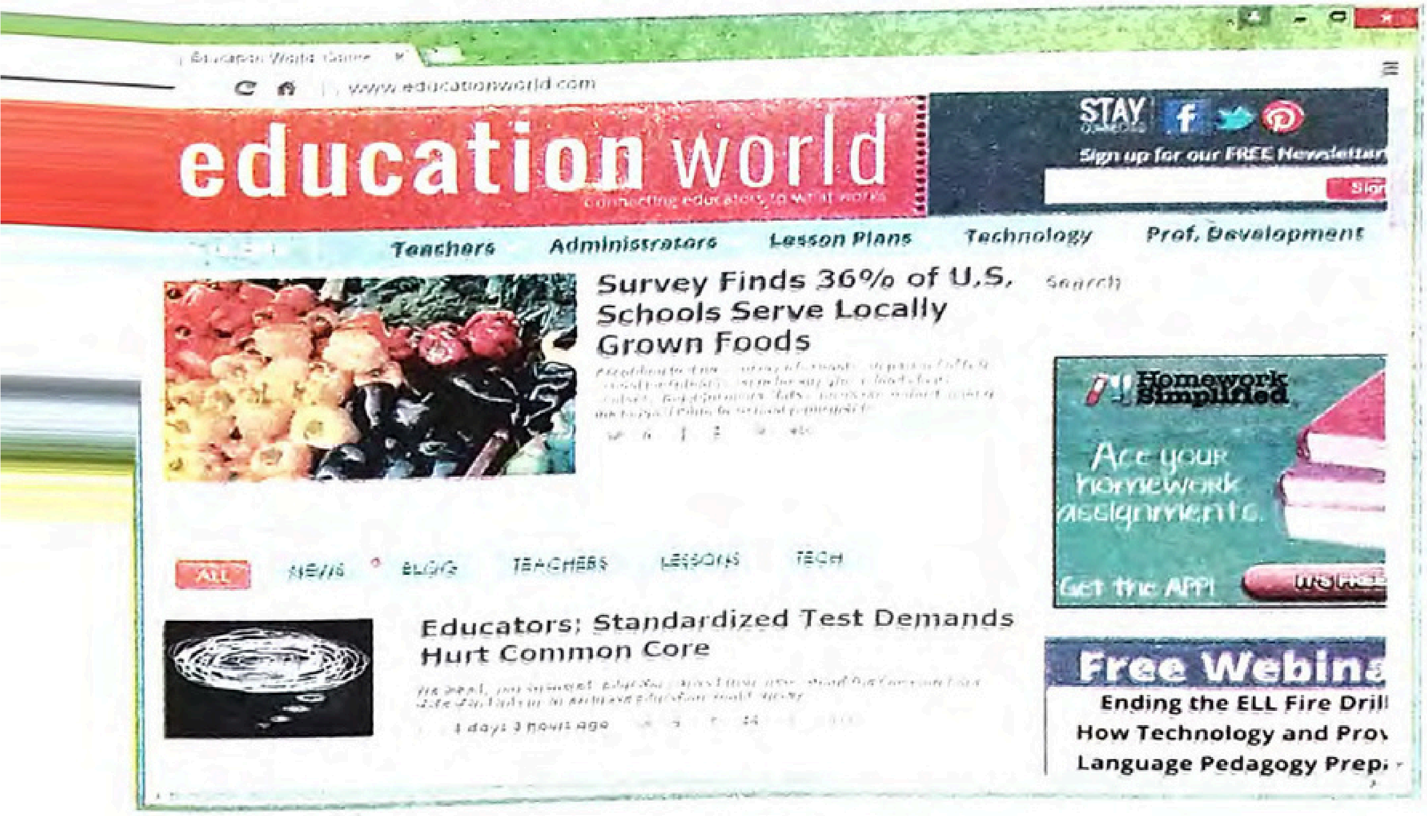
NOT FOR SALE

تعلیمی ویب سائیٹ

تعلیمی ویب سائیٹ تعلیمی مقاصد کے لئے بنائی جاتی ہے۔ اس سائیٹ میں لوگوں کو مختلف موضوعات پر آگاہی دینے او کورسز فراہم کرنے کے لئے حرکت پذیر (Animated)، سلائیڈ پریزنٹیشنز اور ٹیوٹوریل موجود ہوتے ہیں۔ معلومات کو بہت میں پیش کیا جاتا ہے۔ کسی بھی تعلیمی ویب سائیٹ کا مقصد لوگوں کو علم فراہم کرنا ہوتا ہے جو کسی موضوع کو بہتر طور پر سمجھنے کے خ اور علم کا حصول چاہتے ہیں۔

تعلیمی ویب سائیٹ کی کچھ مثالیں ہیں www.educationworld.com، www.ilmkidunya.com، www.knowledgepk.com وغیرہ۔

تعلیمی ویب سائیٹ www.educationworld.com کا ہوم پیج شکل نمبر 7.11 میں دکھایا گیا ہے۔



شکل نمبر 7.11 تعلیمی ویب سائیٹ

ذاتی ویب سائیٹ

ذاتی ویب پیج زیادہ تر معلوماتی اور تفریحی مقاصد کے لئے استعمال ہوتا ہے، ساتھ ہی ساتھ ذاتی کیریئر کی مارکیٹنگ (فرد کی صلاحیتوں، تجربے اور سی وی کی ایک فہرست پر مشتمل)، مشترکہ مفادات رکھنے والے دوسرے لوگوں کے ساتھ سماجی رابطے، یا جگہ کی حیثیت سے بھی استعمال کیا جاسکتا ہے۔

کوئی بھی فرد ذاتی ویب سائیٹ ذاتی استعمال کے لیے تخلیق کرتا ہے اور اسے برقرار رکھتا ہے۔ ذاتی ویب سائیٹ عوام کے لئے کھلی ہوتی ہے

NOT FOR SALE

اور کوئی بھی ان پر تبصرے ملاحظہ کر سکتا ہے۔

کاروباری ویب سائیٹ

کاروباری ویب سائیٹ کاروباری تعلقات کو برقرار رکھنے اور انٹرنیٹ کے توسط سے معلومات، خدمات اور اشیاء کی فروخت کے لئے سہولیات مہیا کرتی ہیں۔ یہ سائیٹ کاروباری دستاویزات کے تبادلے میں سہولت فراہم کرتی ہیں، جیسے سپلائرز اور یوزرز کے مابین آرڈرز یا رسید۔ کاروباری سائیٹوں کے ذریعہ فراہم کردہ خدمات میں سامان کی فروخت، خدمات کی فراہمی جیسے بینکنگ، ٹکٹوں کی بکنگ، اسٹاک مارکیٹ کے لین دین اور یہاں تک کہ فاصلاتی تعلیم بھی شامل ہیں۔ کاروباری ویب سائیٹوں میں سے ایک مثال www.daraz.pk ہے، جو ایک آن لائن بازار اور لاجسٹک کمپنی ہے جو یوزرز کو آن لائن خریداری کی سہولت مہیا کرتی ہے۔

شکل نمبر 7.12 کاروباری ویب سائیٹ

تفریحی ویب سائیٹ

ایک تفریحی ویب سائیٹ ایک آن لائن برانڈ یا ویب سائیٹ ہے جو اپنے یوزرین کے لئے اعلیٰ معیار کے تفریحی مواد تیار کرنے پر مرکوز ہوتی ہے۔

تفریحی ویب سائیٹوں میں نیوز آئٹمز، کمپیوٹر گیمز، کامیڈی، کھیل، کہانیاں، فلمیں، آڈیو اور ویڈیو شامل ہیں۔ مشہور تفریحی ویب سائیٹوں کی مثالوں میں Netflix، youtube، Sound Cloud وغیرہ شامل ہیں۔

NOT FOR SALE

شکل نمبر 7.13 تفریحی ویب سائیٹ

7.2 HTML کا تعارف

HTML کا مطلب ہائپر ٹیکسٹ مارک اپ لینگو یج ہے، جو ویب پیجز تیار کرنے کے لئے ویب پر سب سے زیادہ استعمال ہونے والی لینگو یج ہے۔ HTML کو برنرز لی نے 1991 میں بنایا تھا۔ HTML ڈاکومنٹ کو پڑھنے اور اسے اسکرین پر ویب پیجز کی حیثیت سے ظاہر کرنے کے لئے، انٹرنیٹ ایکسپلورر یا گوگل کروم جیسے براؤزر کا استعمال ہوتا ہے۔

7.2.1 ہائپر ٹیکسٹ مارک اپ لینگو یج

HTML ایک مارک اپ لینگو یج ہے کیونکہ اس میں وہ ٹیگ استعمال ہوتے ہیں جو ویب براؤزر کو بتاتے ہیں کہ ویب پیج کو کیسے ظاہر کرنا ہے۔ ہر پیج میں دوسرے پیج سے رابطہ کی ایک سیریز ہوتی ہے جسے ہائپر لنکس (Hyperlinks) کہتے ہیں۔ انٹرنیٹ پر آپ کو نظر آنے والا ہر ویب پیج HTML کے کسی بھی ورژن کے استعمال سے لکھا جاتا ہے۔ اس کو ہائپر ٹیکسٹ دستاویزات بنانے کے لئے استعمال کیا جاتا ہے جو ٹیکسٹ، تصاویر، آوازیں، ویڈیو کلپس اور سبھی کو ایک جگہ پر لے آتا ہے۔ HTML فائلیں سادہ ٹیکسٹ فائلیں ہیں، لہذا انھیں سادہ ایڈیٹر جیسے Notepad یا Wordpad یا ++Notepad اور Visual Studio Code کا استعمال کرتے ہوئے تخلیق کیا جاسکتا ہے۔

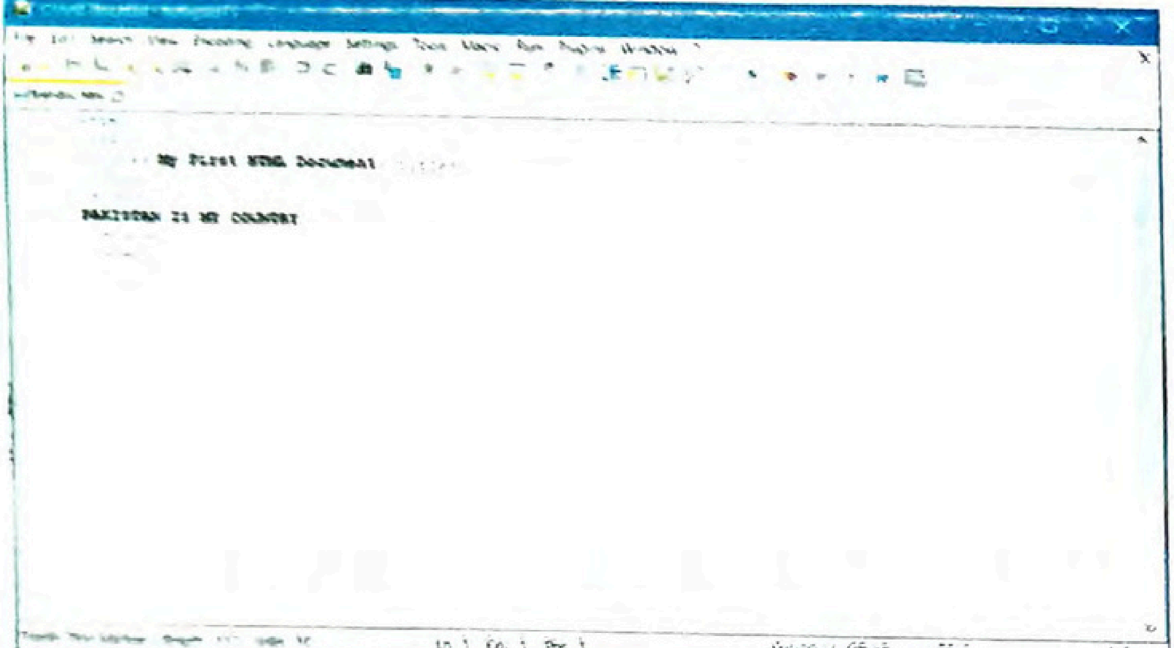
7.2.2 HTML ڈاکومنٹ بنانا اور براؤزر میں دکھانا

اس یونٹ میں ++Notepad کو HTML ڈاکومنٹ بنانے کے لئے استعمال کیا گیا ہے۔ اس ایڈیٹر کو https://notepad-plus-plus.org/downloads/ لنک کا استعمال کرتے ہوئے انٹرنیٹ سے ڈاؤن لوڈ کیا جاسکتا ہے۔

NOT FOR SALE

پہلا HTML ڈاکومنٹ بنانا

Notepad++ پروگرام کھولیں اور شکل 7.14 کے مطابق ٹیکسٹ ٹائپ کریں۔



شکل نمبر 7.14 HTML ڈاکومنٹ

اب اپنے پہلے HTML ڈاکومنٹ کو محفوظ کرنے کے لیے File مینو کو کھولیں اور Save As پر کلک کر کے First-doc.html کے نام سے محفوظ کریں۔

HTML ڈاکومنٹ دکھانا (ویب پیج)

براؤزر ونڈو میں first-doc.html ڈاکومنٹ کو کھولنے کے لئے، اس جگہ پر جائیں جہاں آپ نے اسے محفوظ کیا تھا۔ کھولنے کے لئے اس پر ڈبل کلک کریں۔ براؤزر ونڈو اس کو دکھائے گی جیسا کہ شکل نمبر 7.15 میں دکھایا گیا ہے۔

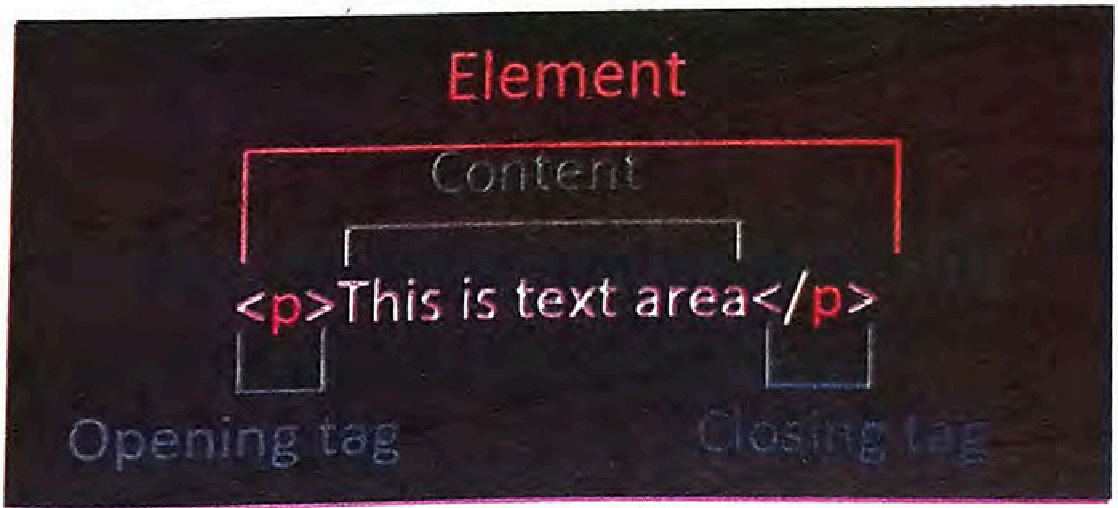
شکل نمبر 7.15 HTML ڈاکومنٹ دکھانے والی براؤزر ونڈو

NOT FOR SALE

7.2.3 HTML ٹیگس کا استعمال

HTML ٹیگز کلیدی الفاظ (Keywords) کی طرح ہیں جو اس بات کی وضاحت کرتے ہیں کہ ویب براؤزر کس طر
فارمیٹ اور ظاہر کرے گا۔ ٹیگز کی مدد سے، ایک ویب براؤزر HTML مواد اور ایک سادہ مواد میں فرق کر سکتا ہے۔ ITML
ٹیگ مختلف کام انجام دیتا ہے۔

HTML ٹیگز دو زاویہ بریکٹ < اور > سے گھرا ہوتا ہے۔ زیادہ تر ٹیگز کے دو حصے ہوتے ہیں، اوپننگ اور کلوزنگ حصہ۔ مث
طور پر، <html> افتتاحی ٹیگ ہے اور </html> کلوزنگ ٹیگ ہے۔ نوٹ کریں کہ اختتامی ٹیگ ابتدائی ٹیگ کی طرح ہی ہے، لیک
میں ایک اضافی فارورڈ سلیش (/) ہوتی ہے۔ ان دونوں ٹیگز کے درمیان ٹیکسٹ مواد (content) کے نام سے جانا جاتا ہے جیسا کہ شکل
7.16 میں دکھایا گیا ہے۔ پہلا ٹیگ اسٹارٹ ٹیگ ہے اور دوسرا ٹیگ سٹاپ ٹیگ ہے۔ HTML ٹیگ انگریزی کے چھوٹے اور بڑے د
حروف میں لکھا جا سکتا ہے۔ اس پورے یونٹ میں انگریزی کے چھوٹے حروف استعمال کیے جائیں گے۔ کچھ ٹیگز میں کچھ اضافی خصوص
بھی ہوتی ہیں۔ یہ اوصاف عام طور پر ابتدائی ٹیگ کے اندر لکھے جاتے ہیں۔



شکل نمبر 7.16 ٹیگز کھولنا اور بند کرنا

7.2.4 HTML ہیڈ اور باڈی ٹیگز

پہلے HTML ڈاکومنٹ میں HTML ٹیگ کو مارک اپ کرنے کے لئے درج ذیل ٹیگز استعمال کیے گئے ہیں۔

<html> </html> ٹیگز

HTML ڈاکومنٹ میں پہلا ٹیگ <html> ہے جو HTML ڈاکومنٹ کے آغاز کی نشاندہی کرتا ہے۔ آخری ٹیگ </html> ہے جو
براؤزر کو بتاتا ہے کہ یہ HTML ڈاکومنٹ کا اختتام ہے۔

<head> </head> ٹیگز

یہ ویب پیج کو عنوان دینے کے لئے استعمال ہوتا ہے۔

NOT FOR SALE

<body> </body>ٹیگز

body ٹیگ کے اندر جو کچھ بھی ٹائپ کیا گیا ہے وہ براؤزر ونڈو میں دکھایا جائے گا جیسا کہ شکل نمبر 7.15 میں دکھایا گیا ہے۔

اگر HTML فائل میں کوئی تبدیلی کی ضرورت ہو تو، اسے ٹیکسٹ ایڈیٹر میں دوبارہ کھول کر تبدیل کیا جا سکتا ہے۔ اسے کھولنے کے لئے، فائل کو منتخب کرنے کے لئے اس پر کلک کریں اور پھر رائٹ کلک کریں، Open with کو منتخب کریں اور شارٹ کٹ مینو میں ٹیکسٹ ایڈیٹر پر کلک کریں۔ اب اس میں ترمیم کریں اور اسے دوبارہ محفوظ کریں۔

7.3 ٹیکسٹ فارمیٹنگ (Text Formating)

HTML نے متعدد ٹیگز مہیا کیے ہیں جن کے استعمال سے آپ اپنے ٹیکسٹ کو عام ٹیکسٹ سے مختلف نظر آنے کے لیے استعمال کر سکتے ہیں، مثال کے طور پر، آپ اس ٹیکسٹ کو بولڈ بنانے کے لئے <b> ٹیگ کا استعمال کر سکتے ہیں، ٹیکسٹ اٹالک بنانے کے لیے <i> ٹیگ اور ٹیکسٹ کو نمایاں کرنے کے لیے <mark> ٹیگ، <ins> یا <del> ٹیگ کو تبدیلی کو ظاہر کرنے کے لیے استعمال کرتے ہیں۔

7.3.1 ٹیکسٹ فارمیٹنگ کی اساسیات (Basics of Text Formatting)

پیج ٹائٹل لگانا (Specifying a Page Title)

<title> </ title > ٹیگز پیج کو عنوان دینے کے لئے استعمال کر سکتے ہیں۔

ان ٹیگز کے درمیان ٹیکسٹ پیج کو عنوان دینے کے لئے استعمال ہوتا ہے۔ یہ ویب براؤزر کے ٹائٹل بار پر ظاہر ہوتا ہے۔

مثال کے طور پر </title> My First HTML Document < title > ویب براؤزر کے ٹائٹل بار پر ٹیکسٹ، **My First HTML Document** ڈاکومنٹ دکھائے گا۔

پیراگراف بنانا (Creating a Paragraph)

<p></p> ٹیگ پیراگراف بنانے کے لئے استعمال ہوتے ہیں۔ HTML کسی پیراگراف سے پہلے اور اس کے بعد خود بخود ایک اضافی خالی لائن شامل کرتا ہے۔

لائن بریک ڈالنا (Inserting Line Break)

 ٹیگ لائن بریک ٹیگ ہے۔ یہ جہاں بھی لکھا جاتا ہے تو یہ اس لائن کو ختم کرنے کے لئے استعمال ہوتا ہے اور اس میں اختتامی ٹیگ نہیں ہوتا ہے۔

خالی جگہیں ڈالنا (Inserting Spaces)

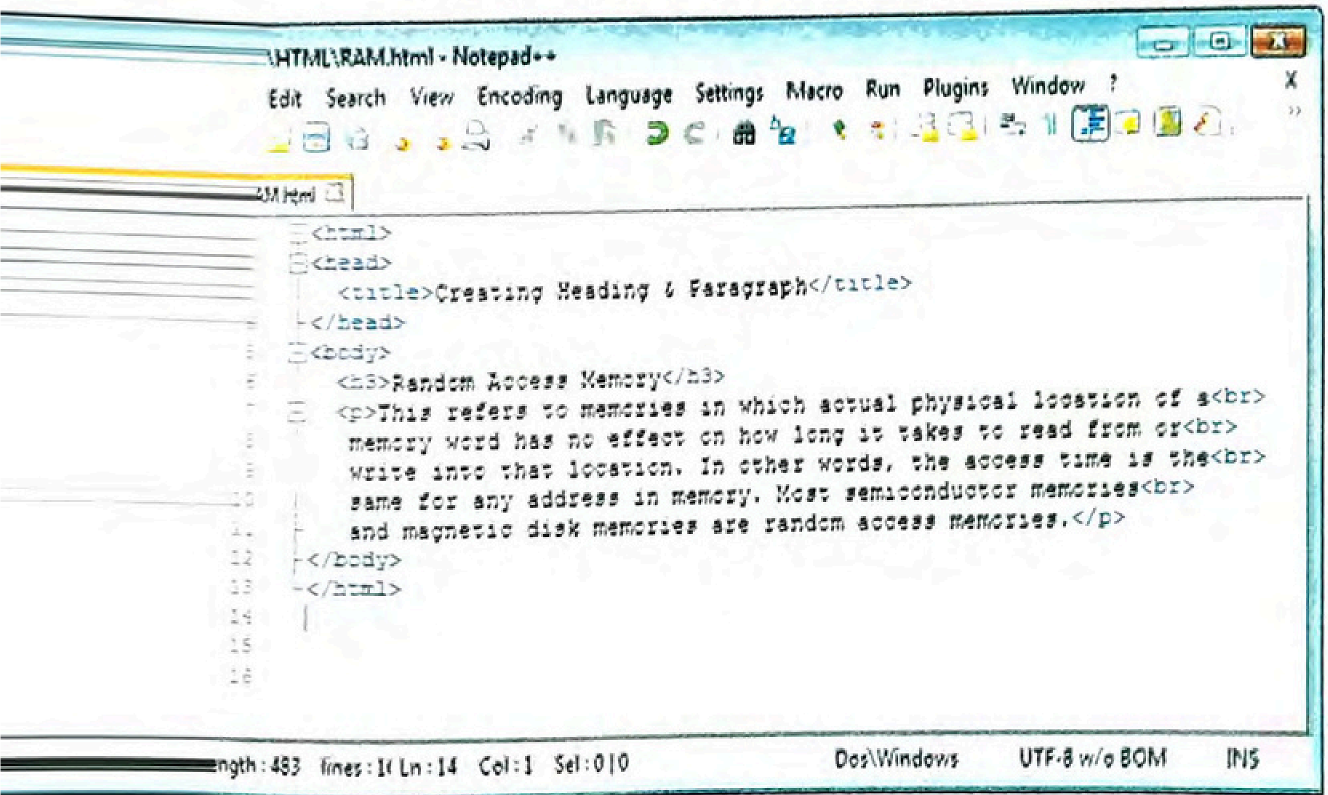
عام طور پر HTML دستاویزات میں خالی جگہوں کی ضرورت ہوتی ہے۔ اگر یوزر کسی بھی ڈاکومنٹ میں کہیں بھی ایک سے زیادہ space ڈالتا ہے تو، براؤزر صرف ایک space دکھائے گا، باقی spaces کو ختم کر دیا جائے گا۔ اگر یوزر کسی HTML ڈاکومنٹ میں بہت سی spaces کو رکھنا چاہتا ہے تو پھر ایک سے زیادہ * * کا استعمال کرنا چاہیئے۔

NOT FOR SALE

عنوانات / ذیلی عنوانات ڈالنا (Inserting Heading / Subheading)

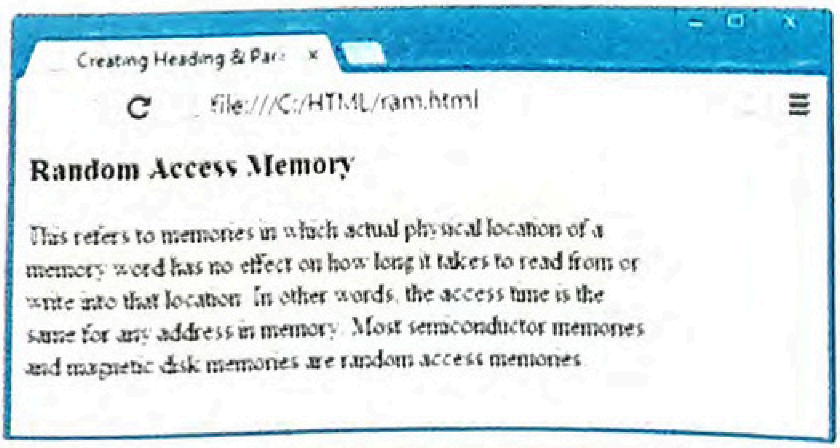
<h1></h1> سے <h6></h6> چھ ہیڈنگ ٹیگ ہیں۔ <h1></h1> ٹیگز سب سے بڑی سرخی اور <6 ٹیگز سب سے چھوٹی سرخی کیلئے استعمال ہوتے ہیں۔

مندرجہ ذیل HTML ڈاکومنٹ یہ ظاہر کرتا ہے کہ ویب پیج میں مذکورہ بالا ٹیگز کو کس طرح استعمال کیا جائے۔



شکل نمبر 7.17 HTML ڈاکومنٹ میں بنیادی ٹیگ کا استعمال

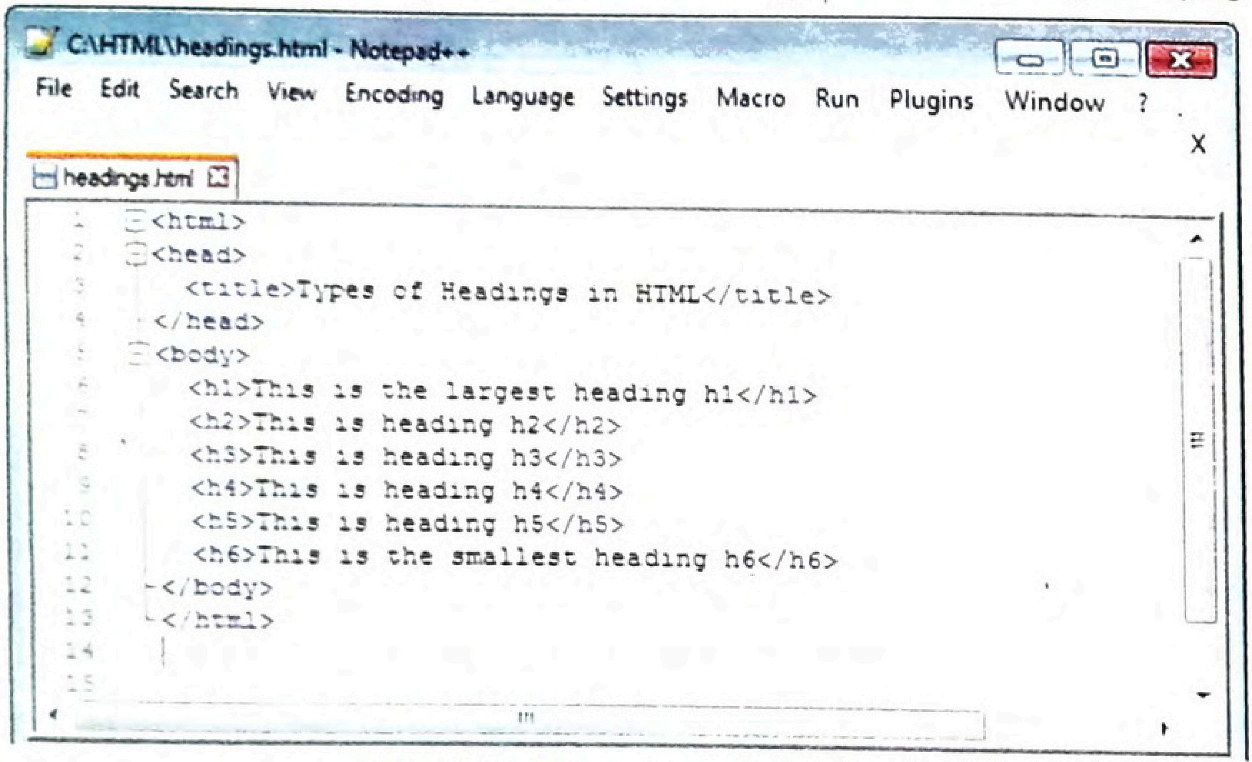
براؤزر ونڈو اسے ایسا دکھائے گی جیسا کہ شکل نمبر 7.18 میں دکھایا گیا ہے۔



شکل نمبر 7.18 براؤزر ونڈو جو ہیڈنگ اور پیراگراف کو دکھا رہی ہے

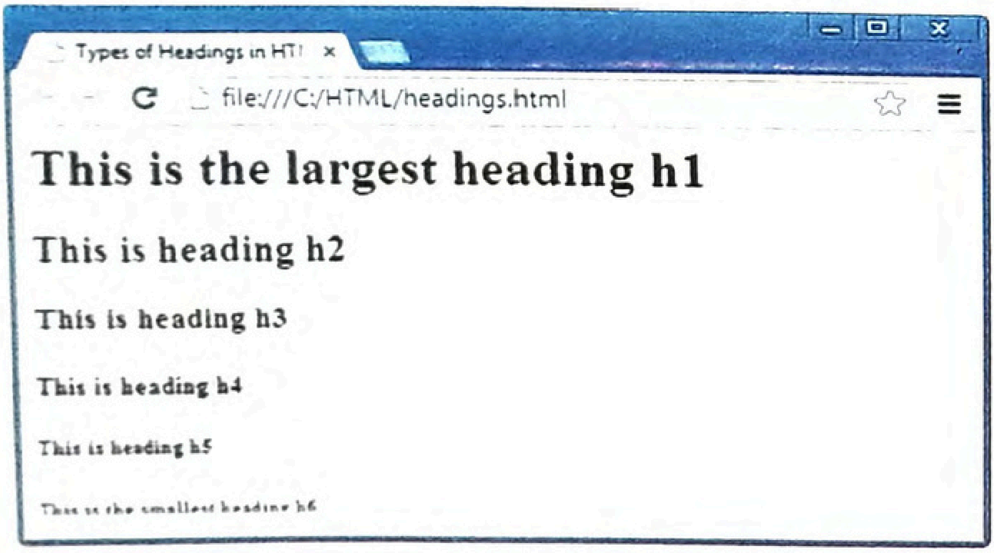
NOT FOR SALE

شکل نمبر 7.19 میں HTML ڈاکومنٹ تمام سرخی والے ٹیگز کے استعمال کو ظاہر کرتی ہے۔



شکل نمبر 7.19 HTML ڈاکومنٹ میں تمام ہیڈنگ ٹیگز کا استعمال

براؤزر ونڈو اس کو دکھائے گی جیسا کہ شکل نمبر 7.20 میں دکھایا گیا ہے۔



شکل نمبر 7.20 براؤزر ونڈو میں دکھائی جانے والی مختلف ہیڈنگز

7.3.2 ٹیکسٹ فارمیٹنگ ٹیگز اور ان کا استعمال

HTML ہمیں ٹیکسٹ فارمیٹ کرنے کی صلاحیت فراہم کرتا ہے بالکل اسی طرح جیسے ہم ایم ایس ورڈ یا کسی بھی ٹیکسٹ ایڈیٹنگ ٹول میں کرتے ہیں۔ ٹیکسٹ فارمیٹنگ ٹیگز کو ٹیکسٹ فارمیٹ کرنے کے لئے استعمال کیا جاتا ہے۔ کچھ عام طور پر استعمال شدہ ٹیکسٹ

NOT FOR SALE

فارمیٹنگ ٹیگ ذیل میں بیان کیے گئے ہیں۔

**Bold **

ٹیگز کے درمیان موجود ٹیکسٹ bold ہوگا اور اس کے آس پاس موجود ٹیکسٹ کے مقابلہ میں نمایاں ہوگا، جیسے ورڈ پروسیسر میں

مثال کے طور پر: Chapter 7 Input Devices

نتیجہ: **Chapter 7 Input Devices**

Underline <u></u>

یہ ٹیگ کے اندر موجود ٹیکسٹ کو اجاگر کریں گے۔ یہ نوٹ کرنا ضروری ہے کہ HTML لنکس کے نیچے پہلے ہی لائن ہوتی ہے اضافی ٹیگ کی ضرورت نہیں ہے۔

مثال کے طور پر: <u>Please Note:</u>

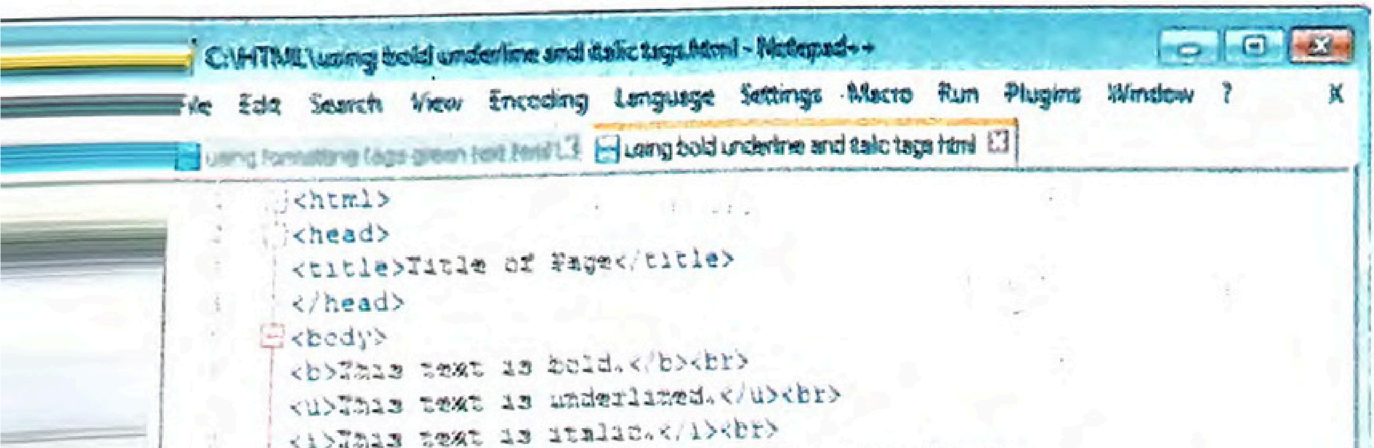
نتیجہ: Please Note

Italic <i></i>

یہ ٹیگ ٹیکسٹ کو ترچھا بنانے کیلئے استعمال کیا جاتا ہے۔

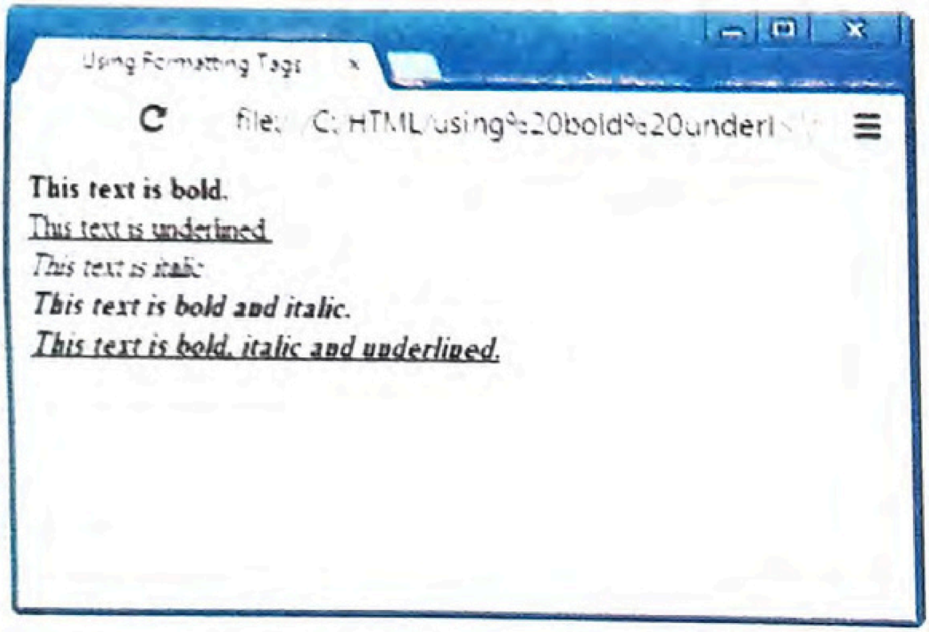
مثال کے طور پر: <i>Pakistan Cricket Team</i>

نتیجہ: *Pakistan Cricket Team*



شکل نمبر 7.21 HTML ڈاکومنٹ میں بولڈ، اٹالک اور انڈر لائن کا استعمال

NOT FOR SALE



شکل نمبر 7.22 ویب پیج پر بولڈ، اٹالک اور انڈر لائن کا استعمال

فونٹ   ٹیگ

فونٹ ٹیگ size، color، اور face کے اوصاف (attribute) کا استعمال کرتے ہوئے ٹیکسٹ کے سائز، کلر، یا فیس کو تبدیل کرنے کے لئے استعمال ہوتا ہے۔

فونٹ کا سائز

size فونٹ کا سائز تبدیل کرنے کے لئے استعمال ہوتا ہے۔ ? کی جگہ 1 سے 7 تک کوئی بھی نمبر لکھیں۔ 1 سب سے چھوٹا ہے اور 7 فونٹ کا سب سے بڑا سائز ہے۔

مثال کے طور پر: Types of Printers

نتیجہ: Types of Printers

فونٹ کا رنگ

Color ٹیگز میں موجود ٹیکسٹ کے رنگ کو تبدیل کرنے کے لئے استعمال کیا جاتا ہے۔ ? کی جگہ brown، black، blue، green، maroon، orange، pink، red، white، yellow وغیرہ لکھ سکتے ہیں۔

مثال کے طور پر: Healthy Cow

نتیجہ: Healthy Cow

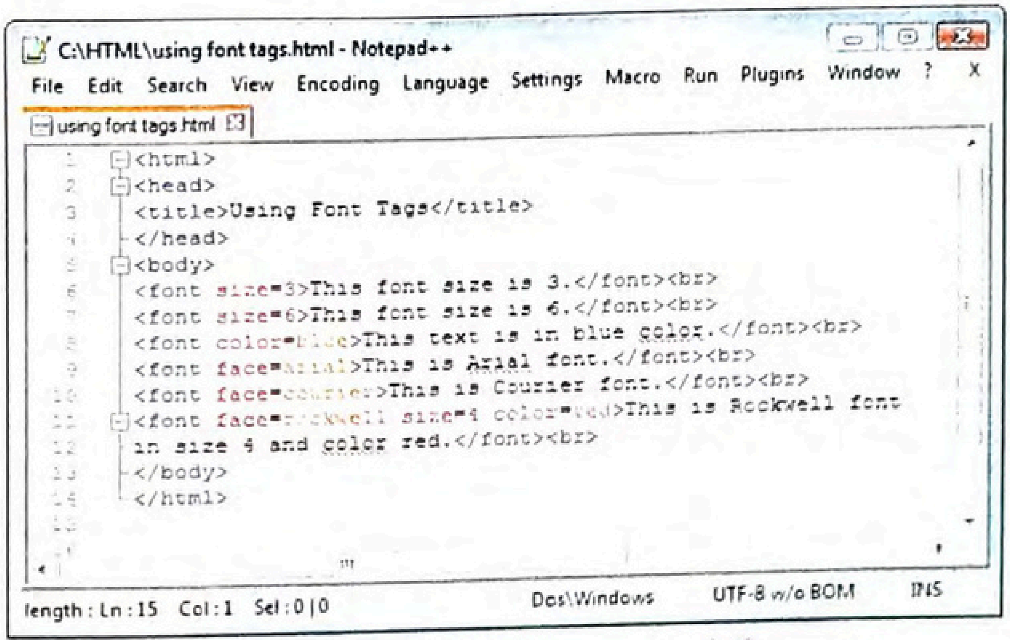
فونٹ فیس

face کا استعمال ٹیکسٹ کے فونٹ face کو تبدیل کرنے کے لئے کیا جاتا ہے جو ٹیگز میں ہوتا ہے۔ ? کو Arial، Courier، Calibri، TimesNewRoman وغیرہ سے تبدیل کر سکتے ہیں۔

NOT FOR SALE

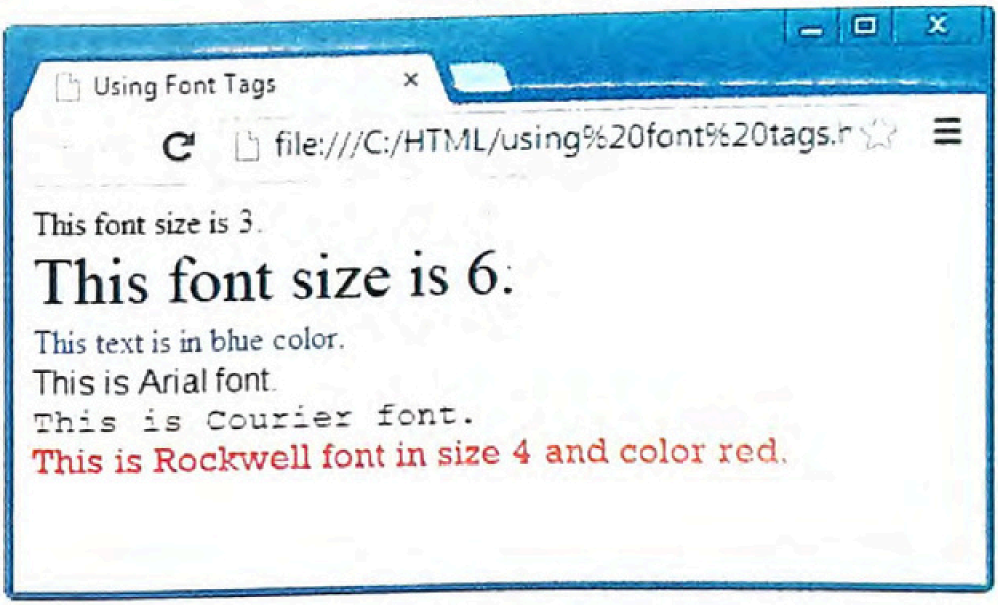
مثال کے طور پر: FORMATTING TAGS

نتیجہ: **FORMATTING TAGS**



شکل نمبر 7.23 HTML ڈاکومنٹ میں font ٹیگ کا استعمال

براؤزر ونڈو اس کو دکھائے گی جیسا کہ شکل نمبر 7.24 میں دکھایا گیا ہے۔



شکل نمبر 7.24 ویب پیج میں فونٹ ٹیگز کا استعمال

**سپر سکرپٹ Superscript **

یہ ٹیگز اپنے درمیان لکھے گئے ٹیکسٹ کو سپر سکرپٹ میں تبدیل کر دینگے۔

مثال کے طور پر: X²

نتیجہ: X^2

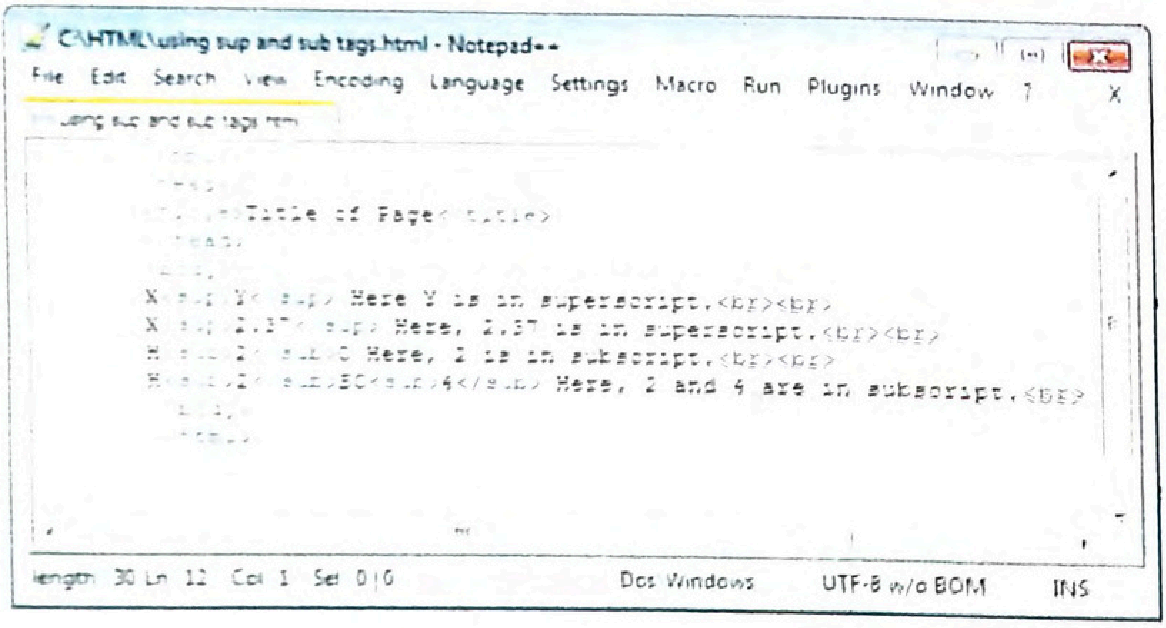
NOT FOR SALE

سب سکرپٹ Subscript

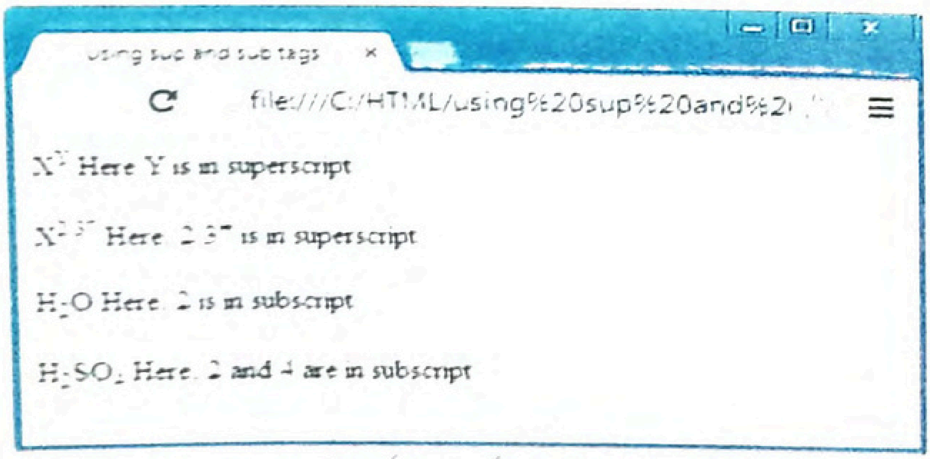
یہ ٹیگز اپنے درمیان لکھے گئے ٹیکسٹ کو سب سکرپٹ میں تبدیل کر دیں گے۔

مثال کے طور پر: O₂

نتیجہ: O_2



شکل نمبر 7.25 HTML کوڈ میں سپر اسکرپٹ اور سب اسکرپٹ کا استعمال



7.26 ویب پیج میں سپر اسکرپٹ اور سب سکرپٹ ٹیگز کا استعمال

7.4 HTML میں فہرست بنانا

فہرستوں کو متعلقہ معلومات کے ٹکڑوں کو ایک ساتھ جوڑنے کے لئے استعمال کیا جاتا ہے تاکہ وہ ایک دوسرے کے ساتھ واضح طور پر وابستہ ہوں اور پڑھنے میں آسانی ہو۔ جدید ویب ڈویلپمنٹ میں، فہرستیں عام طور پر نیویگیشن کے ساتھ ساتھ عمومی مواد کے لیے بھی استعمال ہوتی ہیں۔

NOT FOR SALE

اسٹرکچرل نقطہ نظر سے لسٹ اچھی ہیں کیونکہ وہ ایک ڈاکومنٹ کو سمجھنے میں آسان بنادیتی ہیں۔

HTML میں لسٹ بنانے کے لئے مختلف طریقے استعمال ہوتے ہے۔

7.4.1 لسٹ کی اقسام

جب ہم کسی ویب سائیٹ پر لسٹ استعمال کرنا چاہتے ہیں تو، HTML چار مختلف اقسام کی لسٹ فراہم کرتا unordered list ، ordered list ، definition list اور Nested list ہیں۔

ان آرڈرڈ لسٹ

یہ لسٹ متعلقہ اشیاء کا ایک مجموعہ ہے جس کی کوئی خاص ترتیب نہیں ہوتی۔ اس لسٹ میں شامل ہر شے کے سامنے ایک لگادیا گیا ہے۔ ٹیگز کو ایسی لسٹ بنانے کے لئے استعمال کیا جاتا ہے اور فہرست کی ہر شے کو ٹیگز کے ساتھ لیا جاتا ہے۔

آرڈرڈ لسٹ

اس لسٹ میں لسٹ کے تمام آئٹمز ایک نمبر سے شروع ہوتے ہیں اور نمبر بڑھتی ہوئی ترتیب میں ہوتے ہیں۔

 ٹیگز آرڈرڈ لسٹ بنانے کے لیے استعمال ہوتے ہیں اور فہرست کی ہر شے کو ٹیگز کے ساتھ گھیر لیا جاتا ہے۔

تعریفی لسٹ

HTML ایک لسٹ اسٹائل کی تائید کرتا ہے جس کو Definition list کہا جاتا ہے جہاں اندراجات کسی لسٹ یا انسائیکلوپیڈ کی طرح درج ہیں۔ Definition list ایک لغت، اصطلاحات کی لسٹ، یا دوسرے نام / قیمت کی لسٹ پیش کرنے کا ایک مثالی طریقہ ہے۔

یہ آئٹمز کی لسٹ نہیں ہے۔ یہ اصطلاحات کی وضاحت کے ساتھ لسٹ ہے۔ <dl> </dl>، <dt> </dt> اور <dd> </dd> ٹیگز کا استعمال کرکے definition لسٹ بنائی جاتی ہے۔

<dl> </dl>: یہ ٹیگ definition list بنانے کے لئے استعمال ہوتے ہیں۔

<dt> </dt>: یہ ٹیگ definition list کی ہر اصطلاح کی تعریف کے لئے استعمال ہوتے ہیں۔

<dd> </dd>: یہ ٹیگ تعریف کی فہرست کی ہر اصطلاح کی وضاحت کے لئے استعمال ہوتے ہیں۔

نیسٹڈ لسٹ

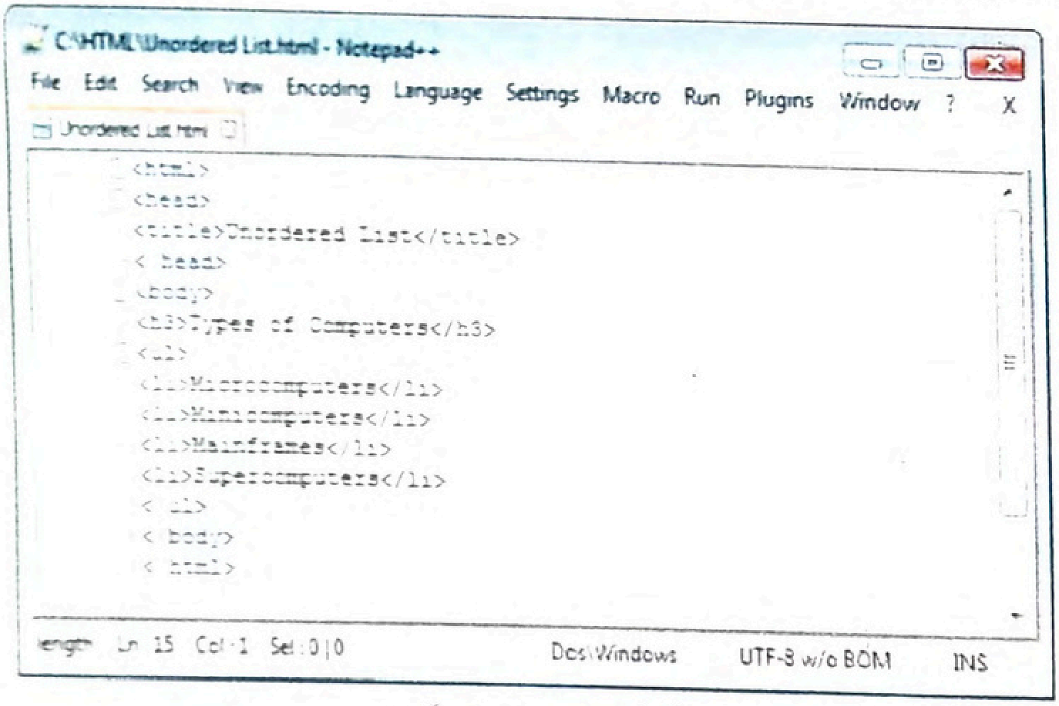
ایک لسٹ میں دوسری لسٹ کو بند کرنے کو نیسٹڈ (Nested) لسٹ کہا جاتا ہے۔ نیسٹڈ لسٹ بنانے کے لیے آرڈرڈ (Ordered) اور ان آرڈرڈ (Unordered) لسٹ کا کوئی بھی امتزاج بنایا جاسکتا ہے۔

NOT FOR SALE

7.4.2 فہرست بنانا

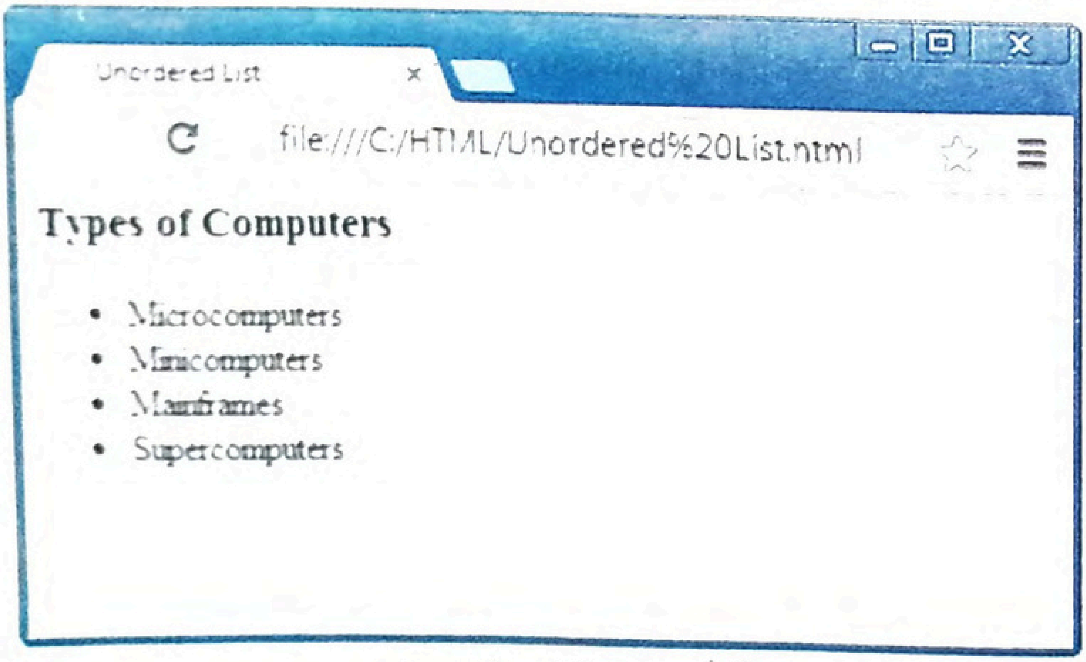
ان آرڈرڈ لسٹ

شکل نمبر 7.27 میں HTML ڈاکومنٹ ان آرڈرڈ لسٹ کی تشکیل کو واضح کرتی ہے۔



شکل نمبر 7.27 ان آرڈرڈ لسٹ کی تیاری

براؤزر ونڈو اس کو دکھائے گی جیسا کہ شکل نمبر 7.28 میں دکھایا گیا ہے۔

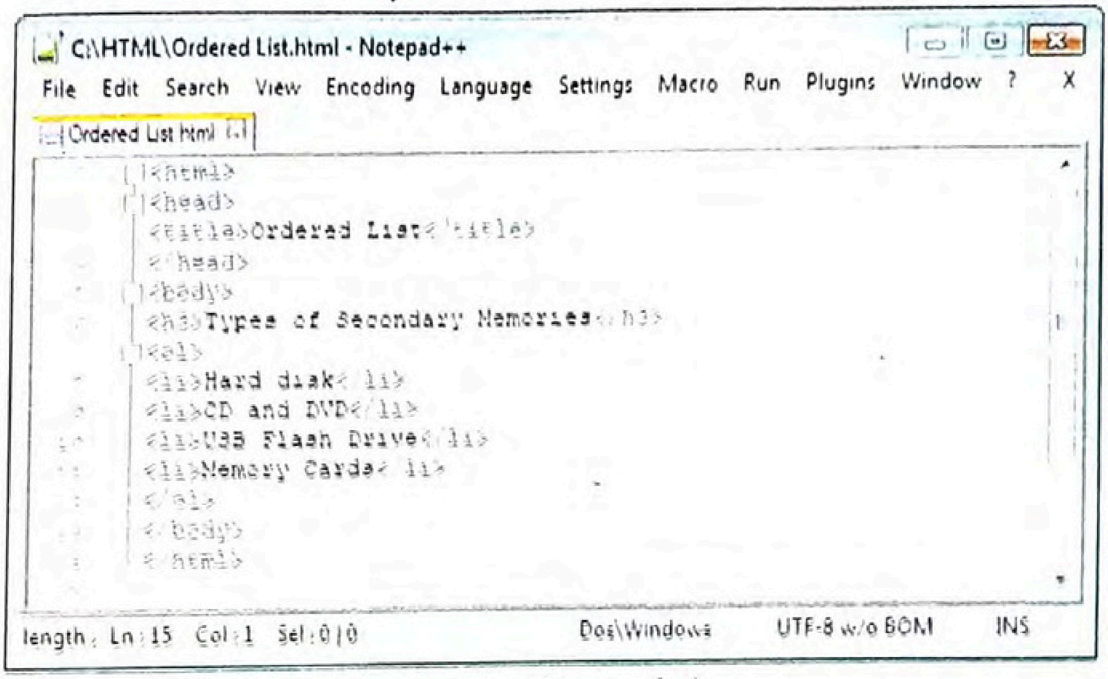


شکل نمبر 7.28 ویب پیج میں ان آرڈرڈ لسٹ کا ڈسپلے

NOT FOR SALE

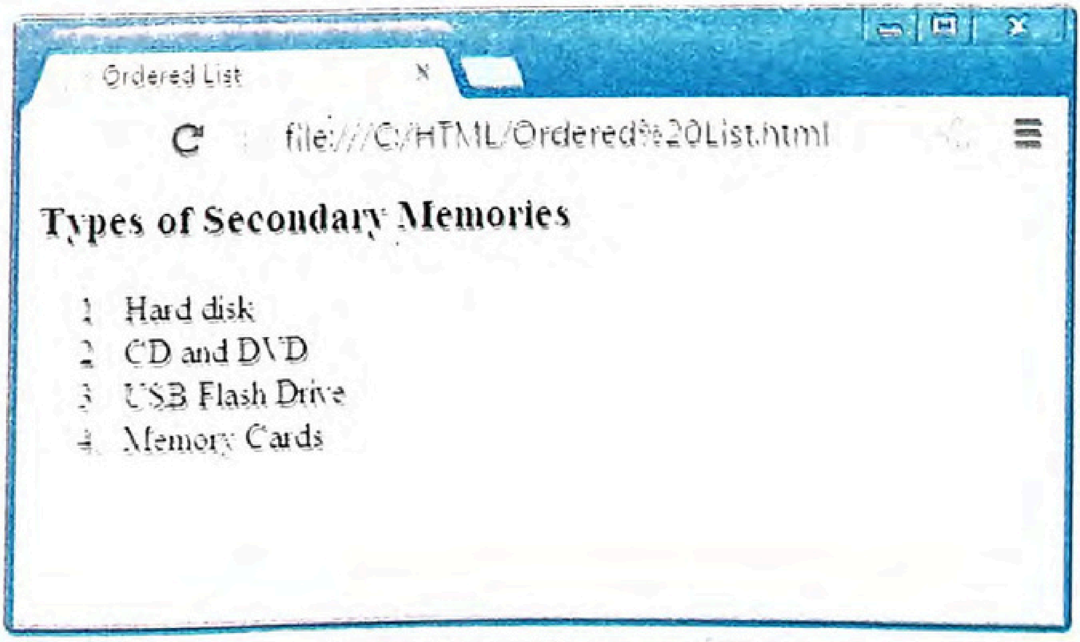
آرڈرڈ لسٹ بنانا

شکل نمبر 7.29 میں موجود HTML ڈاکومنٹ آرڈر شدہ لسٹ کی تشکیل کو ظاہر کرتی ہے۔



شکل نمبر 7.29 آرڈرڈ لسٹ بنانا

براؤزر ونڈو اس کو دکھائے گی جیسا کہ شکل نمبر 7.30 میں دکھایا گیا ہے۔



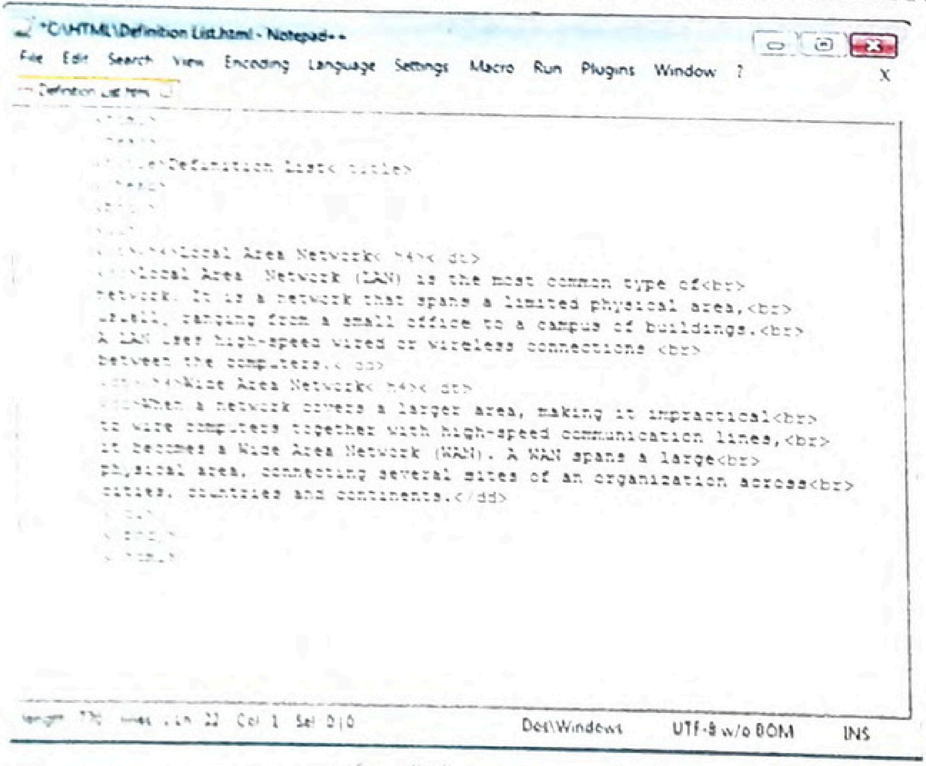
شکل نمبر 7.30 ویب پیج میں آرڈر شدہ لسٹ کا استعمال

تعریفی فہرست / لسٹ بنانا

شکل نمبر 7.31 میں HTML ڈاکومنٹ ایک definition لسٹ تیار کرتی ہے جو LAN اور WAN کے نیٹ ورک کی وضاحت کرتی ہے۔
اس HTML ڈاکومنٹ میں، پوری تعریف کی لسٹ <dl> </dl> ٹیگز کے اندر ہے۔

NOT FOR SALE

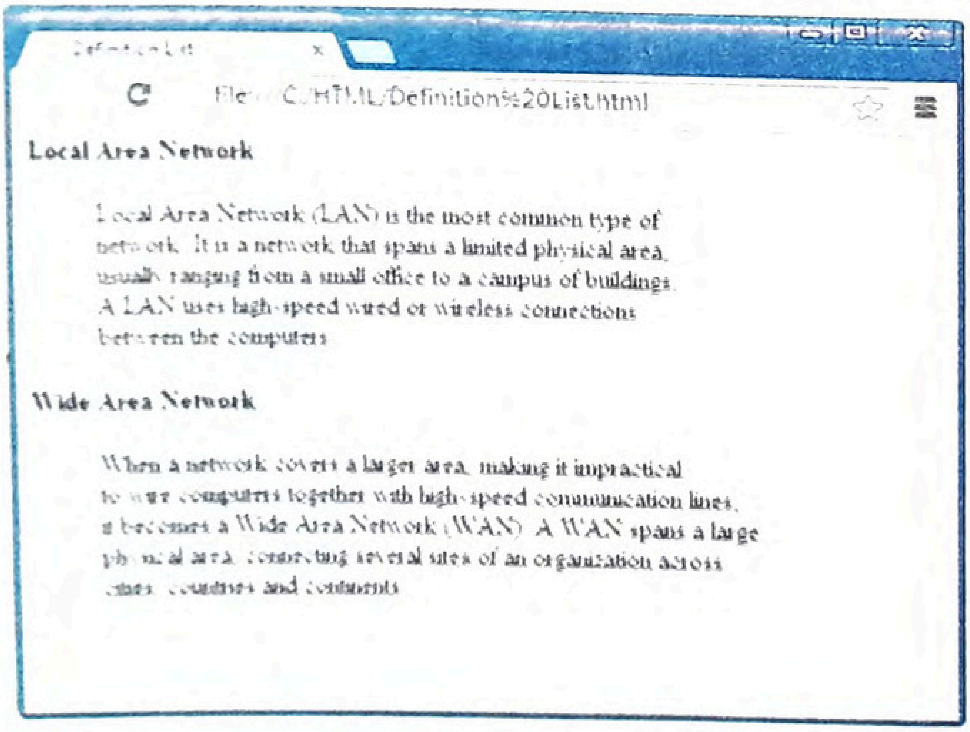
Local Area Network اور Wide Area Network <dt> </dt> ٹیگز کے اندر موجود ہے۔



شکل نمبر 7.31 اصطلاحات کی تعریف کی فہرست بنانا

اصطلاحات کی وضاحت پیراگراف میں <dd> </dd> ٹیگز کے اندر لکھی گئی ہے۔

براؤزر ونڈو اس کو دکھائے گی جیسا کہ شکل نمبر 7.32 میں دکھایا گیا ہے۔



شکل نمبر 7.32 ویب پیج میں تعریفی فہرست کا استعمال

NOT FOR SALE

نیسٹڈ لسٹ

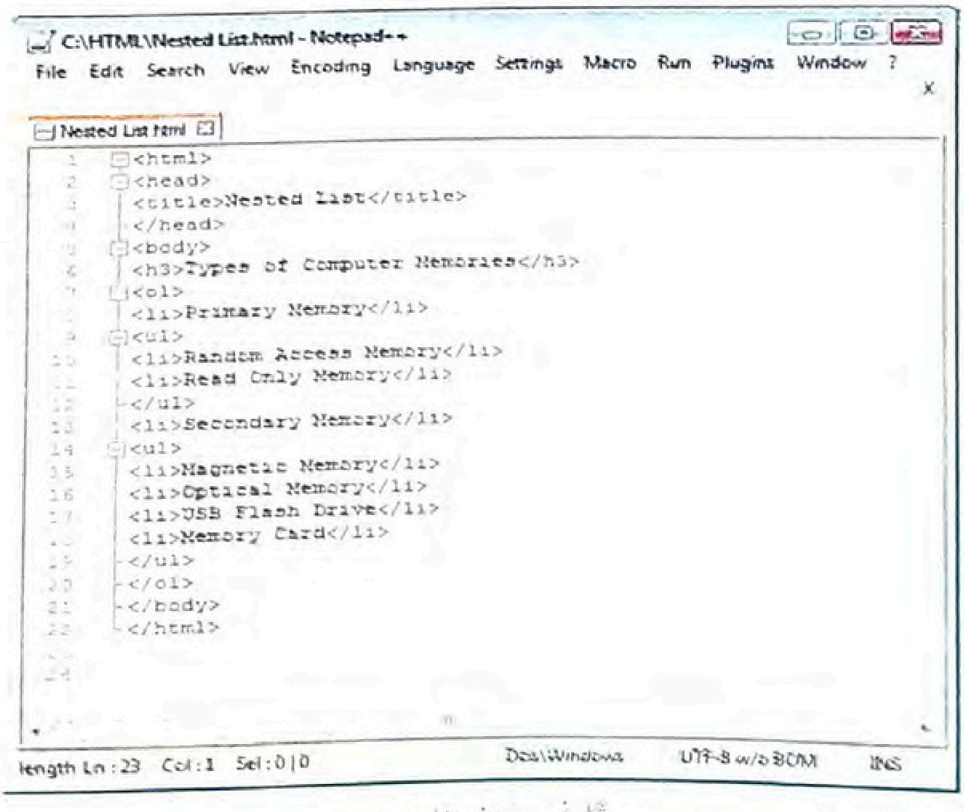
بعض اوقات، کسی ویب پیج میں کسی اور لسٹ کے اندر دوسری لسٹ کو نیسٹڈ کرنے کی ضرورت ہوتی ہے۔ اس سے پڑ ویب پیج میں پیش کردہ معلومات کو سمجھنا آسان ہو جاتا ہے۔ مندرجہ ذیل مثال پر غور کریں:

Computer Memories
nary Memory
dom Access Memory
d Only Memory
ondary Memory
ic Memory
Memory
ash Drive
y Card

اس مثال میں، ایک ان آرڈرڈ لسٹ کو آرڈر لسٹ میں بند کیا گیا ہے۔ شکل نمبر 7.33 میں دکھائے گئے HTML ڈاکومنٹ میں نیـ تیار کی گئی ہے۔

یہاں پر <h3> </h3> ہیڈنگ ٹیگ کو استعمال کر کے ٹائٹل Types of Computer Memories بنایا گیا ہے۔

آرڈرڈ لسٹ کے دو ذیلی عنوانات ہیں جو کہ Primary Memory اور Secondary Memory ہیں۔ اسی طرح پہلے ذیلی عنوا دو اجزاء ہیں جو ان آرڈرڈ ہیں اور دوسرے عنوان کے چار اجزاء ان آرڈرڈ ہیں۔

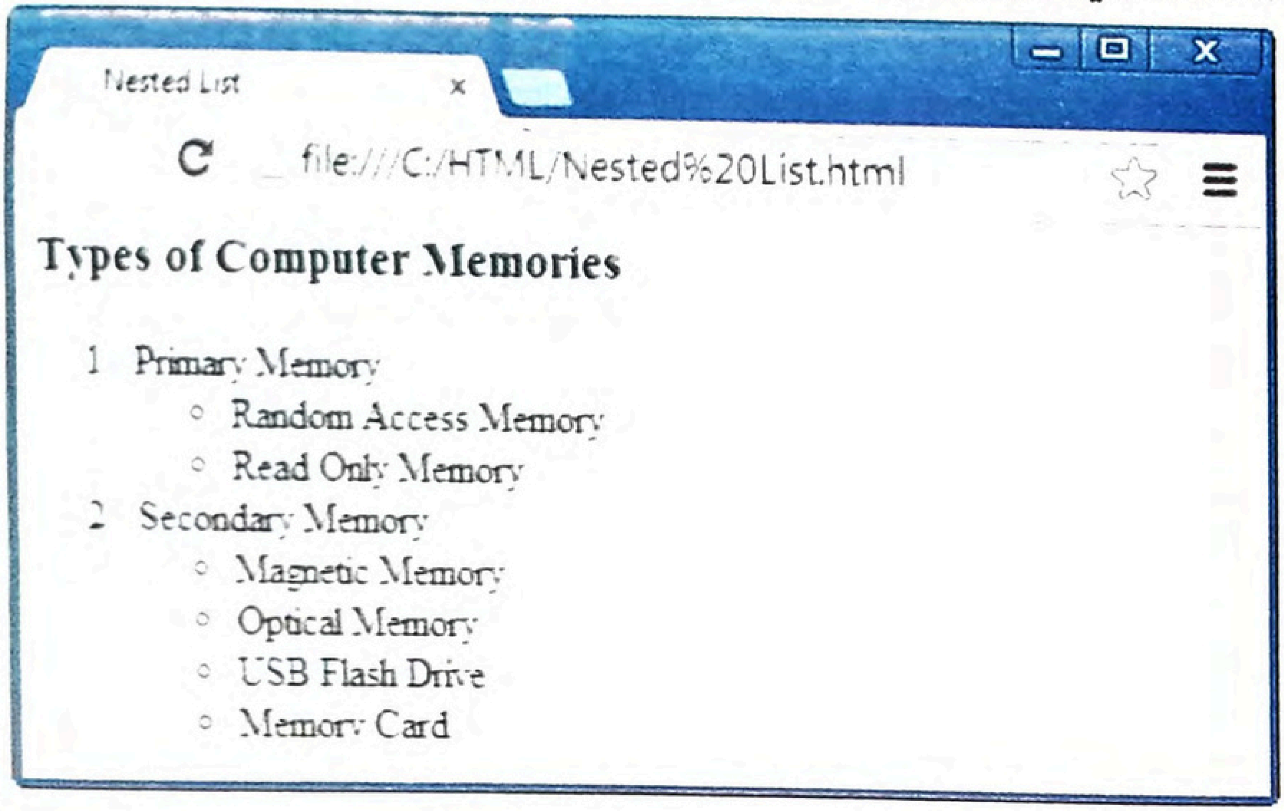


```
<html>
<head>
<title>Nested List</title>
</head>
<body>
<h3>Types of Computer Memories</h3>
<ol>
<li>Primary Memory</li>
<ul>
<li>Random Access Memory</li>
<li>Read Only Memory</li>
</ul>
<li>Secondary Memory</li>
<ul>
<li>Magnetic Memory</li>
<li>Optical Memory</li>
<li>USB Flash Drive</li>
<li>Memory Card</li>
</ul>
</ol>
</body>
</html>
```

شکل نمبر 7.33 نیسٹڈ لسٹ بنانا

NOT FOR SALE

براؤزر ونڈو میں یہ کچھ اس طرح نظر آئے گا جیسے شکل نمبر 7.34 میں دکھایا گیا ہے۔



شکل نمبر 7.34 براؤزر ونڈو میں ویب پیج پر نیسٹڈ لسٹ دکھائی گئی ہے

7.5 تصاویر اور پس منظر (Images and Background)

بہت سارے لوگ زیادہ ٹیکسٹ کو پڑھنے کے بجائے متعلقہ تصاویر والے ویب پیجز کو ترجیح دیتے ہیں جس میں کم وقت اور محنت درکار ہوتی ہے۔ تصاویر کسی ویب سائٹ کو دلآویز بنانے کے لئے اہم عنصر ہیں۔ تصویر کی ایک اہمیت ہے کیونکہ یہ دیکھنے والے کی توجہ حاصل کرلیتی ہے۔ تصویر پلک جھپکتے میں پیچیدہ پیغامات پہنچا سکتی ہے۔ لیکن جب اسے آپ کی ویب سائٹ کے کسی منظر میں شامل کیا جاتا ہے تو، تصاویر اس سے بھی زیادہ اہم کردار ادا کرتی ہیں۔ ویب پیج کے پس منظر کی تصویر آپ کے زائرین کی توجہ حاصل کرتی ہے۔

7.5.1 ویب پیج میں تصویر لگانا (Adding Image in a Web Page)

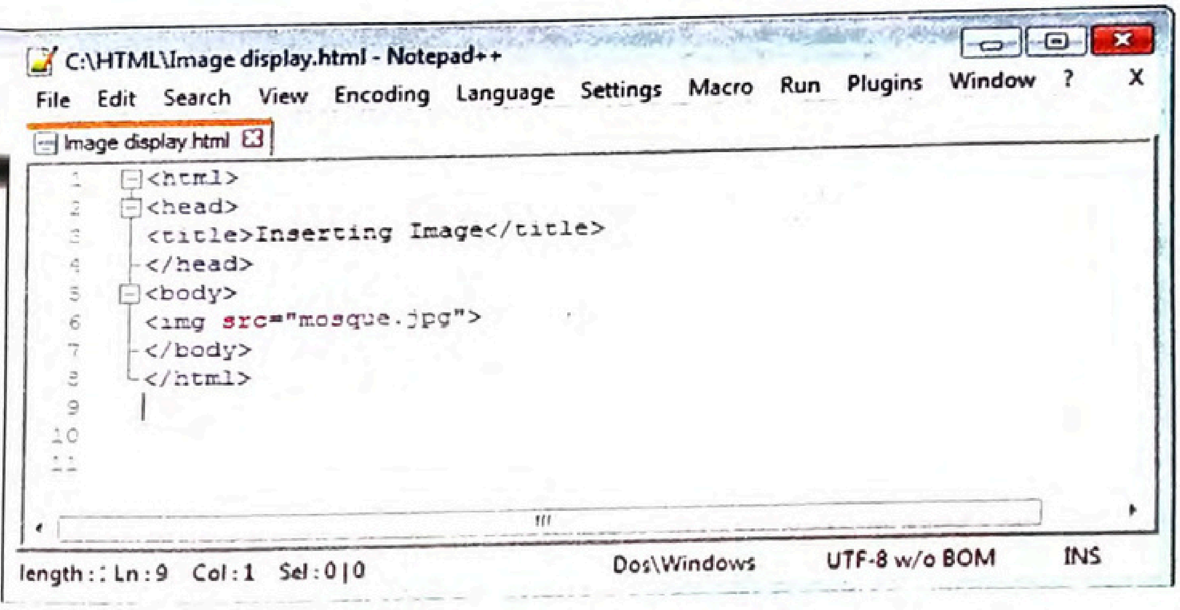
<img> ٹیگ، HTML ڈاکومنٹ میں تصویر شامل کرنے کے لیے استعمال ہوتا ہے۔ اس ٹیگ میں موجود src تصویر کے مقام کے لیے استعمال کیا جاتا ہے۔ تصویر HTML فائل کا حصہ نہیں ہے، یہ ایک علیٰحدہ فائل ہے جو کمپیوٹر کی میموری میں محفوظ ہے۔

<img> tag: یہ HTML ٹیگ ہے جو ویب پیج میں ایک تصویر شامل کرتا ہے۔ اس کا اختتامی ٹیگ نہیں ہوتا۔

src: اس کا مطلب ہے source۔ یہ براؤزر کو تصویر کی جگہ بتاتا ہے۔ <img> ٹیگ میں بہت سی خصوصیات ہیں لیکن src سب سے اہم ہے۔

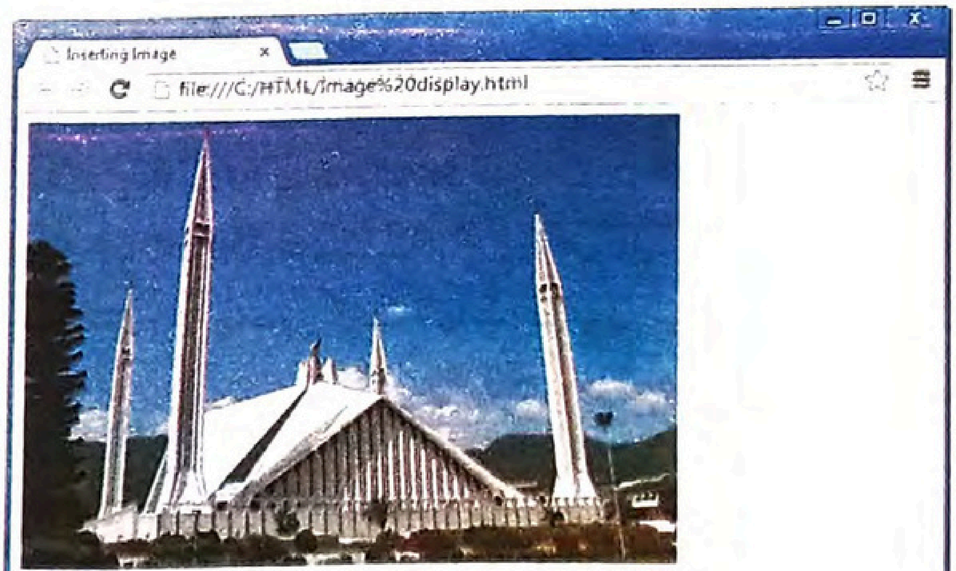
NOT FOR SALE

شکل نمبر 7.35 میں HTML ڈاکومنٹ یہ ظاہر کرتی ہے کہ کس طرح کسی پیج میں تصویر ڈسپلے کرنا ہے۔ فرض کریں کہ تصویر mosques.jpg اور HTML ڈاکومنٹ ایک ہی فولڈر میں ہیں۔



شکل نمبر 7.35 HTML ڈاکومنٹ میں تصویر کا استعمال

براؤزر ونڈو اس کو ایسے دکھائے گی جیسا کہ شکل نمبر 7.36 میں دکھایا گیا ہے۔



شکل نمبر 7.36 براؤزر ونڈو میں دکھائی گئی تصویر

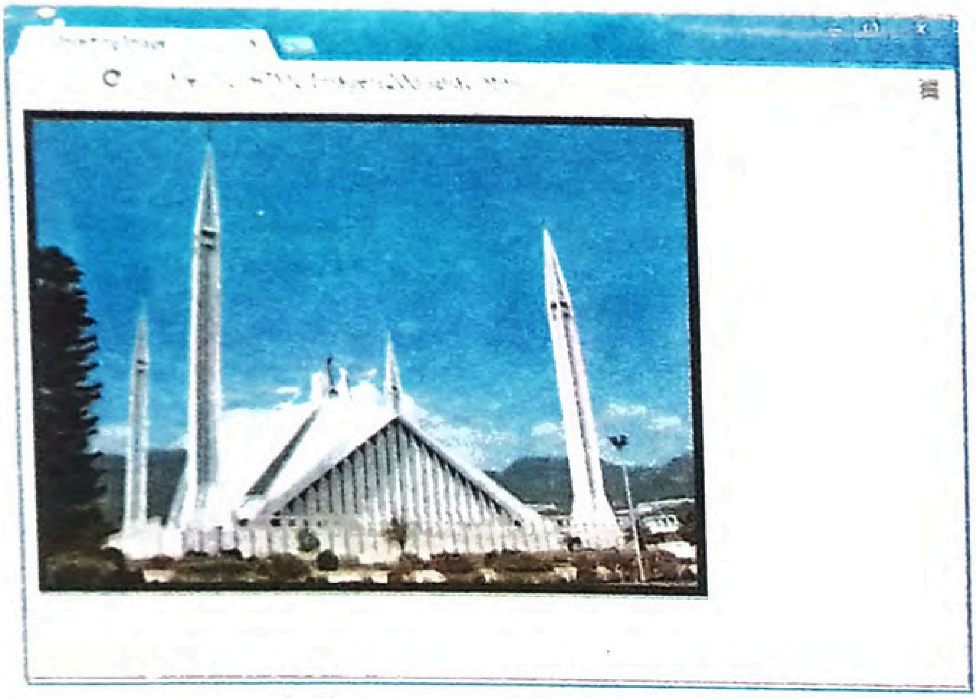
تصویر پر بارڈر لگانا

بارڈر لائنوں کا ایک باکس ہے جو کسی تصویر کے آس پاس ہوتا ہے۔ ذیل میں جیسا کہ <img> ٹیگ border کا استعمال کرتے

ہوئے ایک تصویر کے آس پاس ایک بارڈر مقرر کی جاسکتی ہے۔

```
<img src = "image.jpg" border = 5>
```

یہاں، نمبر 5 بارڈر کی چوڑائی پکسلز میں طے کرتا ہے۔ براؤزر ونڈو تصویر کو بارڈر کے ساتھ دکھائے گی جیسا کہ شکل نمبر 7.37 میں دکھایا گیا ہے۔



شکل نمبر 7.37 براؤزر ونڈو میں بارڈر کے اندر دکھائی گئی تصویر

7.5.2 ویب پیج میں تصویر کا سائز مقرر کرنا

یوزر پیج میں موجود تصویر میں تبدیلی لانے کے لئے <img> ٹیگ میں hight، width اور alt کا استعمال کر سکتا ہے۔

height: تصویر کی اونچائی کو پکسلز میں واضح کرنے کے لئے <img> ٹیگ میں لکھا جاتا ہے۔ یہ تصویر کی اصل اونچائی سے مختلف ہو سکتا ہے۔ براؤزر <img> ٹیگ کے مطابق تصویر کی height کا سائز تبدیل کرتا ہے۔

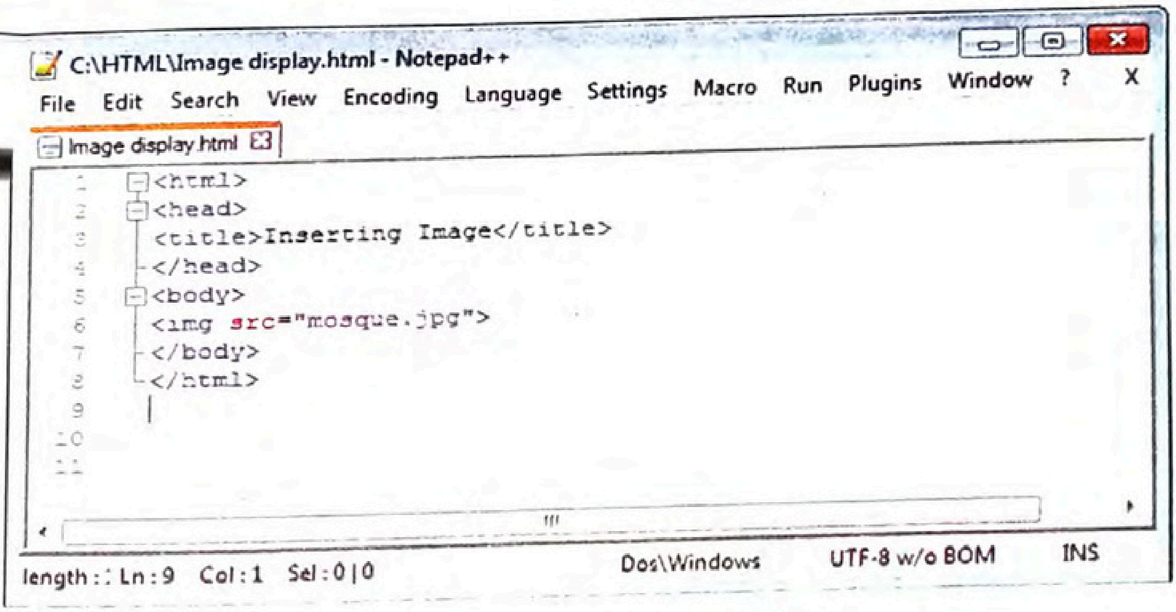
width: تصویر کی چوڑائی کو پکسلز میں واضح کرنے کے لئے <img> ٹیگ میں لکھا جاتا ہے۔ یہ تصویر کی اصل چوڑائی سے مختلف ہو سکتا ہے۔ براؤزر <img> ٹیگ کے مطابق تصویر کی چوڑائی کا سائز تبدیل کرتا ہے۔

alt: alt ایک ٹیکسٹ ہوتا ہے جو کسی تصویر کی وضاحت کے لیے استعمال کیا جاتا ہے۔ alt براہ راست <img> ٹیگ میں لکھا جاتا ہے۔ اگر کسی وجہ سے کسی تصویر کو ظاہر نہیں کیا جاسکتا تو، alt تصویر کے بجائے متبادل ٹیکسٹ دکھاتا ہے۔ یہ یوزر کے لیے اور خاص طور پر بینائی سے محروم یوزر کے لیے متبادل ٹیکسٹ مہیا کرنے کے لئے مفید ہے۔

شکل نمبر 7.38 میں موجود HTML ڈاکومنٹ، ویب پیج میں کسی تصویر کا سائز تبدیل کرنے کے لئے <img> ٹیگ میں height، width اور alt کے استعمال کو ظاہر کرتی ہے۔

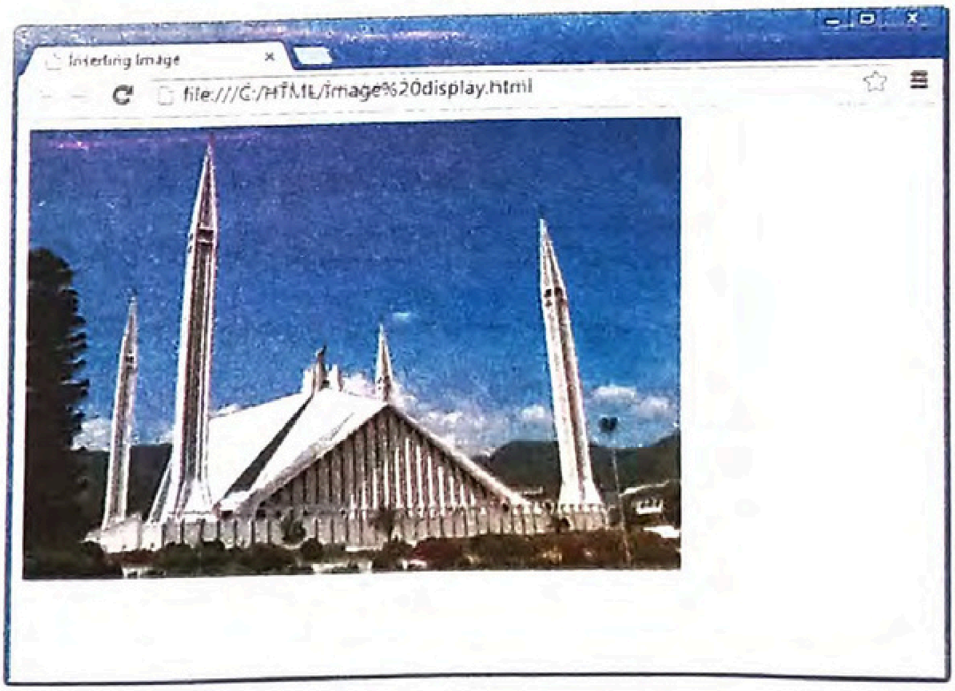
NOT FOR SALE

شکل نمبر 7.35 میں HTML ڈاکومنٹ یہ ظاہر کرتی ہے کہ کس طرح کسی پیج میں تصویر ڈسپلے کرنا ہے۔ فرض کریں کہ تصویر mosques.jpg اور HTML ڈاکومنٹ ایک ہی فولڈر میں ہیں۔



شکل نمبر 7.35 HTML ڈاکومنٹ میں تصویر کا استعمال

براؤزر ونڈو اس کو ایسے دکھائے گی جیسا کہ شکل نمبر 7.36 میں دکھایا گیا ہے۔



شکل نمبر 7.36 براؤزر ونڈو میں دکھائی گئی تصویر

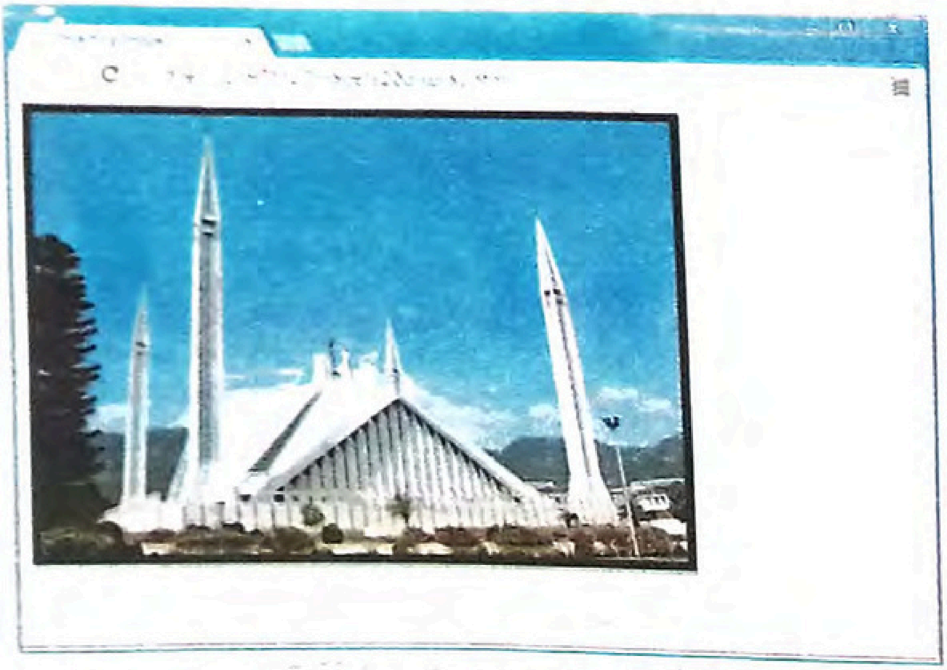
تصویر پر بارڈر لگانا

بارڈر لائنوں کا ایک باکس ہے جو کسی تصویر کے آس پاس ہوتا ہے۔ ذیل میں جیسا کہ <img> ٹیگ border کا استعمال کرتے

ہوئے ایک تصویر کے آس پاس ایک بارڈر مقرر کی جاسکتی ہے۔

```
<img src = "image.jpg" border = 5>
```

یہاں، نمبر 5 بارڈر کی چوڑائی پکسلز میں طے کرتا ہے۔ براؤزر ونڈو تصویر کو بارڈر کے ساتھ دکھائے گی جیسا کہ شکل نمبر 7.37 میں دکھایا گیا ہے۔



شکل نمبر 7.37 براؤزر ونڈو میں بارڈر کے ساتھ دکھائی گئی تصویر

7.5.2 ویب پیج میں تصویر کا سائز مقرر کرنا

یوزر پیج میں موجود تصویر میں تبدیلی لانے کے لئے <img> ٹیگ میں hight، width اور alt کا استعمال کر سکتا ہے۔

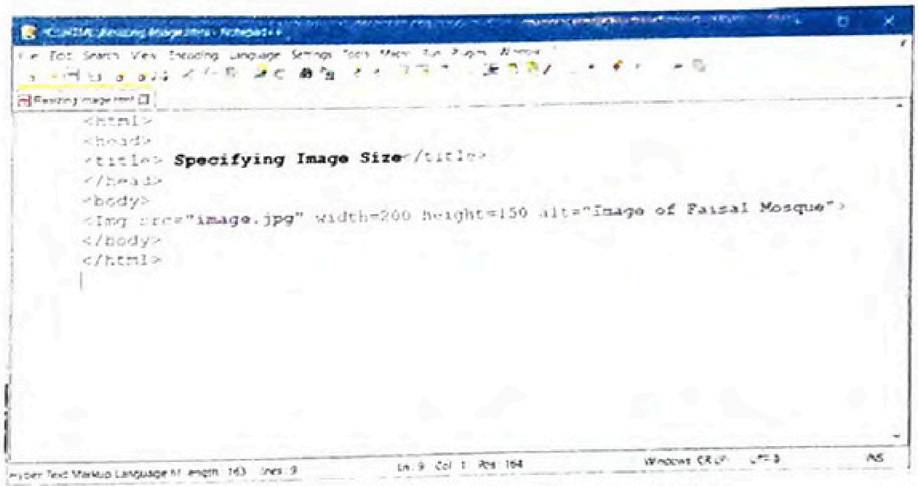
height: تصویر کی اونچائی کو پکسلز میں واضح کرنے کے لئے <img> ٹیگ میں لکھا جاتا ہے۔ یہ تصویر کی اصل اونچائی سے مختلف ہو سکتا ہے۔ براؤزر <img> ٹیگ کے مطابق تصویر کی height کا سائز تبدیل کرتا ہے۔

width: تصویر کی چوڑائی کو پکسلز میں واضح کرنے کے لئے <img> ٹیگ میں لکھا جاتا ہے۔ یہ تصویر کی اصل چوڑائی سے مختلف ہو سکتا ہے۔ براؤزر <img> ٹیگ کے مطابق تصویر کی چوڑائی کا سائز تبدیل کرتا ہے۔

alt: alt ایک ٹیکسٹ ہوتا ہے جو کسی تصویر کی وضاحت کے لیے اسے <im ٹیگ میں لکھا جاتا ہے۔ اگر کسی وجہ سے کسی تصویر کو ظاہر نہیں کیا جا سکے اور خاص طور پر بینائی سے محروم یوزر کے لیے متبادل ٹیکسٹ ہیں

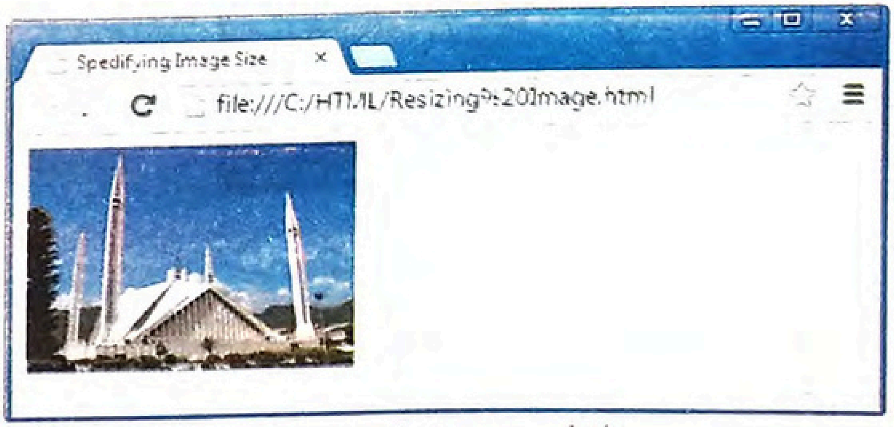
شکل نمبر 7.38 میں موجود HTML width، h اور alt کے استعمال کو ظاہر

FOR SALE



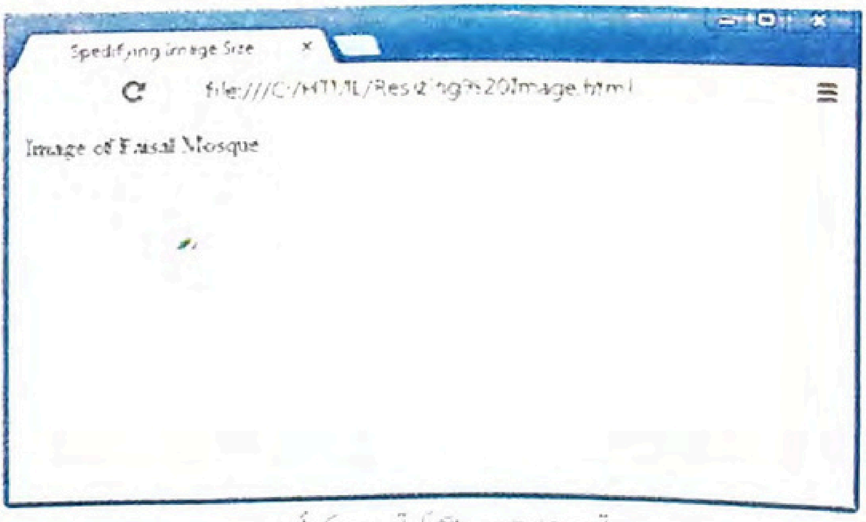
شکل نمبر 7.38 HTML ڈاکومنٹ میں تصویر کے سائز کو بدلنا

براؤزر ونڈو اس کو دکھائے گی جیسا کہ شکل نمبر 7.39 میں دکھایا گیا ہے۔



شکل نمبر 7.39 ویب پیج پر تصویر شدہ سائز

اگر براؤزر کچھ وجوہات کی بناء پر تصویر کو ظاہر نہیں کر سکتا ہے تو، ویب پیج پر ٹیکسٹ ڈسپلے کیا جائے گا جیسا کہ شکل نمبر 7.40 میں دکھایا گیا ہے۔

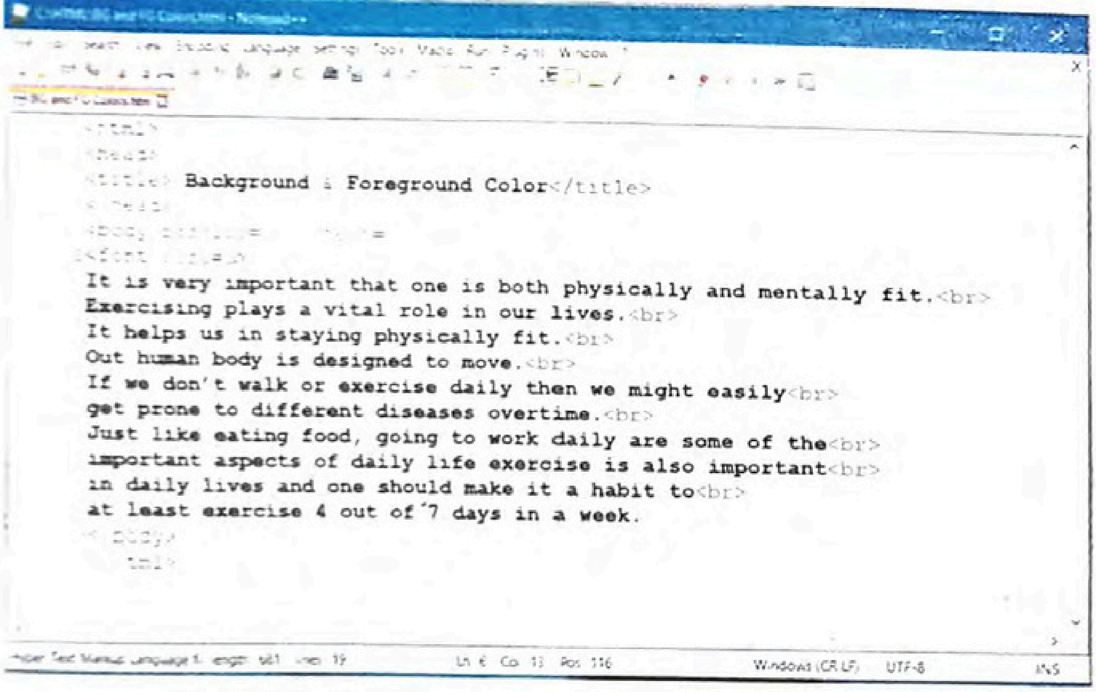


تصویر 7.40 ویب پیج اگر تصویر نہ دکھائی دے

7.5.3 پس منظر اور پیش منظر کا رنگ بدلنا

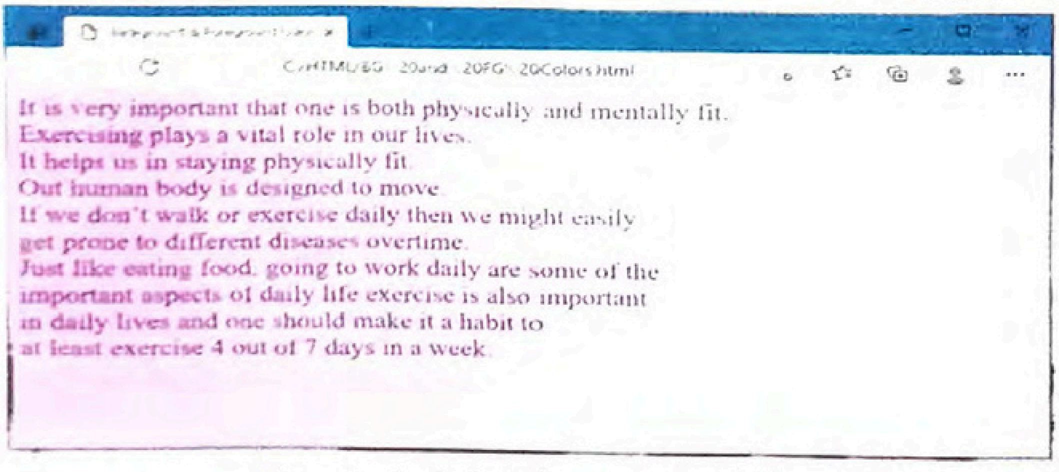
ویب پیج کو اچھی شکل دینے اور دلکش بنانے کے لئے رنگ بہت ضروری ہیں۔ آپ <body> ٹیگ کا استعمال کر کے پیج پر رنگ دکھا سکتے ہیں یا آپ bgcolor کا استعمال کرتے ہوئے انفرادی ٹیگ کیلئے رنگ متعین کر سکتے ہیں۔

کسی پیج میں پس منظر کا رنگ لگانے کے لئے، bgcolor اور پیش منظر کے رنگ کے لئے، <body> ٹیگ میں ٹیکسٹ شامل کریں جیسا کہ شکل نمبر 7.41 میں دکھایا گیا ہے۔



شکل نمبر 7.41 HTML میں پس منظر اور پیش منظر کے رنگوں کا استعمال

اس HTML ڈاکومنٹ میں، پس منظر کا رنگ pink اور پیش منظر کا رنگ blue ہے۔ براؤزر ونڈو اس کو دکھائے گی جیسا کہ شکل نمبر 7.42 میں دکھایا گیا ہے۔



شکل نمبر 7.42 پس منظر اور پیش منظر کے رنگ کے ساتھ ویب پیج

NOT FOR SALE

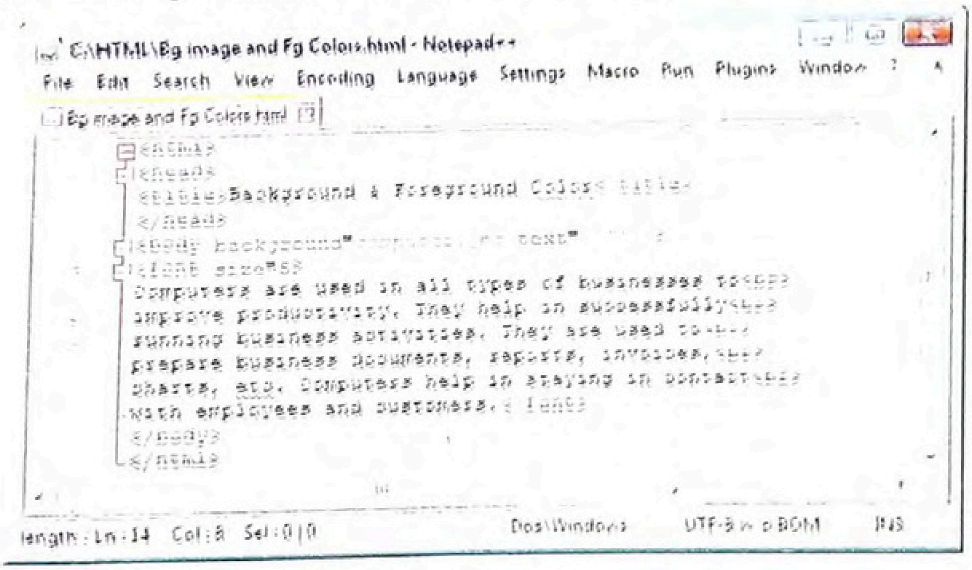
7.5.4 پس منظر میں تصویر لگانا

پس منظر کی تصاویر ویب پیجز میں ایک اہم کردار ادا کرتی ہیں۔ <body> ٹیگ میں background کا استعمال کرتے ہوئے کسی تصویر کو کسی پیج کے پس منظر کے طور پر ترتیب دیا جا سکتا ہے۔

<body background="image.jpg">

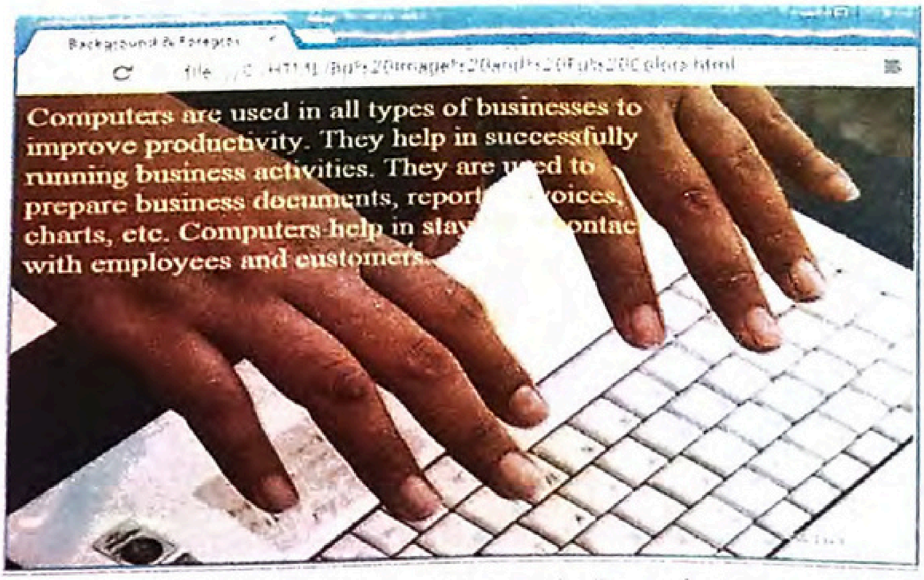
اگر تصویر براؤزر ونڈو سے چھوٹی ہے تو پھر جب تک پوری ونڈو بھر نہیں جاتی ہے تب تک وہ خود ہی اپنے آپ کو دہرائے گی۔

شکل نمبر 7.43 میں موجود HTML ڈاکومنٹ ویب پیج کے پس منظر کے طور پر تصویری فائل computer.jpg کے استعمال کو ظاہر کرتی ہے۔



شکل نمبر 7.43 HTML ڈاکومنٹ کے پس منظر کے لیے تصویر کا استعمال

براؤزر اس کو دکھائے گا جیسا کہ شکل نمبر 7.44 میں دکھایا گیا ہے۔



شکل نمبر 7.44 تصویر کو ویب پیج کے پس منظر کے طور پر استعمال کرنا

NOT FOR SALE

7.6 ہائپر لنک (Hyperlink)

ورلڈ وائڈ ویب ہائپر لنکس پر مشتمل ہے جس میں کھربوں پیجز اور فائلیں ایک دوسرے سے جڑی ہوئی ہیں۔ ویب پیج میں بہت سے ہائپر لنکس ہوتے ہیں، یہ لنک ہر یوزر کو متعلقہ ویب پیج یا تصویر / فائل پر بھیجتا ہے۔

7.6.1 ہائپر لنک کیا ہے؟

ہائپر لنکس تقریباً تمام ویب پیجز میں پائے جاتے ہیں، جس سے یوزر کو ایک پیج سے دوسرے پیج تک پہنچنے میں مدد ملتی ہے۔ اپنے راستے پر کلک کرنے کی اجازت ملتی ہے۔ ٹیکسٹ ہائپر لنکس اکثر نیلے اور سرخ رنگ کے ہوتے ہیں۔ جب یوزر کرسر کو ہائپر لنک کے اوپر منتقل کرتا ہے، چاہے وہ ٹیکسٹ ہو یا ایک تصویر، کرسر چھوٹے ہاتھ میں بدل جاتا ہے۔ اس پر کلک کرنے سے یوزر نئے پیج پر منتقل ہو جاتا ہے۔

7.6.2 اینکر ٹیگ (Anchor Tag)

<a> ٹیگ کو اینکر کے نام سے جانا جاتا ہے۔ یہ ایک ہائپر لنک بنانے کے لئے استعمال کیا جاتا ہے جو href کے ساتھ، ٹیکسٹ یا تصویر ہو سکتا ہے۔

7.6.3 ویب پیج کی طرف ہائپر لنک بنانا

کسی اور ویب پیج کی طرف ہائپر لنک بنانے کا طریقہ یہ ہے:

```
<a href= "URL">text to be displayed</a>
```

مثال کے طور پر:

```
<a href= "http://www.bisep.edu.pk">Peshawar Board</a>
```

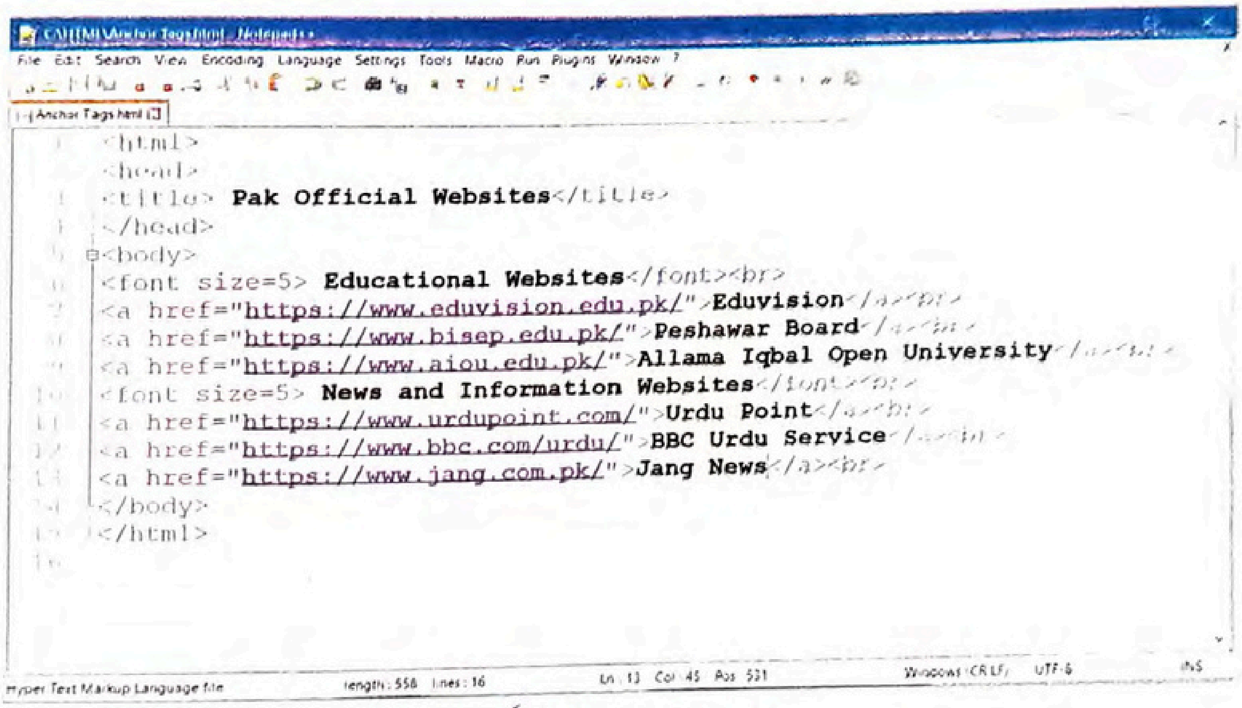
اسے براؤزر ونڈو میں Peshawar Board کے طور پر ظاہر کیا جائے گا اور جب قاری اس پر کلک کرے گا تو، یہ پشاور بورڈ کی ویب سائٹ کھولے گا جس کا URL www.bisep.edu.pk ہے۔

ہائپر لنکس بنانے کی کچھ اور مثالیں ذیل میں دی گئی ہیں۔

```
<a href= "http://www.urdupoint.com">Urdu Point</a>
<a href= "http://www.bbc.com/urdu">BBC Urdu Service</a>
<a href="http://www.aiou.edu.pk">Allama Iqbal Open University</a>
```

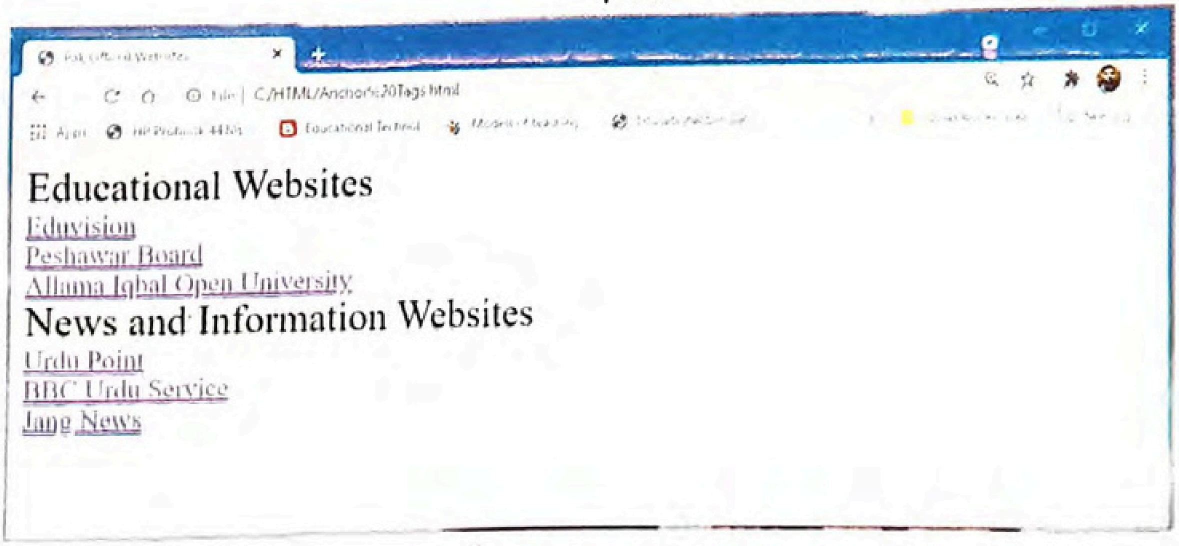
شکل نمبر 7.45 میں HTML ڈاکومنٹ کسی ویب پیج میں ہائپر لنکس کے استعمال کو ظاہر کرتی ہے۔

NOT FOR SALE



HTML 7.45ڈاکومنٹ میں ہائپر لنکس بنانا

براؤزر ونڈو اس کو دکھائے گی جیسا کہ شکل نمبر 7.46 میں دکھایا گیا ہے۔



شکل نمبر 7.46 ویب پیج میں ہائپر لنکس

7.6.4 ویب پیج کے اندر ہائپر لنک بنانا

<a> میں name کا استعمال

اینکر کو اپنے ہی ایٹریبیوٹ name کے ساتھ استعمال کر کے ویب پیج کے اندر ہی ہائپر لنک بنایا جا سکتا ہے۔ یہ دو مرحلوں میں ہو گا۔ پہلے اینکر بنائیں اور پھر اینکر کو ہائپر لنک بنائیں۔

NOT FOR SALE

اینکر بنانا

اینکر ٹیگ ایک HTML ٹیگ ہے اور اسے ٹیکسٹ کے اندر دکھائی دینے والے الفاظ کی وضاحت کرنے کے لئے استعمال کیا جاتا ہے جس پر کلک کیا جاسکتا ہے۔ یہ الفاظ (ہائپر لنکس) عام طور پر ویب پر مزید معلومات کا حوالہ دیتے ہیں۔

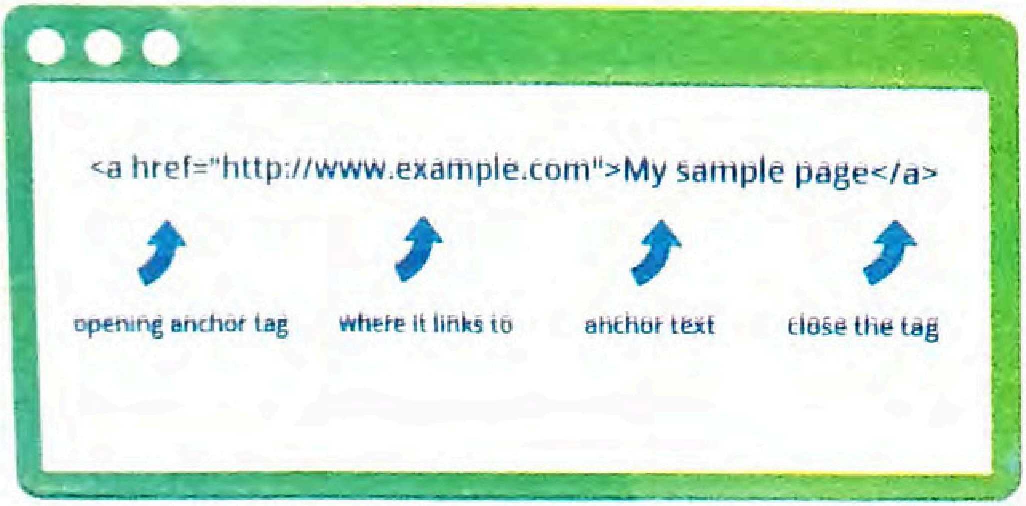
جس پیج پر آپ ہائپر لنک کرنا چاہتے ہو اس جگہ پر اینکر ٹیگ میں لیبل بنائیں۔

لیبل بنانے کے لئے طریقہ یہ ہے:

اینکر کی طرف ہائپر لنک بنانا

ایک ہائپر لنک بنانے کے لیے، ڈاکومنٹ کی منزل (address) معلوم ہونا چاہئے۔ ایک ہائپر لنک آپ کے ویب پیج کو دوسرے ویب پیجز، یا کسی فائل سے منسلک کر سکتا ہے۔ href کے ساتھ عام <a> ٹیگ کا استعمال کرکے یوزر اینکر کو ہائپر لنک دے سکتا ہے۔ href کا مطلب ہے ہائپر ٹیکسٹ ریفرنس۔ یہ href پیج یا ڈاکومنٹ کا ہدف ایڈریس بیان کرتا ہے جیسا کہ شکل نمبر 7.47 میں دکھایا گیا ہے۔ اس کا طریقہ یہ ہے:

text to be displayed



شکل نمبر 7.47 اینکر پر ہائپر لنکس بنانا

علامت کو لیبل کے سامنے رکھنا ضروری ہے۔ مثال کے طور پر، Unit 7 سے پہلے درج ذیل اینکر بنائیں اور یہ وہ جگہ # ہے جہاں آپ ہائپر لنک کرنا چاہتے ہیں۔

اب، نیچے دیئے گئے کسی بھی مقام پر اینکر پر ہائپر لنک بنائیں۔

Unit 7 میں جانے کے لئے درج ذیل لنک پر کلک کریں۔

here

NOT FOR SALE

C:\HTML\Anchor Tags within a doc.html - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

Anchor Tags within a doc.html

```
<html>
<head>
<title>Using Anchors</title>
</head>
<body>
<font size=4>Types of Computers</font><br><br>
<a href=#Microcomputer>1. Microcomputer</a><br>
<a href=#Minicomputer>2. Minicomputer</a><br>
<a href=#Mainframe>3. Mainframe</a><br><br>
1.<a name=Microcomputer></a><b>Microcomputer</b><br>
Microcomputers are the smallest and the cheapest computer.<br>
These computers are the ones, most commonly used in homes<br>
and offices. Microcomputer was introduced in 1970s when<br>
microprocessor was invented. A microprocessor is a single<br>
IC chip that controls the operation of the computer.<br><br>
2.<a name=Minicomputer></a><b>Minicomputer</b><br>
Minicomputers were introduced in the 1960s when IC chips were<br>
introduced. A minicomputer is bigger than a microcomputer but<br>
smaller than a mainframe. These computers can execute billions<br>
of instructions per second. Therefore, they can process more<br><br>
3.<a name=Mainframe></a><b>Mainframe</b><br>
Mainframe is a very large, very powerful and expensive<br>
computers that can support hundreds and thousands of users<br>
at the same time. Mainframes are used in large organizations<br>
such as banks, airlines, big universities, research centers,<br>
and government departments.<br>
</body>
</html>
```

length : 1298 | Ln : 28 Col : 8 Sel : 0 | 0 Dos\Windows UTF-8 w/o BOM INS

شکل نمبر 7.48 HTML ڈاکومنٹ میں ویب پیج کے اندر ہائپر لنکس کا استعمال

براؤزر ونڈو اسے دکھادے گی۔

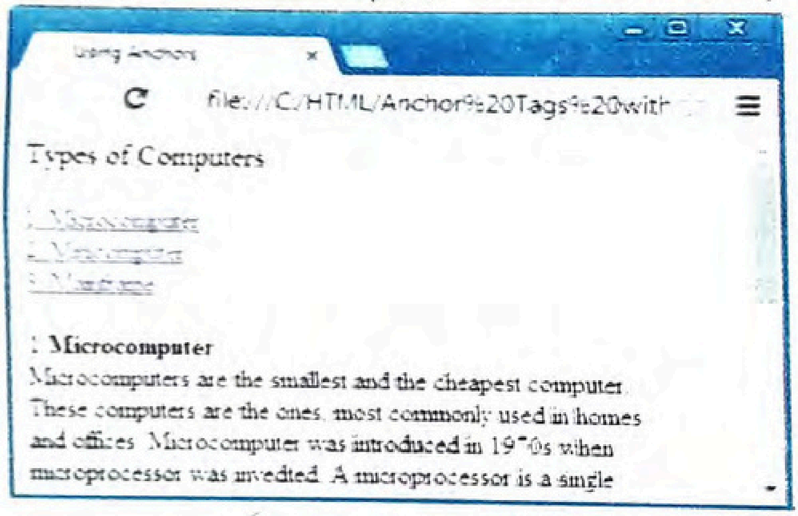
Click here to go to Unit 7.

جب یوزر here لفظ پر کلک کرتا ہے، تو وہ اسے اس مقام پر لے جائے گا جہاں ویب پیج میں Unit 7 شروع ہوتا ہے۔

شکل نمبر 7.48 میں موجود HTML ڈاکومنٹ یہ ظاہر کرتا ہے کہ ویب پیج میں ہائپر لنک کیسے بنائے جائیں۔ اس HTML ڈاکومنٹ میں تین ہائپر لنک ہیں۔ جب یوزر دوسرے ہائپر لنک پر کلک کرتا ہے، تو وہ اسے دوسرے عنوان پر لے جائے گا جو Mini Computer کے بارے میں ہے۔ جب یوزر تیسرے ہائپر لنک پر کلک کرتا ہے، تو وہ اسے تیسری قسم کے کمپیوٹر پر لے جائے گا جو Mainframe ہے۔ اگر یوزر پہلا ہائپر لنک کلک کرتا ہے تو اسکرین پر Microcomputer کے بارے میں پہلا عنوان پہلے ہی ظاہر ہونے کی وجہ سے کچھ نہیں ہوگا۔

NOT FOR SALE

براؤزر ونڈو ویب پیج دکھائے گی جیسا کہ شکل نمبر 7.49 میں دکھایا گیا ہے۔



شکل نمبر 7.49 ایک ویب پیج میں ہائپر لنکس

7.6.5 گرافیکل ہائپر لنک تیار کرنا

اپنے ویب پیج کو مؤثر بنانے کا ایک بہترین طریقہ یہ ہے کہ کچھ گرافکس شامل کریں۔ گرافکس کسی ویب پیج پر رنگ، وسعت اور چمک شامل کرسکتے ہیں اور دیکھنے والوں کی توجہ اور دلچسپی برقرار رکھنے میں مدد کرسکتے ہیں۔

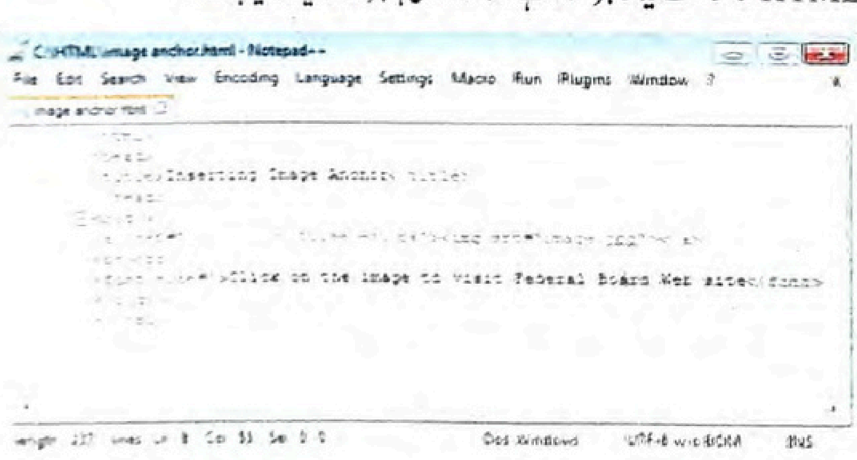
کسی دوسرے ویب پیج پر گرافیکل ہائپر لنک بنانے کے لئے ترکیب یہ ہے:

<img scr=”name of image file”>

مثال کے طور پر: **<img src="image.jpg">**

یہ براؤزر ونڈو میں image.jpg تصویری فائل دکھائے گا۔ جب یوزر ماؤس پوائنٹر کو تصویر پر منتقل کرے گا تو، یہ ایک چھوٹے سے ہاتھ میں بدل جائے گا اور اس تصویر پر کلک کرنے سے فیڈرل بورڈ کی ویب سائٹ کھل جائے گی جس کا URL www.fbise.edu.pk ہے۔

شکل نمبر 7.50 میں HTML ڈاکومنٹ یہ ظاہر کرتا ہے کہ گرافیکل ہائپر لنک کیسے بنایا جائے۔

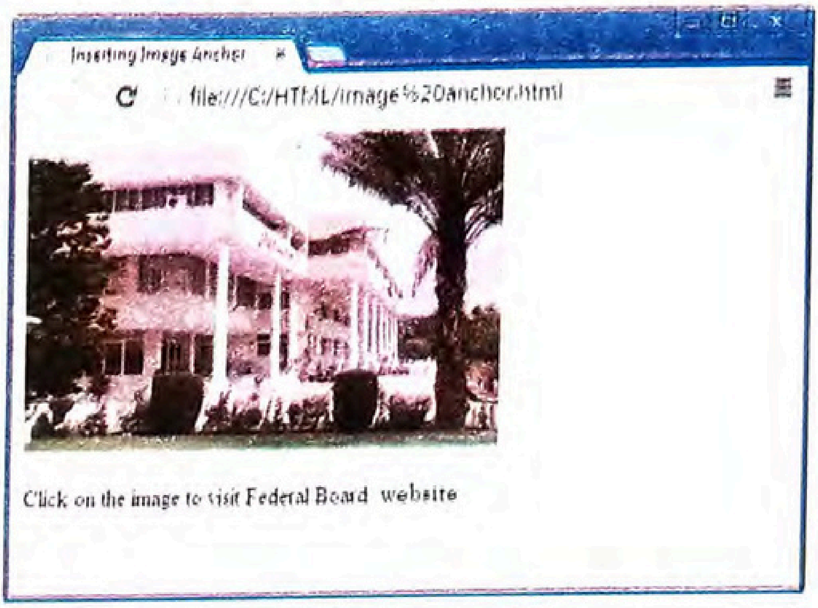


شکل نمبر 7.50 HTML ڈاکومنٹ میں گرافیکل ہائپر لنک کا استعمال

NOT FOR SALE

NOT FOR SALE

براؤزر ونڈو اس کو دکھائے گی جیسا کہ شکل نمبر 7.51 میں دکھایا گیا ہے۔



شکل نمبر 7.51 ویب پیج میں گرافیکل لنک

7.7 ٹیبل / جدول بنانا (Creating Tables)

ٹیبلز کا استعمال اکثر ویب پیجز میں منظم انداز میں معلومات پیش کرنے کے لئے کیا جاتا ہے جسے یوزر کے لیے پڑھنا اور سمجھنا آسان ہوتا ہے۔ ایک ٹیبل قطاروں (Rows) اور کالموں (Columns) پر مشتمل ہوتا ہے۔ ایک ٹیبل آپ کو تیزی سے اور آسانی سے ایسی معلومات تلاش کرنے کی اجازت دیتا ہے جو مختلف قسم کے ڈیٹا کے درمیان کسی نہ کسی طرح کے تعلق کی نشاندہی کرتی ہیں، مثال کے طور پر، افراد کے نام اور ان کی عمر، یا ہفتے کا ایک دن، یا کلاسوں کا ٹائم ٹیبل۔

HTML ٹیبل یوزر کو ڈیٹا یعنی ٹیکسٹ، تصاویر، لنکس، فارم، فارم فیلڈز، دیگر ٹیبلز وغیرہ کو سیلوں اور قطاروں میں ترتیب دینے کی سہولت دیتا ہے۔

ہر ٹیبل کا کوئی نام ہو سکتا ہے جو ٹیبل کے مقصد کی مختصر وضاحت فراہم کرتا ہے۔ ٹیبل کے متعلق تفصیلی معلومات بھی فراہم کی جا سکتی ہیں جو کہ بریلے استعمال کرنے والوں کے لیے آسانی فراہم کرتی ہیں۔

ٹیبل بنانے کے لئے درج ذیل ٹیگز استعمال ہوتے ہیں۔

<table></table> : یہ ٹیگ ٹیبل بنانے کیلئے استعمال ہوتے ہیں۔

<tr></tr> : یہ ٹیگ قطار بنانے کے لیے استعمال ہوتے ہیں۔

<td></td> : یہ ٹیگ ڈیٹا سیل (Data Cell) بنانے کے لیے استعمال ہوں گے۔ ہر قطار کے ایک کالم کو ڈیٹا سیل کہا جاتا ہے۔ ایک ڈیٹا سیل میں ٹیکسٹ، تصویر، پیراگراف وغیرہ ہو سکتے ہیں۔

NOT FOR SALE

امتحان کی ڈیٹ شیٹ بنانے کے لئے شکل نمبر 7.52 میں HTML ڈاکومنٹ دکھایا گیا ہے۔ اس ٹیبل میں 5 قطاریں اور 4 کالم ہیں۔

شکل نمبر 7.52 ٹیبل بنانے کے لئے HTML ڈاکومنٹ

شکل نمبر 7.52 کے HTML ڈاکومنٹ میں دکھایا گیا درج ذیل ٹیگ:

```
<table border=1 cellspacing=3 cellpadding=8>
```

براؤزر کو یہ بتائے گا کہ ایک ٹیبل بارڈر کے ساتھ اور جس کی سیل سپیسنگ (Cell Spacing) 3 پکسلز اور سیل پیڈنگ (Cell Padding) 8 پکسلز کی ہو، ونڈو میں دکھایا جائے۔ سیل سپیسنگ سیل کے درمیان کا فاصلہ اور سیل پیڈنگ سیل کے اندر موجود ٹیکسٹ کا سیل کے بارڈر سے فاصلہ ہے۔

پہلی قطار کے لیے کوڈ، جس میں ٹائٹل لکھا ہوتا ہے، یہ ہے:

```
<tr>
        <td colspan=4 align=center>
        <b>FINAL EXAMINATIONS</b></td>
</tr>
```


NOT FOR SALE

یہاں colspan کو 4 کالموں کو اکٹھا کرنے کے لیے استعمال کیا جاتا ہے اور center ٹائٹل کو کالم کے درمیان میں الائن (Align) کر دے گا۔ یوزر rowspan کو بھی استعمال کر سکتا ہے تاکہ قطار میں موجود سیلوں کو اکٹھا کر سکے۔ ٹائٹل کو دائیں، بائیں کسی بھی سیدھ میں align سے لایا جاسکتا ہے۔

HTML ڈاکومنٹ میں دوسری قطار میں آنے والے ہر کالم کی ہیڈنگ کے لیے کوڈ ذیل میں دیا گیا ہے۔

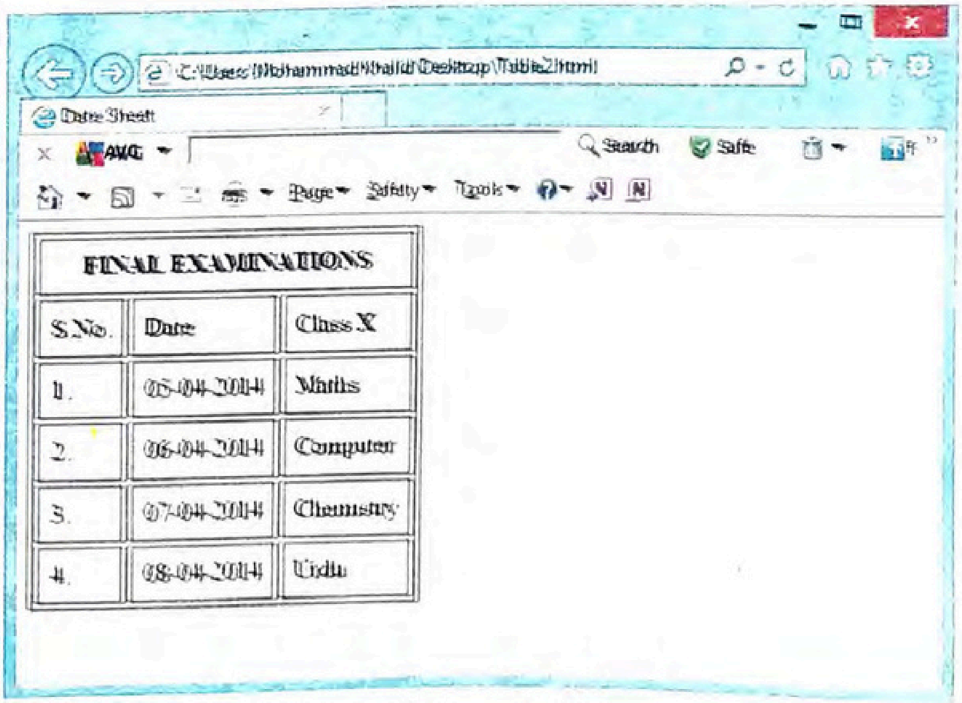
```
<tr>
        <td>S.No.</td>
        <td>Date</td>
        <td>Class X</td>
</tr>
```

دوسری row کے چاروں سیل <tr></tr> ٹیگز کے درمیان اور سیلوں میں موجود ٹیکسٹ <td></td> ٹیگز کے درمیان لکھا جائے گا۔ اگلی row کا ٹیکسٹ کچھ اس طرح سے لکھا جائے گا۔

```
<tr>
<td>1.</td>
<td>05-04-2014</td>
<td>Maths</td>
</tr>
```

اگلی row کا ٹیکسٹ بھی اسی طریقے سے لکھا جائے گا۔

براؤزر ونڈوز حتمی ٹیبل دکھائے گی جیسا کہ شکل نمبر 7.53 میں دکھایا گیا ہے۔



شکل نمبر 7.53 حتمی ٹیبل

NOT FOR SALE

valign سیل کے اندر موجود ٹیکسٹ کی عمودی سیدھ (Vertical Alignment) کی تخصیص کرنے کے لئے استعمال ہوتا ہے۔ مثال کے طور پر

```
<table style="width:50%;">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr style="height:100px">
    <td valign="bottom">January</td>
    <td valign="bottom">RS. 99</td>
  </tr>
</table>
```

Notepad++ ایڈیٹر میں اس کوڈ کو <html> اور <body> ٹیگ کو استعمال کرنے کے بعد، براؤزر ونڈو میں اس کا آؤٹ پٹ کچھ اس طرح نظر آئے گا جیسا کہ شکل نمبر 7.54 میں دکھایا گیا۔

Month	Savings
January	RS. 99

شکل نمبر 7.54 ٹیبل میں valign استعمال ہوا ہے

nowrap ایک بولین ٹیگ ہے۔ جب یہ موجود ہو تو سیل کے اندر موجود ٹیکسٹ لپیٹ (wrap) نہیں پائے گا۔

مثال:

```
<html>
<head>
</head>
  <body>
    <table>
        <tr>
            <th>Poem</th>
            <th>Poem</th>
        </tr>
        <tr>
            <td nowrap>Never increase, beyond what is necessary, the number of entities required
      to explain anything
            </td>
            <td>Never increase, beyond what is necessary, the number of entities required to
      explain anything
            </td>
        </tr>
    </table>
  </body>
</html>
```


NOT FOR SALE

اس کوڈ سے براؤزر ونڈو میں درج ذیل آؤٹ پٹ ظاہر ہو گا جیسا کہ شکل نمبر 7.55 میں دکھایا گیا ہے۔

Poem	Poen
Never increase, beyond what is necessary, the number of entities required to explain anything	Never increase beyond what is necessa the number entities requirec to expla anything

شکل نمبر 7.55 nowrap ایٹریبیوٹ کے ساتھ بنایا گیا ٹیبل

Rowspan ایٹریبیوٹ اس بات کا تعین کرتا ہے کہ کتنی قطاروں پر ایک سیل پھیلا ہونا چاہیے۔ مثال کے طور پر:

```
<html>
<head>
</head>
<body>
 <table>
   <tr>
      <th>Month</th>
      <th>Savings</th>
      <th>Savings for holiday!</th>
   </tr>
   <tr>
     <td>January</td>
     <td>RS. 99</td>
     <td rowspan="2">RS. 50</td>
   </tr>
   <tr>
     <td>February</td>
     <td>RS. 80</td>
   </tr>
 </table>
 </body>
</html>
```

کوڈ کا آؤٹ پٹ براؤزر ونڈو میں، شکل نمبر 7.56 میں دکھایا گیا ہے۔

Month	Savings	Savings for holiday!
January	RS. 99	RS. 50
February	RS. 80	

شکل نمبر 7.56 rowspan ایٹریبیوٹ کے ساتھ بنایا گیا ٹیبل

NOT FOR SALE

7.8 فریم بنانا (Creating Frames)

HTML فریموں کا استعمال آپ کے براؤزر ونڈو کو ایک سے زیادہ حصوں میں تقسیم کر سکتے ہیں جہاں ہر سیکشن الگ HTML ڈاکومنٹ لوڈ کر سکتا ہے۔ براؤزر ونڈو میں فریموں کا ایک مجموعہ Frameset کے نام سے جانا جاتا ہے۔ ونڈو کو قطار اور کالم پر مشتمل ٹیبل کی شکل میں فریموں میں تقسیم کیا گیا ہے۔

7.8.1 فریم (Frame)

براؤزر ونڈو کے اندر ایک فریم ایک چھوٹی مستطیل ونڈو یا پین ہے۔ فریم یوزر کو ایک ہی براؤزر ونڈو میں ایک سے زیادہ HTML ڈاکومنٹ دکھانے کی اجازت دیتا ہے۔ ہر فریم دوسرے سے آزاد ہے۔ HTML ایک <frame> ڈاکومنٹ کو فریم کے اندر رکھنے کے لئے استعمال کیا جاتا ہے۔

7.8.2 فریم سیٹ (Frameset)

frameset اس بات کی وضاحت کرتا ہے کہ براؤزر ونڈو کو فریموں میں کیسے تقسیم کیا جائے۔ کسی پیج پر فریم استعمال کرنے کے لئے ہم <frameset></frameset> استعمال کرتے ہیں۔ frameset ٹیگ وضاحت کرتا ہے، کہ براؤزر ونڈو کو عمودی فریم یا افقی فریموں میں کیسے تقسیم کیا جائے۔

7.8.3 ایک فریم سیٹ تیار کرنا (Creating a Frameset)

براؤزر ونڈو میں فریم بنانے کے لئے <body> ٹیگ کی بجائے <frameset></frameset> اور <frame> ٹیگ ایک ساتھ استعمال کیے جاتے ہیں۔ <frameset> ٹیگ <body> ٹیگ کے اندر نہیں بلکہ header سیکشن میں استعمال ہوتا ہے۔

<frameset></frameset> ٹیگز کالموں یا قطاروں کی ایک سیریز بناتا ہے۔ اس میں دو اوصاف cols اور rows ہیں۔ دونوں ایٹریبیوٹ کا سائنٹکس ایک جیسا ہے۔ cols کالموں کی ایک سیریز کی وضاحت کرتا ہے اور rows قطاروں کی ایک سیریز کی وضاحت کرتا ہے۔

مثال کے طور پر، ٹیگ

```
<frameset cols=30%, 30%, 40%>
```

براؤزر ونڈو کو تین کالموں میں تقسیم کرے گا۔ پہلے دو کالم براؤزر ونڈو کی چوڑائی کے 30% اور تیسرے کالم کو 40% پر سیٹ کریں گے۔ یہ تین عمودی فریم بنائے گا۔

اسی طرح، ٹیگ

```
<frameset rows=50%, 50%>
```


NOT FOR SALE

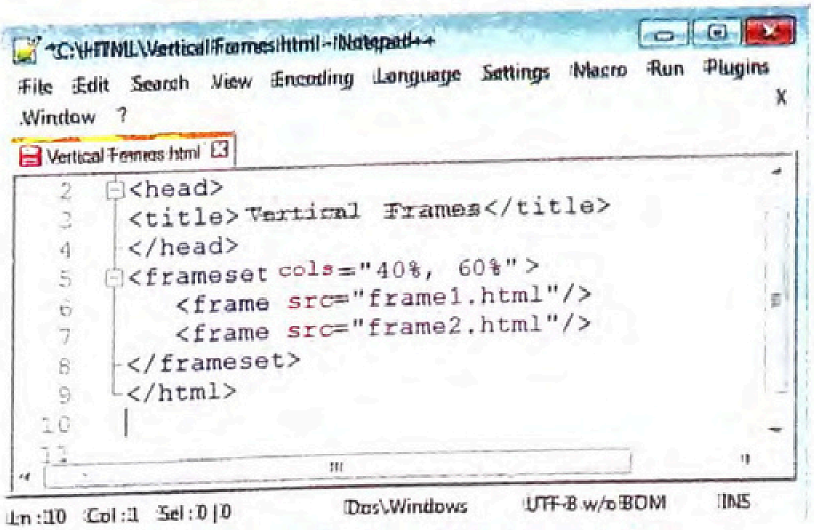
براؤزر ونڈو کو دو برابر سائز والی قطاروں میں تقسیم کرے گا جو کہ براؤزر ونڈو کو دو برابر سائز کے افقی فریموں میں تقسیم کرے گا۔

<frame> ٹیگ <frameset></frameset> ٹیگ کے اندر استعمال کیا جاتا ہے۔ یہ ہر فریم کے مندرجات کی وضاحت کرتا ہے جو <frameset></frameset> ٹیگز کے ذریعے بنائے گئے۔

7.8.4 ایک سے زیادہ فریمز کے ساتھ فریم سیٹ بنانا

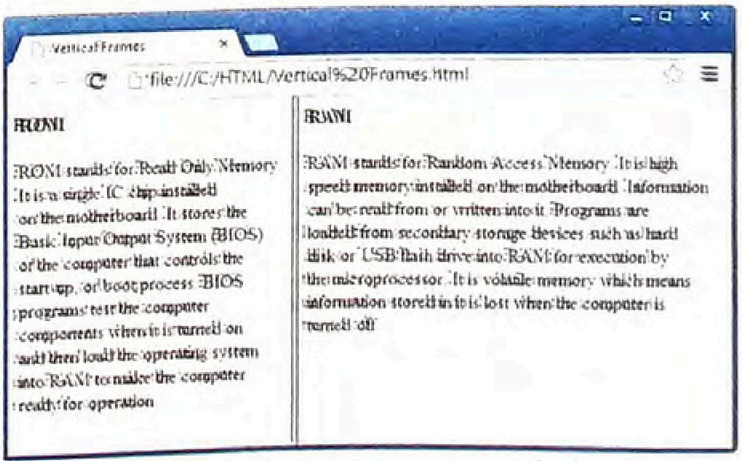
عمودی فریم سیٹ بنانا

شکل نمبر 7.57 میں موجود HTML ڈاکومنٹ عمودی فریم بنانے کا طریقہ ظاہر کرتی ہے۔



شکل نمبر 7.57 عمودی فریم بنانے کے لئے HTML ڈاکومنٹ

یہ براؤزر ونڈو کو دو کالموں میں تقسیم کرے گا۔ پہلا کالم براؤزر ونڈو کی چوڑائی کے 40% اور دوسرا 60% پر سیٹ کیا گیا ہے۔ HTML ڈاکومنٹ frame1.html پہلے کالم میں رکھا جائے گا اور frame2.html دوسرے کالم میں رکھا جائے گا جیسا کہ شکل نمبر 7.58 میں دکھایا گیا ہے۔

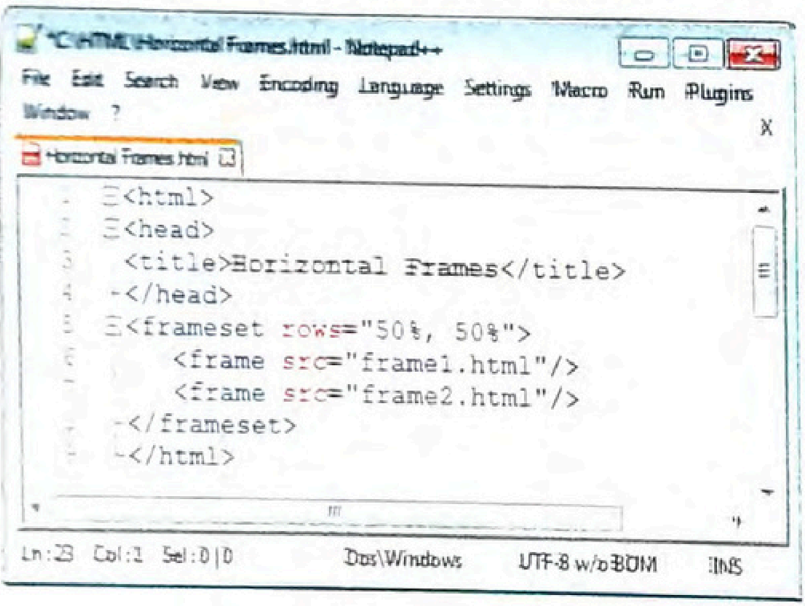


شکل نمبر 7.58 براؤزر ونڈو جس میں دو عمودی فریم ہیں

NOT FOR SALE

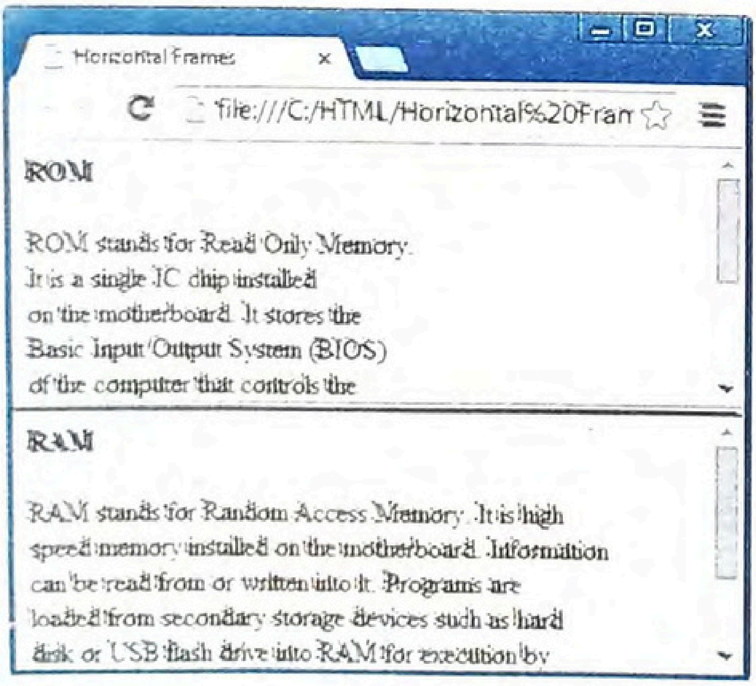
افقی فریمسیٹ بنانا

شکل نمبر 7.59 میں HTML ڈاکومنٹ افقی فریم بنانے کا طریقہ ظاہر کرتی ہے۔



شکل نمبر 7.59 افقی فریم بنانے کے لئے HTML ڈاکومنٹ

یہ براؤزر ونڈو کو دو برابر سائز کے افقی فریموں میں تقسیم کرے گا۔ HTML ڈاکومنٹ frame1.html کو پہلے اور frame2.html دوسرے فریم میں دکھائے گا جیسا کہ شکل نمبر 7.60 میں دکھایا گیا ہے۔



شکل نمبر 7.60: براؤزر ونڈو جس میں دو افقی فریم ہیں

NOT FOR SALE

اہم نکات

- ورلڈ وائڈ ویب (www) انٹرنیٹ پر ایک انفارمیشن سسٹم ہے جو ہائپر ٹیکسٹ لنکس کے ذریعہ ڈاکومنٹ کو دوسرے ڈاکومنٹ سے منسلک کرنے کی اجازت دیتا ہے، جس سے یوزر کو ایک ڈاکومنٹ سے دوسرے ڈاکومنٹ میں منتقل ہو کر معلومات تلاش کرنے میں مدد ملتی ہے۔
- ایک ویب پیج انٹرنیٹ پر ایک ایسا ڈاکومنٹ ہے جس کو ویب براؤزر کے ذریعے حاصل کیا جاسکتا ہے۔
- ایک ویب سائٹ ویب سرور پر موجود متعلقہ ویب پیجز کا ایک مجموعہ ہے۔
- ایک ویب براؤزر وہ سافٹ ویئر ہے جو یوزر کو ویب پر معلومات کو حاصل کرنے کے قابل بناتا ہے۔
- ویب سرور وہ کمپیوٹر ہوتے ہیں جو اپنے پاس موجود ویب سائٹ کے ویب پیجز کی فراہمی کو یقینی بناتے ہیں۔
- URL ایک ایسی فائل کا منفرد ایڈریس ہے جو انٹرنیٹ پر موجود ہوتی ہے۔
- سرچ انجن ایک ویب سائٹ یا سافٹ ویئر سسٹم ہے جو لوگوں کو ورلڈ وائڈ ویب پر معلومات تلاش کرنے کی سہولت دیتا ہے۔
- ہوم پیج کسی ویب سائٹ کا مرکزی یا پہلا ویب پیج ہے۔
- ویب ہوسٹنگ ویب سرورز پر ویب سائٹ اپ لوڈ کرنے کے لئے ایک سہولت ہے، جو ویب ہوسٹنگ کمپنیوں کے ذریعہ فراہم کی جاتی ہے۔
- ویب پورٹلز ایسی ویب سائٹیں ہیں جو مختلف قسم کی خدمات پیش کرتی ہیں۔
- نیوز ویب سائیٹ موجودہ واقعات اور آراء کے بارے میں معلومات فراہم کرتی ہیں۔
- معلوماتی ویب سائیٹ ٹیکسٹ، گرافکس، آواز اور ویڈیوز کی شکل میں کسی بھی عنوان سے متعلق معلومات فراہم کرتی ہیں۔
- تعلیمی ویب سائیٹ تعلیمی مقصد کے لیے تشکیل دی گئیں ہیں۔ ان سائیٹوں میں لوگوں کو مختلف موضوعات پر آگاہی فراہم کرنے کے لئے حرکت پذیر (Animated)، سلائڈ پریزنٹیشنز اور ٹیوٹوریل موجود ہیں۔
- ذاتی ویب سائیٹ ذاتی استعمال کے لئے افراد کے ذریعہ تخلیق اور برقرار رکھی جاتی ہیں۔
- کاروباری ویب سائیٹ انٹرنیٹ کے ذریعے کاروباری تعلقات برقرار رکھنے اور معلومات، خدمات اور اشیاء کو فروخت کرنے کے لئے سہولیات مہیا کرتی ہیں۔
- HTML کا مطلب ہائپر ٹیکسٹ مارک اپ لینگویج ہے۔ یہ ویب پیج بنانے کے لئے استعمال ہوتی ہے۔
- HTML ٹیگ ایک کمانڈ یا کوڈ ہے جو اس بات کی وضاحت کرتا ہے کہ کس طرح ایک ویب پیج فارمیٹ کیا جاتا ہے۔

NOT FOR SALE

مشق

سوال نمبر 1۔ موزوں جواب کا انتخاب کریں۔

i. جو انٹرنیٹ ایڈریس کسی ویب سائٹ کی نشاندہی کرتا ہے کیا کہلاتا ہے؟

(الف) ویب پیج (ب) ویب سائٹ (ج) ویب سرور (د) ویب ایڈریس

ii. ویب پیجز کا مجموعہ جو کسی بھی ویب سرور پر رکھا جاتا ہے ____________ کہلاتا ہے۔

(الف) ویب ایڈریس (ب) ویب سائٹ (ج) ہوم پیج (د) ویب براؤزر

iii. ویب پیجز بنانے کے لیے کونسی لینگوئج استعمال ہوتی ہے؟

(الف) HTML (ب) C لینگوئج (ج) ویب براؤزر (د) [illegible]

iv. ویب پیج پر موجود تصویر یا ٹیکسٹ جو اس ویب پیج کو دوسرے ویب پیج سے منسلک کرتی ہے ____________ کہلاتی ہے۔

(الف) ویب لنک (ب) براؤزر لنک (ج) ہائپر لنک (د) سرچ لنک

v. وہ طریقہ جس سے ویب پیجز کو ویب سرورز پر رکھا جاتا ہے تاکہ دوسرے اس تک رسائی حاصل کر سکیں ____________ کہلاتا ہے۔

(الف) ویب پیج کی تیاری (ب) ویب سرفنگ

(ج) ویب سائٹ انسٹالیشن (د) ویب ہوسٹنگ

سوال نمبر 2۔ ویب سائٹ کیا ہے؟ ویب سائٹس کی مختلف اقسام کو مثالوں سے واضح کریں۔

سوال نمبر 3۔ درج ذیل HTML ٹیگز لکھیں۔

i Paragraph پیراگراف	ii Heading ہیڈنگ	iii Bold بولڈ
iv Underline انڈرلائن	v Italic اٹالک	vi Center text سنٹر ٹیکسٹ
vii Superscript سپر اسکرپٹ	viii Subscript سب اسکرپٹ	ix فانٹ سائز، کلر اور ٹائپ فیس

سوال نمبر 4۔ HTML میں ایک ویب پیج بنائیں جو پرنٹر کی تصویر کو دکھا رہا ہو۔ تصویر کی چوڑائی 250 پکسلز اور اونچائی 150 پکسلز ہو۔

سوال نمبر 5۔ وضاحت کریں کہ کس طرح ویب پیج کے پس منظر میں رنگ اور تصویر لگائی جا سکتی ہے؟

سوال نمبر 6۔ HTML ڈاکومنٹ بنائیں جس میں ایک گرافیکل ہائپر لنک ہو۔

سوال نمبر 7۔ HTML میں ایک ویب پیج بنائیں جس میں ایک ان آرڈرڈ لسٹ اور ایک آرڈرڈ لسٹ ہو۔

سوال نمبر 8۔ ایک HTML ڈاکومنٹ بنائیں جس میں تین مختلف ویب سائٹس کے لیے ہائپر لنک موجود ہوں۔

سوال نمبر 9۔ ایک HTML ڈاکومنٹ بنائیں جس میں کلاس کا ٹائم ٹیبل دکھایا گیا ہو۔

NOT FOR SALE

سوالات کے جوابات

باب 1 پروگرامنگ کا طریقہ کار

سوال نمبر 1۔ موزوں جواب کا انتخاب۔

i	ii	iii	iv	v
b	a	a	c	c

باب 2 C میں پروگرامنگ

سوال نمبر 1۔ موزوں جواب کا انتخاب۔

i	ii	iii	iv	v	vi	vii	viii	ix	x
b	c	d	d	a	b	a	a	b	b

باب 3 اِن پُٹ اور آؤٹ پُٹ کا استعمال

سوال نمبر 1۔ موزوں جواب کا انتخاب۔

i	ii	iii	iv	v	vi	vii	viii	ix	x
d	a	d	c	c	c	b	d	a	a

باب 4 کنٹرول سٹرکچر

سوال نمبر 1۔ موزوں جواب کا انتخاب۔

i	ii	iii	iv	v
a	b	a	c	a

باب 5 لوپ سٹرکچر

سوال نمبر 1۔ موزوں جواب کا انتخاب۔

i	ii	iii	iv	v	vi	vii	viii
b	a	d	c	a	d	a	b

باب 6 کمپیوٹر لاجک اور لاجک گیٹس

سوال نمبر 1۔ موزوں جواب کا انتخاب۔

i	ii	iii	iv	v	vi
a	c	a	c	b	a

باب 7 ورلڈ وائیڈ ویب اور HTML

سوال نمبر 1۔ موزوں جواب کا انتخاب۔

i	ii	iii	iv	v
d	b	a	c	d

NOT FOR SALE

فرہنگ

آ-ا

آپریٹر آپریٹر ایسی علامات ہیں جو ڈیٹا پر کوئی نہ کوئی فعل سرانجام دینے کے لیے استعمال ہوتی ہیں۔

الگورتھم مناسب اور منطقی اقدامات کی وہ فہرست جو کسی مخصوص مسئلے کے حل کے طور پر لکھی جاتی ہیں انہیں الگورتھم کہا جاتا ہے۔

AND گیٹ یہ گیٹ لاجیکل ضرب کرتے ہیں، اس کو عام طور پر AND فنکشن بھی کہا جاتا ہے۔

اخباری ویب سائیٹ اخباری یا خبروں والی ویب سائٹ حالات حاضرہ اور افکار و نظریات سے متعلق معلومات فراہم کرتی ہیں۔

اسمبلی لینگویج اسمبلی لینگویج بھی لو لیول لینگویج ہے۔ یہ مشین لینگویج سے ایک درجہ اوپر ہے۔ اس لینگویج میں بائنری کوڈ کی بجائے علامات کو استعمال کیا جاتا ہے۔ ان علامات کو نمونکس (mnemonics) کہا جاتا ہے۔

اسکیپ سیکونس اسکیپ سیکونس سے مراد حروف کا وہ مجموعہ ہے جو\ سے شروع ہوتے ہیں اور اس کے بعد کوئی بھی دوسرا حرف یا علامت آتی ہے۔ اسکیپ سیکونس آؤٹ پٹ کو ترتیب دینے کے لیے استعمال ہوتے ہیں۔

اِن پُٹ پروگرامنگ کے عمل میں اِن پُٹ سے مُراد وہ ڈیٹا ہے جو کمپیوٹر کو پروسیسنگ کے لیے فراہم کیا جاتا ہے۔

اِن پُٹ اسٹیٹمنٹ C میں ویری ایبل میں ڈیٹا سٹور کرنے کے بہت سے طریقے ہیں۔ اس کے لیے اسائنمنٹ اور اِن پُٹ اسٹیٹمنٹ استعمال ہوتی ہیں۔

انٹرپریٹر انٹرپریٹر بھی ہائی لیول لینگویج کا مترجم ہے جو پروگرام کی اسٹیٹمنٹ کو ایک ایک کرکے ترجمہ کرتا ہے اور ساتھ ہی ساتھ چلاتا بھی ہے۔

انٹیگریٹڈ ڈیویلپمنٹ انوائرمنٹ (IDE) پروگرامنگ کے آلات اور عوامل (Processes) کا ایک مجموعہ ہے جو کمپیوٹر پروگرام بنانے، کمپائل کرنے اور انہیں چلانے کے لیے استعمال کیا جاتا ہے۔

انکریمنٹ انکریمنٹ آپریٹر دوہری جمع کی علامت ++ سے ظاہر کیا جاتا ہے۔ یہ کسی بھی عددی ویری ایبل کی قیمت میں 1 عدد جمع کرنے کے لیے استعمال کیا جاتا ہے۔

ب

بولین فنکشن/ایکسپریشن بولین فنکشن یا ایکسپریشن ویری ایبل یا کانسٹنٹ کا مجموعہ ہوتا ہے جنہیں بولین آپریٹر کی مدد سے آپس میں جوڑا جاتا ہے۔

بائنری آپریٹر ایسا آپریٹر جس کو اپنا عمل مکمل کرنے کے لیے دو رقموں کی ضرورت ہو اسے بائنری (Binary) آپریٹر کہا جاتا ہے۔

بزنس ویب سائیٹ بزنس یا کاروباری ویب سائیٹ انٹرنیٹ کے ذریعے کاروباری تعلقات برقرار رکھنے اور معلومات، خدمات اور اشیاء کو فروخت کرنے کے لئے سہولیات مہیا کرتی ہیں۔

پ

پروسیجرل لینگویج پروسیجرل لینگویج کو تیسری نسل کی زبان (Third Generation Language) بھی کہا جاتا ہے۔ اس زبان میں پروگرام کو پروسیجرز یا موڈیولز کی شکل میں تیار کیا جاتا ہے۔

NOT FOR SALE

پروگرام کمپیوٹر پروگرام ہدایات (Instructions) کا ایسا مجموعہ ہے جو کمپیوٹر کو بتا سکے کہ کیا کرنا ہے اور کیسے کرنا ہے۔

پروگرام سائنٹکس سائنٹکس (Syntax) دراصل کسی بھی پروگرامنگ لینگوئج کے وہ اصول ہیں جن کے تحت کسی بھی لینگوئج میں کوئی بھی پروگرام لکھا جاتا ہے، یہ وہ طریقۂ کار ہے جس سے ہم پروگرام کے اندر صحیح اسٹیٹمنٹ (Statements) لکھ سکتے ہیں۔

پروگرام کا سیمنٹک سیمنٹک (Semantic) پروگرامنگ لینگوئج کی اسٹیٹمنٹ کے مطلب کی وضاحت کرتا ہے۔

پروگرامنگ لینگوئج الفاظ، علامات اور کوڈ کا مجموعہ جن کی مدد سے آپ کوئی بھی پروگرام لکھتے ہیں انہیں پروگرامنگ لینگوئج کہا جاتا ہے۔

پری پروسیسر ڈائریکٹو پروگرام کے حقیقی آغاز سے پہلے کمپائلر کو جو ہدایات دی جاتی ہیں انہیں پری پروسیسر ڈائریکٹیو یا کمپائلر ڈائریکٹیو کہا جاتا ہے۔

ت

تعلیمی ویب سائٹ تعلیمی ویب سائٹ خالصتاً تعلیمی مقاصد کے لیے تیار کی جاتی ہیں۔ ان سائٹ پر اینیمیشن (Animation) سلائیڈ اور ٹیوٹوریل (Tutorial) رکھے جاتے ہیں تاکہ لوگوں کو مختلف موضوعات پر تعلیم دی جا سکے۔

ٹ

ٹائپ کاسٹنگ پروگرام چلنے کے دوران کسی ایک ڈیٹا ٹائپ کو کسی دوسری ڈیٹا ٹائپ میں تبدیل کرنے کے عمل کو ٹائپ کاسٹنگ کیا جاتا ہے۔

ٹروتھ ٹیبل یہ ایسا جدول یا ٹیبل ہے جس میں اِن پُٹ اور آؤٹ پُٹ کے تمام ممکنہ جوڑوں کو اکٹھا کر کے لکھا جاتا ہے۔

ڈ

ڈیٹا ڈیٹا خام اعداد و شمار ہیں جن کو ڈیجیٹل کمپیوٹر پروسیس کرتا ہے۔

ڈیکریمنٹ آپریٹر ڈیکریمنٹ آپریٹر کو دوہری منفی علامت - - سے ظاہر کیا جاتا ہے۔ یہ کسی انٹیجر ویری ایبل کی قیمت سے 1 منفی کرنے کے لیے استعمال ہوتا ہے۔

ذ

ذاتی ویب سائیٹ ذاتی ویب سائٹ افراد کے ذاتی استعمال کے لیے تیار کی جاتی ہے۔ اسے افراد خود ترتیب دیتے اور منظم رکھتے ہیں۔

ر

ریلیشنل آپریٹر ریلیشنل آپریٹر یا نسبتی آپریٹر پروگرام میں کسی بھی قسم کی شرط (Condition) لگانے کے لیے استعمال ہوتا ہے۔ ایک ریلیشنل آپریٹر دو قیمتوں کا آپس میں موازنہ کرتا ہے۔

س

سٹرکچرل لینگوئج ایسی پروگرامنگ لینگوئج جس میں پروگرام کو چھوٹے چھوٹے حصوں، بلاک یا پروسیجرز میں تقسیم کیا جا سکے اسے سٹرکچرڈ پروگرامنگ لینگوئج کہا جاتا ہے۔

سرچ انجن سرچ انجن ایک ویب سائٹ یا سافٹ ویئر سسٹم ہے جو لوگوں کو ورلڈ وائیڈ ویب پر معلومات تلاش کرنے کی سہولت دیتا ہے۔

NOT FOR SALE

ص

صریح حل Candid Solution: مسئلے کو حل کرنے کے لیے بہت سے متبادل ہو سکتے ہیں، تاہم مسئلے کا بہترین اور مناسب ترین حل معلوم کرنا ہی مسئلے کا حل قرار دیا جا سکتا ہے۔

ف

فلو چارٹ: فلو چارٹ الگورتھم کا تصویری (Diagrammatic) یا علامتی (Symbolic) اظہار ہے۔

فلو چارٹ کی علامات: الگورتھم میں موجود مختلف اقدامات کے لیے فلو چارٹ کے اندر مختلف علامات استعمال ہوتی ہیں ان علامات کو فلو چارٹ کی علامات کہا جاتا ہے۔

فارمیٹ سپیسیفائر: فارمیٹ سپیسیفائر تبادلہ کرنے والی کوڈ کا نام ہے۔ یہ آؤٹ پُٹ کی جانے والی قیمت کی ڈیٹا ٹائپ کے مطابق استعمال کیا جاتا ہے۔

ک

کارناف میپ Karnauf (K-Map): کارناف میپ یا K-map بولین ایکسپریشن کا ایک تصویری خاکہ ہے۔ یہ مربع نما خانوں پر مشتمل ہوتا ہے جنھیں سیلز (Cells) کہا جاتا ہے۔ کارناف میپ کی مدد سے ہم ان تمام ویری ایبل کو اکٹھا (Group) کر لیتے ہیں جن میں کوئی نہ کوئی قدر مشترک ہو اس طرح ہم غیر ضروری ویری ایبل کو ختم کر سکتے ہیں۔

کلیدی الفاظ: ایسے الفاظ جو کسی بھی زبان میں مخصوص مقاصد کے لیے استعمال ہوتے ہوں انھیں کلیدی الفاظ یا ریزروڈ ورڈز (Reserved Words) کہا جاتا ہے۔

کمپائلر: کمپائلر ایک ایسا سسٹم سافٹ ویئر ہے جو ہائی لیول پروگرامنگ لینگویج کے سورس کوڈ (Source Code) کو مشین لینگویج آبجیکٹ کوڈ میں ترجمہ کرتا ہے۔

کنڈیشنل آپریٹر: کنڈیشنل آپریٹر ایک ایسا سٹرکچر ہے جو فیصلے کرنے کے لیے استعمال ہوتا ہے۔ یہ بالکل سادہ سے ***if else*** سٹرکچر کی جگہ استعمال کیا جاتا ہے۔

کنٹرول اسٹیٹمنٹ: ایسی اسٹیٹمنٹ جو پروگرام چلنے کے دوران اس کی اسٹیٹمنٹ کی ترتیب کا تعین کرتی ہیں۔ کنٹرول اسٹیٹمنٹ (Control Statement) کہلاتی ہیں۔

کنٹرول سٹرکچرز: کنٹرول سٹرکچرز پروگرام کے چلنے کی عمومی ترتیب کو بدلنے کے لیے استعمال ہوتے ہیں۔

ل

لاجک گیٹ: لاجک گیٹ ایسا الیکٹرانک سرکٹ ہے جو ایک یا ایک سے زیادہ سگنلز کو بطور ان پُٹ لیتا ہے اور ان سے ایک ہی آؤٹ پُٹ مہیا کرتا ہے۔

لنکر: ایسا کمپیوٹر پروگرام جو کمپائلر سے بنائے گئے بہت سارے آبجیکٹ کوڈ ایک جگہ جمع کر کے انھیں ایک ہی پروگرام کی شکل دیتا ہے تاکہ وہ ایک ہی پروگرام کے طور پر چلائے جا سکیں، لنکر (Linker) کہلاتا ہے۔

NOT FOR SALE

لوپ	لوپ ہدایات کا ایسا تسلسل ہے جو کہ بار بار چلایا جاتا ہے یہاں تک کہ ایک مخصوص حالت (Condition) واقع ہو جا
لوڈر	ایک ایسا پروگرام ہے جو C لینگویج کے پروگرام کو میموری میں لوڈ کرتا ہے اور پھر اس کو چلاتا ہے۔
لو لیول لینگویج	لو لیول لینگویج کمپیوٹر کے ہارڈویئر کے زیادہ قریب ہیں۔

م

متغیر	ایسی مقدار جس کی قیمت پروگرام چلنے کے دوران تبدیل ہو جائے اسے متغیر یا ویری ایبل (Variable) کہا جاتا ہے۔
مستقل	مستقل یا کانسٹنٹ وہ مقدار ہے جس کی قیمت پروگرام کے چلنے کے دوران تبدیل نہیں ہوتی۔ حقیقت میں کانسٹنٹ اس قیم کا نام ہے جو پروگرام میں براہ راست ٹائپ کی گئی ہو۔
مسئلہ	مسئلہ ایک رکاوٹ، مشکل یا چیلنج یا کوئی بھی ایسی صورت حال ہے جس سے نکلنا مقصود ہو۔
مشینی زبان	زبان کی وہ قسم جس میں ہم ہدایات کو بائنری شکل میں لکھتے ہیں مشینی زبان کہلاتی ہے۔
معلوماتی ویب سائٹ	معلوماتی ویب سائٹ کسی بھی موضوع پر معلومات فراہم کرتی ہیں جو کہ ٹیکسٹ، گرافکس، آواز اور ریڈیو وغیرہ کی شکل میں ہوتی ہیں۔
فنکشن	وہ چھوٹا سا پروگرام جو کسی خاص مقصد کے لیے استعمال ہو اُسے کمپیوٹر کی زبان میں فنکشن کہا جاتا ہے۔

ن

NAND گیٹ	وہ گیٹ جس کی آؤٹ پُٹ 0 ہوتی ہے جب اس کی تمام ان پُٹ 1 ہوں، NAND گیٹ کہلاتا ہے۔
نیسٹڈ لوپ	ایک لوپ کے اندر دوسری لوپ کو بند کرنے کے عمل کو نیسٹڈ لوپ کہا جاتا ہے۔
NOR گیٹ	ایسا گیٹ جس کی تمام اِن پُٹ اگر 0 ہوں اور اس کی آؤٹ پُٹ 1 آئے تو اسے NOR گیٹ کہتے ہیں۔
NOT گیٹ	NOT گیٹ کسی بھی ان پُٹ کو معکوس (Inverse) کرنے کے لیے استعمال ہوتا ہے۔

و

ورلڈ وائیڈ ویب	ورلڈ وائیڈ ویب (www) انٹرنیٹ پر موجود ایک انفارمیشن سسٹم ہے جو اس بات کی سہولت فراہم کرتا ہے کہ ڈاکومنٹس کو آپس میں ایک دوسرے کے ساتھ مربوط (Link) کیا جا سکے۔
ویب براؤزر	ایک ویب براؤزر وہ سافٹ ویئر ہے جو یوزر کو ویب پر معلومات کو حاصل کرنے کے قابل بناتا ہے۔
ویب ہوسٹنگ	ویب ہوسٹنگ ویب سرورز پر ویب سائٹ اپ لوڈ کرنے کے لئے ایک سہولت ہے، جو ویب ہوسٹنگ کمپنیوں کے ذریعہ فراہم کی جاتی ہے۔
ویب پیج	ایک ویب پیج انٹرنیٹ پر ایک ایسا ڈاکومنٹ ہے جس کو ویب براؤزر کے ذریعے حاصل کیا جا سکتا ہے۔
ویب پورٹل	ویب پورٹلز ایسی ویب سائٹیں ہیں جو مختلف قسم کی خدمات پیش کرتی ہیں۔

NOT FOR SALE

ویب سرور ویب سرور وہ کمپیوٹر ہوتے ہیں جو اپنے پاس موجود ویب سائیٹ کے ویب پیجز کی فراہمی کو یقینی بناتے ہیں۔

ویب سائیٹ ایک ویب سائیٹ ویب سرور پر موجود متعلقہ ویب پیجز کا ایک مجموعہ ہے۔

ہ

ہائی لیول لینگویج ہائی لیول لینگویج انگریزی الفاظ پر مشتمل زبان ہے۔ پروگرام کی ہدایات انگریزی کے الفاظ میں لکھی جاتی ہیں۔

ہوم پیج ہوم پیج ویب سائیٹ کا وہ سب سے پہلا پیج ہے جو اس وقت براؤزر میں کھلتا ہے جب اسے ویب سائیٹ لنک تک رسائی حاصل ہو۔

ہیڈر فائل ہیڈر فائل سٹینڈرڈ لائبریری فنکشن کا ایک مجموعہ ہے جو مختلف افعال انجام دینے کے لیے استعمال ہوتا ہے۔

H TML ہائپر ٹیکسٹ مارک اپ لینگویج ہے۔ یہ ویب پیج کی تیاری کے لیے استعمال ہوتی ہے۔

HTML ٹیگ HTML ٹیگ ایک کمانڈ یا کوڈ ہے جو یہ تعین کرتا ہے کہ ویب پیج کا فارمیٹ کیسا ہوگا۔

ی - ے

یونری آپریٹر ایسا آپریٹر جس کو اپنا عمل مکمل کرنے کے لیے صرف ایک رقم چاہیئے ہو، اسے یونری آپریٹر کہا جاتا ہے۔

URL URL یعنی یونیفارم ریسورس لوکیٹر، انٹرنیٹ پر موجود قابل رسائی فائل کا منفرد ایڈریس ہے۔

NOT FOR SALE

انڈیکس

آ-ا

آپریٹر 63
آؤٹ پُٹ فنکشن 52
آرڈرڈ لسٹ 156
اینکر 167
اسمبلی لینگویج 26
اسائنمنٹ آپریٹر 65
انٹیگریٹڈ ڈویلپمنٹ انوائرنمنٹ 33
اسکیپ سیکونس 62
if اسٹیٹمنٹ 79
الگورتھم 6
AND گیٹ 112
اِن پُٹ فنکشن 55
انٹرپریٹر 33
ان آرڈرڈ لسٹ 156
انکریمنٹ آپریٹر 66
ایڈیٹر 34
اریتھ میٹک آپریٹر 64

ب

بائنری آپریٹر 72
بریک (break) اسٹیٹمنٹ 102

پ

پرنٹ (printf) فنکشن 52
پروسیجرل لینگویج 28
پروگرام 25
پروگرام کا مفہوم 25
پروگرام کی ترکیب 25
پروگرامنگ لینگویجز 25
پری پروسیسر ڈائریکٹیو 33
puts() فنکشن 53

ت

تبصرے یا کمنٹس 40
تعلیمی ویب سائٹ 144
تفریحی ویب سائٹ 145

ٹ

ٹیکسٹ فارمیٹنگ 149
ٹروتھ ٹیبل 113
ٹائپ کاسٹنگ 45

ح

حسابی آپریٹر 64
LMTH 146

ڈ

ڈیکریمنٹ آپریٹر 68
do-while لوپ 99
ڈیٹا ٹائپ 43

ذ

ذاتی ویب سائٹ 144

س

سرچ انجن 139

NOT FOR SALE

scanf() فنکشن 55
سنگل کمنٹ 40
سٹرکچرڈ لینگوئج 28
switch اسٹیٹمنٹ 83

ص

صریح حل 05

ف

فلوچارٹ 13
فلوچارٹ کی ضروریات 14
فلوچارٹ کی علامات 15
for لوپ 95
فارمیٹ سپیسیفائر 60
فریم 177
فریم سیٹ 177

ک

کمپائلر 32
کمپاؤنڈ اسائنمنٹ آپریٹر 65
کنڈیشنل آپریٹر 78
continue اسٹیٹمنٹ 103
کنٹرول اسٹیٹمنٹ 78
کارناف میپ (K-map) 122
کلیدی الفاظ 37
کاروباری ویب سائٹ 145

گ

getch() 56
getchar() 57
getche() 57
gets() 58
گرافیکل ہائپر لنک 171

ل

لاجک گیٹس 111
لسٹ کی اقسام 156
لاجیکل آپریٹرز 70
لوپ سٹرکچر 94
لو لیول لینگوئج 26

م

معلوماتی ویب سائٹ 143
مشینی زبان 26
main() فنکشن 39
مسئلہ 2
مسئلے کا تجزیہ 3
مسئلے کے حل میں الگورتھم کا کردار 6

ن

NAND گیٹ 116
نیسٹڈ لسٹ 156
نیسٹڈ لوپ 104
XNOR گیٹ 118
NOT گیٹ 113
نسبتی (Relational) آپریٹر 69

و

ویری ایبلز 41
ویب براؤزر 136
ویب ہوسٹنگ 140
ویب پیج 134

NOT FOR SALE

ویب پورٹل 141
ویب سائٹ 135
while() لوپ 97
world wide web (www) 134
web سرور 137

ہ

ہیڈر فائل 36
ہائی لیول لینگوتج 27
ہوم پیج 140
ہائپر ٹیکسٹ مارک اپ لینگوتج 146

ی-ے

Unary Operators 71
138.Uniform Resource Locator (URL)

NOT FOR SALE

ضمیمہ نمبر 1

اشکال کی فہرست

شکل	صفحہ نمبر
شکل نمبر 1.1 مسئلے کو سمجھنا	2
شکل نمبر 1.2 مسئلے کا تجزیہ کرنا	3
شکل نمبر 1.3 مسئلے کے حل کے لیے منصوبہ بندی	4
شکل نمبر 1.4 ممکنہ جوابات (Brainstorming)	4
شکل نمبر 1.5 بہترین حل کا چناؤ کرنا	5
شکل نمبر 1.6 الگورتھم کی کارکردگی	7
شکل نمبر 1.7 پانچ نمبروں کا اوسط معلوم کرنے کا فلو چارٹ	13
شکل نمبر 2.1 لو لیول اور ہائی لیول پروگرامنگ لینگویجز	26
شکل نمبر 2.2 اسمبلی لینگویج میں ترجمے کا عمل	27
شکل نمبر 2.3 مختلف پروگرامنگ لینگویجز	27
شکل نمبر 2.4 کمپیوٹر لینگویجز	28
شکل نمبر 2.5 ہائی لیول لنگویجز	29
شکل نمبر 2.6 ہائی لیول لینگویج کی خصوصیات	30
شکل نمبر 2.7 مقبول ہائی لیول پروگرامنگ لینگویجز	32
شکل نمبر 2.8 کمپائلر اور انٹرپرٹر کیسے کام کرتے ہیں	33
شکل نمبر 2.9 C لینگویج کا IDE	34
شکل نمبر 2.10 سورس کوڈ کو کمپائل کر کے مشین کوڈ میں تبدیل کرنے کا عمل	35
شکل نمبر: 2.11 لنکر کا عمل	35
شکل نمبر: 2.12 کمپائل اور لنک کرنے والا C پروگرام	36
شکل نمبر C2.13 زبان میں ہیڈر فائلیں	37
شکل نمبر 2.14 C لینگویج میں پری پروسیسر کی مثال	38
شکل نمبر 2.15 سنگل لائن اور ملٹی لائن کمنٹس	41

NOT FOR SALE

شکل	صفحہ نمبر
شکل نمبر 2.16 کمپیوٹر میموری میں ویری ایبل	41
شکل نمبر 2.17 ویری ایبل اور کانسٹنٹ کی اقسام	42
شکل نمبر 2.18 ٹائپ کاسٹنگ کی ترتیب	45
شکل نمبر 3.1 cout آبجیکٹ	54
شکل نمبر: cin 3.2 کا استعمال	59
شکل نمبر 3.3 تقسیم کا عمل	65
شکل نمبر if 4.1 سٹرکچر	80
شکل نمبر if-else 4.2 کا سٹرکچر	82
شکل نمبر switch-case 4.3 سٹرکچر	85
شکل نمبر 4.4 نیسٹڈ if-else کا ترکیب	87
شکل نمبر for 5.1 لوپ کی ترکیب	96
شکل نمبر while 5.2 لوپ کا کام کرنے کا انداز	98
شکل نمبر do-while 5.3 لوپ کا کام کرنے کا انداز	100
شکل نمبر break 5.4 اسٹیٹمنٹ کا کام کرنے کا طریقہ کار	102
شکل نمبر continue 5.5 اسٹیٹمنٹ کا طریقہ کار	104
شکل نمبر 5.6 نیسٹڈ لوپ	105
شکل نمبر 6.1 دو اِن پُٹ والا AND گیٹ	112
شکل نمبر 6.2 دو اِن پُٹ والا OR گیٹ	113
شکل نمبر NOT 6.3 گیٹ	113
شکل نمبر 6.4 دو اِن پُٹ والا NOR گیٹ	116
شکل نمبر 6.5 دو اِن پُٹ والا NAND گیٹ	117
شکل نمبر 6.6 دو اِن پُٹ والا XOR گیٹ	117
شکل نمبر 6.7 دو اِن پُٹ والا XNOR گیٹ	118
شکل نمبر 6.8 بنیادی گیٹس کی مدد سے NAND گیٹ بنانا	118
شکل نمبر OR 6.9 گیٹ اور NOT گیٹ کا استعمال کر کے NOR گیٹ بنانا	119

NOT FOR SALE

شکل	صفحہ نمبر
شکل نمبر 6.10 XOR گیٹ بنانا	120
شکل نمبر 6.11 XNOR گیٹ بنانا	120
شکل نمبر 6.12 K-map اور ٹروتھ ٹیبل	123
شکل نمبر 6.13 1 کے گروپ والا میپ	123
شکل نمبر 6.14 گروپ میں ترچھے سیل شامل نہیں	123
شکل نمبر 6.15 دکھایا گیا ہے کہ گروپ میں 1، 2، 4، 8 سیل یا عمومی طور پر 2n سیل ہوں گے	124
شکل نمبر 6.16 گروپ جہاں تک ممکن ہو بڑے بڑے ہونے چاہیئں	124
شکل نمبر 6.17 ہر سیل جس میں 1 ہے ضرور کسی نہ کسی گروپ میں ہو گا	124
شکل نمبر 6.18 K-map کے اطراف میں موجود سیل آپس میں گروپ بنا سکتے ہیں	125
شکل نمبر 7.1 ویب پیج	135
شکل نمبر 7.2 اسلامیہ کالج کی ویب سائٹ	135
شکل نمبر (a)7.3 گوگل کروم	136
شکل نمبر (b)7.3 انٹرنیٹ ایکسپلورر	136
شکل نمبر (c)7.3 مائیکروسافٹ ایج براؤزر	137
شکل نمبر (d)7.3 موزیلا فائر فاکس	137
شکل نمبر 7.4 ویب براؤزر اور ویب سرور	138
شکل نمبر 7.5 فیڈرل بورڈ کا ویب ایڈریس	139
شکل نمبر 7.6: گوگل سرچ انجن کا استعمال کرتے ہوئے معلومات کی تلاش	139
شکل نمبر 7.7 www.kptbb.gov.pk کا ہوم پیج	140
شکل نمبر (a) 7.8 ویب ہوسٹنگ کا عمل	141
شکل نمبر: (b)7.8 علامہ اقبال اوپن یونیورسٹی کا ویب پورٹل	142
شکل نمبر 7.9: روزنامہ مشرق کی ویب سائٹ	143
شکل نمبر 7.10: معلوماتی ویب سائٹ	143
شکل نمبر 7.11: تعلیمی ویب سائٹ	144
شکل نمبر 7.12: کاروباری ویب سائٹ	145

NOT FOR SALE

شکل	صفحہ نمبر
شکل نمبر 7.13: تفریحی ویب سائٹ	146
شکل نمبر 7.14: ایک HTML ڈاکومنٹ	147
شکل نمبر 7.15: HTML ڈاکومنٹ دکھانے والی براؤزر ونڈو	147
شکل نمبر 7.16: ٹیگز کھولنا اور بند کرنا	148
شکل نمبر 7.17: HTML ڈاکومنٹ میں بنیادی ٹیگ کا استعمال	150
شکل نمبر 7.18: براؤزر ونڈو جو ہیڈنگ اور پیراگراف کو دکھا رہی ہے	150
شکل نمبر 7.19 HTML ڈاکومنٹ میں تمام ہیڈنگ ٹیگز کا استعمال	151
شکل نمبر 7.20: براؤزر ونڈو میں دکھائی جانے والی مختلف ہیڈنگز	151
شکل نمبر 7.21 HTML ڈاکومنٹ میں بولڈ، اٹالک اور انڈر لائن کا استعمال	152
شکل نمبر 7.22 ویب پیج پر بولڈ، اٹالک اور انڈر لائن کا استعمال	153
شکل نمبر 7.23 HTML ڈاکومنٹ میں font ٹیگ کا استعمال	154
شکل نمبر 7.24: ویب پیج میں فونٹ ٹیگز کا استعمال	154
شکل نمبر 7.25 HTML ڈاکومنٹ میں سپر اسکرپٹ اور سبسکرپٹ کا استعمال	155
7.26: ویب پیج میں سپر اسکرپٹ اور سبسکرپٹ ٹیگز کا استعمال	155
شکل نمبر 7.27: ان آرڈرڈ لسٹ کی تیاری	157
شکل نمبر 7.28: ویب پیج میں ان آرڈرڈ لسٹ کا استعمال	157
شکل نمبر 7.29: آئٹمز کا آرڈرڈ لسٹ بنانا	158
شکل نمبر 7.30: ویب پیج میں آرڈر شدہ لسٹ کا استعمال	158
شکل نمبر 7.31: اصطلاحات کی تعریف کی فہرست بنانا	159
شکل نمبر 7.32: ویب پیج میں تعریفی فہرست کا استعمال	159
شکل نمبر 7.33 انیسٹڈ لسٹ بنانا	160
شکل نمبر 7.34: براؤزر ونڈو میں ویب پیج پر نیسٹڈ لسٹ دکھائی گئی ہے	161
شکل نمبر 7.35: HTML ڈاکومنٹ میں تصویر کا استعمال	162
شکل نمبر 7.36: براؤزر ونڈو میں دکھائی گئی تصویر	162
شکل نمبر 7.37: براؤزر ونڈو میں بارڈر کے اندر دکھائی گئی تصویر	163

شکل	صفحہ نمبر
شکل نمبر 7.38: HTML ڈاکومنٹ میں تصویر کے سائز کو بدلنا	164
شکل نمبر 7.39: ویب پیج پر تصویر شدہ سائز	164
تصویر 7.40: ویب پیج اگر تصویر نہ دکھائی دے	164
شکل نمبر 7.41: HTML میں پس منظر اور پیش منظر کے رنگوں کا استعمال	165
شکل نمبر 7.42: پس منظر اور پیش منظر کے رنگ کے ساتھ ویب پیج	165
شکل نمبر 7.43: HTML ڈاکومنٹ کے پس منظر کے لیے تصویر کا استعمال	166
شکل نمبر 7.44: تصویر کو ویب پیج کے پس منظر کے طور پر استعمال کرنا	166
7.45: HTML ڈاکومنٹ میں ہائپر لنکس بنانا	168
شکل نمبر 7.46: ویب پیج میں ہائپر لنکس	168
شکل نمبر 7.47: اینکر پر ہائپر لنکس بنانا	169
شکل نمبر 7.48: HTML ڈاکومنٹ میں ویب پیج کے اندر ہائپر لنکس کا استعمال	170
شکل نمبر 7.49: ایک ویب پیج میں ہائپر لنکس	171
شکل نمبر 7.50 HTML ڈاکومنٹ میں گرافیکل ہائپر لنک کا استعمال	171
شکل نمبر 7.51: ویب پیج میں گرافیکل لنک	172
شکل نمبر 7.52: ٹیبل بنانے کے لئے HTML ڈاکومنٹ	173
شکل نمبر 7.53: حتمی ٹیبل	174
شکل نمبر 7.54: ٹیبل میں valign استعمال ہوا ہے	175
شکل نمبر 7.55: nowrap ایٹریبیوٹ کے ساتھ بنایا گیا ٹیبل	176
شکل نمبر 7.56: rowspan ایٹریبیوٹ کے ساتھ بنایا گیا ٹیبل	176
شکل نمبر 7.57: عمودی فریم بنانے کے لئے HTML ڈاکومنٹ	178
شکل نمبر 7.58: براؤزر ونڈو جس میں دو عمودی فریم ہیں	178
شکل نمبر 7.59: افقی فریم بنانے کے لئے HTML ڈاکومنٹ	179
شکل نمبر 7.60: براؤزر ونڈو جس میں دو افقی فریم ہیں	179

NOT FOR SALE

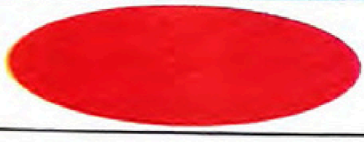
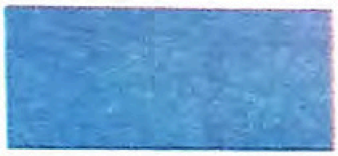
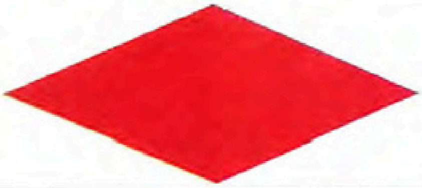
جدول کی فہرست

جدول	صفحہ نمبر
جدول نمبر 6.1 دو اِن پُٹ والے AND گیٹ کا ٹروتھ ٹیبل	114
جدول نمبر 6.2 تین اِن پُٹ والے AND گیٹ کا ٹروتھ ٹیبل	114
جدول نمبر 6.3 دو اِن پُٹ والے OR گیٹ کا ٹروتھ ٹیبل	115
جدول نمبر 6.4 تین اِن پُٹ والے OR گیٹ کا ٹروتھ ٹیبل	115
جدول نمبر 6.5 NOT گیٹ کا ٹروتھ ٹیبل	115
جدول نمبر 6.6 دو اِن پُٹ والے NOR گیٹ کا ٹروتھ ٹیبل	116
جدول نمبر 6.7 دو اِن پُٹ والے NAND گیٹ کا ٹروتھ ٹیبل	117
جدول نمبر 6.8 دو اِن پُٹ والے XOR گیٹ کا ٹروتھ ٹیبل	117
جدول نمبر 6.9 دو اِن پُٹ والے XNOR گیٹ کا ٹروتھ ٹیبل	118
جدول نمبر 6.10 AND اور NOT گیٹ کو استعمال کر کے NAND گیٹ کا ٹروتھ ٹیبل	119
جدول نمبر 6.11 OR گیٹ اور NOT گیٹ کا استعمال کر کے NOR گیٹ بنانا	119
جدول نمبر 6.12 XOR گیٹ کا ٹروتھ ٹیبل	120
جدول نمبر 6.13 $Z = F(A,B) = A\bar{B} + AB$ کے لیے ٹروتھ ٹیبل	125
جدول نمبر 6.14 $Z = F(A,B) = \bar{A}\bar{B}+A\bar{B}+A\bar{B}$ کے لیے ٹروتھ ٹیبل	126

NOT FOR SALE

ضمیمہ نمبر 3

نشانات کی فہرست

نام	نشان
آغاز	
پروسیس	
اِن پُٹ / آؤٹ پُٹ	
فیصلہ	
تیر	

ضمیمہ نمبر 4

Notepad++ کمانڈز کی فہرست

مقصد	کمانڈ	مقصد	کمانڈ
Select All	Ctrl+A	Copy	Ctrl+C
Save	Ctrl+S	Cut	Ctrl+X
Open	Ctrl+O	Paste	Ctrl+V
New	Ctrl+N	Undo	Ctrl+Z
		Redo	Ctrl+Y

NOT FOR SALE

کتابیات

- Introduction to Algorithms Part 2 Third Edition by Cormen
- Algorithm Design by Jon Kleinberg and Éva Tardos
- The C Programming Language by Brian W. Kernighan, Dennis M. Ritchic
- C Programming Absolute Beginner's Guide by Dean Miller and Greg Perry
- C Programming: A Modern Approach by K.N. King
- C Programming in 12 Easy Lessons by Greg Perry
- C for Dummies Vol. I by Dan Gookin
- C for Dummies Vol. II by Dan Gookin
- The C Programming Language. 2nd Edition by Brian Kernighan and Dennis Ritchie
- Teach Yourself MAC C++ in 21 Days by Namie Shammas Sams Publishing.
- Turbo C++ by Jowl Adams, Sanford Leestma, Larry Nyhoff, Prentice Hall, Inc., USA.
- C++ How to Program Deitel & Detiel, Prntice-Hall, Inc., New Jersey.
- Let Us C, by Yashwant Kanetkar, BPB Publications, New Delhi.
- Introducing C to Pascal Programmers by Namir Shammas, John Wiley & Sons, Inc.,
- Programming with C++ by John Hubbard, McGraw-Hill Companies, Inc.
- Turbo C++ by Norman E. Smith, CDP. Wordware Publishing, Inc.
- Turbo C++ by Example by Greg Perry, Marcus Johnson. SM Printer, Karachi.
- Mastering C by Craig Bolon, BPB Publications. New-Delhi.
- Programming on-line help by Clay Riley. BPB Publications, New-Delhi.
- Object-Oriented Programming in C++ by Robert Lafore, Waite Group Cote Madera.
- HTML Intermediate Skills & Drills by Mary Gillen
- HTML 4 for Dummies Quick Reference by Eric Ray
- New Perspectives on Creating Web Pages with HTML Comprehensive by Patrick Carey & Joan Carey
- New Perspectives on Creating Web Pages with HTML - Introductory by Patrick Carey
- Using HTML 4, 3rd Ed. by Lois Paterson et al.
- HTML 3.2 Programmer's Reference, 2nd Ed. by Robert Mullen
- Teach Yourself HTML Publishing on the World Wide Web by Mac Bride
- Digital Logic and Computer Design by M Morris Mano
- Principle of digital Electronics by Malvino & Leach Modern Digital Electronics by R.P.Jain
- Digital Fundamentals 10th Edition Thomas by L Floyd
- Digital Integrated Electronics by H. Taub and D. Schilling
- Contemporary Logic Design by Randy H. Katz and Gaetano Borriello

NOT FOR SALE

مصنفین کے بارے میں

محمد خالد او-پی-ایف بوائز کالج برائے طلباء، اسلام آباد میں شعبہ کمپیوٹر سائنس کے سربراہ ہیں۔ انہوں نے پشاور یونیورسٹی سے کمپیوٹر سائنس میں ایم ایس کی ڈگری حاصل کی اور انسٹی ٹیوٹ آف ایجوکیشن اینڈ ریسرچ، پشاور یونیورسٹی سے بی-ایڈ کی گرڈی حاصل کی۔ وہ گذشتہ اٹھارہ سال سے کمپیوٹر سائنس کے معلم کے طور پر خدمات انجام دے رہے ہیں۔ وہ ثانوی سے اعلیٰ سطح تک تعلیم کا وسیع تجربہ رکھتے ہیں۔ وہ کمپیوٹر سائنس کے تعلیمی میدان میں ہر سطح پر فروغ کے لیے کئی قومی اور بین الاقوامی سیمیناروں اور ورکشاپس میں شرکت کر چکے ہیں۔

وہ 2007 سے قومی نصاب مرتب کرنے والی ٹیم کے رکن ہیں۔ انہوں نے کمپیوٹر ایجوکیشن اور کمپیوٹر سائنس کا قومی نصاب مرتب کرنے میں بھرپور حصہ لیا ہے۔

ایک مصنف کی حیثیت سے وہ مختلف ٹیکسٹ بک بورڈز کے لئے گریڈ VI سے بارہویں جماعت تک کمپیوٹر سائنس کی بہت سی درسی کتابیں لکھ چکے ہیں۔

ان کے تجربے اور مہارت کو مد نظر رکھتے ہوئے، یہ کتاب طلباء اور اساتذہ دونوں کے لئے ایک اثاثہ ثابت ہوگی۔

خالد اسد گورنمنٹ ڈگری کالج و ڈیگہ، پشاور میں کمپیوٹر سائنس کے اسسٹنٹ پروفیسر ہیں۔ انہوں نے پشاور یونیورسٹی سے کمپیوٹر سائنس میں ایم ایس سی کی ڈگری حاصل کی اور ایم ایڈ کی ڈگری علامہ اقبال اوپن یونیورسٹی، اسلام آباد سے حاصل کی۔ کمپیوٹر سائنس کے میدان میں کالج کی سطح پر ان کا تدریسی تجربہ 13 سال ہے۔

وہ وزارتِ موسمیاتی تبدیلی، حکومت پاکستان میں بطور اسسٹنٹ ڈائریکٹر (آئی ٹی) خدمات انجام دے چکے ہیں۔ اس کے علاوہ ڈائریکٹوریٹ آف ہائر ایجوکیشن، خیبر پختونخوا میں ڈپٹی ڈائریکٹر (آئی ٹی) بھی رہے۔

ایک مصنف کی حیثیت سے وہ انٹرمیڈیٹ اور بیچلر لیول تک کمپیوٹر سائنس کی کتابیں لکھ چکے ہیں۔

NOT FOR SALE

Khyber Pakhtunkhwa Textbook Board

Phase – V, Hayatabad Peshawar

Phone: 091-9217714-15 Fax: 091-9217163, Website: www.kptbb.gov.pk

TEXTBOOK FEEDBACK FORM

S#	Statement	Remarks
1.	Overall the book is interesting and user friendly.	+32,767 سے -32768
2.	The language and content of the book is age / grade appropriate and the content is free of grammatical and punctuation errors.	+2,147,483,647 سے -2147,483648
3.	Content is supported with examples from real life / culture.	+65,535 سے 0
4.	Contents / texts are authentic and updated.	
5.	Pictures / diagrams / graphs / illustrations are informative, relevant and clear if not, then identify them.	
6.	Activities, projects and additional work is suggested for reinforcement of concepts.	
7.	Assessment achievements are thought provoking and comprise cognitive, psychomotor and effective skills.	
8.	The textbook is easy to be covered within academic year.	

Page No.	Observation/comments	Suggested amendment along with rationale
int	2	+32,767 سے -32,768
long	4	+2,147,483,647 سے -2147,483,648
[illegible]	2	+65,535 سے 0

General Remarks: ______________________

____

Name & Address: ______________________

Contact No. & Email: ______________________

"THE SEEKING OF KNOWLEDGE IS OBLIGATORY FOR EVERY MUSLIM"
(AL-TIRMIDHI # 74)

"Stop Corruption"
"Save the Nation."

قومی ترانہ

پاک سرزمین شاد باد کشورِ حسین شاد باد
تو نشانِ عزمِ عالی شان ارضِ پاکستان!
مرکزِ یقین شاد باد

پاک سرزمین کا نظام قوتِ اخوتِ عوام
قوم، ملک، سلطنت پائندہ تابندہ باد
شاد باد منزلِ مراد

پرچمِ ستارہ و ہلال رہبرِ ترقی و کمال
ترجمانِ ماضی شانِ حال جانِ استقبال!
سایۂ خدائے ذوالجلال

Leading Books Publishers
Jamrud Road, Peshawar

Printed by	Quantity	Code No.
Silver Classic Printing	11,865	PSP/SCP-12-49-2324(O)