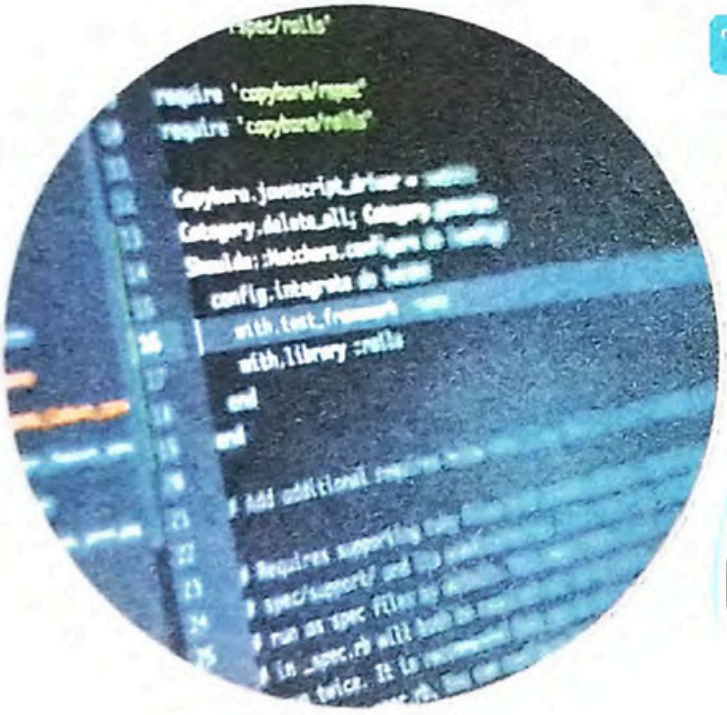
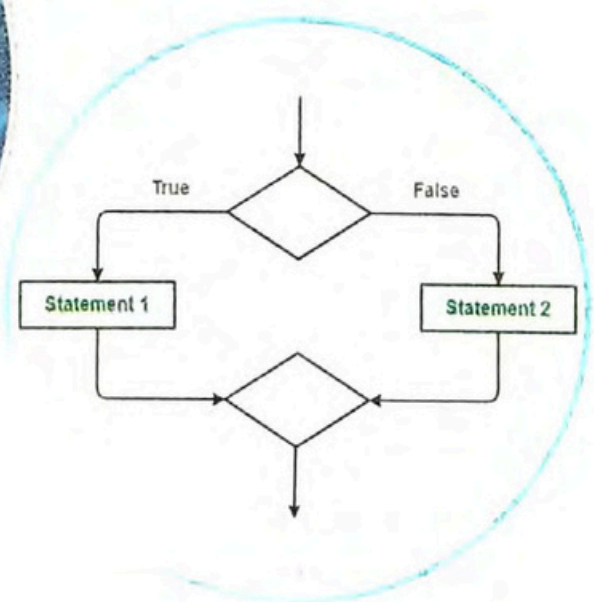
باب نمبر
4

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۔





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 چاہیے۔ 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

اوقات پروگرام کو کسی خاص حالت یا کنڈیشنل 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۔

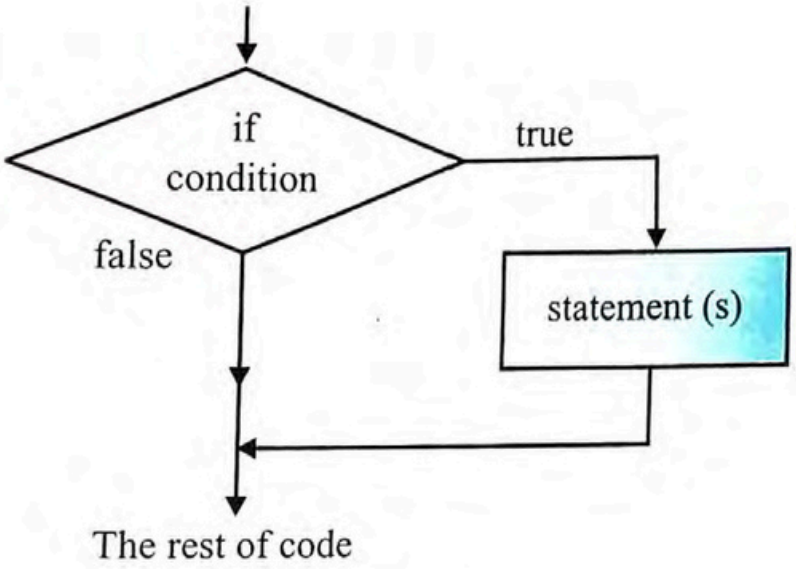
شرط (Relational Expression) کی شکل میں لکھی جاتی ہے۔ اگر شرط کا نتیجہ درست (True) ہے تو، اسٹیٹمنٹ یا اسٹیٹمنٹس کا مجموعہ جو کہ if اسٹیٹمنٹ کے بعد آتا ہے اسے چلا دیا جاتا ہے ورنہ نظر انداز کر دیا جاتا ہے۔ if اسٹیٹمنٹ کا سائنٹکس درج ذیل ہے:

```
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 فلوچارٹ if اسٹیٹمنٹ کے عمل کو ظاہر کرتا ہے۔



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 ایکلی 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 دو طرفہ سلیکشن اسٹیٹمنٹ بھی کہا جاتا ہے۔

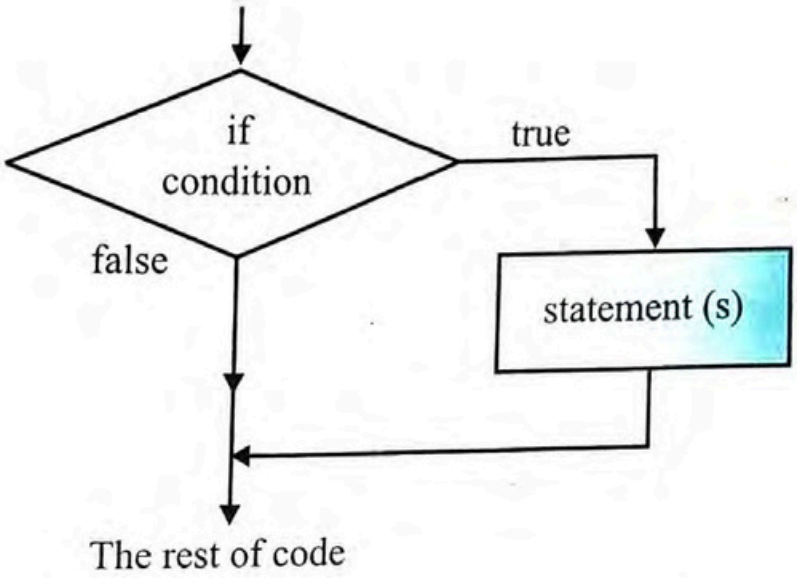
سائنٹکس:

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;
```



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");
}
```

```
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:

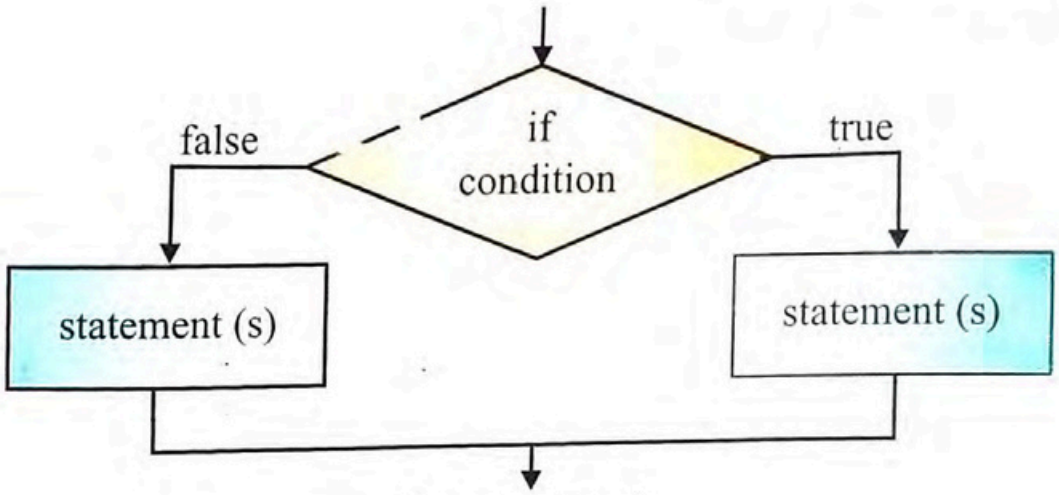
```
Syntax:

 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;
```

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

درج ذیل فلوچارٹ 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 پُٹ 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 پُٹ

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

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

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۔

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 پڑھی جائیں گی۔ switch کا سائنٹکس درج ذیل ہے:

سائنٹکس:

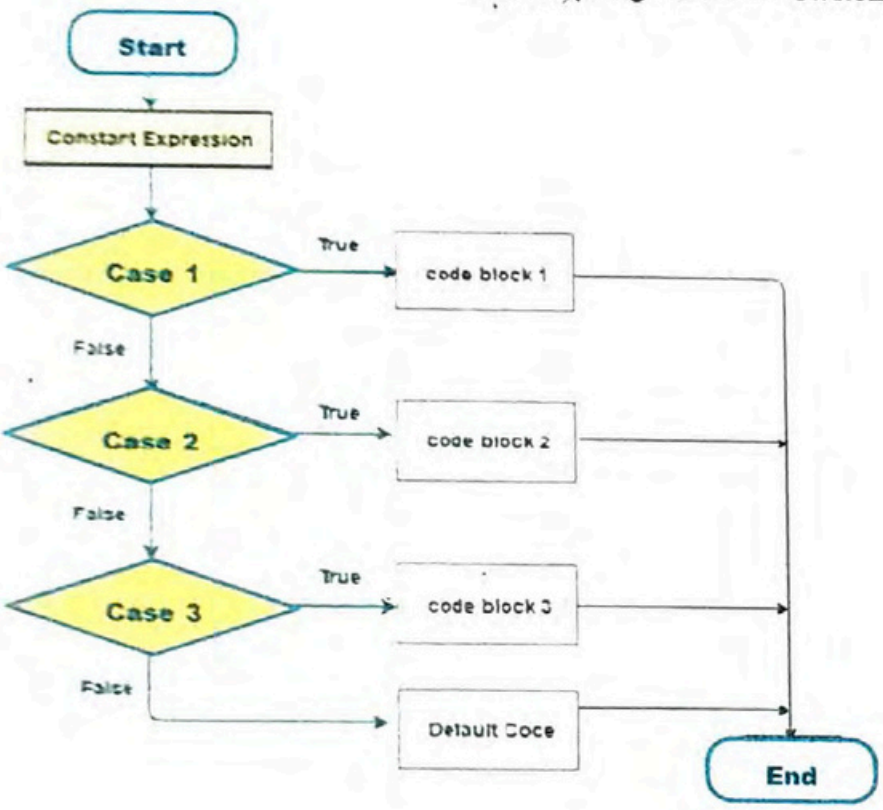
```
switch(expression)
{
        case const-1:
                statements;
                break;
        case const-2:
                statements;
                break;
        .
        .
        case const-n:
                statements;
                break;
        default:
        statements;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

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
```

```
        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 پر اسس زیادہ تر سلیکشن کنٹرول سٹرکچر میں استعمال ہوتے ہیں۔

سائنٹکس:

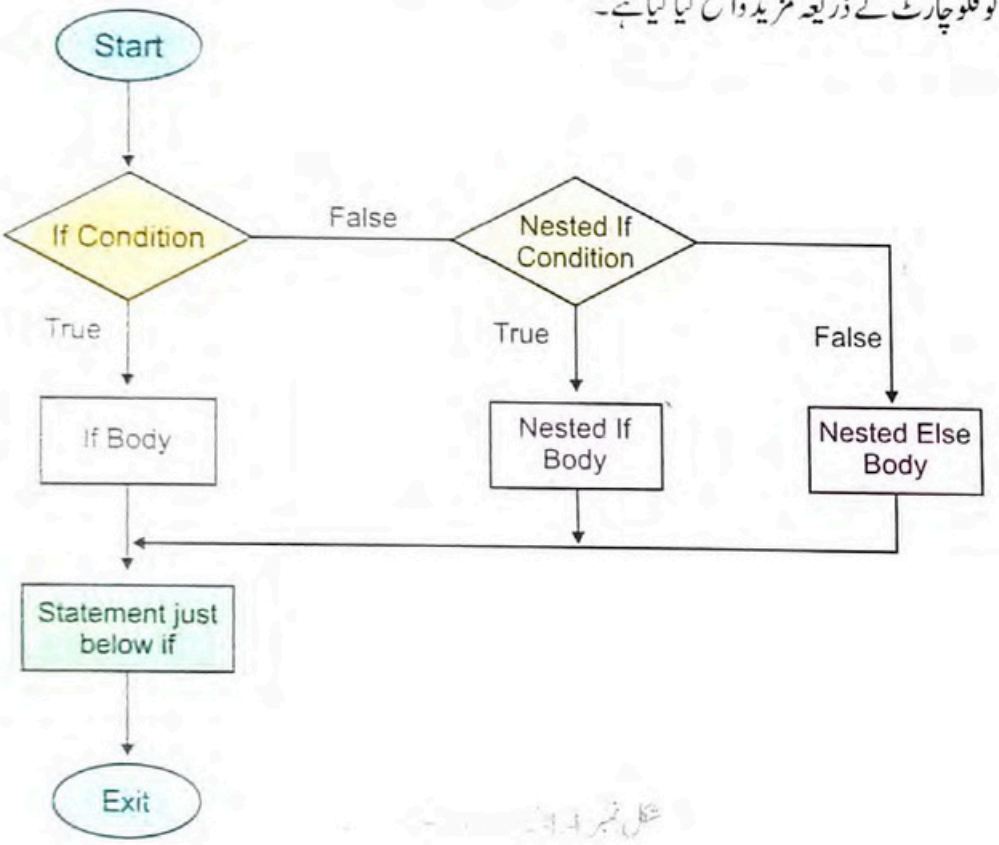
```
if (Condition1)
{
   if(Condition2)
   {
      Statement1;
   }
   else
   {
      Statement2;
   }
}
else
{
   if(Condition3)
   {
      Statement3;
   }
   else
   {
      Statement4;
   }
}
```

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

ہے 4 نہیں ہے

میں Statement 2 پر عملدرآمد کیا جائے گا۔ اسی طرح، اگر Condition 1 کا نتیجہ غلط ہے تو پھر 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 فلوچارٹ 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
```

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

مذکورہ پروگرام میں، جو عدد ان پٹ ہو گا 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 سیلیکشن 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 4 کے فلوچارٹس 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;
```

ان پٹ

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

```
    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 پٹ

```
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 پٹ 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 پٹ

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

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))
```

```
    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 ")
}
```