

- اس باب کے اختتام تک، آپ اس قابل ہو جائیں گے کہ:
- بنیادی پروگرامنگ کے تصورات کو سمجھ سکیں اور پانٹھن ڈیولپمنٹ انوائرنمنٹ سیٹ اپ کر سکیں۔
- بنیادی پانٹھن سینٹیکس اور ساخت لکھ اور سمجھ سکیں، جس میں ویری ایبلز، ڈیٹا ٹائپس، اور ان پٹ / آؤٹ پٹ آپریٹرز شامل ہیں۔
- پانٹھن میں مختلف آپریٹرز اور ایکسپریشنز، بشمول حسابی (arithmetic)، موازنہ (comparison)، اسائنمنٹ (assignment)، منطقی (logical) آپریٹرز، اور آپریٹرز کی ترجیح کا استعمال کر سکیں۔

انشائی طرز سوالات

مختصر کی نئی امتحانی تکنیکس (Knowledge, Understanding, Application, Analytical & Conceptual) کی دشمنی میں مرتب کیے گئے انشائی طرز سوالات

تعارف

سوال: آپ پانٹھن لینگوئج کے بارے میں کیا جانتے ہیں؟
جواب: پانٹھن پروگرامنگ کی دنیا میں خوش آمدید! پانٹھن ایک مقبول زبان ہے جو اپنی سادگی اور پڑھنے کی صلاحیت کے لیے جانی جاتی ہے، جو اسے ابتدائی اور پیشہ ور افراد دونوں کے لیے مثالی بناتی ہے۔ اس باب میں، ہم بنیادی باتوں سے آغاز کریں گے، جس میں آپ کا ڈیولپمنٹ ماحول ترتیب دینا اور بنیادی پروگرامنگ تصورات سیکھنا شامل ہے۔ اس باب کے آخر تک، آپ کو پانٹھن کی جامع سمجھ حاصل ہو جائے گی، جو آپ کو اپنے پروگرام لکھنے، جانچنے اور ڈیبگ کرنے کے قابل بنائے گی۔

پانٹھن پروگرامنگ کا تعارف

3.1

سوال 1: پانٹھن لینگوئج کیا ہے؟ پانٹھن کے اطلاق کے مختلف شعبوں کو بیان کریں۔ نیز مختلف لینگوئجز کے مقابلے میں پانٹھن کے فوائد بھی بیان کریں۔

جواب: تعریف:

پانٹھن ایک وسیع پیمانے پر استعمال ہونے والی اعلیٰ سطحی (high-level) پروگرامنگ زبان ہے جو اپنی سادگی اور پڑھنے کی صلاحیت کے لیے مشہور ہے۔
پانٹھن کے اطلاقات: یہ ایک درمیان میں زبان ہے اور مختلف شعبوں میں قابل اطلاق ہے، جس میں ویب ڈیولپمنٹ، ڈیٹا کا تجزیہ اور مصنوعی ذہانت وغیرہ شامل ہیں۔
پانٹھن کے فوائد: چاہے آپ ایک سادہ اسکرپٹ بنا رہے ہوں یا کوئی پیچیدہ سافٹ ویئر تیار کر رہے ہوں، پانٹھن کی صارف دوست نوعیت آپ کو تیزی اور مؤثر طریقے سے آغاز کرنے میں مدد دیتی ہے۔

سوال 2: کمپیوٹر پروگرامنگ کی تعریف کریں۔ نیز ایک مثال دیں۔

جواب: کمپیوٹر پروگرامنگ ہدایات کا ایک مجموعہ بنانے کا عمل ہے جو کمپیوٹر کو بتاتا ہے کہ کوئی کام کیسے انجام دینا ہے۔ یہ ہدایات ایک پروگرامنگ زبان میں لکھی جاتی ہیں جسے کمپیوٹر سمجھ سکتا ہے اور اس پر عمل کر سکتا ہے۔

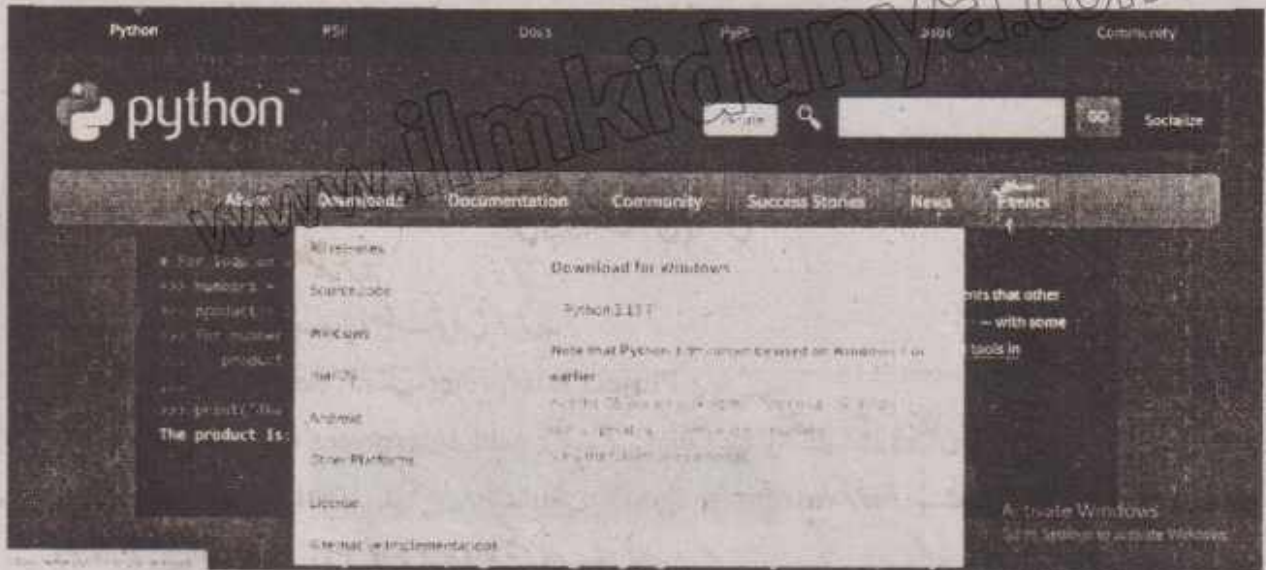
مثال: پلے اسٹور سے موبائل ایپ انسٹال کرنا۔ جب ہم کسی بھی موبائل ایپ پر انسٹال بٹن کلک کرتے ہیں تو ہم کمپیوٹر کو مخصوص کام کرنے کے لیے واضح اور درست ہدایات دیتے ہیں۔

نوٹ: پائٹھن انسٹال کرتے وقت، یقینی بنائیں کہ 'Add Python to PATH' والے باکس کو چیک کریں۔ اس سے کمانڈ لائن سے پائٹھن چلانا آسان ہو جاتا ہے۔ ہم پائٹھن پروگرام لکھنے اور چلانے کے لیے آن لائن سرور بھی استعمال کر سکتے ہیں۔

سوال 3: پائٹھن ڈیولپمنٹ انوائرنمنٹ ترتیب دینے سے کیا مراد ہے؟ نیز پائٹھن آئی ڈی ای (IDE) ترتیب دینے کا طریقہ کار بھی بیان کریں۔

جواب: ڈیولپمنٹ انوائرنمنٹ سے مراد کمپیوٹر کو پائٹھن کوڈ مؤثر طریقے سے لکھنے، چلانے اور ڈیبگ کرنے کے لیے تیار کرنے کا عمل ہے۔ اس میں ضروری سافٹ ویئر، ٹولز اور لائبریریز کو انسٹال اور کنفیگر کرنا شامل ہے جو ڈیولپمنٹ کو ہموار اور زیادہ موثر بناتے ہیں۔ ہم <https://www.python.org/> سے پائٹھن ڈاؤن لوڈ اور انسٹال کر سکتے ہیں۔

1- پائٹھن انسٹالیشن کے مراحل: <https://www.python.org/> پر جائیں۔ 'Downloads' سیکشن میں جائیں اور اپنے آپریٹنگ سسٹم (ونڈوز، مک، لینکس، یا یونیکس) کے لیے پائٹھن کا تازہ ترین ورژن مثلاً Python 3.13.xia1 پر کلک کریں۔



فصل 3.1: پائٹھن کا ہوم پیج

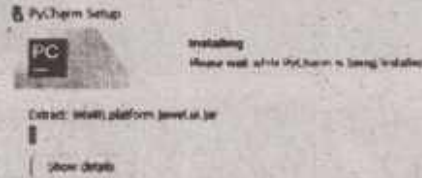
انسٹالر خود بخود ڈاؤن لوڈ ہو جائے گا۔

2- پائٹھن انسٹال کریں:

- ڈاؤن لوڈ کردہ .exe فائل کو چلائیں۔
- انسٹالیشن کے دوران 'Add Python to PATH' والے باکس کو چیک کریں۔
- 'Install Now' منتخب کریں یا اگر ضرورت ہو تو انسٹالیشن کو حسب ضرورت بنائیں (مثلاً انسٹالیشن ڈائریکٹری منتخب کریں)۔
- انسٹالیشن مکمل ہونے کا انتظار کریں، پھر 'Close' پر کلک کریں۔

3- آئی ڈی ای (IDE) سیٹ اپ کرنا:

- ایک مقبول IDE جیسے PyCharm، VS Code، یا IDLE (جو پائیتھن کے ساتھ شامل ہے) ڈاؤن لوڈ اور انسٹال کریں۔
- PyCharm پائیتھن ڈیولپمنٹ کے لیے ایک طاقتور IDE ہے، جو مفت کیونٹی ایڈیشن یا پیڈ پیشر ورانہ ایڈیشن میں دستیاب ہے۔ اپنے آپریٹنگ سسٹم (ونڈوز، میک او ایس، یا لینکس) کے لیے مناسب ورژن ڈاؤن لوڈ کرنے کے لیے <https://www.jetbrains.com/pycharm/> پر جائیں۔



فصل 3.3: انسٹالیشن پراس

4- IDE کو پائیتھن کے ساتھ کنفیگر کرنا:

- PyCharm کھولیں اور ایک نیا پروجیکٹ شروع کریں۔
- 'Configure' یا 'Settings' مینو میں، 'Project Interpreter' پر جائیں۔
- 'Add Interpreter' منتخب کریں، اور اپنے پائیتھن ایگزیکیوٹبل کا راستہ منتخب کریں مثلاً ونڈوز پر `C:\Python312\python.exe` یا لینکس / میک او ایس پر `/usr/bin/python3` اگر خود بخود نہ ملے تو اپنے انسٹالیشن فولڈر میں براؤز کریں۔
- سیکنگو کو لاگو کریں اور PyCharm کو انٹرپرائز کوآئرنیکس کرنے دیں۔



فصل 3.4: پائیتھن IDE

5- کمنٹیشن کی ٹیپنگ:

- اپنے IDE میں ایک نئی فائل بنائیں مثلاً test.py
 - ایک سادہ اسکرپٹ لکھیں: print("Hello, Python!")
 - اسکرپٹ کو چلائیں PyCharm/IDLE میں 'Run' بٹن استعمال کریں یا Code VSI میں ٹاسک سیٹ اپ کرنے کے بعد FS دبائیں۔
 - اگر آؤٹ پٹ ظاہر ہو جائے مثلاً "Hello, Python!" تو آپ کا IDE صحیح طریقے سے کنفیگ ہو گیا ہے۔
- سوال 4: گوگل کولیب (Google Colab) کیا ہے؟ نیز گوگل کولیب استعمال کرنے کا طریقہ بھی لکھیں۔

جواب: پائیتھن استعمال کرنے کے لیے ایک اور بہترین آپشن گوگل کولیب ہے۔ یہ ایک مفت کلاؤڈ بیسڈ پلیٹ فارم ہے جو جیو پیٹرنوٹ (Jupyter notebook) کا ماحول فراہم کرتا ہے جس میں پائیتھن اور NumPy, Pandas، جیسی مشہور لائبریریاں پہلے سے انسٹال شدہ ہوتی ہیں۔

گوگل کولیب استعمال کرنے کا طریقہ: شروع کرنے کے لیے، <https://colab.research.google.com/> پر جائیں، گوگل اکاؤنٹ سے سائن ان کریں، اور ایک نیا نوٹ بک بنائیں۔ کسی مقامی انسٹالیشن کی ضرورت نہیں ہے، جو اسے ابتدائی افراد یا محدود وسائل والی مشینوں پر کام کرنے والوں کے لیے مثالی بناتا ہے۔ آپ سبزیز میں پائیتھن کو ڈکھ اور چلا سکتے ہیں۔



فصل 3.5: گوگل کولیب IDE

بنیادی پائیتھن سٹینکس اور سٹرکچر

3.2

سوال 5: بنیادی پائیتھن سٹینکس (syntax) اور سٹرکچر (structure) بیان کریں؟

جواب: پائیتھن لینگویج اپنی سادگی اور پڑھنے کی صلاحیت کی وجہ سے مقبول ہے۔ درج ذیل مثال ملاحظہ کریں: print("Hello! World")

وضاحت: اس مثال میں، پرنٹ (print) فنکشن کو ڈیبل کوئیشن مارکس میں موجود پیغام کو آؤٹ پٹ کرنے کے لیے استعمال کیا گیا ہے۔ یہ پائیتھن کے سیدھے سادے سٹینکس کو ظاہر کرتا ہے، جہاں پرنٹ جیسے فنکشنز اعمال انجام دینے کے لیے استعمال ہوتے ہیں، جیسے کہ متن ظاہر کرنا۔

سوال 6: پائیتھن کمنٹس (comments) بیان کریں؟ نیز اس کی ہر قسم کی مثال دیں۔

جواب: کمنٹس وہ لائنیں ہیں جنہیں پائیتھن انٹراپرائز نہیں سمجھتا۔ وہ پروگرامر کے لیے وضاحتیں یا نوٹس فراہم کرنے کے لیے استعمال ہوتے ہیں۔ پائیتھن میں دو قسم کے کمنٹس استعمال ہوتے ہیں:

- سنگل لائن کمنٹس (Single line comments)
- ملٹی لائن کمنٹس (Multi-line comments)

سنگلی لائن کمنٹس # علامت سے شروع ہوتے ہیں۔ اس قسم کا کمنٹ صرف ایک لائن میں وضاحتی ریمارکس شامل کرنے کے لیے استعمال ہوتا ہے۔
مثال:

```
print ("K2 is the second-highest mountain in the world")
#This line shows one message.
```

ملٹی لائن کمنٹس: ملٹی لائن کمنٹس "" علامت سے شروع ہوتے ہیں اور "" علامت پر ختم ہوتے ہیں۔ اس قسم کا کمنٹ اس وقت استعمال ہوتا ہے جب وضاحت ایک سے زیادہ لائنوں میں لکھی گئی ہو۔
مثال:

""

This is a comment.

""

```
print ("Edhi Foundation operates the world's largest volunteer
ambulance network.")
```

کیا آپ جانتے ہیں؟

ہم پائیتھون میں ملٹی لائن کمنٹس کے لیے "" (ٹرپل کوٹس) استعمال کر سکتے ہیں لیکن دراصل یہ ملٹی لائن سٹرنگ لٹریز ہوتے ہیں جو رن ٹائم پر نظر انداز کر دیے جاتے ہیں۔

سوال 7: پائیتھن میں ویری ایبل (variable) کی تعریف کریں؟ نیز ویری ایبل کا نام رکھنے کے قواعد بھی لکھیں۔
جواب: تعریف:

پروگرامنگ میں ویری ایبل کمپیوٹر کی میموری میں ایک ذخیرہ کنٹینر کا کام کرتا ہے۔ یہ ڈیٹا کو ذخیرہ کرنے کی اجازت دیتا ہے جسے بعد میں کوڈ میں بازیافت کیا جاسکتا ہے۔
ویری ایبل کیوں کہلاتا ہے: ویری ایبل کی قدر پروگرام کی عمل آوری کے دوران تبدیل ہو سکتی ہے، اسی لیے اسے ویری ایبل کہا جاتا ہے۔

مثال:

```
age = "Hello, dear students!"
print(age)
age = 71
print ("Azam lived for", age, "years")
age = 60
print ("Iqbal lived for", age, "years")
```

Output:

```
Azam lived for 71 years
Iqbal lived for 60 years
```

وضاحت: یہ مثال ظاہر کرتی ہے کہ کس طرح ویری ایبل 'age' کو اعظم اور اقبال کی عمروں کی نمائندگی کے لیے مختلف اقدار تفویض کی گئی ہیں، جو پھر آؤٹ پٹ کے حصے کے طور پر پرنٹ کی جاتی ہیں۔
 پائیتھن میں ویری ایبل نام رکھنے کے قواعد: پائیتھن میں ویری ایبل کے نام ورنج ذیل قواعد کے پابند ہوتے ہیں:
 شروع: نام کا آغاز ایک حرف (a-z, A-Z) یا انڈر اسکور (_) سے ہونا چاہیے۔
 دیگر حروف: دیگر حروف میں حروف، ہندسے (0-9)، یا انڈر اسکور (_) شامل ہو سکتے ہیں۔

وری ایبل کے نام: ویری ایبل کے نام کیس سینٹیو ہوتے ہیں، جس کا مطلب ہے کہ age اور Age دو مختلف ویری ایبل تصور کیے جاتے ہیں۔
 کی ورڈز: پائیتھن کی کی ورڈز (reserved keywords)، جیسے if, while, for وغیرہ، کو ویری ایبل کے نام کے طور پر استعمال نہیں کیا جاسکتا۔

کیا آپ جانتے ہیں؟

پائیتھن میں درج ذیل کچھ محفوظ الفاظ ہیں:

if, else, elif, for, while, break, continue, and, or, not, in, is, True, False, None.

ہم انہیں ویری ایبل کے نام کے طور پر استعمال نہیں کر سکتے۔

سوال 8: پائیتھن میں استعمال ہونے والی ویری ایبل کی اقسام بیان کریں۔

جواب: ویری ایبل کی اقسام:

پائیتھن میں، آپ مختلف اقسام کا ڈیٹا ذخیرہ کرنے کے لیے مختلف اقسام کے ویری ایبل بنا سکتے ہیں۔ یہاں ویری ایبل کی کچھ عام اقسام ہیں:
 انٹجر (int): یہ ڈیٹا ٹائپ مکمل اعداد کو ذخیرہ کرنے کے لیے استعمال ہوتی ہے۔

مثال: age = 17

فلوئیٹنگ پوائنٹ (float): یہ ڈیٹا ٹائپ اعشاریہ اعداد کو ذخیرہ کرتی ہے۔

مثال: Price = 19.99

سٹرنگ (str): یہ پائیتھن میں استعمال ہونے والی ڈیٹا ٹائپ ہے۔ یہ ڈیٹا ٹائپ متن کو ذخیرہ کرنے کے لیے استعمال ہوتی ہے۔

مثال: "Ali" = name

بولین (bool): یہ ڈیٹا ٹائپ بولین ویلیوز یعنی True یا False کو ذخیرہ کرنے کے لیے استعمال ہوتی ہے۔

مثال: is_student = True

سوال 9: پائیتھن میں استعمال ہونے والے ان پٹ اور آؤٹ پٹ فنکشنز بیان کریں۔

جواب: ان پٹ اور آؤٹ پٹ آپریشن آپ کو صارف کے ساتھ تعامل کرنے کی اجازت دیتے ہیں۔ آپ صارف سے ڈیٹا داخل کرنے کو کہہ سکتے ہیں (ان پٹ) اور صارف کو معلومات ظاہر کر سکتے ہیں (آؤٹ پٹ)۔

ان پٹ: ہم صارف کا ان پٹ حاصل کرنے کے لیے input() فنکشن استعمال کرتے ہیں۔ input() فنکشن اسکرین پر ایک پیغام ظاہر کرتا ہے اور صارف کے پچھناپ کے انشور کے ساتھ ڈیٹا کو ذخیرہ کرتا ہے۔ صارف کے داخل کردہ متن کو پچھرا ایک ویری ایبل میں ذخیرہ کیا جاتا ہے۔

مثال:

Code:

```
name = input("Enter your name: ")
print("Hello, " + name + "!")
```

Output:

Enter your name: Maryam Ashraf

Hello, Maryam Ashraf!

وضاحت: کوڈ کی یہ لائن صارف سے ان کا نام درج کرنے کو کہتی ہے اور اسے name میں ذخیرہ کرتی ہے۔
آؤٹ پٹ: اسکرین پر معلومات ظاہر کرنے کے لیے print() فنکشن استعمال کریں۔ print() فنکشن ایک یا زیادہ آرگیومنٹس لیتا ہے اور انہیں ظاہر کرتا ہے۔

مثال:

Code:

```
name = input("Enter your name: ")
```

```
print("Hello, " + name + " !")
```

Output:

Enter your name: Maryam Ashraf

Hello, Maryam Ashraf!

وضاحت: کوڈ کی یہ لائن ایک خوش آمدید کا پیغام ظاہر کرتی ہے جس میں صارف کا نام شامل ہوتا ہے۔

سوال 10: پانچھن میں انچر اور فلوٹ ان پٹ کو بیان کریں۔

جواب: انچر اور فلوٹ ان پٹس کو ہینڈل کرتا:

عدد دی ان پٹس کو ہینڈل کرنے کے لیے، آپ آپ int() یا float() فنکشنز استعمال کرتے ہیں تاکہ ان پٹ سٹرنگز کو انچر یا فلوٹنگ پوائنٹ نمبرز

میں تبدیل کیا جاسکے۔

انچر ان پٹس:

```
user_age = int(input("Enter your age: "))
```

```
print("Your age is:", user_age)
```

Output:

Enter your age: 16

Your age is: 16

فلوٹ ان پٹس:

```
user_height = float(input("Enter your height in meters: "))
```

```
print("Your height is", user_height, "meters")
```

Output:

Enter your height in meters: 1.5

Your height is 1.5 meters

کیا آپ جانتے ہیں؟

پانچھن میں انڈینیشن آپ کے کوڈ کی ساخت اور حدود (scope) کی تعریف کرتا ہے۔ غلط انڈینیشن آپ کے کوڈ میں خرابیوں کا سبب بن سکتی ہے۔

سوال 11: آپریٹرز اور ایکسپریشن کی تعریف کریں۔ نیز آپریٹرز کی اقسام تفصیل سے بیان کریں۔
 جواب: آپریٹرز: آپریٹرز وہ علامتیں ہیں جو آپریٹرز پر مختلف آپریٹیشنز کرنے کے لیے استعمال ہوتی ہیں۔
 ایکسپریشن: ایکسپریشن آپریٹرز، آپریٹرز اور ویریبلز کا ایک مجموعہ ہے جس کے نتیجے میں کوئی ویلیو حاصل ہوتی ہے۔
 آپریٹرز کی اقسام:

حسابی آپریٹرز (Arithmetic Operators): حسابی آپریٹرز بنیادی ریاضیاتی عمل جیسے جمع، تفریق، ضرب، تقسیم، ماڈیولس، ایکسپونینشن، اور فلور ڈویژن کو انجام دینے کے لیے استعمال ہوتے ہیں۔

مثال:

تقابلی آپریٹرز (Comparison Operators): تقابلی آپریٹرز دو ویریبلز یا ایکسپریشنز کا موازنہ کرنے کے لیے استعمال ہوتے ہیں۔ وہ ان کے درمیان تعلق کو متعین کرتے ہیں، جیسے مساوات، عدم مساوات، بڑا ہے، چھوٹا ہے، وغیرہ۔ یہ آپریٹرز موازنہ کے نتیجے کی بنیاد پر بولین ویلیو True یا False لوٹاتے ہیں۔

مثال:

Define variables

a = 10

b = 3

print(a, "+", b, "=", a + b) # Output: 10 + 3 = 13
 print(a, "*", b, "=", a * b) # Output: 10 * 3 = 30
 print(a, "/", b, "=", a / b) # Output: 10 / 3 = 3.3333333333333335
 print(a, "//", b, "=", a // b) # Floor division # Output: 10 // 3 = 3
 print(a, "%", b, "=", a % b) # Modulus # Output: 10 % 3 = 1
 print(a, "**", b, "=", a ** b) # ** represents power operator # Output: 10 ** 3 = 1000

وضاحت: مندرجہ بالا کوڈ دو متغیرات x اور y کی تعریف کر کے اور ان کا مختلف آپریٹرز جیسے <, <=, >, >=, !=, == استعمال کرتے ہوئے موازنہ کر کے پائنتھون کے مقابلہ آپریٹرز کو ظاہر کرتا ہے۔

اسائنمنٹ آپریٹر (Assignment Operator): اسائنمنٹ آپریٹرز متغیرات کو ویریبلز تفویض کرنے کے لیے استعمال ہوتے ہیں۔ سب سے عام اسائنمنٹ آپریٹر مساوی کی علامت (=) ہے، جو دائیں جانب کی ویریبل کو بائیں جانب کے متغیر کو تفویض کرتا ہے۔

مُرکب اسائنمنٹ آپریٹر (Compound Assignment Operator): مُرکب اسائنمنٹ آپریٹرز بھی ہوتے ہیں جیسے +=, -=, *=, /= اور **=، جو ریاضیاتی عمل کو تفویض کے ساتھ یکجا کرتے ہیں۔

مثال:

Define initial values

a = 10

b = 5

Assignment

x = a; print("a = ", x) # Output: a = 10

Addition assignment

a += b; print("a after addition = ", a) # Output: a = 15

Subtraction assignment

a -= b; print("a after subtraction = ", a)

Output: a = 10

Multiplication assignment

a *= b; print("a after multiplication = ", a)

Output: a = 50

Division assignment

a /= b; print("a after division = ", a)

Output: a = 10.0

Floor division assignment

a //= b; print("a after floor division = ", a)

Output: a = 2.0

Modulus assignment

a %= b; print("a after modulus = ", a)

Output: a = 2.0

Exponentiation assignment

a **= b; print("a after exponentiation = ", a)

Output: a = 32.0

وضاحت: مندرجہ بالا کوڈ پانچوں میں مرکب تفویضی آپریٹرز کے استعمال کو ظاہر کرتا ہے۔ یہ متغیر 'a' کو ایک ویلیو تفویض کر کے شروع ہوتا ہے۔ پھر یہ مختلف تفویضی عملوں کو ظاہر کرتا ہے جن میں جمع، تفریق، ضرب، تقسیم، فلور ڈویژن، ماڈیولس، اور ایکسپونینشن شامل ہیں۔ ہر عمل 'a' کی ویلیو کو اپ ڈیٹ کرتا ہے اور نتائج کو اپ ڈیٹ شدہ ویلیوز کے ساتھ پرنٹ کیا جاتا ہے۔ کوڈ کمپائلر ہر عمل کا آؤٹ پٹ ظاہر کرتے ہیں۔

منطقی آپریٹرز (Logical Operators): منطقی آپریٹرز متعدد شرائط یا ایکسپریشنز کو یکجا کرنے کے لیے استعمال ہوتے ہیں۔ سب سے عام منطقی آپریٹرز 'and'، 'or' اور 'not' ہیں۔ وہ منطقی عمل کو انجام دینے اور ان میں شامل ایکسپریشنز کی تشخیص کی بنیاد پر بولین ویلیوز لوٹانے کے لیے استعمال ہوتے ہیں۔

مثال:

Define variables

x = True

y = False

Logical AND

logical_and = x and y

print(x, "and", y, "=", logical_and)

Output: True and False = False

Logical OR

logical_or = x or y

print(x, "or", y, "=", logical_or)

Output: True or False = True

Logical NOT

logical_not_x = not x

print("not", x, "=", logical_not_x)

Output: not True = False

logical_not_y = not y

print("not", y, "=", logical_not_y)

Output: not False = True

وضاحت: مندرجہ بالا کوڈ پانچوں میں منطقی آپریٹرز کے استعمال کو ظاہر کرتا ہے۔ یہ دو بولین متغیرات x اور y کی تعریف کرتا ہے اور ان پر 'and'، 'or' اور 'not' منطقی عمل انجام دیتا ہے۔ ان منطقی عملوں کے نتائج پرنٹ کیے جاتے ہیں، جو یہ ظاہر کرتے ہیں کہ ایکسپریشنز True ہیں یا False۔

سوال 12: پانچوں میں آپریٹر پر کی ترجیح (operator precedence) کی تعریف کریں۔ پانچوں میں آپریٹر پر کی ترجیح کی ترتیب بیان کریں۔ نیز مثال دیں۔

جواب: آپریٹر پریڈنس اس ترتیب کا تعین کرتا ہے جس میں کسی ایکسپریشن میں آپریٹرز انجام دیے جاتے ہیں۔ پانچوں میں آپریٹرز کے ساتھ ریاضی میں

بھی، بعض آپریٹرز کی پریسڈنس زیادہ ہوتی ہے اور ان کا اندازہ دوسروں سے پہلے کیا جاتا ہے۔ اس کو سمجھنا اس بات کو یقینی بنانے میں مدد کرتا ہے کہ آپ کے حسابات درست طریقے سے کیے جائیں۔

- 1- قوسین (): سب سے زیادہ پریسڈنس۔ قوسین کے اندر موجود آپریٹرز پہلے کیے جاتے ہیں۔
 $4 * (3 + 2)$ کی ویلیو 20 ہوتی ہے۔
- 2- ایکسپونینشن: ** پاور آپریٹرز اگلے نمبر پر کیے جاتے ہیں۔
 $2 + 2 * 2 ** 3 = 18$ (not 32)
- 3- ضرب *، تقسیم / اور ماڈیولس %: یہ آپریٹرز اگلے نمبر پر آتے ہیں، بائیں سے دائیں کی ترتیب سے انجام دیے جاتے ہیں۔
- 4- جمع + اور تفریق -: ان کی پریسڈنس ضرب اور تقسیم کے مقابلے میں کم ہوتی ہے۔

مثال:

$$4 * 3 + 10 / 2 - 11 \% 3$$

• $4 * 3$ کی ویلیو 12 ہے

• $10 / 2$ کی ویلیو 5.0 ہے

• $11 \% 3$ کی ویلیو 2 ہے

• پھر جمع اور تفریق کریں: $12 + 5.0 - 2 = 15.0$

مثال: آپریٹرز پریسڈنس کو ظاہر کرنے کے لیے درج ذیل کوڈ پر غور کریں

```
print("1. ARITHMETIC OPERATORS:")
print(f"3 + 2 * 5 = {3 + 2 * 5}") # Multiplication before addition
print(f"(3 + 2) * 5 = {(3 + 2) * 5}") # Parentheses change order
print(f"2 ** 3 * 4 = {2 ** 3 * 4}") # Exponentiation before multiplication
print(f"15 / 3 * 2 = {15 / 3 * 2}") # Division and multiplication (left to right)
print(f"15 // 4 + 2 = {15 // 4 + 2}") # Floor division before addition
print(f"11 // 4 + 2 = {11 // 4 + 2}") # Floor division before addition
print(f"17 % 5 * 2 = {17 % 5 * 2}") # Modulo before multiplication
```

Output:

== PYTHON OPERATOR PRECEDENCE DEMONSTRATION ==

1. ARITHMETIC OPERATORS:

$$3 + 2 * 5 = 13$$

$$(3 + 2) * 5 = 25$$

$$2 ** 3 * 4 = 32$$

$$15 / 3 * 2 = 10.0$$

$$15 // 4 + 2 = 5$$

$$11 // 4 + 2 = 4$$

$$17 \% 5 * 2 = 4$$

کیا آپ جانتے ہیں؟

پانچویں میں سٹرنگ کے اندر متغیرات اور ایکسپریشنز کو ایمبیڈ کرنے (سٹرنگ انٹرپولیشن) کے لیے f کو سٹرنگ سے پہلے استعمال کیا جاتا ہے۔

ایکسپونینشن (Exponentiation) دائیں سے بائیں ایسی ہی ہوتی ہے، یعنی جب متعدد ایکسپونینشن آپریٹرز ہوتے ہیں تو پائنتھون ان کا اندازہ دائیں سے بائیں کرتا ہے۔

معروضی طرز سوالات

اضافی کثیر الانتخابی سوالات

مختصر تعریفی نوعی انتخابی تفکیک (Knowledge, Understanding, Application, Analytical & Conceptual) کی روشنی میں مرتب کیے گئے کثیر الانتخابی سوالات

تعارف

- 1- پانچھن مندرجہ ذیل کی وجہ سے مقبول ہے: (A) سادگی (B) پڑھنے کی صلاحیت (C) آسانی سے سمجھنا (D) تمام بالا
- 2- پانچھن کوڈ میں مسائل تلاش کرنا کہلاتا ہے: (A) ڈیٹ (B) ڈیٹنگ (C) چینگ (D) ان میں سے کوئی نہیں

3.1 پانچھن پروگرامنگ کا تعارف

- 3- دوسری زبانوں کے مقابلے میں پانچھن ہے: (A) زیادہ پیچیدہ (B) ڈیٹنگ کرنے میں آسان (C) عملدرآمد میں تیز (D) موثر زبان
- 4- پانچھن آج کل زیادہ کامیاب کیوں ہے؟ (A) یہ سمجھنے میں آسان ہے (B) پیچیدہ مسائل کو حل کرنا مشکل (C) ورشائل اپیلی کیشنز (D) پیشہ ورانہ زبان
- 5- پانچھن پروگرامنگ میں مندرجہ ذیل میں سے کیا ہے؟ (A) پیچیدہ ڈھانچہ (B) پیچیدہ ڈھانچے کو سنبھالنے کی صلاحیت (C) ابتدائی زبان (D) پیشہ ورانہ زبان
- 6- ہدایات کا مجموعہ بنانے کا عمل: (A) سافٹ ویئر (B) پروگرامنگ (C) عملدرآمد (D) ڈیٹنگ
- 7- پانچھن کوڈ کو چلانے اور ڈیٹنگ کرنے کے لیے کمپیوٹر کو تیار کرنا کہلاتا ہے: (A) پانچھن چلانا (B) پانچھن ترتیب دینا (C) ترقیاتی ماحول (D) پانچھن ڈیٹنگ کرنا
- 8- سافٹ ویئر ٹول انشال کرنے کا مطلب ہے: (A) ضروری ٹولز کی ترتیب (B) ضروری ٹولز اور لائبریری کی ترتیب (C) ضروری لائبریری کی ترتیب (D) آپریٹنگ سسٹم لائبریری کی ترتیب
- 9- ترقیاتی ماحول کا کیا فائدہ ہے؟ (A) کوڈ چلانے اور بنانے کو موثر بنانا (B) ڈیٹنگ کو آسانی سے ڈیٹنگ کرنے میں مدد دینا (C) بڑے منصوبوں کو سنبھالنے میں مدد دینا (D) تمام بالا

- 10- پائتھن کو کماؤ لائن سے چلانے کی چیز سے آسان ہوتا ہے؟
 (A) سورس کوڈ (B) پائتھن کوڈ Path میں شامل کرنا
 (C) آجیکٹ کوڈ (D) پائتھن لائبریری میں شامل کرنا
- 11- پائتھن کا اب تک کا بہترین ورژن کون سا ہے؟
 Python 3.14x (D) Python 3.13x (C) Python 3.12x (B) Python 3.11x (A)
- 12- پائتھن کا تازہ ترین ورژن کس کے لیے مطابقت رکھتا ہے؟
 (A) ونڈوز (B) میک او ایس (C) لینکس (D) تمام
- 13- انسٹالیشن کے لیے صارف کو _____ فائل چلانی ہوتی ہے؟
 .dat (A) .exe (B) .doc (C) .setup (D)
- 14- انسٹالیشن ڈائریکٹری سے مراد ہے:
 (A) پائتھن لائبریری (B) پائتھن ڈائریکٹری
 (C) پائتھن سورس کوڈ محفوظ کرنے کا مقام (D) ان میں سے کوئی نہیں
- 15- مشہور پائتھن IDE ہے:
 PyCharm (A) VS Code (B) IDLE (C) تمام بالا (D)
- 16- پائتھن کے لیے مندرجہ ذیل میں سے کون سا پائتھن ورژن ڈائریکٹری ہے؟
 PyCharm (A) VS Code (B) IDLE (C) تمام بالا (D)
- 17- PyCharm میں انٹرپرائز شامل کرنے کے لیے کون سا آئیکن استعمال ہوتا ہے؟
 (A) نیو پروجیکٹ آئیکن (B) ریمر آئیکن (C) نیو اسکرپٹ آئیکن (D) ان میں سے کوئی نہیں
- 18- Python _____ مترجم استعمال کرتی ہے:
 (A) اسمبلر (B) کمپائلر (C) انٹرپرائز (D) تمام
- 19- PyCharm اور IDLE دونوں میں اسکرپٹ چلانے کے لیے کون سی شارٹ کلید استعمال ہوتی ہے؟
 F1 (A) F5 (B) F2 (C) F9 (D)
- 20- Colab Google ایک ہے:
 (A) سرچ انجن (B) کلاؤڈ بیسڈ پلیٹ فارم (C) ویب براؤزر (D) Python IDE
- 21- Colab Google ماحول فراہم کرتا ہے:
 (A) Matplotlib (B) IDE (C) Jupyter Notebook (D) ان میں سے کوئی نہیں
- 22- Matplotlib ایک ہے:
 (A) کمپائلر (B) IDE (C) انٹرپرائز (D) بلٹ ان لائبریری
- 23- پائتھن میں print() ایک ہے:
 (A) فنکشن (B) کی ورڈ (C) متغیر (D) لائبریری

24- print ("") اسٹینٹ میں ڈبل کوٹس کے اندر لکھی گئی چیز کہلاتی ہے:

(D) سٹیکس

(C) پیراگراف

(B) سٹرنگ

(A) پیغام

25- سنگل لائن کمنٹس کی علامت کون سی ہے؟

//

(D)

(B) "

(A) #

26- کس قسم کے کمنٹس کو بند کرنے کی ضرورت ہوتی ہے؟

(D) کوئی نہیں

(C) دونوں

(B) ملٹی لائن

(A) سنگل لائن

27- ملٹی لائن کمنٹ کو یہ بھی کہتے ہیں:

(D) ان میں سے کوئی نہیں

(C) ملٹی لائن کمانڈ

(B) ملٹی لائن سٹرنگ لٹریچر

(A) ملٹی لائن سٹرنگ

28- مندرجہ ذیل میں سے کون سا متغیر نہیں ہے؟

(D) Air

(C) Var

(B) 11variable

(A) variable11

29- ایک متغیر میں حروف تہجی، ہندسے اور _____ آ سکتے ہیں:

(D) انڈر سکور

(C) رپورٹس

(B) کریکٹرز

(A) علامت

30- پاکھن پروگرامنگ میں Age اور age ہیں:

(D) ان کا ڈیٹا ٹائپ مختلف ہے

(B) دو ایک جیسے متغیرات

(A) دو مختلف متغیرات

31- ان پٹ فنکشن میں، مگر متغیر کا ڈیٹا ٹائپ نہیں بتایا جاتا، تو پاکھن اسے _____ ٹائپ سمجھتا ہے؟

(D) bool

(C) string

(B) float

(A) int

32- مندرجہ ذیل میں سے کون سا آؤٹ پٹ فنکشن ہے؟

(D) float()

(C) int()

(B) input()

(A) print()

33- عددی ان پٹ کو اینڈل کرنے کے لیے ہم استعمال کرتے ہیں:

(D) ان میں سے کوئی نہیں

(C) a,b دونوں

(B) float()

(A) int()

آپریٹرز اور ایکسپریشنز

3.3

34- 10% 3 کا آؤٹ پٹ بتائیں:

(D) 10

(C) 1

(B) 3

(A) 0

35- اینکرل ڈویژن کی علامت:

(D) **

(C) %

(B) //

(A) /

36- مندرجہ ذیل میں سے کون سا پاور آپریٹر ہے؟

(D) **

(C) %

(B) //

(A) /

37- تقابلی آپریٹر جواب دیتا ہے:

(D) بولین فارم میں

(C) فلوئڈ پوائنٹ میں

(B) ایکسپونینشل فارم میں

(A) انٹر فارم میں

38- تقابلی آپریٹر میں = کہلاتا ہے:

(D) بڑا برابر

(C) چھوٹا برابر

(B) برابر

(A) تا برابر

39- مندرجہ ذیل میں سے کون سا سائنس آپریٹر ہے؟

>=

(D) = (C)

(B)

=! (A)

40- اگر $a=10$, $b=3$ اور $b=a$ کا آرگٹ پٹ کیا ہے؟

100000 (D)

10000 (C)

1000 (B)

100 (A)

41- لاجیکل آپریٹرز استعمال ہوتے ہیں:

(B) نتائج کا تجزیہ کرنے کے لیے

(A) ایکپریشن حل کرنے کے لیے

(C) متعدد ایکپریشنز کو یکجا کرنے کے لیے

(D) ان میں سے کوئی نہیں

42- پائتھن میں لاجیکل - اینڈ آپریٹر کی علامت ہے:

&& (D)

not (C)

or (B)

and (A)

43- پائتھن میں لاجیکل - اور کی علامت ہے:

|| (D)

or (C)

&& (B)

and (A)

44- مندرجہ ذیل میں سے کس کی اہمیت سب سے زیادہ ہے؟

% (D)

/ (C)

* (B)

** (A)

45- مندرجہ ذیل میں سے کون سا سائنس ایسوی ایٹ ہے؟

** (D)

+

(B)

* (A)

جوابات :-

B -10	D -9	B -8	C -7	B -6	B -5	C -4	D -3	A -2	D -1
B -20	B -10	C -18	B -17	A -16	D -15	C -14	B -13	D -12	C -11
A -30	D -11	B -28	B -27	B -26	A -25	B -24	A -23	D -22	C -21
B -41	C -39	A -38	D -37	D -36	B -35	C -34	C -33	A -32	C -31
					D -45	A -44	C -43	A -42	C -41

اضافی مختصر سوالات

محکمہ تعلیم کی نئی امتحانی تکنیکس (Knowledge, Understanding, Application, Analytical & Conceptual) کی روشنی میں مرتب کیے گئے مختصر جوابی سوالات

تعارف

1- پائتھن کی تعریف کریں۔

جواب: پائتھن ایک مقبول لینگویج ہے جو اپنی سادگی اور پڑھنے میں آسانی کے لیے جانی جاتی ہے، جو اسے ابتدائی اور پیشہ ورانہ افراد دونوں کے لیے مثالی بناتی ہے۔

پائتھن پروگرامنگ کا تعارف

3.1

2- پائتھن کے استعمالات لکھیں۔

جواب: یہ ایک در سائل زبان ہے اور مختلف شعبوں میں قابل اطلاق ہے، بشمول:

3- پائتھن کے فوائد کیا ہیں؟

جواب: چاہے آپ ایک سادہ اسکرپٹ بنارہے ہوں یا ایک پیچیدہ سافٹ ویئر تیار کر رہے ہوں، پائتھن کی صارف دوست نوعیت آپ کو جلدی اور مؤثر طریقے سے شروع کرنے میں مدد دیتی ہے۔

4- کمپیوٹر پروگرامنگ کی تعریف کریں۔

جواب: کمپیوٹر پروگرامنگ ہدایات کا ایک مجموعہ بنانے کا عمل ہے جو کمپیوٹر کو بتاتا ہے کہ کوئی کام کیسے انجام دینا ہے۔

مثال: ہلے اسٹور سے موبائل ایپ انسٹال کرنا۔

5- پائتھن انسٹال کرتے وقت کون سا آپشن چیک کرنا ضروری ہے اور کیوں؟

جواب: پائتھن انسٹال کرتے وقت، اس باکس کو چیک کرنا یقینی بنائیں جس پر لکھا ہو "Add Python to PATH"۔ یہ کمانڈ لائن سے پائتھن چلانے کو آسان بناتا ہے۔ ہم پائتھن پروگرام لکھنے اور چلانے کے لیے آن لائن خدمات بھی استعمال کر سکتے ہیں۔

6- پائتھن انسٹال کرنے کا طریقہ کار لکھیں۔

جواب: پائتھن انسٹال کرنے کا طریقہ کار درج ذیل ہے:

• ڈاؤن لوڈ کر دہرہ.exe فائل کو چلائیں۔

• انسٹالیشن کے دوران "Add Python to PATH" باکس کو چیک کریں۔

• "Install Now" کو منتخب کریں یا اگر ضرورت ہو تو انسٹالیشن کو حسب ضرورت بنائیں (مثلاً، انسٹالیشن ڈائریکٹری کا انتخاب کریں)۔

• انسٹالیشن مکمل ہونے کا انتظار کریں، پھر 'Close' پر کلک کریں۔

7- گوگل کو ایب کی تعریف کریں۔

جواب: یہ ایک مفت کلاؤڈ میڈ پلیٹ فارم ہے جو ایک جو پیرنوٹ بک ماحول فراہم کرتا ہے جس میں پائتھن اور مقبول لائبریریاں جیسے Numpy، pandas اور matplotlib پہلے سے انسٹال شدہ ہوتی ہیں۔

8- گوگل کو ایب آن لائن کیسے استعمال کیا جاتا ہے؟

جواب: شروع کرنے کے لیے، <https://colab.research.google.com/> پر جائیں، گوگل اکاؤنٹ سے سائن ان کریں، اور ایک نیا نوٹ بک بنائیں۔ کوئی مقامی انسٹالیشن درکار نہیں ہے، جو اسے ابتدائی افراد یا محدود وسائل والی مشینوں پر کام کرنے والوں کے لیے مثالی بناتا ہے۔ آپ سبزیز میں پانی تھون کو ڈلکھ اور اس پر عمل درآمد کر سکتے ہیں۔

بنیادی پائتھن سٹیمیکس اور سٹرکچر

3.2

9- بنیادی پائتھن سٹیمیکس اور سٹرکچر کی وضاحت کریں۔

جواب: پائتھن زبان اپنی سادگی اور پڑھنے میں آسانی کی وجہ سے مقبول ہے۔ درج ذیل مثال دیکھیں: `print("Hello! World")` وضاحت: اس مثال میں، پرنٹ فنکشن کو ڈیٹا کوئیشن مارکس میں بند پیغام کو آؤٹ پٹ کرنے کے لیے استعمال کیا گیا ہے۔ یہ پائتھن کے سیدھے سادے سٹیمیکس کو ظاہر کرتا ہے، جہاں پرنٹ جسے فنکشن کو اعمال انجام دینے کے لیے استعمال کیا جاتا ہے، جیسے کہ متن کی نمائش کرتا۔

10- سٹنگل لائن کمنٹ کی مثال دے کر تعریف کریں۔

جواب: سٹنگل لائن کمنٹس # علامت سے شروع ہوتے ہیں۔ اس قسم کا کمنٹ صرف ایک لائن میں وضاحتی ریمارکس شامل کرنے کے لیے استعمال ہوتا ہے۔

مثال:

```
print ("K2 is the second-highest mountain in the world")  
#This line shows one message.
```

11- ملٹی لائن کمنٹ کی مثال دے کر تعریف کریں۔

جواب: ملٹی لائن کمنٹس "" علامت سے شروع ہوتے ہیں اور "" علامت پر ختم بھی ہوتے ہیں۔ اس قسم کا کمنٹ اس وقت استعمال ہوتا ہے جب وضاحت ایک سے زیادہ لائنوں میں لکھی گئی ہو۔

مثال:

""

یہ ایک کمنٹ ہے۔

""

```
print ("Edhi Foundation operates the world's largest volunteer  
ambulance network.")
```

12- ملٹی لائن کمنٹ کو بند کرنا کیوں ضروری ہے؟

جواب: ملٹی لائن کمنٹ کو بند کرنا ضروری ہے کیونکہ اگر ہم اسے بند نہیں کرتے تو پائی تھون ٹرانسلیٹر نیچے کی تمام لائنوں کو بطور کمنٹ شامل کر لیتا ہے اور یہ لائنیں عمل میں نہیں آتیں جس کا مطلب ہے کہ آؤٹ پٹ آپ کی مطلوبہ شکل میں ظاہر نہیں ہوگا۔

13- پروگرامنگ میں ویری ایبل کیوں استعمال کیا جاتا ہے؟ ایک مثال دیں۔

جواب: پروگرامنگ میں ویری ایبل کمپیوٹر کی میموری میں ایک اسٹوریج کنٹینر کا کام کرتا ہے۔ یہ ڈیٹا کو ذخیرہ کرنے کی اجازت دیتا ہے جسے بعد میں کوڈ میں بازیافت کیا جاسکتا ہے۔

مثال:

```
age = "Hello, dear students!"  
print(age)  
age = 71  
print ("Azam lived for", age, "years")
```

14- ویری ایبل کا نام رکھنے کے لیے کوئی سے دو اصول لکھیں۔

جواب: پانچھن میں ویری ایبل کے نام درج ذیل اصولوں کی پابندی کرتے ہیں:

- نام کا آغاز حرف (a-z, A-Z) یا انڈر سکور (_) سے ہونا چاہیے۔
- ویری ایبل کے نام کیس سنسیٹو ہوتے ہیں، یعنی age اور Age دو مختلف ویری ایبل تصور کیے جائیں گے۔
- 15- ویری ایبل کی کوئی سے دو قسمیں لکھیں۔

جواب:

• انٹیجر (int): یہ ڈیٹا ٹائپ پورے نمبر کو ذخیرہ کرنے کے لیے استعمال ہوتا ہے۔

مثال: age = 17

• فلوئٹنگ پوائنٹ (float): یہ ڈیٹا ٹائپ اعشاریہ نمبر کو ذخیرہ کرتا ہے۔

مثال: Price = 19.99

- 16- ان پٹ اور آؤٹ پٹ فنکشنز کے مقصد کی وضاحت کریں۔
 جواب: ان پٹ اور آؤٹ پٹ آپریٹرز آپ کو صارف کے ساتھ تعامل کرنے کی اجازت دیتے ہیں۔ آپ صارف سے ڈیٹا داخل کرنے کے لیے کہہ سکتے ہیں (ان پٹ) اور صارف کو معلومات ظاہر کر سکتے ہیں (آؤٹ پٹ)۔
 17- انیچر اور فلٹ ڈیٹا ٹائپ میں فرق کریں۔

جواب: انیچر ڈیٹا ٹائپ:

- پورے نمبرز (مثبت، منفی، یا صفر) کی نمائندگی کرتا ہے۔
- انیچر کی کوئی حد نہیں ہوتی (unlimited precision)
- فلور ڈویژن آپریٹر // استعمال کرنے سے انیچر واپس آتا ہے۔

فلٹ ڈیٹا ٹائپ:

- حقیقی اعداد کی نمائندگی کرتا ہے جنہیں اعشاریہ نقطہ کی ضرورت ہوتی ہے یا وہ ایکسپونینشل شکل میں ظاہر کیے جاتے ہیں۔
- فلٹس ڈبل پریجن (عام طور پر 64-bit) رکھتے ہیں۔
- فلٹ پر مشتمل کوئی بھی ریاضیاتی عمل عام طور پر فلٹ نتیجہ پیدا کرے گا۔

آپریٹرز اور ایکسپریشنز

3.3

- 18- آپریٹرز اور ایکسپریشن کی تعریف کریں۔
 جواب: آپریٹرز: آپریٹرز وہ علامات ہیں جو آپریٹرز پر مختلف کارروائیاں کرنے کے لیے استعمال ہوتی ہیں۔
 ایکسپریشن: ایک ایکسپریشن آپریٹرز، اوپرینڈز اور ویلیوز کا مجموعہ ہوتا ہے تاکہ نتائج پیدا کیے جاسکیں۔
 19- حسابی آپریٹرز کی تعریف کریں۔
 جواب: حسابی آپریٹرز بنیادی ریاضیاتی کارروائیوں جیسے جمع، تفریق، ضرب، تقسیم، ماڈیولس، ایکسپونینیشن، اور فلور ڈویژن کو انجام دینے کے لیے استعمال ہوتے ہیں۔
 20- فلور ڈویژن، ماڈیولس اور پاور آپریٹرز کی مثال لکھیں۔
 جواب:

$$a = 10$$

$$b = 3$$

print(a, "/", b, "=", a // b) # Floor division # Output: 10 // 3 = 3

print(a, "%", b, "=", a % b) # Modulus # Output: 10 % 3 = 1

print(a, "**", b, "=", a ** b) # ** represents power operator # Output: 10 ** 3 = 1000

- 21- کمپیوٹر آپریٹرز کی تعریف کریں۔
 جواب: کمپیوٹر آپریٹرز وہ ویلیوز یا ایکسپریشنز کا موازنہ کرنے کے لیے استعمال ہوتے ہیں۔ وہ ان کے درمیان تعلق کا تعین کرتے ہیں، جیسے برابری، عدم برابری، بڑا، چھوٹا، وغیرہ۔ آپریٹرز موازنہ کے نتیجے کی بنیاد پر بولین ویلیوز True یا False واپس کرتے ہیں۔

- 22- منطقی آپریٹرز کیوں استعمال کیے جاتے ہیں؟
 جواب: منطقی آپریٹرز متعدد شرائط یا ایکسپریشنز کو یکجا کرنے کے لیے استعمال ہوتے ہیں۔ مسابغے عام لاجیکل آپریٹرز 'and'، 'or'، 'not' ہیں۔ وہ لاجیکل کارروائیاں کرنے اور شامل ایکسپریشنز کی تشخیص کی بنیاد پر بولین ویلیوز واپس کرنے کے لیے استعمال ہوتے ہیں۔

23- کیا آپ اسائنمنٹ آپریٹر کیا ہے مثال دیں؟
جواب: کیا آپ اسائنمنٹ آپریٹر نہیں ہوتے ہیں جیسے $=$, $+=$, $-=$, $*=$, $/=$, $\% =$, $**=$ ، جو ریاضیاتی کارروائیوں کو اسائنمنٹ کے ساتھ ملاتے ہیں۔

مثال:

ابتدائی ویلیوز کی تعریف

$$a = 10$$

$$b = 5$$

$$a += b;$$

print("a after addition = ", a)

Output: a = 15

$$a -= b;$$

print("a after subtraction = ", a)

Output: a = 10

جمع اسائنمنٹ

تفریق اسائنمنٹ

24- آپریٹر پر کی ترجیح کی تعریف کریں۔

جواب: آپریٹر پر سیڈنس ایک ایکسپریشن میں کارروائیوں کے انجام پانے کی ترتیب کا تعین کرتا ہے۔ پائیتھن کے ساتھ ساتھ ریاضی میں بھی، بعض آپریٹرز کی پر سیڈنس زیادہ ہوتی ہے اور ان کا اندازہ دوسروں سے پہلے لگایا جاتا ہے۔ اس کو سمجھنا اس بات کو یقینی بنانے میں مدد کرتا ہے کہ آپ کے حسابات درست طریقے سے کیے گئے ہیں۔

$$25- 8 + 3 * 4 ** 2$$

جواب: مرحلہ وار:

$$1- 2 ** 2 = 4$$

$$8 + 3 * 4 - 4$$

$$2- 3 * 4 = 12$$

$$8 + 12 - 4$$

$$3- 8 + 12 = 20$$

$$20 - 4 = 16$$

$$20 - 4 = 16$$

نتیجہ: 16



حل مشقی سوالات

کثیر الانتخابی سوالات

i- درج ذیل میں سے کون سا مرحلہ بنیادی پروگرامنگ کے عمل کا حصہ نہیں ہے؟

(الف) کوڈ لکھنا

(ب) مرتب کرنا/ تشریح کرنا

(ج) عمل درآمد (Execute) کرنا

(د) غلطیوں کو نظر انداز کرنا

ii-

کماڈ لائن سے آسانی سے چلانے کے لیے پائیتھن انسٹال کرنے کا وقت آپ کو کیا کرنا چاہیے؟

(الف) Add Python to PATH کو آن چیک کریں

(ب) ایک مختلف IDE کا انتخاب کریں

(ج) Add Python to PATH کو چیک کریں

(د) صرف IDE انسٹال کریں

- iii- print(10/3) کا نتیجہ کیا ہوگا؟
 (الف) 3.333 (ب) 3 (ج) 1 (د) 4
- iv- $x = 123$ میں x کی ڈیٹا ٹائپ کیا ہے؟
 (الف) انیچر (ب) فلوٹ (ج) انٹگر (د) بولین
- v- پانچھن میں کس آپریٹر کی سب سے زیادہ ترجیح (highest precedence) ہے؟
 (الف) جمع (+) (ب) ضرب (*) (ج) قوت (**) (د) قوسین ()
- vi- $2 < 1$ and $3 > 5$ کا نتیجہ کیا ہے؟
 (الف) True (ب) False (ج) Error (د) None
- vii- $++$ آپریٹر کیا کرتا ہے؟
 (الف) دو نمبروں کو جمع کرتا ہے (ب) قدروں کا موازنہ کرتا ہے
 (ج) جمع کرتا ہے اور تفویض (assign) کرتا ہے (د) قدروں کو ضرب دیتا ہے
- viii- کون سا فنکشن سٹرنگ ان پٹ کو انیچر میں تبدیل کرتا ہے؟
 (الف) str() (ب) int() (ج) Float() (د) input()
- ix- درج ذیل کوڈ کا نتیجہ کیا ہے؟
`age = 25; print(" Age : ", age)`
 (الف) Age:25 (ب) 25 (ج) Age (د) Age
- x- پانچھن میں کمنٹس (comments) کے لیے کون سی علامت استعمال ہوتی ہے؟
 (الف) // (ب) /* (ج) # (د) -

جوابات :-

- i- ii- iii- iv- v- vi- vii- viii- ix- x- ج

مختصر سوالات

1- پروگرامنگ میں کمنٹس کیوں اہم ہیں؟

جواب: تبصرے اہم ہیں کیونکہ:

- وہ ہمارے کوڈ کی منطق کو دوسرے صارفین کو سمجھنے میں مدد دیتے ہیں۔
- وہ اسے لکھنے کے برسوں بعد بھی ہمارے اپنے کوڈ کو سمجھنے میں مدد دیتے ہیں۔

2- پانچھن میں متغیرات کی تین بنیادی اقسام پر بحث کریں۔

جواب: انیچر (int):

یہ ڈیٹا ٹائپ پورے نمبر کو ذخیرہ کرنے کے لیے استعمال ہوتی ہے۔

مثال: `age = 17`

فلوٹنگ پوائنٹ (float):

یہ ڈیٹا ٹائپ اعشاریہ نمبر کو ذخیرہ کرتی ہے۔

مثال: `Price = 19.99`

اسٹرنگ (str): یہ پائتھن میں استعمال ہونے والی ڈیٹا ٹائپ ہے۔ یہ ڈیٹا ٹائپ متن کو ذخیرہ کرتی ہے۔

مثال name = "Ali"

3- پائتھن میں integer اور float ڈیٹا اقسام کے درمیان فرق بیان کریں۔

جواب: انٹجر ڈیٹا ٹائپ:

• پورے نمبرز (مثبت، منفی، یا صفر) کو ظاہر کرتی ہے۔

• انٹجر کی کوئی حد نہیں ہوتی

• فلور ڈویژن آپریٹر // استعمال کرنے سے انٹجر ملتا ہے۔

فلوٹ ڈیٹا ٹائپ

• حقیقی نمبرز کو ظاہر کرتی ہے جنہیں اعشاریہ نقطہ کی ضرورت ہوتی ہے یا وہ کفائی شکل میں ظاہر کیے جاتے ہیں۔

• فلوٹس ڈبل پریسیژن عام طور پر 64-bit رکھتے ہیں۔

• فلوٹ پر مشتمل کوئی بھی ریاضیاتی عمل عام طور پر فلوٹ نتیجہ دے گا۔

4- منطقی آپریٹرز کیا ہیں اور تینوں کے نام بتائیں؟

جواب: لاجیکل آپریٹرز متعدد شرائط یا کچھ مشن کو یکجا کرنے کے لیے استعمال ہوتے ہیں۔ سب سے عام لاجیکل آپریٹرز 'and'، 'or'، اور 'not' ہیں۔ یہ لاجیکل آپریٹرز کرنے اور بولین واپس لانے کے لیے استعمال ہوتے ہیں۔

لاجیکل آپریٹرز کی اقسام:

• And آپریٹر (and)

• Or آپریٹر (or)

• Not آپریٹر (not)

5- آپ پائتھن میں صارف سے ان پٹ کیسے حاصل کرتے ہیں؟

جواب: صارف سے ان پٹ حاصل کرنے کے لیے ہم input() فنکشن استعمال کرتے ہیں۔ input() فنکشن اسکرین پر ایک پیغام دکھاتا ہے اور صارف کے کچھ ٹائپ کر کے انٹر دبانے کا انتظار کرتا ہے۔ صارف کی درج کردہ عبارت پھر ایک متغیر میں محفوظ ہو جاتی ہے۔

مثال:

کوڈ:

```
name = input("Enter your name: ")
```

```
print("Hello, " + name + "!")
```

Enter your name: Maryam Ashraf

Hello, Maryam Ashraf!

6- پائتھن میں print() سٹیٹمنٹ کا مقصد کیا ہے؟

جواب: اسکرین پر معلومات ظاہر کرنے کے لیے print() فنکشن استعمال کریں۔ print() فنکشن ایک یا زیادہ آرگومنٹس لیتا ہے اور انہیں ظاہر کرتا ہے۔

مثال:
کوڈ:

```
name = input("Enter your name: ")  
print("Hello, " + name + "!")
```

آؤٹ پٹ:

Enter your name: Maryam Ashraf

Hello, Maryam Ashraf!

7- آپریٹر پر کی ترجیح کیا ہے؟ ایک مثال دیں۔

جواب: آپریٹر پر کی ترجیح اس ترتیب کا تعین کرتا ہے جس میں کسی ایکسپریشن میں آپریٹرز کیسے جاتے ہیں۔ پانچھن کے ساتھ ساتھ ریاضی میں بھی، کچھ آپریٹرز کی ترجیح زیادہ ہوتی ہے اور ان کا اندازہ دوسروں سے پہلے کیا جاتا ہے۔

$$4 * 3 + 10 / 2 - 11 \% 3$$

4 * 3 کا اندازہ 12 لگتا ہے۔

10 / 2 کا اندازہ 5.0 لگتا ہے۔

11 % 3 کا اندازہ 2 لگتا ہے۔

$$12 + 5.0 - 2 = 15.0$$

8- پانچھن میں متغیرات کے نام رکھنے کے کوئی تین اہم اصول کیا ہیں؟

جواب: شروع کرنا: نام کا آغاز ایک حرف (a-z, A-Z) یا انڈر اسکور (_) سے ہونا چاہیے۔
بعد والے حروف: بعد والے حروف میں حروف، ہندسے (0-9)، یا انڈر اسکور (_) شامل ہو سکتے ہیں۔
متغیر کے نام: متغیر کے نام کیسے سنسیٹو ہوتے ہیں، یعنی Age اور age دو مختلف متغیرات تصور کیے جاتے ہیں۔

تفصیلی سوالات

1- پانچھن میں بنیادی ڈیٹا کی اقسام کیا ہیں؟ مثالوں کے ساتھ Booleans, strings, floats, integers کی وضاحت کریں۔

جواب: جواب کے لیے دیکھیے سوال نمبر 8

2- پانچھن میں حسابی آپریٹرز کیا ہیں۔ ہر آپریٹر کی مثالیں فراہم کریں اور وضاحت کریں کہ آپ انہیں کب استعمال کریں گے۔

جواب: جواب کے لیے دیکھیے سوال نمبر 11

3- بیان کریں کہ پانچھن میں صارفین سے ان پٹ کیسے حاصل کیا جائے اور آؤٹ پٹ کیسے دکھایا جائے۔ مثالوں کے ساتھ input() اور print() فنکشنز کی وضاحت کریں۔

جواب: جواب کے لیے دیکھیے سوال نمبر 9

4- وضاحت کریں کہ تقابلی اور منطقی آپریٹرز (conditional statements) میں مل کر کیسے کام کرتے ہیں۔ ایک عمل مثال فراہم کریں۔

جواب: جواب کے لیے دیکھیے سوال نمبر 11

5- پانچ نمبروں میں مشروطہ کیس کیس ہیں؟ مناسب سیٹنگس اور عملی مثالوں کے ساتھ else-if اور else,if کی وضاحت کریں۔

جواب: جواب کے لیے پونٹ نمبر 4 کا سوال نمبر 2 اور 3 دیکھیے۔
6- پانچ نمبروں میں ایک ایسا پروگرام لکھیں جو صارف سے ایک نام لے اور دوسرے نام کو خوش آمدید (greeting) کا پیغام دکھائے۔
جواب: کوڈ:

Program to take user's name and display a greeting message

Taking input from the user

```
name = input("Please enter your name: ")
```

Displaying the greeting message

```
print("Hello, " + name + "! Welcome to Python programming.")
```

7- پانچ نمبروں میں ایک ایسا پروگرام لکھیں جو صارف سے دو نمبر لے اور ان کا حاصل جمع دکھائے۔
جواب: کوڈ:

```
num1 = input("Enter first number: ")
```

```
num2 = input("Enter second number: ")
```

```
print("The sum is: " + num1 + num2)
```

7- پانچ نمبروں میں ایک ایسا پروگرام لکھیں جو صارف سے پانچ نمبر لے اور ان کا اوسط دکھائے۔
جواب: کوڈ:

Program to take five numbers from user and display their average

```
num1 = float(input("Enter first number: "))
```

```
num2 = float(input("Enter second number: "))
```

```
num3 = float(input("Enter third number: "))
```

```
num4 = float(input("Enter fourth number: "))
```

```
num5 = float(input("Enter fifth number: "))
```

```
sum = num1 + num2 + num3 + num4 + num5
```

```
print("sum:", sum)
```

8- درج ذیل ایکسپریشنز کو حل کریں:

a) $8+3*4-2**2$

b) $20\%7*3+10//3$

c) $2*3**2\%4+10-6/2$

جواب:

a) $8+3*4*2**2$

مرحلہ وار:

پہلے قوت: $2**2=4$

$$8+12=20$$

$$20-4=16$$

b) $20\%7*3+10//3$

نتیجہ: 16

مرحلہ وار:

ماڈیولو اور ضرب (یکساں ترجیح، بائیں سے دائیں):
 پہلا: $20\%7=6$ (جب 20 کو 7 سے تقسیم کیا جائے تو باقی)
 ایکسپریشن بن گئی: $6*3+10//3$

ضرب: $6*3=18$

ایکسپریشن بن گئی: $18+10//3$

فلور ڈویژن: $10//3=3$

ایکسپریشن بن گئی: $18+3$

جمع: $18+3=21$

نتیجہ: 21

c) $2*3**2\%4+10-6/2$

مرحلہ وار:

پہلے قوت: $3**2=9$

ایکسپریشن بن گئی: $2*9\%4+10-6/2$
 ضرب، ماڈیولو، تقسیم (یکساں ترجیح، بائیں سے دائیں):

پہلا: $2*9=18$

ایکسپریشن بن گئی: $18\%4+10-6/2$

اکلا: $18\%4=21$

ایکسپریشن بن گئی: $2+10-6/2$

اکلا: $6/2=3.0$ (فلوٹ نتیجہ)

ایکسپریشن بن گئی: $2+10-3.0$

جمع اور تفریق (بائیں سے دائیں):

$2+10=12$

$12-3.0=9.0$

نتیجہ: 9.0