یونٹ 4

پائتھن میں کنٹرول اسٹرکچرز

حاصلاتِ تعلم:

اس باب کے اختتام تک، آپ اس قابل ہو جائیں گے کہ:

- پائتھن میں کنٹرول اسٹرکچرز کو نافذ کر سکیں، جیسے کہ فیصلہ سازی کی سٹیٹمنٹس اور لوپس۔
- پائتھن کے ماڈیولز، فنکشنز، اور بلٹ ان ڈیٹا اسٹرکچرز جیسے کہ لسٹس (Lists) کے ساتھ کام کر سکیں۔
- لسٹس (Lists) پر مختلف آپریشنز پرفارم کر سکیں۔

انشائی طرز سوالات

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

تعارف

سوال: کنٹرول سٹرکچر کیا ہے؟ اس کی اہمیت بیان کریں۔

جواب: پروگرامنگ میں، ہمیں اکثر اپنے پروگرام کے بہاؤ کو مختلف شرائط کی بنیاد پر کنٹرول کرنے یا کچھ کاموں کو متعدد بار دہرانے کی ضرورت ہوتی ہے۔ یہ وہ مقام ہے جہاں کنٹرول سٹرکچرز کام آتے ہیں۔

کنٹرول سٹرکچرز کی اہمیت: کنٹرول سٹرکچرز اس لیے ضروری ہیں کیونکہ یہ ہمارے پروگراموں میں فیصلہ سازی کے عمل اور کاموں کی تکرار کو منظم کرنے میں ہماری مدد کرتے ہیں۔ کنٹرول سٹرکچرز کی دو اہم اقسام ہیں: فیصلہ سازی اور تکرار۔

4.1 فیصلہ سازی

سوال 1: فیصلہ سازی کی تعریف کریں؟ پائتھن میں استعمال ہونے والے فیصلہ جملوں کی اقسام کے نام بتائیں۔

جواب: پروگرامنگ میں فیصلہ سازی پروگرام کو شرائط کی بنیاد پر مختلف اقدامات منتخب کرنے کی اجازت دیتی ہے۔ یہ اسی طرح ہے جیسے ہم حقیقی زندگی میں فیصلے کرتے ہیں۔

مثال: اگر آپ بیک بٹن دبا کر کسی ایپلیکیشن (موبائل ایپ) سے باہر جانے کی کوشش کرتے ہیں، تو ایپ پوچھتی ہے؟

کیا آپ باہر جانا چاہتے ہیں: ہاں یا نہیں

یہ سافٹ ویئر کے ذریعے اپنائے جانے والے فیصلہ سازی کے ڈھانچے کو ظاہر کرتا ہے۔

شرط (Condition):

شرط ایک ایکسپریشن ہے جس میں آپریڈز ریشنل آپریٹرز یا لاجیکل آپریٹرز یا دونوں کا مجموعہ ہوتا ہے جو بولین شکل میں نتیجہ دیتا ہے، یا تو سچ ہو یا جھوٹ۔ تمام فیصلہ سازی کے جملے شرط پر مبنی ہوتے ہیں۔

فیصلہ جملوں کی اقسام:

- if statement

www.ilmkidunya.com

- if else statement

سوال 2: if statement کی اس کے سنٹیکس کے ساتھ تعریف کریں؟ if statement کی حدود بیان کریں۔

جواب: if statement شرائط کی بنیاد پر فیصلے کرتا ہے۔ اگر شرط سچ ہے، تو یہ کوڈ کا ایک بلاک چلاتا ہے۔

سنٹیکس:

```
if condition
    code to run if the condition is true
```

انڈینٹیشن:

پائیتھن میں انڈینٹیشن آپ کے کوڈ کے ڈھانچے یا اسکوپ کی وضاحت کرتی ہے۔ پائیتھن میں مناسب انڈینٹیشن لازمی ہے۔ غلط انڈینٹیشن آپ کے کوڈ میں خرابی (Indentation Error) کا سبب بن سکتی ہے۔

مثال 1:

```
Age = 21
if Age > 18:
    print("You are a teenager!!!!!!!")
```

مثال 2:

```
temperature = 20
if temperature > 25:
print("It's a hot day")
```

if statement کی حد:

if statement تب آؤٹ پٹ دکھاتا ہے جب مخصوص شرط سچ ہو، اگر شرط جھوٹی ہو تو اسکرین پر کچھ ظاہر نہیں ہوگا۔

سوال 3: if statement کی اس کے سنٹیکس کے ساتھ تعریف کریں؟ ایک مثال لکھیں۔

جواب: if-else statement ہمیں اجازت دیتا ہے کہ اگر شرط سچ ہو تو کوڈ کا ایک بلاک چلائیں اور اگر شرط جھوٹی ہو تو دوسرا بلاک چلائیں۔

if-else statement کا سنٹیکس:

```
if condition:
    code to run if the condition is true
else:
    code to run if the condition is false
```

مثال:

```
temperature = 15
if temperature > 30:
    print("It's a hot day")
else:
    print("It's not a hot day")
```

آؤٹ پٹ:

```
It's not a hot day
```

سوال 4: نیسٹڈ کنڈیشن کی اس کے سنٹیکس کے ساتھ تعریف کریں؟ ایک مثال لکھیں۔

جواب: بعض اوقات، ہمیں کسی شرط کے اندر متعدد شرائط کو چیک کرنے کی ضرورت ہوتی ہے۔ اسے نیسٹنگ کہتے ہیں۔

لہٰذا، if statement جس کے جسم میں دوسری if statement ہو، نیسٹڈ if کہلاتا ہے۔ یہ ڈھانچہ پیچیدہ متعدد شرائط کو لاگو کرنے کے لیے استعمال ہوتا ہے۔

نیسٹڈ if statement کا سنٹیکس:

```
if condition1:
    if condition2:
        code to run if both condition1 and condition2 are true
    else:
        code to run if condition1 is true but condition2 is false
else:
    code to run if condition1 is false
```

مثال: اگر موسم بارانی ہے اور درجہ حرارت 15 ڈگری سے کم ہے، تو ہم برساتی کوٹ پہنتے ہیں۔ اگر صرف بارانی ہے، تو ہم چھتری لیتے ہیں۔ اگر موسم بارانی نہیں ہے، تو ہم دن کا لطف اٹھاتے ہیں۔

Code:

```
weather = "rainy"
temperature = 10
if weather == "rainy":
    if temperature < 15:
        print("Wear a raincoat")
    else:
        print("Take an umbrella")
else:
    print("Enjoy your day!")
```

Output:

```
Wear a raincoat
```

وضاحت: اس مثال میں، کوڈ چیک کرتا ہے کہ آیا موسم بارانی ہے۔ اگر ہے، تو یہ پھر چیک کرتا ہے کہ آیا درجہ حرارت 15 ڈگری سے کم ہے۔ اگر دونوں شرائط سچ ہیں، تو یہ "Wear a raincoat" پرنٹ کرتا ہے۔ اگر موسم بارانی ہے لیکن درجہ حرارت 15 ڈگری سے کم نہیں ہے، تو یہ "Take an umbrella" پرنٹ کرتا ہے۔ اگر موسم بارانی نہیں ہے، تو یہ "Enjoy your day!" پرنٹ کرتا ہے۔

4.2 لوپنگ کنسٹرکٹ

سوال 5: لوپ کی تعریف کریں؟ وہائل لوپ کی سنٹیکس اور مثال کے ساتھ وضاحت کریں۔

جواب: اگر ہم کسی ایک سٹیٹمنٹ یا سٹیٹمنٹس کے سیٹ کو ایک سے زیادہ بار دہرانا چاہتے ہیں، تو ہم لوپ استعمال کرتے ہیں۔ لوپ کم سے کم ان پٹ کی مدد سے زیادہ سے زیادہ آؤٹ پٹ حاصل کرنے میں ہماری مدد کرتے ہیں۔ پائیتھن میں ہم دو قسم کے لوپ استعمال کرتے ہیں:

- while لوپ
- for لوپ

while لوپ:

while لوپ اس وقت تک چلتا ہے جب تک شرط سچ ہو۔ یہ ہر بار بار چلنے سے پہلے شرط کو چیک کرتا ہے اور جب شرط مزید سچ نہیں رہتی تو رک جاتا ہے۔ بنیادی طور پر، while لوپ استعمال کیا جاتا ہے جہاں ہمیں پہلے سے معلوم نہ ہو کہ ہمارا لوپ کتنی بار دہرایا جائے گا۔

www.ilmkidunya.com

while لوپ کا سنٹیکس:

```
while condition:
    code to run while the condition is true
```

مثال: ایک نمبر میں 1 اس وقت تک شامل کریں جب تک یہ 10 تک نہ پہنچ جائے۔

```
number = 1
while number < 10:
    print(number)
    number += 1
```

وضاحت: اس مثال میں، کوڈ نمبر 1 سے شروع ہوتا ہے۔ 'وہائل' لوپ چیک کرتا ہے کہ آیا نمبر 10 سے کم ہے۔ اگر ہے، تو پروگرام نمبر کی موجودہ قیمت پرنٹ کرتا ہے اور پھر اس میں 1 کا اضافہ کرتا ہے۔ یہ لوپ اس وقت تک جاری رہتا ہے جب تک نمبر 10 تک نہ پہنچ جائے، جس کے بعد لوپ رک جاتا ہے۔

سوال 6: for لوپ کی سنٹیکس اور مثال کے ساتھ تعریف اور وضاحت کریں۔

جواب: for لوپ کوڈ کے ایک بلاک کو ایک مخصوص تعداد میں دہراتا ہے۔ یہ عام طور پر کسی ترتیب (جیسے لسٹ، ٹیپل، یا سٹرنگ) پر iterate کرنے کے لیے استعمال ہوتا ہے۔

for لوپ کا سنٹیکس:

```
for variable in sequence:
    code to run for each element in the sequence
```

مثال: دوستوں کی فہرست میں ہر دوست کو 'اسلام علیکم' کہیں۔

```
friends = ["Sami", "Raza", "Moosa"]
for friend in friends:
    print("Welcome to ", friend)
```

آؤٹ پٹ:

```
Welcome to: Sami
Welcome to: Raza
Welcome to: Moosa
```

وضاحت: اس مثال میں، کوڈ فہرست میں موجود ہر دوست پر جاتا ہے اور ہر ایک کے لیے خوش آمدید کا پیغام پرنٹ کرتا ہے۔

سوال 7: Range فنکشن اور end سٹیٹمنٹ کی تعریف کریں۔ نیز مثالیں لکھیں۔

جواب: Range فنکشن: ہم range() فنکشن استعمال کر سکتے ہیں تاکہ اعداد کا تسلسل پیدا کیا جا سکے، جو اکثر فار لوپس میں استعمال ہوتا ہے۔

رینج فنکشن کا سنٹیکس:

```
range(start)
range(start, stop)
range(start, stop, step)
```

مثال: پہلے 5 پورے نمبر پرنٹ کریں۔

```
for i in range(5):
    print(i)
```

آؤٹ پٹ:

```
0
1
```

2
3
4

end سٹیٹمنٹ: پائتھن میں، اینڈ سٹیٹمنٹ کوئی علیحدہ کمانڈ نہیں ہے، بلکہ یہ ()print فنکشن کا ایک اختیاری پیرامیٹر ہے۔ ڈیفالٹ طور پر، پائتھن میں ہر ()print ایک نئی لائن کیریکٹر کے ساتھ ختم ہوتا ہے، لیکن آپ end پیرامیٹر استعمال کر کے تبدیل کر سکتے ہیں کہ آخر میں کیا پرنٹ ہو۔

مثال 1: 'end' سٹیٹمنٹ استعمال کرتے ہوئے دو سٹرنگز کو ایک ہی لائن پر پرنٹ کریں۔

```
print("Azhar Shah", end=" ")
print("Ahsan")
```

آؤٹ پٹ:

```
Azhar Shah Ahsan
print("Ayyan", end="***")
print("Naveed")
```

آؤٹ پٹ:

```
Ayyan***Naveed
```

مثال 2: 'end' سٹیٹمنٹ استعمال کرتے ہوئے 1 سے 5 تک کے نمبر ایک ہی لائن پر پرنٹ کریں۔

```
for i in range(1, 6):
        print(i, end=" ")
```

آؤٹ پٹ:

```
1 2 3 4 5
```

تشریح: اوپر والا کوڈ 0 سے 4 تک کے نمبر تیار کرتا ہے اور ہر نمبر پرنٹ کرتا ہے۔

کیا آپ جانتے ہیں؟

پائتھن میں سٹرنگز یا تو سنگل کوٹیشن 'مارکس یا ڈبل کوٹیشن' مارکس کے درمیان ہوتی ہیں۔ جیسے 'Ayyan' وہی ہے جو 'Ayyan' ہے۔

سرگرمی

1- ایک فار لوپ ()range کے ساتھ لکھیں جو 2 سے 10 تک کے زوج اعداد کو ایک ہی لائن پر پرنٹ کرے۔
2- ایک پائتھن پروگرام لکھیں جو 3 کے پہلے 10 ضربوں کو فار لوپ اور ()range فنکشن کے استعمال سے پرنٹ کرے۔

حل:

```
for i in range(2, 11, 2):
        print(i, end=" ")
```

آؤٹ پٹ:

```
2 4 6 8 10
```

حل:

```
for i in range(3, 31, 3):
        print(i, end=" ")
```

www.ilmkidunya.com

آؤٹ پٹ:

3 6 9 12 15 18 21 24 27 30

سوال 8: نیسٹڈ لوپ کی تعریف کریں؟ نیز مثالیں لکھیں۔

جواب: نیسٹڈ لوپ دوسرے لوپ کے اندر موجود لوپ ہوتے ہیں۔ بیرونی لوپ کے ہر بار چلنے کے لیے، اندرونی لوپ مکمل طور پر چلتا ہے۔ ہم نیسٹڈ لوپ استعمال کرتے ہیں جب ہمیں کاموں کے اندر دہرائے جانے والے کام انجام دینے ہوں۔

مثال 1: بنیادی نمبر پیٹرن کے لیے نیسٹڈ لوپ

```
    print("=== Basic Number Pattern ===")
for i in range(3):    # Outer loop
    for j in range(4):  # Inner loop
    print(f"i={i}, j={j}")
  print("---")    # Separator after each outer loop
```

آؤٹ پٹ:

```
=== Basic Number Pattern ===
i=0, j=0
i=0, j=1
i=0, j=2
i=0, j=3
---
i=1, j=0
i=1, j=1
i=1, j=2
i=1, j=3
---
i=2, j=0
i=2, j=1
i=2, j=2
i=2, j=3
---
```

مثال 2: مثلثی پیٹرن میں ستارے پرنٹ کرنے کے لیے نیسٹڈ لوپ

```
print("=== Right-Angled Triangle ====")
n = 5
for i in range(1, n+1):
  for j in range(i):    # Inner loop runs 'i' times
    print("*", end=" ")
  print()    # New line after each row
```

آؤٹ پٹ:

```
    *
    **
    ***
    ****
    *****
```

www.ilmkidunya.com

4.3 پائتھن میں لائبریریز

سوال 9: پائتھن میں ڈیٹا سٹرکچر اور لائبریری سے کیا مراد ہے؟ پائتھن میں استعمال ہونے والی مختلف لائبریریوں کا اطلاق کریں۔

جواب: ڈیٹا سٹرکچر:

ڈیٹا سٹرکچر سے مراد ڈیٹا کو منظم اور ذخیرہ کرنے کا ایک خاص فارمیٹ یا طریقہ ہے۔ پائیتھن میں لسٹ اور ٹیپل مشہور ڈیٹا سٹرکچر ہیں۔

لسٹ: پائیتھن میں، لائبریریاں ٹول باکسز کی طرح ہوتی ہیں جو مفید ٹولز سے بھری ہوتی ہیں، جو آپ کو مختلف مسائل حل کرنے میں مدد دیتی ہیں بغیر اس کے کہ آپ کو سب کچھ خود ہی بنانا پڑے۔ لائبریریاں پہلے سے تیار شدہ ٹول کٹس کی طرح ہوتی ہیں جنہیں آپ استعمال کر سکتے ہیں بغیر اس کے کہ آپ خود تمام کوڈ لکھیں۔

مثال: بے ترتیب نمبر جنریٹ کرنے کے لیے random لائبریری درآمد کریں۔

```
import random
number = random.randint(1, 10)
print("The random number is:", number)
```

آؤٹ پٹ:

```
The random number is: 3
```

مثال: شماریاتی حسابات کرنے کے لیے statistics لائبریری درآمد کریں۔

```
import statistics
data = [23, 45, 67, 89, 12, 44, 56]
mean_value = statistics.mean(data)
print("The mean value is:", mean_value)
```

آؤٹ پٹ:

```
The mean value is: 48
```

وضاحت: statistics لائبریری بنیادی شماریاتی حسابات کرنے کے لیے ایک بہترین ٹول ہے، جیسے کہ اعداد کے سیٹ کا اوسط، میڈین اور موڈ معلوم کرنا۔ یہ خاص طور پر ڈیٹا تجزیہ کے کاموں میں مفید ہو سکتی ہے۔

مختصراً، ان لائبریریوں کو استعمال کر کے، آپ وقت اور محنت بچا سکتے ہیں، جس سے آپ ہاتھ میں موجود مخصوص مسئلے کو حل کرنے پر توجہ مرکوز کر سکتے ہیں بجائے اس کے کہ پہیہ دوبارہ ایجاد کریں۔

> **سمارٹ ٹپس**
>
> لائبریریاں درآمد کرتے وقت، یقینی بنائیں کہ صرف وہی درآمد کریں جس کی آپ کو ضرورت ہے۔ آپ مزید لائبریریاں https://docs.python.org/3/ پر دریافت کر سکتے ہیں۔

4.4 پائتھن میں لسٹیں

سوال 10: لسٹ کی تعریف کریں؟ مثال کے ساتھ لسٹ بنانے، اس تک رسائی حاصل کرنے اور اس میں ترمیم کرنے کو بیان کریں۔

جواب: پائیتھن میں، لسٹ ایک ورسٹائل ڈیٹا سٹرکچر ہے جو اشیاء کے مجموعے کو محفوظ کر سکتی ہے۔ آپ لسٹ کو آسانی سے بنا سکتے ہیں، اس تک رسائی حاصل کر سکتے ہیں اور اس میں ترمیم کر سکتے ہیں۔

لسٹ بنانا: لسٹ کو مربع بریکٹ [] کے اندر اشیاء کو کوما سے الگ کر کے بنایا جاتا ہے۔ لسٹ میں مختلف اقسام کی اشیاء شامل ہو سکتی ہیں، جیسے نمبر، سٹرنگز، یا یہاں تک کہ دیگر لسٹیں۔

www.ilmkidunya.com

انڈیکس نمبر: ہر لسٹ عنصر کا ایک انڈیکس نمبر ہوتا ہے۔ جب لسٹ بنائی جاتی ہے تو ہر عنصر کو بطور ڈیفالٹ ایک انڈیکس نمبر تفویض ہو جاتا ہے۔ یہ انڈیکس نمبر اس لسٹ میں محفوظ تمام عناصر تک رسائی حاصل کرنے کے لیے استعمال ہوتا ہے۔

مثال: اپنے پسندیدہ پھلوں کی لسٹ بنائیں۔

```
fruits = ["Mango", "Apple", "Banana"]
        print(fruits)
```

آؤٹ پٹ:

```
['Mango', 'Apple', 'Banana']
```

وضاحت: یہ کوڈ 'fruits' نامی ایک لسٹ بناتا ہے جس میں تین عناصر ہوتے ہیں اور پھر لسٹ کو پرنٹ کرتا ہے۔

لسٹ کی اشیاء تک رسائی:

آپ لسٹ میں اشیاء تک ان کے انڈیکس کا حوالہ دے کر رسائی حاصل کر سکتے ہیں، جو 0 سے شروع ہوتا ہے۔

مثال: پھلوں کی لسٹ سے دوسری شے تک رسائی حاصل کریں اور پرنٹ کریں۔

```
fruits = ["Mango", "Apple", "Banana"]
print(fruits[1])
```

آؤٹ پٹ:

```
Apple
```

وضاحت: کوڈ 'fruits' لسٹ بناتا ہے جس میں 'Mango'، 'Apple'، اور 'Banana' ہوتے ہیں، پھر انڈیکس '1' کا استعمال کرتے ہوئے دوسری شے 'Apple' پرنٹ کرتا ہے۔

لسٹ میں ترمیم کرنا: آپ لسٹ کی اشیاء میں ان کے انڈیکس کے ذریعے رسائی حاصل کر کے اور نئی قیمت تفویض کر کے ترمیم کر سکتے ہیں۔

مثال: لسٹ میں پہلی شے کو 'Orange' میں تبدیل کریں اور ایک نیا پھل 'Pineapple' شامل کریں۔

```
fruits = ["Mango", "Apple", "Banana"]
fruits[0] = "Orange"
fruits.append("Pineapple")
print(fruits)
```

آؤٹ پٹ:

```
['Orange', 'Apple', 'Banana', 'Pineapple']
```

وضاحت: کوڈ fruits لسٹ کے پہلے عنصر کو 'Orange' میں تبدیل کرتا ہے۔ یہ آخر میں 'Pineapple' کا اضافہ کرتا ہے، اور تازہ ترین لسٹ پرنٹ کرتا ہے۔

سرگرمی

ایک پائیتھن پروگرام لکھیں جو random اور statistics لائبریریاں درآمد کرے۔ random کا استعمال کرتے ہوئے 1 اور 50 کے درمیان 10 بے ترتیب نمبروں کی لسٹ بنائیں۔ پھر، statistics کا استعمال کرتے ہوئے ان نمبروں کا اوسط (mean) شمار کریں اور پرنٹ کریں۔

حل:

```
import random
import statistics
```

www.ilmkidunya.com

```
# Generate a list of 10 random numbers between 1 and 50
random_numbers = []
for i in range(10):
    random_numbers.append(random.randint(1, 50))
# Print the list of random numbers
print("Random numbers:", random_numbers)
# Calculate and print the mean (average)
mean_value = statistics.mean(random_numbers)
print("Mean (average):", mean_value)
```

آؤٹ پٹ:

```
Random numbers: [23, 45, 12, 38, 7, 42, 19, 31, 8, 27]
Mean (average): 25.2
```

سوال 11: پائیتھن کوڈ کا استعمال کرتے ہوئے لسٹ پر لاگو ہونے والے append()، remove()، sort() اور reverse() آپریشنز کی وضاحت کریں؟

جواب: پائیتھن لسٹ کے ساتھ کام کرنے کے لیے کئی بلٹ ان میتھڈز فراہم کرتا ہے۔ یہاں کچھ مفید میتھڈز ہیں:

- append (item): لسٹ کے آخر میں ایک شے کا اضافہ کرتا ہے۔
- remove(item): لسٹ سے کسی شے کی پہلی موجودگی کو حذف کرتا ہے۔
- sort(item): لسٹ کو صعودی ترتیب میں ترتیب دیتا ہے۔
- reverse(item): لسٹ کی ترتیب کو الٹ دیتا ہے۔

-1 append() فنکشن:

یہ آپریشن لسٹ کے آخر میں نئی شے کا اضافہ کرتا ہے۔

مثال: طلباء کی لسٹ میں نیا طالب علم شامل کریں اور لسٹ پرنٹ کریں۔

```
students = ["Ahmed", "Sara", "Ali"]
students.append("Hina")
print(students)
```

آؤٹ پٹ:

```
['Ahmed', 'Ali', 'Sara', 'Hina']
```

وضاحت: کوڈ طلباء کی لسٹ بناتا ہے، اس میں 'Hina' کا اضافہ کرتا ہے، اور تازہ ترین لسٹ پرنٹ کرتا ہے۔

-2 remove () فنکشن:

یہ آپریشن لسٹ سے کسی شے کو حذف کرنے کے لیے استعمال ہوتا ہے۔

مثال: طلباء کی لسٹ سے ایک طالب علم کو حذف کریں اور لسٹ پرنٹ کریں۔

```
students = ["Ahmed", "Sara", "Ali", "Hina", "Sara"]
students.remove("Sara")
print(students)
```

آؤٹ پٹ:

```
['Ahmed', 'Ali', 'Hina', 'Sara']
```

وضاحت: کوڈ طلباء کی لسٹ بناتا ہے، لسٹ سے 'Sara' کی پہلی موجودگی کو حذف کرتا ہے، اور تازہ ترین لسٹ پرنٹ کرتا ہے۔ نوٹ کریں کہ صرف پہلی مماثل شے حذف ہوتی ہے۔

www.ilmkidunya.com

-3 ()Sort فنکشن:

یہ آپریشن اشیاء کو مخصوص ترتیب میں ترتیب دینے کے لیے استعمال ہوتا ہے۔

مثال: نمبروں کی لسٹ کو صعودی ترتیب میں ترتیب دیں۔

```
numbers = [34, 12, 89, 5, 23]
numbers.sort()
print(numbers)
```

آؤٹ پٹ:

```
[5, 12, 23, 34, 89]
```

وضاحت: کوڈ نمبروں کی لسٹ بناتا ہے، ()sort میتھڈ کا استعمال کرتے ہوئے انہیں صعودی ترتیب میں ترتیب دیتا ہے، اور ترتیب شدہ لسٹ پرنٹ کرتا ہے۔

-4 ()reverse فنکشن:

یہ آپریشن لسٹ کی اشیاء کی ترتیب کو الٹنے کے لیے استعمال ہوتا ہے۔

مثال4: پھلوں کی لسٹ کی ترتیب الٹ دیں۔

```
fruits = ["Apple", "Banana", "Orange", "Mango"]
fruits.reverse()
print(fruits)
```

آؤٹ پٹ:

```
['Mango', 'Orange', 'Banana', 'Apple']
```

وضاحت: کوڈ پھلوں کی لسٹ بناتا ہے، ()reverse میتھڈ کا استعمال کرتے ہوئے عناصر کی ترتیب الٹ دیتا ہے، اور الٹی ہوئی لسٹ پرنٹ کرتا ہے۔

سرگرمی 1:

تصور کریں کہ آپ اپنی پسندیدہ کتابوں کی فہرست مرتب کر رہے ہیں:

["To Kill a Mockingbird", "1984", "The Great Gatsby", "Pride and Prejudice"]

پائیتھن کا استعمال کرتے ہوئے درج ذیل کام کریں:

- فہرست میں ایک نئی کتاب "Alchemist" کا اضافہ کریں۔
- "1984" کو "Brave New World" سے تبدیل کریں۔
- "The Great Gatsby" کو فہرست سے حذف کریں۔
- اس فہرست کو کتابوں کی ایک اور فہرست: ["War and Peace", "Hamlet"] کے ساتھ ضم کریں۔
- کتابوں کی حتمی فہرست پرنٹ کریں۔

ان کاموں کو انجام دینے کے لیے پائیتھن کوڈ لکھیں اور کتابوں کی حتمی فہرست پرنٹ کریں۔

حل:

کوڈ:

```
# Starting list
books = ["To Kill a Mockingbird", "1984", "The Great Gatsby", "Pride and Prejudice"]
# Perform all operations
books.append("Alchemist")
books[books.index("1984")] = "Brave New World"
```

www.ilmkidunya.com

```
books.remove("The Great Gatsby")
books.extend(["War and Peace", "Hamlet"])
# Print final result
print("Final list of books:", favorite_books)
```

4.5 ٹیسٹنگ اور ڈیبگنگ

سوال 12: ٹیسٹنگ اور اس کی اقسام کی وضاحت کریں؟

جواب: ٹیسٹنگ آپ کے کوڈ کو مختلف ان پٹس کے ساتھ چلانے کا عمل ہے تاکہ یہ چیک کیا جا سکے کہ آیا یہ توقع کے مطابق کام کرتا ہے۔ اس کا مقصد کوڈ کو حقیقی دنیا کی ایپلیکیشنز میں استعمال کرنے سے پہلے کسی بھی مسئلے کو تلاش کرنا اور ٹھیک کرنا ہے۔

ٹیسٹنگ کی اقسام:

یونٹ ٹیسٹنگ: کوڈ کے انفرادی حصوں (جیسے فنکشنز یا کلاسز) کو الگ تھلگ کر کے جانچتی ہے۔ پائیتھن کا unittest ماڈیول عام طور پر اس کے لیے استعمال ہوتا ہے۔

انٹیگریشن ٹیسٹنگ: چیک کرتی ہے کہ کوڈ کے مختلف حصے ایک ساتھ کیسے کام کرتے ہیں۔

فنکشنل ٹیسٹنگ: اس بات کی توثیق کرتی ہے کہ سافٹ ویئر صارف کے نقطہ نظر سے توقع کے مطابق برتاؤ کرتا ہے۔

ریگریشن ٹیسٹنگ: اس بات کو یقینی بناتی ہے کہ نئی تبدیلیاں موجودہ فعالیت کو متاثر نہ کریں۔

سوال 13: ڈیبگنگ اور اس کی تکنیکوں کی وضاحت کریں؟

جواب: ڈیبگنگ آپ کے کوڈ میں موجود ایررز (بگز) کو تلاش کرنے اور ٹھیک کرنے کا عمل ہے۔ اس میں مسائل کی بنیادی وجہ کی نشاندہی کرنا اور ضروری تبدیلیاں کرنا شامل ہے۔

عام ڈیبگنگ تکنیکیں:

پرنٹ سٹیٹمنٹس: کوڈ کے مختلف مراحل پر متغیرات کی قدروں کو چیک کرنے کے لیے پرنٹ سٹیٹمنٹس کا اضافہ کرنا۔

ڈیبگنگ ٹولز: کوڈ میں قدم بہ قدم جانے، متغیرات کا معائنہ کرنے، اور عملدرآمد کے بہاؤ کو سمجھنے کے لیے pdb (پائیتھن ڈیبگر) جیسے ٹولز کا استعمال کرنا۔

ایرر میسجز: مسئلے کا ماخذ تلاش کرنے کے لیے ایرر میسجز کو پڑھنا اور ان کی تشریح کرنا۔

معروضی طرز سوالات

اضافی کثیر الانتخابی سوالات

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

تعارف

1- کنٹرول سٹرکچر اس لیے ضروری ہے:

(A) فیصلہ سازی کو منظم کرنے میں ہماری مدد کرتا ہے
(B) بار بار ہونے والے کام کو منظم کرنے میں ہماری مدد کرتا ہے
(C) پیچیدہ پروگراموں کو منظم کرنے میں ہماری مدد کرتا ہے
(D) اوپر کے تمام

-2 تکراری سٹرکچر کو یہ بھی کہتے ہیں:

(A) iteration (B) loop (C) a,b دونوں (D) کوئی نہیں

4.1 فیصلہ سازی

-3 کنڈیشنل سٹیٹمنٹ کی کتنی اقسام ہیں؟

(A) 1 (B) 2 (C) 3 (D) 4

-4 فیصلہ سازی پروگرامر کو مختلف اقدامات منتخب کرنے کی اجازت دیتی ہے اس کی بنیاد پر:

(A) شرط (B) ایکسپریشن (C) تشخیص (D) لوپ

-5 if (5 > 7): کا آؤٹ پٹ if سٹیٹمنٹ کے ڈھانچے کے مطابق ہے:

(A) True (B) False (C) کوئی آؤٹ پٹ نہیں (D) Yes

-6 پائیتھن میں ______ آپ کے کوڈ کا ڈھانچہ یا بلاک متعین کرتی ہے:

(A) سنٹیکس (B) آپریٹر (C) بلاک (D) انڈینٹیشن

-7 اگر غلط انڈینٹیشن موجود ہو تو یہ کس کا سبب بن سکتی ہے؟

(A) لاجیکل ایرر (B) سنٹیکس ایرر (C) رن ٹائم ایرر (D) لاجیکل ایرر

-8 if سٹیٹمنٹ کا سنٹیکس:

(A) if condition then: Block (B) if condition: Block

(C) if block: condition (D) if condition do: Block

-9 else کی ورڈ کو ظاہر کرنے کے لیے استعمال کیا جاتا ہے:

(A) درست سٹیٹمنٹ (B) غلط سٹیٹمنٹ (C) اس کے متوازی (D) اس کے عمودی

-10 temp > 30 کا آؤٹ پٹ ہے:

(A) True (B) False (C) Error (D) 30

-11 پائیتھن میں int() ایک ہے:

(A) ڈیٹا ٹائپ (B) کی ورڈ (C) فنکشن (D) انٹیجر کا سنٹیکس

-12 if سٹیٹمنٹ کے اندر if سٹیٹمنٹ:

(A) Multiple if (B) if-elif-else (C) if-else (D) ifNested

-13 اندرونی if بلاک صرف اس وقت چلتا ہے جب:

(A) دونوں شرائط جھوٹی ہوں (B) دونوں شرائط سچی ہوں (C) دونوں شرائط برابر ہوں (D) ان میں سے کوئی نہیں

-14 پائیتھن میں 5<10 کا نتیجہ:

(A) 5 (B) 10 (C) True (D) False

-15 اندرونی else میں لکھا بلاک صرف اس وقت چلتا ہے جب:

(A) شرط1- سچ ہے لیکن شرط2- جھوٹی ہے (B) شرط1- جھوٹی ہے لیکن شرط2- سچ ہے

(C) دونوں سچ ہیں (D) دونوں جھوٹی ہیں

www.ilmkidunya.com

16- نیسٹڈ کنڈیشنز کا استعمال کرتے ہوئے کتنی شرائط کو ظاہر کیا جا سکتا ہے؟

(A) 1 (B) 2 (C) 3 (D) کوئی بھی تعداد

4.2 لوپنگ کنسٹرکٹ

17- پائتھن میں لوپ کی کتنی اقسام استعمال ہوتی ہیں؟

(A) 1 (B) 2 (C) 3 (D) 4

18- جب تکرار کی کل تعداد معلوم ہو ______ لوپ استعمال کیا جاتا ہے؟

(A) while (B) whiledo (C) Nested (D) For

19- n=1 while n<4: Print(n) کا آؤٹ پٹ

(A) 210 (B) 4321 (C) 321 (D) 432

20- جب تکرار کی کل تعداد معلوم نہ ہو، ہم ______ لوپ استعمال کرتے ہیں؟

(A) while (B) do-while (C) Nested (D) For

21- ______ لوپ ترتیب (sequence) یا رینج استعمال کرتا ہے:

(A) while (B) do-while (C) For (D) All

22- ______ لوپ اس وقت تک دہراتا ہے جب تک شرط False نہ ہو جائے:

(A) while (B) For (C) Do-while (D) All

23- کون سا لوپ تکرار کے لیے لسٹ استعمال کرتا ہے؟

(A) while (B) Do-while (C) for (D) Nested

24- ______ فنکشن اعداد کا تسلسل حاصل کرنے یا پیدا کرنے کے لیے استعمال ہوتا ہے؟

(A) input function (B) output function (C) range function (D) Built-in function

25- رینج فنکشن میں سٹیپ (step) انجام دیتا ہے:

(A) جمع (B) تفریق (C) ضرب (D) ان پٹ پر منحصر ہے

26- درج ذیل کوڈ کا آؤٹ پٹ: for i in range(3): Print(i)

(A) 3210 (B) 210 (C) 321 (D) 0123

27- رینج فنکشن ______ سنگیمنٹس استعمال کرتا ہے:

(A) 1 (B) 2 (C) 3 (D) 4

28- پائتھن میں سٹرنگ کو نشان زد کیا جاتا ہے:

(A) سنگل کوٹس سے (B) ڈبل کوٹس سے (C) ٹرپل کوٹس سے (D) a,b دونوں

29- پائتھن میں قدر 'ayan' اور 'Ayan' ہیں:

(A) ایک جیسی (B) مختلف (C) دونوں (D) کوئی نہیں

30- لوپ کے اندر موجود دوسرے لوپ کو کہتے ہیں:

(A) while (B) Do-while (C) Nested (D) For

www.ilmkidunya.com

-31 اگر بیرونی لوپ 3 تکرار میں ختم ہوتا ہے اور اندرونی لوپ کی رینج (5) ہے تو تکرار کی کل تعداد کتنی ہوگی؟

(A) 3 (B) 5 (C) 15 (D) 12

4.3 پائیتھن میں لائبریریز

-32 لائبریریاں ______ ماڈیولز ہوتی ہیں:

(A) بلٹ ان (B) یوزر ڈیفائن (C) کمپائلر ڈیفائن (D) فنکشن

-33 پائیتھن میں لائبریریاں شامل کرنے کے لیے کون سا کی ورڈ استعمال کیا جاتا ہے؟

(A) Matplotlib (B) Random (C) import (D) Add

-34 Randomایک ہے:

(A) فنکشن (B) لائبریری (C) ماڈیولس (D) کمپائلر

-35 ()Randintایک ہے:

(A) فنکشن (B) لائبریری (C) ماڈیولس (D) کمپائلر

-36 (Statistics.mean(dataاس کوڈ کی لائن میں dataایک ہے:

(A) فنکشن (B) لائبریری (C) لسٹ (D) ارے

4.4 پائیتھن میں لسٹیں

-37 لسٹ ایک ہے:

(A) ارے (B) فنکشن (C) ماڈیولس (D) ڈیٹا سٹرکچر

-38 لسٹ میں شامل ہو سکتے ہیں:

(A) ایک جیسی اشیاء (B) واحد اشیاء (C) مختلف اشیاء (D) ان میں سے کوئی نہیں

-39 لسٹ میں، ہر عنصر کا ہوتا ہے:

(A) انڈیکس (B) قدر (C) رسائی (D) پوزیشن

-40 لسٹ میں شروع ہونے والا انڈیکس نمبر ہے:

(A) 1 (B) 2 (C) 0 (D) -1

-41 ہم لسٹ میں عنصر تک اس کے ذریعے رسائی حاصل کر سکتے ہیں:

(A) قدر (B) نام (C) انڈیکس (D) تمام

-42 لسٹ سے عناصر کو حذف کرنے کے لیے کون سا فنکشن استعمال کیا جاتا ہے؟

(A) append() (B) remove() (C) sort() (D) reverse()

-43 اشیاء کو مخصوص ترتیب میں ترتیب دینے کے لیے کون سا فنکشن استعمال کیا جاتا ہے؟

(A) append() (B) remove() (C) sort() (D) reverse()

-44 لسٹ میں عنصر شامل کرنے کے لیے کون سا فنکشن استعمال کیا جاتا ہے؟

(A) append() (B) remove() (C) sort() (D) reverse()

-45 لسٹ کی ترتیب کو تبدیل کرنے کے لیے کون سا فنکشن استعمال کیا جاتا ہے؟

(A) append() (B) sort() (C) reverse() (D) b,c دونوں

-46 خرابیوں کو تلاش کرنے اور ہٹانے کا عمل:

Test (A)   Debug (B)   Find (C)   Acknowledge (D)

-47 ٹیسٹنگ کی اقسام:

1 (A)   2 (B)   3 (C)   4 (D)

-48 ٹیسٹنگ کی کون سی قسم ایک حصے کو الگ سے جانچتی ہے؟

unit (A)   Integration (B)   Functional (C)   Regression (D)

-49 کون سی ٹیسٹنگ اس بات کو یقینی بناتی ہے کہ نئی تبدیلیاں فعالیت کو متاثر نہ کریں؟

unit (A)   Integration (B)   Functional (C)   Regression (D)

-50 ٹیسٹنگ کی کون سی قسم جانچتی ہے کہ مختلف حصے ایک ساتھ کیسے کام کرتے ہیں؟

unit (A)   Integration (B)   Functional (C)   Regression (D)

-51 ڈیبگنگ تکنیکوں کی اقسام:

(A)   2 (B)   3 (C)   4 (D)

-52 pdb ایک ہے:

Python Translator (A)   Python Module (B)   Python Debugger (C)   ان میں سے کوئی نہیں (D)

جوابات

A	-10	C	-9	D	-8	B	-7	D	-6	D	-5	A	-4	B	-3	D	-2	C	-1
C	-20	A	-19	B	-18	B	-17	D	-16	A	-15	C	-14	B	-13	C	-12	C	-11
C	-30	C	-29	A	-28	C	-27	D	-26	B	-25	C	-24	B	-23	C	-22	B	-21
D	-40	C	-39	B	-38	A	-37	B	-36	C	-35	A	-34	A	-33	C	-32	D	-31
D	-50	D	-49	C	-48	B	-47	C	-46	C	-45	C	-44	B	-43	A	-42	A	-41
																C	-52	C	-51

اضافی مختصر سوالات

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

تعارف

-1 کنٹرول سٹرکچر کی کیا اہمیت ہے؟

جواب: کنٹرول سٹرکچرز اس لیے ضروری ہیں کیونکہ یہ ہمارے پروگراموں میں فیصلہ سازی کے عمل اور کاموں کی تکرار کو منظم کرنے میں ہماری مدد کرتے ہیں۔ کنٹرول سٹرکچرز کی دو اہم اقسام ہیں: فیصلہ سازی اور لوپنگ۔

www.ilmkidunya.com

www.ilmkidunya.com

-2 فیصلہ سازی کی تعریف کریں۔

جواب: پروگرامنگ میں فیصلہ سازی پروگرام کو شرائط کی بنیاد پر مختلف اقدامات منتخب کرنے کی اجازت دیتی ہے۔ یہ اسی طرح ہے جیسے ہم حقیقی زندگی میں فیصلے کرتے ہیں۔

مثال: اگر آپ بیک بٹن دبا کر کسی ایپلیکیشن (موبائل ایپ) سے باہر جانے کی کوشش کرتے ہیں، تو ایپ پوچھتی ہے۔

'کیا آپ باہر جانا چاہتے ہیں': ہاں یا نہیں

یہ سافٹ ویئر کے ذریعے اپنائے جانے والے فیصلہ سازی کے ڈھانچے کو ظاہر کرتا ہے۔

-3 کنڈیشن کی مثال کے ساتھ تعریف کریں۔

جواب: کنڈیشن ایک ایکسپریشن ہے جس میں آپریڈز ریشنل آپریٹرز یا لاجیکل آپریٹرز یا دونوں کا مجموعہ ہوتا ہے جو بولین شکل میں نتیجہ دیتا ہے، یا توسچ ہو یا جھوٹ۔

مثال:

- 5 > 7 False
- 5==5 Ture

-4 پائتھن پروگرامنگ میں انڈینٹیشن کی کیا اہمیت ہے؟

جواب: پائتھن میں انڈینٹیشن آپ کے کوڈ کے ڈھانچے یا اسکوپ کی وضاحت کرتی ہے۔ پائتھن میں مناسب انڈینٹیشن لازمی ہے۔ غلط انڈینٹیشن آپ کے کوڈ میں خرابی (Indentation Error) کا سبب بن سکتی ہے۔

-5 if statement کی کیا حد ہے؟

جواب if statement تب آؤٹ پٹ دکھاتا ہے جب مخصوص شرط سچ ہو، اگر شرط جھوٹی ہو تو اسکرین پر کچھ بھی ظاہر نہیں ہوگا۔

-6 if-else statement کا سنٹیکس لکھیں۔

جواب:

```
if condition:
        code to run if the condition is true
else:
        code to run if the condition is false
```

-7 Nested if کی تعریف کریں۔

جواب: if statement جس کے جسم میں دوسری if statement ہو، if nested کہلاتا ہے۔ یہ ڈھانچہ پیچیدہ متعدد شرائط کو لاگو کرنے کے لیے استعمال ہوتا ہے۔

Nested if statement کا سنٹیکس:

```
if condition1:
    if condition2:
        code to run if both condition1 and condition2 are true
    else:
        code to run if condition1 is true but condition2 is false
else:
    code to run if condition1 is false
```

8- لوپ کی تعریف کریں اور پائتھن میں استعمال ہونے والی لوپ کی اقسام کے نام بتائیں۔

جواب: اگر ہم کسی ایک سٹیٹمنٹ یا سٹیٹمنٹس کے سیٹ کو ایک سے زیادہ بار دہرانا چاہتے ہیں، تو ہم لوپ استعمال کرتے ہیں۔ لوپ کم سے کم ان پٹ کی مدد سے زیادہ سے زیادہ آؤٹ پٹ حاصل کرنے میں ہماری مدد کرتے ہیں۔ پائیتھن میں ہم دو قسم کے لوپ استعمال کرتے ہیں:

- while لوپ
- for لوپ

9- for لوپ اور while لوپ کے درمیان فرق کریں۔

جواب: for لوپ:

- استعمال ہوتا ہے جب تکرار کی کل تعداد پہلے سے معلوم ہو۔
- اسے کاؤنٹر کنٹرول لوپ بھی کہتے ہیں۔
- سنٹیکس:

```
for variable in sequence:
code to run for each element
```

- while لوپ
- استعمال ہوتا ہے جب تکرار کی کل تعداد پہلے سے معلوم نہ ہو۔
- اسے سینٹینل کنٹرول لوپ بھی کہتے ہیں۔
- سنٹیکس:

```
while condition:
code to run while the condition is true
```

10- رینج فنکشن کی تعریف کریں۔

جواب: ہم ()range فنکشن استعمال کرسکتے ہیں تاکہ اعداد کا تسلسل پیدا کیا جاسکے، جو اکثر فار لوپس میں استعمال ہوتا ہے۔

رینج فنکشن کا سنٹیکس:

```
range(start)
range(start, stop)
range(start, stop, step)
```

مثال: پہلے 5 پورے نمبر پرنٹ کریں۔

```
for i in range(5):
    print(i)
```

آؤٹ پٹ:

```
0
1
2
3
4
```

11- نیسٹڈ لوپ کی تعریف کریں۔

جواب: نیسٹڈ لوپ دوسرے لوپ کے اندر موجود لوپ ہوتے ہیں۔ بیرونی لوپ کے ہر بار چلنے کے لیے، اندرونی لوپ مکمل طور پر چلتا ہے۔ ہم نیسٹڈ لوپ استعمال کرتے ہیں جب ہمیں کاموں کے اندر دہرائے جانے والے کام انجام دینے ہوں۔

www.ilmkidunya.com

12- ڈیٹا سٹرکچر کی تعریف کریں۔

جواب: ڈیٹا سٹرکچر سے مراد ڈیٹا کو منظم اور ذخیرہ کرنے کا ایک خاص فارمیٹ یا طریقہ ہے۔ پائیتھن میں، لسٹ، ٹیپل مشہور ڈیٹا سٹرکچر ہیں۔

13- لسٹ کی تعریف کریں۔

جواب: پائیتھن میں، لائبریریاں ٹول باکسز کی طرح ہوتی ہیں جو مفید ٹولز سے بھری ہوتی ہیں، جو آپ کو مختلف مسائل حل کرنے میں مدد دیتی ہیں بغیر اس کے کہ آپ کو سب کچھ خود ہی بنانا پڑے۔ لائبریریاں پہلے سے تیار شدہ ٹول کٹس کی طرح ہوتی ہیں جنہیں آپ استعمال کر سکتے ہیں بغیر اس کے کہ آپ خود تمام کوڈ لکھیں۔

14- پائتھن میں لائبریری درآمد کرنے کی ایک مثال بیان کریں۔

جواب: کوڈ:

```
import random
number = random.randint(1, 10)
print("The random number is:", number)
```

آؤٹ پٹ:

```
The random number is: 3
```

15- پائتھن میں statistics لائبریری درآمد کرنے کی ایک مثال لکھیں۔

جواب:

```
import statistics

data = [23, 45, 67, 89, 12, 44, 56]
mean_value = statistics.mean(data)
print("The mean value is:", mean_value)
```

آؤٹ پٹ:

```
The mean value is: 48
```

4.4 پائتھن میں لسٹیں

16- لسٹ بنانے کی وضاحت کریں۔

جواب: لسٹ کو مربع بریکٹ [ ] کے اندر اشیاء کو کوما سے الگ کر کے بنایا جاتا ہے۔ لسٹ میں مختلف اقسام کی اشیاء شامل ہو سکتی ہیں، جیسے نمبر، سٹرنگز، یا یہاں تک کہ دیگر لسٹیں۔

مثال: اپنے پسندیدہ پھلوں کی لسٹ بنائیں۔

کوڈ:

```
fruits = ["Mango", "Apple", "Banana"]
print(fruits)
Output:
['Mango', 'Apple', 'Banana']
```

آؤٹ پٹ:

```
['Mango', 'Apple', 'Banana']
```

www.ilmkidunya.com

17- لسٹ کے عنصر میں ترمیم کرنے کی وضاحت کریں۔

جواب: آپ لسٹ کی اشیاء میں ان کے انڈیکس کے ذریعے رسائی حاصل کر کے اور نئی قیمت تفویض کر کے ترمیم کر سکتے ہیں۔

مثال: لسٹ میں پہلی شے کو 'Orange' میں تبدیل کریں اور ایک نیا پھل 'Pineapple' شامل کریں۔

کوڈ:

```
fruits = ["Mango", "Apple", "Banana"]
fruits[0] = "Orange"
fruits.append("Pineapple")
print(fruits)
```

آؤٹ پٹ:

```
['Orange', 'Apple', 'Banana', 'Pineapple']
```

18- append میتھڈ کی وضاحت کریں۔

جواب: append() فنکشن:

یہ آپریشن لسٹ کے آخر میں نئی شے کا اضافہ کرتا ہے۔

مثال: طلباء کی لسٹ میں نیا طالب علم شامل کریں اور لسٹ پرنٹ کریں۔

کوڈ:

```
students = ["Ahmed", "Sara", "Ali"]
students.append("Hina")
print(students)
```

آؤٹ پٹ:

```
['Ahmed', 'Ali', 'Sara', 'Hina']
```

4.5 ٹیسٹنگ اور ڈیبگنگ

19- ٹیسٹنگ کی تعریف کریں اور اس کی اقسام کے نام بتائیں۔

جواب: ٹیسٹنگ آپ کے کوڈ کو مختلف ان پٹس کے ساتھ چلانے کا عمل ہے تاکہ یہ چیک کیا جا سکے کہ آیا یہ توقع کے مطابق کام کرتا ہے۔ اس کا مقصد کوڈ کو حقیقی دنیا کی ایپلیکیشنز میں استعمال کرنے سے پہلے کسی بھی مسئلے کو تلاش کرنا اور ٹھیک کرنا ہے۔

ٹیسٹنگ کی اقسام:

- یونٹ ٹیسٹنگ
- انٹیگریشن ٹیسٹنگ
- فنکشنل ٹیسٹنگ
- ریگریشن ٹیسٹنگ

20- ڈیبگنگ کی تعریف کریں اور اس کی تکنیکوں کے نام بتائیں۔

جواب: ڈیبگنگ آپ کے کوڈ میں موجود ایررز (بگز) کو تلاش کرنے اور ٹھیک کرنے کا عمل ہے۔ اس میں مسائل کی بنیادی وجہ کی نشاندہی کرنا اور ضروری تبدیلیاں کرنا شامل ہے۔

عام ڈیبگنگ تکنیکیں:

- پرنٹ سٹیٹمنٹس
- ڈیبگنگ ٹولز
- ایرر میسجز

www.ilmkidunya.com

حل مشقی سوالات

کثیر الانتخابی سوالات

-i پائتھن میں range() فنکشن کیا کرتا ہے؟

(الف) اعداد کی ایک لسٹ بناتا ہے
(ب) اعداد کی ایک ترتیب بناتا ہے
(ج) اعداد کا حاصل جمع نکالتا ہے
(د) اعداد کی ایک رینج پرنٹ کرتا ہے

-ii درج ذیل کوڈ کا نتیجہ کیا ہوگا؟

```
count = 0
for i in range(2):
    for j in range(3):
        count += 1
print(count)
```

(الف) 4 (ب) 5 (ج) 6 (د) 7

-iii پائتھن میں لسٹ کے آخر میں ایک آئٹم شامل کرنے کے لیے کون سا میتھڈ استعمال ہوتا ہے؟

(الف) insert() (ب) append() (ج) add() (د) extend()

-iv پروگرامنگ میں کنٹرول سٹرکچرز کا مقصد کیا ہے؟

(الف) صرف کوڈ کو لمبا کرنا
(ب) فیصلہ سازی اور کاموں کی ریپٹیشن کو منظم کرنا
(ج) متغیرات بنانا
(د) لائبریریز امپورٹ کرنا

-v کون سی سٹیٹمنٹ استعمال ہوتی ہے جب کوئی شرط غلط (false) ہو تو کوڈ کو چلایا جائے؟

(الف) if (ب) elif (ج) else (د) while

-vi آپ print() کو نئی لائن شامل کرنے سے کیسے روکتے ہیں؟

(الف) end="" استعمال کریں
(ب) newline=False استعمال کریں
(ج) break استعمال کریں
(د) continue استعمال کریں

-vii رینڈم اعداد (random numbers) بنانے کے لیے کون سی لائبریری استعمال ہوتی ہے؟

(الف) Datetime (ب) statistics (ج) random (د) math

-viii list.append() کیا کرتا ہے؟

(الف) پہلی آئٹم کو ہٹاتا ہے
(ب) آخر میں ایک آئٹم شامل کرتا ہے
(ج) لسٹ کو ترتیب دیتا ہے
(د) لسٹ کو الٹ (reverse) دیتا ہے

-ix range() فنکشن کا مقصد کیا ہے؟

(الف) اعداد کی ایک لسٹ بنانا (ب) اعداد کی ایک ترتیب پیدا کرنا (ج) قدروں کی رینج کا حساب لگانا (د) اعداد کو ترتیب دینا

-x کون سی ڈیبگنگ تکنیک آؤٹ پٹ سٹیٹمنٹس کو شامل کرنے پر مشتمل ہے؟

(الف) یونٹ ٹیسٹنگ (ب) پرنٹ سٹیٹمنٹس (ج) ایرر میسجز (د) انٹیگریشن ٹیسٹنگ

جوابات

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

www.ilmkidunya.com

-1 for لوپ میں range() فنکشن کے استعمال کی وضاحت کریں۔

جواب: ہم range() فنکشن کو اعداد کا تسلسل پیدا کرنے کے لیے استعمال کر سکتے ہیں، جو اکثر for لوپس میں استعمال ہوتا ہے۔

مثال: پہلے 5 مکمل اعداد پرنٹ کریں۔

```
for i in range(5):
        print(i)
```

-2 پائتھن لسٹوں میں append اور remove میتھڈز کے درمیان کیا فرق ہے؟

جواب: append() فنکشن:

یہ آپریشن لسٹ کے آخر میں نیا آئٹم شامل کرتا ہے۔

مثال: طلباء کی لسٹ میں ایک نیا طالب علم شامل کریں اور لسٹ پرنٹ کریں۔

کوڈ:

```
students = ["Ahmed", "Sara", "Ali"]
students.append("Hina")
print(students)
```

remove () فنکشن:

یہ آپریشن لسٹ سے کسی آئٹم کو حذف کرنے کے لیے استعمال ہوتا ہے۔

مثال: طلباء کی لسٹ سے ایک طالب علم کو حذف کریں اور لسٹ پرنٹ کریں۔

کوڈ:

```
students = ["Ahmed", "Sara", "Ali", "Hina", "Sara"]

students.remove("Sara")

print(students)
```

-3 ڈیبگنگ کی تعریف کریں۔

جواب: ڈیبگنگ آپ کے کوڈ میں موجود ایررز (بگز) کو تلاش کرنے اور انہیں ٹھیک کرنے کا عمل ہے۔ اس میں مسائل کی بنیادی وجہ کی نشاندہی کرنا اور ضروری تبدیلیاں کرنا شامل ہے۔

-4 range() فنکشن کے تین پیرامیٹرز کیا ہیں؟

جواب: range فنکشن کا سنٹیکس:

range(start, stop, step)

اس سنٹیکس میں start پیرامیٹر اعداد کے تسلسل کی شروعات بتاتا ہے۔ stop اعداد کا تسلسل پیدا کرنے کی آخری حد بتاتا ہے اور step اعداد کے درمیان اضافہ بتاتا ہے۔

-5 لسٹ کے تین میتھڈز اور ان کے مقاصد کے نام بتائیں۔

جواب: append (item): لسٹ کے آخر میں کوئی آئٹم شامل کرتا ہے۔

remove(item): لسٹ سے کسی آئٹم کی پہلی موجودگی کو حذف کرتا ہے۔

sort(item): لسٹ کو صعودی ترتیب میں ترتیب دیتا ہے۔

www.ilmkidunya.com

6- آپ پائتھن میں ایک لائبریری کو کیسے امپورٹ کرتے ہیں؟

جواب: Python میں لائبریری کو "import" کی ورڈ استعمال کرتے ہوئے درآمد کیا جاسکتا ہے:

مثال:

```
import random
number = random.randint(1, 10)
print("The random number is:", number)
```

7- یونٹ ٹیسٹنگ کیا ہے اور یہ کیوں اہم ہے؟

جواب: یہ کوڈ کے انفرادی حصوں (جیسے فنکشنز یا کلاسز) کو الگ تھلگ کر کے جانچتا ہے۔ Python کا unittest ماڈیول عام طور پر اس کے لیے استعمال ہوتا ہے۔ یہ اہم ہے کیونکہ پروگرامر تخلیق کے وقت ہر انفرادی جز کو جانچ سکتا ہے، اس سے کام کی افادیت میں اضافہ ہوتا ہے۔

8- آپ ایک لسٹ میں تیسرے عنصر تک کیسے رسائی حاصل کرتے ہیں؟

جواب: لسٹ میں بطور ڈیفالٹ 0 سے شروع ہونے والے انڈیکس نمبر ہوتے ہیں۔ اگر ہمارے پاس 4 عناصر کی لسٹ ہے تو ہم انڈیکس نمبر 2 کی مدد سے تیسرے عنصر تک رسائی حاصل کر سکتے ہیں۔

مثال:

```
marks = [76, 75, 77, 78]
print(marks[2]
```

آؤٹ پٹ:

```
77
```

9- ()print میں end پیرامیٹر کا مقصد کیا ہے؟

جواب: Python میں، end کوئی علیحدہ کمانڈ نہیں ہے، بلکہ یہ ()print فنکشن کا ایک اختیاری پیرامیٹر ہے۔ بطور ڈیفالٹ، Python میں ہر ()print ایک نئی لائن کریکٹر کے ساتھ ختم ہوتا ہے، لیکن آپ end پیرامیٹر کا استعمال کر کے آخر میں پرنٹ ہونے والی چیز کو تبدیل کر سکتے ہیں۔

مثال:

"end" اسٹیٹمنٹ کا استعمال کرتے ہوئے دو سٹرنگز کو ایک ہی لائن پر پرنٹ کریں۔

کوڈ:

```
print("Azhar Shah", end=" ")
print("Ahsan")
```

آؤٹ پٹ:

```
Azhar Shah Ahsan
```

10- ہم لسٹ میں سے ایک عنصر تک رسائی کیسے حاصل کر سکتے ہیں؟

جواب: لسٹ میں بطور ڈیفالٹ 0 سے شروع ہونے والے انڈیکس نمبر ہوتے ہیں۔ اگر ہم لسٹ سے عناصر تک رسائی حاصل کرنا چاہتے ہیں تو ہم انڈیکس نمبر کی مدد سے عنصر تک رسائی حاصل کر سکتے ہیں۔

مثال:

```
marks = [76, 75, 77, 78]
print(marks[2]
```

آؤٹ پٹ:

```
77
```

www.ilmkidunya.com

11- آپ ایک لسٹ میں موجود آئٹم میں کیسے ترمیم کرتے ہیں؟

جواب: آپ لسٹ آئٹمز کو ان کے انڈیکس کے ذریعے رسائی حاصل کر کے اور ایک نئی ویلیو تفویض کر کے تبدیل کر سکتے ہیں۔

مثال: لسٹ میں پہلی آئٹم کو "Orange" میں تبدیل کریں اور ایک نیا فروٹ "Pineapple" شامل کریں۔

کوڈ:

```
fruits = ["Mango", "Apple", "Banana"]
fruits[0] = "Orange"
fruits.append("Pineapple")
print(fruits)
```

آؤٹ پٹ:

```
['Orange', 'Apple', 'Banana', 'Pineapple']
```

تفصیلی سوالات

1- پائتھن میں فیصلہ سازی سٹیٹمنٹس کی مختلف اقسام کی مثالوں کے ساتھ وضاحت کریں۔

جواب: پروگرامنگ میں فیصلہ سازی پروگرام کو کنڈیشنز کی بنیاد پر مختلف اعمال کا انتخاب کرنے کی اجازت دیتی ہے۔ یہ اسی طرح ہے جیسے ہم حقیقی زندگی میں فیصلے کرتے ہیں۔

فیصلہ سازی کے اسٹیٹمنٹس کی اقسام:

- if statement
- if else statement

if Statement:

if اسٹیٹمنٹ کنڈیشنز کی بنیاد پر فیصلے کرتا ہے۔ اگر کنڈیشن درست ہوتی ہے، تو یہ کوڈ کا ایک بلاک چلاتا ہے۔

سنٹیکس:

```
if condition:
    code to run if the condition is true
```

انڈینٹیشن:

Python میں انڈینٹیشن آپ کے کوڈ کے ڈھانچے یا سکوپ کی وضاحت کرتی ہے۔ Python میں مناسب انڈینٹیشن لازمی ہے۔ غلط انڈینٹیشن آپ کے کوڈ میں ایرر (Indentation Error) کا سبب بن سکتی ہے۔

مثال:

```
Age = 21
if Age > 18:
        print("You are a teenager!!!!!!!")
```

if - else:

اسٹیٹمنٹ ہمیں ایک بلاک کوڈ کو چلانے کی اجازت دیتا ہے اگر کنڈیشن درست ہو اور دوسرا بلاک اگر کنڈیشن غلط ہو۔

if-else اسٹیٹمنٹ کا سنٹیکس:

```
if condition:
    code to run if the condition is true
else:
    code to run if the condition is false
```

www.ilmkidunya.com

مثال:

```
temperature = 15
if temperature > 30:
        print("It's a hot day")
else:
        print("It's not a hot day")
```

آؤٹ پٹ:

```
It's not a hot day
```

-2 while لوپس اور for لوپس کا موازنہ اور تضاد کریں۔

جواب: While loop:

while لوپ اس وقت تک چلتا ہے جب تک کوئی کنڈیشن درست ہو۔ یہ ہر تکرار سے پہلے کنڈیشن کو چیک کرتا ہے اور جب کنڈیشن مزید درست نہیں رہتی تو رک جاتا ہے۔ بنیادی طور پر، while لوپ اس وقت استعمال ہوتا ہے جب ہم پہلے سے نہیں جانتے کہ ہمارا لوپ کتنی بار دہرایا جائے گا۔

while لوپ کا سنٹیکس:

```
while condition:
        code to run while the condition is true
```

مثال: ایک عدد میں 1 شامل کریں جب تک یہ 10 تک نہ پہنچ جائے۔

```
    number = 1
while number < 10:
    print(number)
    number += 1
```

وضاحت: اس مثال میں، کوڈ نمبر 1 سے شروع ہوتا ہے۔ 'while' لوپ چیک کرتا ہے کہ آیا نمبر 10 سے کم ہے۔ اگر ہے، تو پروگرام نمبر کی موجودہ ویلیو پرنٹ کرتا ہے اور پھر اس میں 1 کا اضافہ کرتا ہے۔ یہ لوپ اس وقت تک جاری رہتا ہے جب تک نمبر 10 تک نہ پہنچ جائے، جس پر لوپ رک جاتا ہے۔

For Loop:

for لوپ کوڈ کے ایک بلاک کو ایک مخصوص تعداد میں دہراتا ہے۔ یہ عام طور پر کسی ترتیب (جیسے لسٹ، ٹپل، یا سٹرنگ) پر تکرار کرنے کے لیے استعمال ہوتا ہے۔

for لوپ کا سنٹیکس:

```
for variable in sequence:
code to run for each element in the sequence
```

مثال:

دوستوں کی لسٹ میں ہر دوست کو 'As-Salaam-Alaikum' کہیں۔

```
friends = ["Sami", "Raza", "Moosa"]
for friend in friends:
        print("Welcome to ", friend)
```

آؤٹ پٹ:

```
Welcome to: Sami
Welcome to: Raza
Welcome to: Moosa
```

وضاحت: اس مثال میں، کوڈ لسٹ میں موجود ہر دوست پر جاتا ہے اور ہر ایک کے لیے ایک سلام کا پیغام پرنٹ کرتا ہے۔

www.ilmkidunya.com

-3 پائتھن لسٹوں کے ساتھ کام کرنے کے طریقے کو بیان کریں، بشمول ان کو بنانے اور ان میں ترمیم کرنے کا طریقہ۔ آئٹمز شامل کرنے، ہٹانے اور عناصر تک رسائی حاصل کرنے کے لیے مثالیں فراہم کریں۔

جواب: جواب کے لیے دیکھیے سوال نمبر 10

-4 پروگرامنگ میں ٹیسٹنگ اور ڈیبگنگ کی اہمیت کی وضاحت کریں۔ مختلف ڈیبگنگ تکنیکس پر بحث کریں۔

جواب: ٹیسٹنگ آپ کے کوڈ کو مختلف ان پٹس کے ساتھ چلانے کا عمل ہے تاکہ یہ جانچا جا سکے کہ آیا یہ توقع کے مطابق برتاؤ کرتا ہے۔ مقصد یہ ہے کہ کوڈ کو حقیقی دنیا کی ایپلیکیشنز میں استعمال کرنے سے پہلے کسی بھی مسئلے کو تلاش کر کے ٹھیک کیا جائے۔

ڈیبگنگ آپ کے کوڈ میں موجود ایررز (بگز) کو تلاش کرنے اور انہیں ٹھیک کرنے کا عمل ہے۔ اس میں مسائل کی بنیادی وجہ کی نشاندہی کرنا اور ضروری تبدیلیاں کرنا شامل ہے۔

عام ڈیبگنگ تکنیکیں:

پرنٹ اسٹیٹمنٹس: کوڈ کے مختلف مراحل پر متغیرات کی ویلیوز چیک کرنے کے لیے پرنٹ اسٹیٹمنٹس کا اضافہ کرنا۔

ڈیبگنگ ٹولز: pdb (Python Debugger)

جیسے ٹولز کا استعمال کرتے ہوئے کوڈ میں قدم بہ قدم جانا، متغیرات کا معائنہ کرنا، اور عمل درآمد کے بہاؤ کو سمجھنا۔

ایرر میسجز: ایرر میسجز کو پڑھنا اور ان کی تشریح کرنا تاکہ مسئلے کا ماخذ معلوم کیا جا سکے۔

-5 پائتھن لائبریریز پروگرامنگ کی کارکردگی (efficiency) کو کیسے بڑھاتی ہیں؟ ایک مثال فراہم کریں۔

جواب: کوڈ:

جواب کے لیے دیکھیے سوال نمبر 9

-6 ایک پائتھن پروگرام لکھیں جو ایک لوپ کا استعمال کرتے ہوئے 1 اور 20 کے درمیان تمام طاق اعداد (Odd number) کو پرنٹ کرے۔

جواب: کوڈ:

```
# Print odd numbers using step parameter
print("Odd numbers from 1 to 20:")
for i in range(1, 21, 2):
        print(i, end=" ")
```

-7 ایک پائتھن پروگرام لکھیں جو درج ذیل کام انجام دے:

(a) 1 اور 20 کے درمیان 5 رینڈم اعداد کی ایک لسٹ بنائے۔

(b) پیدا کردہ اعداد کا اوسط (mean) نکالنے اور پرنٹ کرنے کے لیے statistics لائبریری کا استعمال کرے۔

جواب: کوڈ:

```
import random
import statistics

random_numbers = []
for i in range(5):
        random_numbers.append(random.randint(1, 20))

print("Generated random numbers:", random_numbers)

mean_value = statistics.mean(random_numbers)
print("Mean (average) of the numbers:", mean_value)
```

8- ایک پائتھن پروگرام لکھیں جو درج ذیل آپریشنز انجام دے:

- مشہور پاکستانی کھانوں کی ایک لسٹ بنائیں: ["Biryani", "Nihari", "Karahi", "Seekh Kebabs"]۔
- لسٹ میں ایک نیا کھانا "Quorma" شامل کریں۔
- "Karahi" کو "Chicken Karahi" میں تبدیل کریں۔
- لسٹ میں سے "Nihari" کو ہٹا دیں۔

جواب: کوڈ:

```
pakistani_dishes = ["Biryani", "Nihari", "Karahi", "Seekh Kebabs"]
print("Initial list:", pakistani_dishes)

pakistani_dishes.append("Quorma")
pakistani_dishes[pakistani_dishes.index("Karahi")] = "Chicken Karahi"
pakistani_dishes.remove("Nihari")
print("Final list:", pakistani_dishes)
```

9- نیسٹڈ لوپس کا استعمال کرتے ہوئے تین قطاروں (rows) اور پانچ کالموں (columns) میں ستاروں (*) کو مستطیل پیٹرن میں پرنٹ کرنے کے لیے کوڈ لکھیں۔

جواب: کوڈ:

```
for row in range(3):
    for col in range(5):
        print("*", end="")
    print()
```

آؤٹ پٹ:

```
*****
*****
*****
```

10- درج ذیل کوڈ کا آؤٹ پٹ لکھیں:

```
print("=== Multiplication Table (1-5) ====")
for i in range(2, 4):    # Outer: 2 to 3
    for j in range(1, 11):  # Inner: 1 to 10
        print(f"{i} x {j} = {i*j:2d}", end=" ")
    print()    # New line after each table
```

جواب:

=== Multiplication Table (1-5) ====

2 x 1 = 2 2 x 2 = 4 2 x 3 = 6 2 x 4 = 8 2 x 5 = 10 2 x 6 = 12 2 x 7 = 14 2 x 8 = 16 2 x 9 = 18 2 x 10 = 20

3 x 1 = 3 3 x 2 = 6 3 x 3 = 9 3 x 4 = 12 3 x 5 = 15 3 x 6 = 18 3 x 7 = 21 3 x 8 = 24 3 x 9 = 27 3 x 10 = 30

www.ilmkidunya.com