

- اس باب کے اختتام تک، آپ اس قابل ہو جائیں گے کہ:
- پانچھن میں کنٹرول اسٹرکچرز کو نافذ کر سکیں، جیسے کہ فیصلہ سازی کی شیٹس اور لوپس۔
- پانچھن کے ماڈیولز، فنکشنز، اور بلٹ ان ڈیٹا اسٹرکچرز جیسے کہ لسٹس (Lists) کے ساتھ کام کر سکیں۔
- لسٹس (Lists) پر مختلف آپریشنز پر فارم 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 کرنا پڑتا 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۔

فیصلہ سازی

4.1

سوال 1: فیصلہ سازی کی تعریف کریں؟ پانچھن 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 بن 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 اپنا جواب دینے والے فیصلہ سازی کے ڈھانچے کو ظاہر کرتا ہے۔

شرط (Condition):

شرط ایک ایکسپریشن ہے جس میں آپریٹرز، ریشل آپریٹرز یا لاجیکل آپریٹرز 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

if else statement •
سوال 2: if statement کی اس کے سنیکس 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۔
سنیکس:

if condition

code to run if the condition is true

انڈینیشن:

پانچھن میں انڈینیشن آپ کے کوڈ کے ڈھانچے یا اسکوپ کی وضاحت کرتی ہے۔ پانچھن میں مناسب انڈینیشن لازمی ہے۔ غلط انڈینیشن 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 بھی ظاہر نہیں ہوگا۔

سوال 3: if statement کی اس کے سنیکس 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 سنیکس:

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: عیڈ کنڈیشن کی اس کے سنیکس 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 ٹیسٹنگ کہتے ہیں۔

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

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۔

لوپنگ کنسٹرکٹ

4.2

سوال 5: لوپ کی تعریف کریں؟ وہاں لوپ کی سٹیکس اور مثال کے ساتھ وضاحت کریں۔

جواب: اگر ہم کسی ایک شیمنٹ یا شیمنٹس 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۔ پائیکشن 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 بار بار 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۔

while condition:

code to run while the condition is true

```
number = 1
```

```
while number <= 10:
```

```
    print(number)
```

```
    number += 1
```

وضاحت: اس مثال میں، کوڈ نمبر 1 سے شروع ہوتا ہے۔ وہاں 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 سٹیکس 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، ٹیبل، یا سٹرنگ) پر iterate کرنے کے لیے استعمال ہوتا ہے۔

for لوپ کا سٹیکس:

for variable in sequence:

code to run for each element in the sequence

```
friends = ["Sami", "Raza", "Moosa"]
```

```
for friend in friends:
```

```
    print("Welcome to ", friend)
```

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

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 ٹیٹنٹ کی تعریف کریں۔ نیز مثالیں لکھیں۔

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

```
range(start)
```

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

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

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

```
    print(i)
```

مثال: پہلے 5 پورے نمبر پرنٹ کریں۔

آؤٹ پٹ:

```
0
```

```
1
```

2
3
4

end شیئٹ: پانچھن میں، اینڈ شیئٹ کوئی علیحدہ مکائنڈ نہیں ہے، بلکہ یہ print() فنکشن کا ایک اختیاری پیرامیٹر ہے۔ ڈیفالٹ طور پر، پانچھن میں ہر print() ایک نئی لائن کیئر کیئر کے ساتھ ختم ہوتا ہے، لیکن آپ end پیرامیٹر استعمال کر کے تبدیلی کر سکتے ہیں کہ آخر میں کیا پرنٹ ہو۔ مثال 1: 'end' شیئٹ استعمال کرتے ہوئے دو سرنگز 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' شیئٹ 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؟

پانچھن میں سرنگز یا تو سیکل کوئیشن 'مارکس' یا ڈبل کوئیشن 'مارکس' کے درمیان ہوتی ہیں۔ جیسے 'Ayyan' وہی ہے جو 'Ayyan' ہے۔

سری

1- ایک فارلوپ range() کے ساتھ لکھیں جو 2 سے 10 تک کے زوج اعداد کو ایک ہی لائن پر پرنٹ کرے۔

2- ایک پانچھن پروگرام لکھیں جو 3 کے پہلے 10 ضربوں کو فارلوپ 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=" ")
```

3 6 9 12 15 18 21 24 27 30

سوال 8: میڈل لوپ کی تعریف کریں؟ نیز مثالیں لکھیں۔
جواب: میڈل لوپ دوسرے لوپ کے اندر موجود لوپ ہوتے ہیں۔ بیرونی لوپ کے برابر چلنے کے لیے، اندرونی لوپ مکمل طور پر چلتا ہے۔ ہم میڈل لوپ استعمال کرتے ہیں جب ہمیں کاموں کے اندر دہرائے جانے والے کام انجام دینے ہوں۔
مثال 1: بنیادی نمبر پیٹرن کے لیے میڈل 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 میڈل 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:

```
*
```

```
**
```

```
***
```

```
****
```

```
*****
```

سوال 9: پانچھن میں ڈیٹا سٹرکچر اور لائبریری سے کیا مراد ہے؟ پانچھن 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۔ پانچھن میں، لسٹ، ٹیپل، مشہور ڈیٹا سٹرکچر ہیں۔
لسٹ: پانچھن میں، لائبریریوں ٹول باکس 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۔ لائبریریوں 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 درآ کرے۔

```
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 درآ کرے۔

```
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۔
مختصر، 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 پیہر دوبارہ ایجاد کریں۔

سمارٹ ٹپس:

لائبریریوں درآ کرے وقت، یقینی بنائیں کہ صرف وہی درآ کرے جس کی آپ کو ضرورت ہے۔ آپ مزید لائبریریوں <https://docs.python.org/3/> پر دریافت کر سکتے ہیں۔

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: پانچھن میں، لسٹ ایک درشائل 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۔

انڈیکس نمبر: جیسے کہ 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' شامل 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 1

ایک پائیتھن پروگرام لکھیں جو random اور statistics نمبریں دیا کرتا ہے۔ 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
```

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: پائٹھن کوڈ کا استعمال کرتے ہوئے لسٹ پر لاگو ہونے والے reverse(), sort(), remove(), append() آپریشنز کی وضاحت کریں؟

جواب: پائٹھن لسٹ کے ساتھ کام کرنے کے لیے کئی بلٹ ان میٹھڈز فراہم کرتا ہے۔ یہاں کچھ مفید میٹھڈز 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)
```

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

آؤٹ پٹ:

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)
```

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

آؤٹ پٹ:

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 مائل شے حذف ہوتی ہے۔

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() میٹھڈ 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۔
مثال 4: 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() میٹھڈ 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"]

پایتھن 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 پایتھن 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"
```

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

ٹیسٹ اور ڈیٹنگ

4.5

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

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

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

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

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

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

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

جواب: ڈیٹنگ آپ کے کوڈ میں موجود ایررز (بگ) 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 ڈیٹنگ تکنیکیں:

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

ڈیٹنگ ٹولز: کوڈ میں قدم بہ قدم جانے، تغیرات کا معائنہ کرنے، اور عملدرآمد کے بہاؤ کو سمجھنے کے لیے pdb (پائتھن ڈیباگر) جیسے ٹولز کا استعمال کرنا۔

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

معروضی طرز سوالات

اضافی کثیر الانتخابی سوالات

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

تعارف

1- کنٹرول سٹرپچر 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

(D) کوئی نہیں

(C) a,b دونوں

(B) loop

(A) iteration

فیصلہ سازی

4.1

کنڈیشنل شیڈنگ کی کتنی اقسام ہیں؟

(D) 4

(C) 3

(B) 2

(A) 1

فیصلہ سازی پروگرام کو کثرت اقدامات منتخب کرنے کی اجازت دیتی ہے اس کی بنیاد پر:

(D) لوپ

(C) تشخیص

(B) ایکپریشن

(A) شرط

if (5 > 7) کا آؤٹ پٹ if شیڈنگ کے لحاظ سے کیا ہے:

(D) Yes

(C) کوئی آؤٹ پٹ نہیں

(B) False

(A) True

پائیکسن میں _____ آپ کے کوڈ کا لحاظ نہ ہونا یا ہلاک متعین کرتی ہے:

(D) انڈینٹیشن

(C) ہلاک

(B) آپریٹر

(A) سنیکس

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

(D) لاجیکل ایرر

(C) دن ٹائم ایرر

(B) سنیکس ایرر

(A) لاجیکل ایرر

if شیڈنگ کا سنیکس:

if condition: Block (B)

if condition then: Block (A)

if condition do: Block (D)

if block: condition (C)

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

(D) اس کے عمودی

(C) اس کے متوازی

(B) غلط شیڈنگ

(A) درست شیڈنگ

temp > 30 کا آؤٹ پٹ ہے:

(D) 30

(C) Error

(B) False

(A) True

پائیکسن میں lint() ایک ہے:

(D) انجیر کا سنیکس

(C) فنکشن

(B) کی ورڈ

(A) ڈیٹا ٹائپ

if شیڈنگ کے اندر if شیڈنگ:

(D) ifNested

(C) if-else

(B) if-elif-else

(A) Multiple if

اعددوں کی ہلاک صرف اس وقت چلتا ہے جب:

(D) ان میں سے کوئی نہیں

(C) دونوں شرائط برابر ہوں

(B) دونوں شرائط سچی ہوں

(A) دونوں شرائط جھوٹی ہوں

پائیکسن میں 5 < 10 کا نتیجہ:

(D) False

(C) True

(B) 10

(A) 5

اعددوں کی ہلاک صرف اس وقت چلتا ہے جب:

(B) شرط 1- جھوٹی ہے لیکن شرط 2- سچی ہے

(A) شرط 1- سچی ہے لیکن شرط 2- جھوٹی ہے

(D) دونوں جھوٹی ہیں

(C) دونوں سچی ہیں

- 16- میڈکنڈیشنز کا استعمال کرتے ہوئے کتنی شرائط کو ظاہر کیا جاسکتا ہے؟
 (A) 1 (B) 2 (C) 3 (D) کوئی بھی تعداد

(C) 3

(B) 2

(A) 1

لوپنگ کنسٹرکٹ

4.2

- 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- `Print: 4 < n while n = 1` 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 ریج استعمال کرتا ہے:
 (A) while (B) do-while (C) For (D) All
- 22- _____ لوپ اس وقت تک دہراتا ہے جب تک `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- ریج 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- ریج فنکشن _____ سٹیمپز 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) ٹریپل 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

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

12 (D)

15 (C)

5 (B)

3 (A)

پانچھن میں لائبریری

4.3

32- لائبریریاں _____ ماڈیولز ہوتی ہیں:

(D) فنکشن

(C) کمپائلر ڈیفائن

(B) یوزر ڈیفائن

(A) بلٹ ان

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

Add (D)

import (C)

Random (B)

Matplotlib (A)

34- Random ایک ہے:

(D) کمپائلر

(C) ماڈیولس

(B) لائبریری

(A) فنکشن

35- Randint() ایک ہے:

(D) کمپائلر

(C) ماڈیولس

(B) لائبریری

(A) فنکشن

36- Statistics.mean(data) اس کوڈ کی لائن میں data ایک ہے:

(D) ارے

(C) لسٹ

(B) لائبریری

(A) فنکشن

پانچھن میں لیں

4.4

37- لسٹ ایک ہے:

(D) ڈیٹا سٹرکچر

(C) ماڈیولس

(B) فنکشن

(A) ارے

38- لسٹ میں شامل ہو سکتے ہیں:

(D) ان میں سے کوئی نہیں

(C) مختلف اشیاء

(B) واحد اشیاء

(A) ایک جیسی اشیاء

39- لسٹ میں، ہر عنصر کا ہوتا ہے:

(D) پوزیشن

(C) رسائی

(B) قدر

(A) انڈیکس

40- لسٹ میں شروع ہونے والا انڈیکس نمبر ہے:

(D) -1

(C) 0

(B) 2

(A) 1

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

(D) تمام

(C) انڈیکس

(B) نام

(A) قدر

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

reverse() (D)

sort() (C)

remove() (B)

append() (A)

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

reverse() (D)

sort() (C)

remove() (B)

append() (A)

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

reverse() (D)

sort() (C)

remove() (B)

append() (A)

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

b,c دونوں (D)

reverse() (C)

sort() (B)

append() (A)

Acknowledge (D)

Find (C)

Debug (B)

Test (A)

یونٹنگ کی اقسام:

4 (D)

3 (C)

2 (B)

1 (A)

یونٹنگ کی کون سی قسم ایک حصے کو الگ سے جانچتی ہے؟

Regression (D)

Functional (C)

Integration (B)

unit (A)

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

Regression (D)

Functional (C)

Integration (B)

unit (A)

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

Regression (D)

Functional (C)

Integration (B)

unit (A)

یونٹنگ تکنیکوں کی اقسام:

4 (D)

3 (C)

2 (B)

1 (A)

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

تجزیاتی سوالات (Knowledge, Understanding, Application, Analytical & Conceptual) کی روشنی میں مرتب کیے گئے مختصر جوابی سوالات

تعارف

کنٹرول سٹرکچر کی کیا اہمیت ہے؟

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

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 بن 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 اپنا نئے 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 آپریٹرز مثل آریٹریز یا لاجیکل آریٹریز یا دونوں کا مجموعہ ہوتا ہے جو بولین شکل میں نتیجہ دیتا ہے، یا تو جی ہو یا جھوٹ۔

مثال:

- $5 > 7$ False
- $5 == 5$ True

4- پانچن پروگرامنگ میں انڈینٹیشن کی کیا اہمیت ہے؟

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

5- if statement کی کیا اہمیت ہے؟

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

6- if-else statement کا سٹیکس 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 سٹیکس:

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 پانچوں میں استعمال ہونے والی لوپ کی اقسام کے نام بتائیں۔
جواب: اگر ہم کسی ایک سینٹ یا ٹینٹس کے سیٹ کو ایک سے زیادہ بار دہرائنا 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۔ پانچوں 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۔

• سنکس:

for variable in sequence:

code to run for each element

• while

• استعمال ہوتا ہے جب تکرار کی کل تعداد پہلے سے معلوم نہ ہو۔

• اسے سینٹیل کنٹرول لوپ بھی کہتے ہیں۔

• سنکس:

while condition:

code to run while the condition is true

10- رینج فنکشن کی تعریف کریں۔

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

رینج فنکشن کا سنکس:

range(start)

range(start, stop)

range(start, stop, step)

for i in range(5):

print(i)

0

1

2

3

4

مثال: پہلے 5 پورے نمبر پرنٹ کریں۔

آؤٹ پٹ:

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۔

12- ڈیٹا سٹرکچر کی تعریف کریں۔

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

13- لسٹ کی تعریف کریں۔

جواب: پانچھن میں، لائبریریاں نول 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 نول کس کی طرح ہوتی ہیں جنہیں آپ استعمال کر سکتے ہیں بغیر اس کے کہ آپ خود تمام کوڈ لکھیں۔

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

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

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']

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

جواب: آپ لسٹ کی اشیاء میں ان کے اندر ایسے مقامات حاصل کر کے اور نئی قیمت تفویض کر کے ترمیم کر سکتے ہیں۔

مثال: لسٹ میں پہلی شے '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 میٹھڈ 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']

ٹیسٹنگ اور ڈیبلنگ

4.5

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 جاسکے کہ آیا یہ توقع کے مطابق کام کرتا ہے۔ اس کا مقصد کوڈ کو حقیقی دنیا کی اپیلیکیشنز میں استعمال کرنے سے پہلے کسی بھی مسئلے کو تلاش کرنا اور ٹھیک کرنا ہے۔

ٹیسٹنگ کی اقسام:

- یونٹ ٹیسٹنگ
- انٹیگریشن ٹیسٹنگ
- سسٹم ٹیسٹنگ
- ریگریژن ٹیسٹنگ

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

جواب: ڈیبلنگ آپ کے کوڈ میں موجود ایررز (بگ) 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 ڈیبلنگ تکنیکیں:

- پرنٹ سٹینٹس
- ڈیبلنگ ٹولز
- ایرر میسجز

-i پانچن 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 پانچن میں لسٹ کے آخر میں ایک آئٹم شامل کرنے کے لیے کون سا میٹھ استعمال ہوتا ہے؟

- (الف) insert() (ب) append() (ج) add() (د) extend()
پروگرامنگ میں کنٹرول سٹرکچر کا مقصد کیا ہے؟

- (الف) صرف کوڈ کو لمبا کرنا
(ب) فیصلہ سازی اور کاموں کی ریپیٹیشن کو منظم کرنا
(ج) متغیرات بنانا
(د) لاگو پروگرام پورٹ کرنا

-v کون سی سینٹ 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() کوئی 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 ریٹنڈم 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 ڈیپلٹنگ تکنیک آؤٹ پٹ سے فائنل کوڈ کو منظم کرتی ہے؟

- (الف) پونٹ ٹیسٹنگ
(ب) پرنٹ ٹیسٹنگ
(ج) ایرر میسج
(د) انٹیگریشن ٹیسٹنگ

جوابات :-

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

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 پر پرنٹ کریں۔

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

2- پانچھن لسٹوں میں append اور remove میٹھڈز 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 پر 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 پر پرنٹ کریں۔

کوڈ:

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

3- ڈیٹنگ کی تعریف کریں۔

جواب: ڈیٹنگ آپ کے کوڈ میں موجود ایررز (بگ) 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 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 میٹھڈز 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۔

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

جواب: Python میں لائبریری کو "import" کی مدد سے 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 کرکٹر 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" سینٹنسٹ 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

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

جواب: آپ لسٹ آئٹمز کو ان کے انڈیکس کے ذریعے رسائی حاصل کر سکتے ہیں 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']

تقسیم کی مثال

1- پائٹھن میں فیصلہ سازی سٹیمٹس 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 سٹیمٹس کی اقسام

- if statement
- if else statement

if Statement:

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

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:

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

if-else سٹینٹ کا سٹیکس:

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

temperature = 15

if temperature > 30:

print("It's a hot day")

else:

print("It's not a hot day")

It's not a hot day

مثال: آؤٹ پٹ:

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 سنگمس:

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، ٹیبل، یا سٹرنگ) پر تکرار کرنے کے لیے استعمال ہوتا ہے۔

استعمال ہوتا ہے۔

for لوپ کا سنگمس:

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)

Welcome to: Sami

Welcome to: Raza

Welcome to: Moosa

آؤٹ پٹ:

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۔

3- پانچھن لکھوں کے ساتھ کام کرنے کے طریقے کو بیان کریں، بشمول ان کو بنانے اور ان میں ترمیم کرنے کا طریقہ۔ آٹھ 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 ڈیباگنگ کی اہمیت کی وضاحت کریں۔ مختلف ڈیباگنگ ٹیکنیکس پر بحث کریں۔

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

ڈیباگنگ آپ کے کوڈ میں موجود ایررز (بگ) 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 ڈیباگنگ ٹیکنیکس:

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

ڈیباگ 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۔

ایر میسج: ایر میسج کو پڑھنا اور ان کی تشریح کرنا تاکہ مسئلے کا ماخذ معلوم کیا جاسکے۔

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

جواب: کوڈ:

جواب کے لیے دیکھیے سوال نمبر 9

6- ایک پانچھن 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 پانچھن پروگرام لکھیں جو درج ذیل کام انجام دے:

(a) 1 اور 20 کے درمیان 5 ریٹنڈم 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 پانچھن پروگرام لکھیں جو درج ذیل آپریشنز انجام دے:

- -["Biryani", "Nihari", "Karahi", "Seekh Kebabs"]
- لست میں ایک نیا کھانا "Quorma" شامل کریں۔
- "Karahi" کو "Chicken Karahi" میں تبدیل کریں۔
- لست 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- میٹرکس 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
```