Top of Form

Question 1: **Correct**

**The XYZ organics company needs a simple program that their call center will use to enter survey data for a new coffee variety. The program must accept input and return the average rating based on a five-star scale.The output must be rounded to two decimal places.**

**Consider the code:**

1. **sum=count=done=0**
2. **average=0.0**
3. **while(done != -1):**
4. **rating=float(input('Enter Next Rating(1-5),-1 for done'))**
5. **if rating == -1:**
6. **break**
7. **sum+=rating**
8. **count+=1**
9. **average=float(sum/count)**
10. **#Line-1**

**Which of the following print() statement should be placed at Line-1 to meet requirement?**

* ​

print('The average star rating for the new coffee is:{:.2f}'.format(average))

**(Correct)**

* ​

print('The average star rating for the new coffee is:{:.2d}'.format(average))

* ​

print('The average star rating for the new coffee is:{:2f}'.format(average))

* ​

print('The average star rating for the new coffee is:{:2.2d}'.format(average))

**Explanation**

As the output required to rounded to 2 decimal places and it is float value we should use print('The average star rating for the new coffee is:{:.2f}'.format(average))

Bottom of Form

Top of Form

Question 2: **Correct**

**Consider the following statements:**

1. **1. print('V:{:.2f}'.format(123.45678)) will print to the console V:123.46**
2. **2. print('V:{:.2f}'.format(123.4)) will print to the console V:123.40**
3. **3. print('V:{:8.2f}'.format(1.45678)) will print to the console V: 1.46**
4. **4. print('V:{:08.2f}'.format(1.45678)) will print to the console V:00001.46**

**Which of the above statements are True?**

* ​

only 1 and 2

* ​

only 1 and 3

* ​

only 2 and 4

* ​

1,2,3 and 4

**(Correct)**

**Explanation**

{:.2f}==>After decimal point, the value will be rounded to 2 digits  
{:8.2f}==>Total 8 length and after decimal point value will be rounded to 2 digits.  
Numbers are right aligned by default and vacant spaces are filled with blank space.  
{:08.2f}===>Total 8 length and after decimal point value will be rounded to 2 digits.  
Numbers are right aligned by default and vacant spaces are filled with digit 0.

Bottom of Form

Top of Form

Question 3: **Correct**

**We are developing a sports application. Our program should allow players to enter their name and score. The program will print player name and his average score. Output must meet the following requirements:**

**The user name must be left aligned. If the user name is fewer than 20 characters ,additional space must be added to the right. The average score must be 3 places to the left of decimal point and one place to the right of decimal point ( like YYY.Y).**

**Consider the code:**

1. **name=input('Enter Your Name:')**
2. **score=0**
3. **count=0**
4. **sum=0**
5. **while(score != -1):**
6. **score=int(input('Enter your scores: (-1 to end)'))**
7. **if score==-1:**
8. **break**
9. **sum+=score**
10. **count+=1**
11. **average\_score=sum/count**
12. **#Line-1**

**Which print statement we have to take at Line-1 to meet requirements.**

* ​

print('%-20s,Your average score is: %4.1f' %(name,average\_score))

**(Correct)**

* ​

print('%-20f,Your average score is: %4.1f' %(name,average\_score))

* ​

print('%-20s,Your average score is: %1.4f' %(name,average\_score))

* ​

print('%-20s,Your average score is: %4.1s' %(name,average\_score))

**Explanation**

'%4.1f': Minimum 4 length  
After decimal point all digits rounded to 1 digit  
If the number less than 4 length then spaces will be added at left hand side '%04.1f'  
Minimum 4 length  
After decimal point all digits rounded to 1 digit  
If the number less than 4 length then 0s will be added at left hand side  
name=input('Enter Some Name:')  
print('%-20s' %name)  
# It will consider minimum 20 length,if it is less than 20 then spaces will be padded at right hand side  
print('%20s' %name)  
# It will consider minimum 20 length,if it is less than 20 then spaces will be padded at left hand side

Bottom of Form

Top of Form

Question 4: **Correct**

**Consider the following code:**

1. **numbers=[0,1,2,3,4,5,6,7,8,9]**
2. **index=0**
3. **while (index<10)#Line-1**
4. **print(numbers[index])**
6. **if numbers(index) = 6#Line-2**
7. **break**
8. **else:**
9. **index += 1**

**To print 0 to 6,which changes we have to perform in the above code?**

* ​
  1. Line-1 should be replaced with
  2. while(index<10):

**(Correct)**

* ​
  1. Line-2 should be replaced with
  2. if numbers[index]==6:

**(Correct)**

* ​
  1. Line-2 should be replaced with
  2. if numbers[index]=6:
* ​
  1. Line-1 should be replaced with
  2. while(index>0):

**Explanation**

At Line-1, invalid syntax,because colon(:) is missing.  
At Line-2 we should use == operator for comparison and colon(:) is missing.

Bottom of Form

Top of Form

Question 5: **Correct**

**You are writing a Python program to validate employee numbers.**

**The employee number must have the format dd-ddd-dddd and consists of only numbers and dashes.The program must print True if the format is correct,otherwise print False.**

1. **employee\_number=input('Enter Your Employee Number(dd-ddd-dddd):')**
2. **parts=employee\_number.split('-')**
3. **valid=False**
4. **if len(parts) == 3:**
5. **if len(parts[0])==2 and len(parts[1])==3 and len(parts[2])==4:**
6. **if parts[0].isdigit() and parts[1].isdigit() and parts[2].isdigit():**
7. **valid=True**
8. **print(valid)**

**Which of the following is True about this code**

* ​

It will throw error because misuse of split() method

* ​

It will throw error because misuse of isDigit() method

* ​

There is no error but it won't fulfll our requirement.

* ​

No changes are required for this code and it can fulfill requirement.

**(Correct)**

**Explanation**

Every method call invoked properly and there is no error. This code can fulfill requirement without any changes.

Bottom of Form

Top of Form

Question 6: **Incorrect**

**You are coding a math utility by using python.**

1. **You are writing a function to compute roots**
2. **The function must meet the following requirements**
3. **If a is non-negative, return a\*\*(1/b)**
4. **If a is negative and even, return "Result is an imaginary number"**
5. **if a is negative and odd,return -(-a)\*\*(1/b)**

**Which of the following root function should be used?**

* ​
  1. def root(a,b):
  2. if a>=0:
  3. answer=a\*\*(1/b)
  4. elif a%2 == 0:
  5. answer="Result is an imaginary number"
  6. else:
  7. answer=-(-a)\*\*(1/b)
  8. return answer

**(Correct)**

* ​
  1. def root(a,b):
  2. if a>=0:
  3. answer=a\*\*(1/b)
  4. elif a%2 != 0:
  5. answer="Result is an imaginary number"
  6. else:
  7. answer=-(-a)\*\*(1/b)
  8. return answer
* ​
  1. def root(a,b):
  2. if a>=0:
  3. answer=a\*\*(1/b)
  4. if a%2 == 0:
  5. answer="Result is an imaginary number"
  6. else:
  7. answer=-(-a)\*\*(1/b)
  8. return answer
* ​
  1. def root(a,b):
  2. if a>=0:
  3. answer=a\*\*(1/b)
  4. elif a%2 == 0:
  5. answer=-(-a)\*\*(1/b)
  6. else:
  7. answer="Result is an imaginary number"
  8. return answer

**(Incorrect)**

**Explanation**

Make sure you should remember if condition fails then only elif will be executed.

Bottom of Form

Top of Form

Question 7: **Correct**

**We are developing a online shopping application. Consider the code**

1. **d =input('Enter day of the week:')**
2. **discount\_percentage = 3**
4. **if d== 'Monday':**
5. **discount\_percentage+=5**
7. **elif d== 'Tuesday':**
8. **discount\_percentage+=7**
10. **elif d== 'Saturday':**
11. **discount\_percentage+=10**
13. **elif d== 'Sunday':**
14. **discount\_percentage+=20**
15. **else:**
16. **discount\_percentage+=2**

**To get 5 as discount\_percentage,which of the following input should be provided end user?**

* ​

Monday

* ​

Tuesday

* ​

Thursday

**(Correct)**

* ​

Saturday

* ​

Sunday

**Explanation**

if user enters Monday then discount\_percentage will be 8  
if user enters Tuesday then discount\_percentage will be 10  
if user enters Thursday then discount\_percentage will be 5  
if user enters Saturday then discount\_percentage will be 13  
if user enters Sunday then discount\_percentage will be 23

Bottom of Form

Top of Form

Question 8: **Correct**

**We are developing gold loan application for XYZ company.**

1. **amount=float(input('Enter Loan Amount:'))**
2. **interest\_rate=0**
3. **if amount > 0 and amount<= 50000:**
4. **interest\_rate = 10**
6. **elif amount > 50000 and amount<100000:**
7. **interest\_rate = 12**
9. **elif amount >= 100000 and amount<150000:**
10. **interest\_rate = 16**
12. **else:**
13. **interest\_rate = 22**

**For which of the following user input interest\_rate will be 12.**

* ​

50000

* ​

50001

**(Correct)**

* ​

100000

* ​

100001

* ​

150000

**Explanation**

For 50000 amount interest\_rate is 10  
For 50001 amount interest\_rate is 12  
For 100000 amount interest\_rate is 16  
For 100001 amount interest\_rate is 16  
For 150000 amount interest\_rate is 22

Bottom of Form

Top of Form

Question 9: **Correct**

**We are developing an application for leave approval in XYZ Company.**

1. **days=int(input('Enter number of days for leave:'))**
2. **cause=input('Enter the cause:')**
3. **if days==1:**
4. **print('Leave will be approved immediately')**
6. **elif days>1 and days<=3:**
7. **if cause=='Sick':**
8. **print('Leave will be approved immediately')**
9. **else:**
10. **print('Needs Lead Approval')**
12. **elif days>3 and days<5:**
13. **if cause=='Sick':**
14. **print('Needs Manager Approval')**
15. **else:**
16. **print('Needs Director Approval')**
18. **elif days>=5 and days<=10:**
19. **print('Needs Director Approval')**

**In which of the following cases 'Needs Director Approval' will be printed to the console?**

* ​

days = 2 and cause='Sick'

* ​

days = 3 and cause='Personal'

* ​

days = 4 and cause='Sick'

* ​

days = 4 and cause='Official'

**(Correct)**

**Explanation**

If days = 2 and cause='Sick' then 'Leave will be approved immediately' will be printed to the console.  
If days = 3 and cause='Personal' then 'Needs Lead Approval' will be printed to the console.  
If days = 4 and cause='Sick' then 'Needs Manager Approval' will be printed to the console.  
If days = 4 and cause='Official' then 'Needs Director Approval' will be printed to the console.

Bottom of Form

Top of Form

Question 10: **Correct**

**Consider the following code:**

1. **marks=[30,40,50,45,50,100]**
2. **average=sum(marks)//len(marks)**
3. **grades={1:'A',2:'B',3:'C',4:'D'}**
4. **if average>=90 and average<=100:**
5. **key=1**
6. **elif average>=80 and average<90:**
7. **key=2**
8. **elif average>=50 and average<80:**
9. **key=3**
10. **else:**
11. **key=4**
12. **print(grades[key])**

**Which grade will be printed to the console?**

* ​

A

* ​

B

* ​

C

**(Correct)**

* ​

D

**Explanation**

average=sum(marks)//len(marks)=315//6=52  
For 52 average key is 3 and corresponding grade is C

Bottom of Form

Top of Form

Question 11: **Incorrect**

**Consider the code:**

1. **a=12**
2. **b=4**
3. **s='He shall not be happy if he does not work'**

**In which of the following cases result value will become 9**

* ​

result=3 if None else a/b

* ​

result=s.find('not') if s else None

**(Correct)**

* ​

result=s.rfind('not') if s else None

* ​

result=5 if len(s)>4 else 6

**(Incorrect)**

**Explanation**

result=3 if None else a/b==>3.0 (None is treated as False)  
result=s.find('not') if s else None==>9(find() method returns the index of first match from left hand side)  
result=s.rfind('not') if s else None===>33(rfind() method returns the index of first match from right hand side)  
result=5 if len(s)>4 else 6===>5

Bottom of Form

Top of Form

Question 12: **Correct**

**We are developing loan collection agent application. Consider the code:**

1. **collected\_amount=3000**
2. **commission=0**
3. **if collected\_amount <= 2000:**
4. **commission=50**
5. **elif collected\_amount> 2500 and collected\_amount<3000:**
6. **commission=100**
7. **elif collected\_amount>2500:**
8. **commission=150**
9. **if collected\_amount>=3000:**
10. **commission+=200**

**What will be the value of commission?**

* ​

350

**(Correct)**

* ​

200

* ​

150

* ​

100

**Explanation**

As the value of collected amount is 3000 the following statements will be executed.  
commission=0  
commission=150  
commission+=200  
Hence the value of commission will become 350.

Bottom of Form

Top of Form

Question 13: **Correct**

**You are developing online shopping application.**

1. **Consider the code:**
2. **order\_value=1500**
3. **state='ap'**
4. **delivery\_charge=0**
5. **if state in ['up','mp','ts']:**
6. **if order\_value<=1000:**
7. **delivery\_charge=50**
8. **elif order\_value>1000 and order\_value<2000:**
9. **delivery\_charge=100**
10. **else:**
11. **delivery\_charge=150**
12. **else:**
13. **delivery\_charge=25**
14. **if state in ['lp','kp','ap']:**
15. **if order\_value>1000:**
16. **delivery\_charge+=20**
17. **if order\_value<2000 and state in ['kp','ap']:**
18. **delivery\_charge+=30**
19. **else:**
20. **delivery\_charge+=15**
21. **print(delivery\_charge)**

**What is the result?**

* ​

65

* ​

75

**(Correct)**

* ​

85

* ​

55

**Explanation**

As the state is 'ap' and order\_value is 1500 the following lines of the code will be executed.  
delivery\_charge=0  
delivery\_charge=25  
delivery\_charge+=20  
delivery\_charge+=30  
Hence the value of delivery\_charge is 75.

Bottom of Form

Top of Form

Question 14: **Correct**

**Consider the code:**

1. **l=[10,20,[30,40],[50,60]]**
2. **count=0**
3. **for i in range(len(l)):**
4. **if type(l[i])==list:**
5. **count=count+1**
6. **print(count)**

**What is the result?**

* ​

1

* ​

2

**(Correct)**

* ​

3

* ​

4

**Explanation**

list,set,tuple,dict are python's inbuilt variables to represent data structure types.

Bottom of Form

Top of Form

Question 15: **Incorrect**

**Consider the code:**

1. **l=[10,(20,),{30},{},{},[40,50]]**
2. **count=0**
3. **for i in range(len(l)):**
4. **if type(l[i])==list:**
5. **count+=1**
6. **elif type(l[i])==tuple:**
7. **count+=2**
8. **elif type(l[i])==set:**
9. **count+=3**
10. **elif type(l[i])==dict:**
11. **count+=4**
12. **else:**
13. **count+=5**
14. **print(count)**

**What is the result?**

* ​

17

**(Incorrect)**

* ​

18

* ​

19

**(Correct)**

* ​

20

**Explanation**

list,set,tuple,dict are python's inbuilt variables to represent data structure types.

Bottom of Form

Top of Form

Question 16: **Correct**

**Consider the code:**

1. **t = (2,4,6,8,10,12)**
2. **d = {1:'A',2:'B',3:'C',4:'D',5:'E',6:'F'}**
3. **result=1**
4. **for t1 in t:**
5. **if t1 in d:**
6. **result+=t1**
7. **print(result)**

**What is the result?**

* ​

12

* ​

13

**(Correct)**

* ​

19

* ​

6

**Explanation**

The elements 2,4,6 from the tuple present in dict as keys. Hence these elements will be added to result. Hence result is 13(=1+2+4+6)

Bottom of Form

Top of Form

Question 17: **Correct**

**Consider the code:**

1. **t = (2,4,6,8,10,12)**
2. **d = {1:'A',2:'B',3:'C',4:'D',5:'E',6:'F'}**
3. **result=1**
4. **for t1 in t:**
5. **if t1 in d:**
6. **continue**
7. **else:**
8. **result+=t1**
9. **print(result)**

**What is the result?**

* ​

29

* ​

30

* ​

31

**(Correct)**

* ​

32

**Explanation**

The elements 8,10,12 from the tuple are not present in dict as keys. Hence these elements will be added to result. Hence result is 31(=1+8+10+12)

Bottom of Form

Top of Form

Question 18: **Incorrect**

**Consider the code:**

1. **values = [[3, 4, 5, 1], [33, 6, 1, 2]]**
2. **v = values[0][0]**
3. **for lst in values:**
4. **for element in lst:**
5. **if v > element:**
6. **v = element**
7. **print(v)**

**What is the result?**

* ​

3

* ​

2

**(Incorrect)**

* ​

1

**(Correct)**

* ​

4

**Explanation**

The above code is for finding minimum element present in the nested list.

Bottom of Form

Top of Form

Question 19: **Incorrect**

**Consider the code**

1. **def get\_names():**
2. **names=['Sunny','Bunny','Chinny','Vinny','Pinny']**
3. **return names[2:]**
5. **def update\_names(elements):**
6. **new\_names=[]**
7. **for name in elements:**
8. **new\_names.append(name[:3].upper())**
9. **return new\_names**
11. **print(update\_names(get\_names()))**

**What is the result?**

* ​

['CHI', 'VIN', 'PIN']

**(Correct)**

* ​

['VIN', 'PIN']

* ​

['CH', 'VI', 'PI']

**(Incorrect)**

* ​

['SU', 'BU']

**Explanation**

names[2:] returns all names from 2 index to end of list. i.e ['Chinny','Vinny','Pinny']. name[:3].upper() selects first 3 characters from name and will convert to upper case. Hence the output is : ['CHI', 'VIN', 'PIN']

Bottom of Form

Top of Form

Question 20: **Incorrect**

**Consider the following code**

1. **def my\_list(x):**
2. **lst.append(a)**
3. **return lst**
5. **my\_list('chicken')**
6. **my\_list('mutton')**
7. **print(my\_list('fish'))**
9. **to print the following to the console**
10. **['chicken','mutton','fish']**

**x should be replaced with**

* ​

a,lst=[]

**(Correct)**

* ​

a,lst=()

* ​

a,lst={}

* ​

a,lst=None

**(Incorrect)**

**Explanation**

The required output is of list type and append() method is applicable only for list and hence we should take  
a,lst=[]

Bottom of Form

Top of Form

Question 21: **Incorrect**

**Consider the following code:**

1. **def f1(x=0,y=0):**
2. **return x+y**

**Which of the following method calls are valid?**

* ​

f1()

**(Correct)**

* ​

f1('10','20')

**(Correct)**

* ​

f1(10)

**(Correct)**

* ​

f1('10')

**Explanation**

To use + operator both arguments should be either number type or string type. f1()==>returns 0  
f1('10','20')===>returns '1020'  
f1(10)===>returns 10  
f1('10')===>TypeError: must be str, not int. We cannot use + operator between '10' and 0

Bottom of Form

Top of Form

Question 22: **Incorrect**

**Consider the following code:**

1. **def f1(x=0,y=0):**
2. **return x\*y**

**Which of the following method calls are valid?**

* ​

f1()

**(Correct)**

* ​

f1('10','20')

* ​

f1(10)

**(Correct)**

* ​

f1('10')

**(Correct)**

**Explanation**

To use \* operator both arguments should be either number type, or one is string type and other is int type.  
Hence f1('10','20') is invalid. TypeError: can't multiply sequence by non-int of type 'str'

Bottom of Form

Top of Form

Question 23: **Incorrect**

**Consider the following code:**

1. **numbers=[100,20,10,70,50,60,40,30,90,80]**
2. **#Insert Code Here**
3. **print('The highest Number:{} and Least Number:{}'.format(high,low))**

**Which of the following code should be inserted to print Highest Number as 100 and Least Number as 10**

* ​
  1. def find\_numbers():
  2. numbers.sort()
  3. return numbers[0],numbers[-1]
  4. low,high=find\_numbers()

**(Correct)**

* ​
  1. def find\_numbers():
  2. numbers.sort()
  3. return numbers[0],numbers[len(numbers)]
  4. low,high=find\_numbers()
* ​
  1. def find\_numbers():
  2. numbers.sort()
  3. return numbers[0],numbers[-1]
  4. low=find\_numbers()
  5. high=find\_numbers()
* ​
  1. def find\_numbers():
  2. numbers.sort()
  3. return numbers[2],numbers[0]
  4. low,high=find\_numbers()

**(Incorrect)**

**Explanation**

After sorting numbers list will become [10,20,30,40,50,60,70,80,90,100]. Hence numbers[0] represents low value and numbers[-1] represents highest value.

Bottom of Form

Top of Form

Question 24: **Correct**

**Consider the code:**

1. **numbers=[100,20,10,70,50,60,40,30,90,80]**
2. **def find\_numbers():**
3. **numbers.sort()**
4. **return numbers[0],numbers[-1]**
6. **low=find\_numbers()**
7. **high=find\_numbers()**
8. **#Line-1**

**To print 10 100 to the console which of the following code we have to take at Line-1**

* ​

print(low,high)

* ​

print(low[0],high[-1])

**(Correct)**

* ​

print(low[-1],high[0])

* ​

print(low[2],high[0])

**Explanation**

In the above code, low and high are tuple type.The content of low=(10,100) and high=(10,100).Hence to print 10 100 we should use print(low[0],high[-1])

Bottom of Form

Top of Form

Question 25: **Correct**

**Consider the code:**

1. **def calculate(amount=6,factor=3):**
2. **if amount>6:**
3. **return amount\*factor**
4. **else:**
5. **return amount\*factor\*2**

**Which of the following function calls returns 30**

* ​

calculate()

* ​

calculate(10)

**(Correct)**

* ​

calculate(5,2)

* ​

calculate(1)

**Explanation**

print(calculate())==>36  
print(calculate(10))==>30  
print(calculate(5,2))==>20  
print(calculate(1))==>6

Bottom of Form

Top of Form

Question 26: Skipped

**Consider the following code**

1. **def fib\_seq(n):**
2. **if n==0:**
3. **return 0**
4. **elif n==1:**
5. **return 1**
6. **else:**
7. **return fib\_seq(n-1)+fib\_seq(n-2)**
8. **for i in range(7):**
9. **print(fib\_seq(i),end=',')**

**What is the result?**

* ​

0,1,1,2,3,5,8,

**(Correct)**

* ​

0,1,2,4,8,16,32,

* ​

0,1,0,2,0,4,0,

* ​

None of these

**Explanation**

The above program generates the first 7 fibonacci numbers. Hence the output is: 0,1,1,2,3,5,8,

Bottom of Form

Top of Form

Question 27: Skipped

**You are developing a Python application for online game.**

1. **You need to create a function that meets the following criteria:**
2. **The function is named update\_score**
3. **The function receives the current score and a value.**
4. **The function adds the value to the current score.**
5. **The function returns the new score.**

**Which of the following is valid function to fulfill this requirement?**

* ​
  1. update\_score(score,value):
  2. new\_score=score+value
  3. return new\_score
* ​
  1. def update\_score(score,value):
  2. new\_score=score+value
  3. return new\_score

**(Correct)**

* ​
  1. def update\_score(score,value):
  2. new\_score=score+value
  3. pass new\_score
* ​
  1. def update\_score():
  2. new\_score=score+value
  3. return new\_score

**Explanation**

We should declare a function with def keyword. A function can return value by using return keyword. As per our requirement, compulsory the function should take some arguments and return new score. Hence the following function can fulfill our requirement. def update\_score(score,value):  
new\_score=score+value  
return new\_score

Bottom of Form

Top of Form

Question 28: Skipped

**The XYZ company is creating a program that allows customers to log the number of miles biked.The program will send messages based on how many miles the customer logs. Consider the following python code:**

1. **Line-1:**
2. **name=input('Enter Your Name:')**
3. **return name**
5. **Line-2:**
6. **calories=miles\*calories\_per\_mile**
7. **return calories**
9. **distance=int(input('How many miles did you bike this week:'))**
10. **burn\_rate=50**
11. **biker=get\_name()**
12. **calories\_burned=calc\_calories(distance,burn\_rate)**
13. **print(biker,", You burned about",calories\_burned," calories")**

**The lines Line-1 and Line-2 should be replaced with:**

* ​
  1. Line-1 should be replaced with
  2. def get\_name():

**(Correct)**

* ​
  1. Line-1 should be replaced with
  2. def get\_name(name):
* ​
  1. Line-1 should be replaced with
  2. def get\_name(biker):
* ​
  1. Line-2 should be replaced with
  2. def calc\_calories(miles,calories\_per\_mile):

**(Correct)**

* ​
  1. Line-2 should be replaced with
  2. def calc\_calories(miles,burn\_rate):
* ​
  1. Line-2 should be replaced with
  2. def calc\_calories():

**Explanation**

The following are valid function declarations to fulfill our requirement: def get\_name():  
name=input('Enter Your Name:')  
return name  
def calc\_calories(miles,calories\_per\_mile):  
calories=miles\*calories\_per\_mile  
return calories

Bottom of Form

Top of Form

Question 29: Skipped

**You work for a company that distributes media for all ages.**

**You are writing a function that assigns a rating based on a user's age.The function must meet the following requirements.**

**Anyone 18 years old or older receives a rating of "A"**

**Anyone 13 or older,but younger than 18, receives a rating of "T"**

**Anyone 12 years old or younger receives a rating of "C"**

**If the age is unknown ,the rating is set to "C"**

**Which of the following code meets above requirements:**

* ​
  1. def get\_rating(age):
  2. if age>=18:
  3. rating="A"
  4. elif age>=13:
  5. rating="T"
  6. else:
  7. rating="C"
  8. return rating

**(Correct)**

* ​
  1. def get\_rating(age):
  2. if age>=18:
  3. rating="A"
  4. if age>=13:
  5. rating="T"
  6. else:
  7. rating="C"
  8. return rating
* ​
  1. def get\_rating(age):
  2. if age>18:
  3. rating="A"
  4. elif age>13:
  5. rating="T"
  6. else:
  7. rating="C"
  8. return rating
* ​
  1. def get\_rating(age):
  2. if age>=18:
  3. rating="A"
  4. elif age>=13:
  5. rating="T"
  6. else:
  7. rating="C"
  8. pass rating

**Explanation**

The correct function to fulfill above requirements is  
def get\_rating(age):  
if age>=18:  
rating="A"  
elif age>=13:  
rating="T"  
else:  
rating="C"  
return rating

Bottom of Form

Top of Form

Question 30: Skipped

**Consider the following Python Code:**

1. **def count\_letter(letter,word\_list):**
2. **count=0**
3. **for word in word\_list:**
4. **if letter in word:**
5. **count +=1**
6. **return count**
7. **word\_list=['apple','pears','orange','mango']**
8. **letter=input('Enter some alphabet symbol:')**
9. **letter\_count=count\_letter(letter,word\_list)**
10. **print(letter\_count)**

**If the user provides input 'a' then what is the result?**

* ​

1

* ​

2

* ​

3

* ​

4

**(Correct)**

**Explanation**

The above program prints the number of occurrences of specified alphabet symbol in the given word\_list.

Bottom of Form

Top of Form

Question 31: Skipped

**Consider the code:**

1. **startmsg = "hello"**
2. **endmsg = ""**
3. **for i in range(0,len(startmsg)):**
4. **endmsg = startmsg[i] + endmsg**
5. **print(endmsg)**

**What is the result?**

* ​

olleh

**(Correct)**

* ​

hello

* ​

IndexError

* ​

hlelo

**Explanation**

It is the program to reverse content of the given String.

Bottom of Form

Top of Form

Question 32: Skipped

**Consider the code:**

1. **x = [13,4,17,10]**
2. **w = x[1:]**
3. **u = x[1:]**
4. **y = x**
5. **u[0] = 50**
6. **y[1] = 40**
7. **print(x)**

**What is the result?**

* ​

[13, 40, 17, 10]

**(Correct)**

* ​

[50, 40, 10]

* ​

[13,4,17,10]

* ​

[50,40,17,10]

**Explanation**

In the above example both x and y are pointing to the same object.By using y if we are changing the content the changes will be reflected to x. By using slice operator a new object will be created.

Bottom of Form

Top of Form

Question 33: Skipped

**You want to add comments to your code so that other team members can understand it.**

**What should you do?**

* ​

Place the comments after the #sign on any line

**(Correct)**

* ​

Place the comments after the last line of the code separated by a blank line

* ​

Place the comments before the first line of code separated by a blank line

* ​

Place the comments inside parentheses anywhere

**Explanation**

If any line starts with #sign then it acts as python single line comment.

Bottom of Form

Top of Form

Question 34: Skipped

**We are creating a function to calculate the power of a number by using python.**

**We have to ensure that the function is documented with comments.**

**Consider the code(Line numbers included for reference):**

1. **01 # The calc\_power function calculates exponents**
2. **02 # x is the base**
3. **03 # y is the exponent**
4. **04 # The value of x raided to the y power is returned**
5. **05 def calc\_power(x, y):**
6. **06 comment="#Return the value"**
7. **07 return x\*\*y #raise x to the power y**

**Which of the following statements are true?**

* ​

Lines 01 through 04 will be ignored for syntax checking

**(Correct)**

* ​

The hash sign(#) is optional for lines 01 and 03.

* ​

The String in line 06 will be interpreted as a comment

**Explanation**

If any line starts with #sign then it acts as python single line comment.

Bottom of Form

Top of Form

Question 35: Skipped

**You are writing a python script to convert student marks into grade. The grades are defined as follows:**

1. **90 through 100 --> A grade**
2. **80 through 89 --> B grade**
3. **70 through 79 --> C grade**
4. **65 through 69 --> D grade**
5. **0 through 64 --> E grade**

**And developed application is :**

1. **# Grade Converter**
2. **marks=int(input('Enter Student Marks:'))**
3. **if marks >=90: #Line-1**
4. **grade='A'**
5. **elif marks>=80: #Line-2**
6. **grade='B'**
7. **elif marks>=70: #Line-3**
8. **grade='C'**
9. **elif marks>=65:**
10. **grade='D'**
11. **else:**
12. **grade='E'**
13. **print('Your grade is:',grade)**

**Which of the following changes should be performed to fulfill the requirement?**

* ​
  1. Line-1 should be replaced with
  2. if marks <= 90:
* ​
  1. Line-2 should be replaced with
  2. if marks>=80 and marks <= 90 :
* ​
  1. Line-3 should be replaced with
  2. if marks>=70 and marks <= 80 :
* ​

No Changes are  required.

**(Correct)**

**Explanation**

No changes are required for the above program. Make sure you should remember if condition fails then only elif will be executed.

Bottom of Form

Top of Form

Question 36: Skipped

**You are developing a Python application for an online product distribution company. You need the program to iterate through a list of products and escape when a target product ID is found.**

**Which of the following code can fulfill our requirement**

* ​
  1. productIdList=[0,1,2,3,4,5,6,7,8,9]
  2. index=0
  3. while index<len(productIdList):
  4. print(productIdList[index])
  5. if productIdList[index]==6:
  6. break
  7. else:
  8. index+=1

**(Correct)**

* ​
  1. productIdList=[0,1,2,3,4,5,6,7,8,9]
  2. index=1
  3. while index<len(productIdList):
  4. print(productIdList[index])
  5. if productIdList[index]==6:
  6. break
  7. else:
  8. index+=1
* ​
  1. productIdList=[0,1,2,3,4,5,6,7,8,9]
  2. index=0
  3. while index<len(productIdList):
  4. print(productIdList[index])
  5. if productIdList[index]==6:
  6. continue
  7. else:
  8. index+=1
* ​
  1. productIdList=[0,1,2,3,4,5,6,7,8,9]
  2. index=0
  3. while index<len(productIdList):
  4. print(productIdList[index])
  5. if productIdList[index]==6:
  6. break
  7. else:
  8. continue

**Explanation**

While developing the application, we have to consider the following syntactical things.  
The body of the while loop will be executed as long as condition is True.  
Inside while loop to break loop execution based on some condition, we should go for break statement. The index of the first element inside list is 0.  
Hence the following code fulfills our requirement  
productIdList=[0,1,2,3,4,5,6,7,8,9]  
index=0  
while index<len(productidlist):  
 print(productIdList[index])  
if productIdList[index]==6:  
break  
else:  
index+=1  
</len(productidlist):

Bottom of Form

Top of Form

Question 37: Skipped

**You are writing a python program that displays all prime numbers from 2 to 200. Which of the following is the proper code to fulfill our requirement?**

* ​
  1. p=2
  2. while p<=200:
  3. is\_prime=True
  4. for i in range(2,p):
  5. if p % i == 0:
  6. is\_prime=False
  7. break
  9. if is\_prime==True:
  10. print(p)
  11. p=p+1

**(Correct)**

* ​
  1. p=2
  2. is\_prime=True
  3. while p<=200:
  4. for i in range(2,p):
  5. if p % i == 0:
  6. is\_prime=False
  7. break
  9. if is\_prime==True:
  10. print(p)
  11. p=p+1
* ​
  1. p=2
  2. while p<=200:
  3. is\_prime=True
  4. for i in range(2,p):
  5. if p % i == 0:
  6. is\_prime=False
  7. break
  9. if is\_prime==False:
  10. print(p)
  11. p=p+1
* ​
  1. p=2
  2. while p<=200:
  3. is\_prime=True
  4. for i in range(2,p):
  5. if p % i == 0:
  6. is\_prime=False
  7. break
  9. if is\_prime==True:
  10. print(p)

**Explanation**

A positive integer greater than 1 which has no other factors except 1 and the number itself is called a prime number.  
2, 3, 5, 7 etc. are prime numbers as they do not have any other factors. But 6 is not prime (it is composite) since, 2 x 3 = 6.  
The following code fulfills our requirement  
p=2  
while p<=200:  
is\_prime=True  
for i in range(2,p):  
if p % i == 0:  
is\_prime=False  
break  
if is\_prime==True:  
print(p)  
p=p+1

Bottom of Form

Top of Form

Question 38: Skipped

**You created the following program to locate a conference room and display room name.**

1. **rooms={1:'Left Conference Room',2:'Right Conference Room'}**
2. **room=input('Enter the room number:')**
3. **if not room in rooms:#Line-3**
4. **print('Room does not exist')**
5. **else:**
6. **print('The room name is:'+rooms[room])**

**team reported that the program sometimes produces incorrect results.**

**You need to troubleshoot the program. Why does Line-3 Fails to find the rooms?**

* ​

Invalid Syntax

**(Correct)**

* ​

Mismatched data type(s)

* ​

Misnamed variable(s)

* ​

None of these

**Explanation**

To meet the requirement we have to write Line-3 as if room not in rooms:

Bottom of Form

Top of Form

Question 39: Skipped

**The XYZ Book Company needs a way to determine the cost that a student will pay for renting a Book.**

**The Cost is dependent on the time of the Book is returned.**

**However there are also special rates on Saturday and Sundays.**

**The Fee Structure is shown in the following list:**

**The cost is $3.00 per night.**

**If the Book is returned after 9PM, the student will be charged an extra day.**

**If the Book is rented on a Sunday,the student will get 50% off for as long as they the keep the book.**

**If the Book is rented on a Saturday,the student will get 30% off for as long as they keep the book.**

**We need to write the code to meet this requirements.**

1. **# XYZ Book Rented Amount Calculator**
2. **ontime=input('Was Book returned before 9 pm? y or n:').lower()**
3. **days\_rented=int(input('How many days was book rented?'))**
4. **day\_rented=input('What day the Book rented?').capitalize()**
5. **cost\_per\_day=3.00**
6. **if ontime == 'n':**
7. **days\_rented=days\_rented+1**
8. **if day\_rented=='Sunday':**
9. **total=(days\_rented\*cost\_per\_day)\*0.5**
10. **elif day\_rented=='Saturday':**
11. **total=(days\_rented\*cost\_per\_day)\*0.7**
12. **else:**
13. **total=(days\_rented\*cost\_per\_day)**
14. **print('The Cost of Book Rental is:$',total)**

**If the Book rented on 'Sunday',the number of days Book rented is 5 and Book returned after 9PM then what is the result?**

* ​

The Cost of Book Rental is:$ 7.0

* ​

The Cost of Book Rental is:$ 8.0

* ​

The Cost of Book Rental is:$ 9.0

**(Correct)**

* ​

The Cost of Book Rental is:$ 10.0

**Explanation**

If the Book rented on 'Sunday',the number of days Book rented is 5 and Book returned after 9PM then the following lines will be executed.  
cost\_per\_day=3.00  
days\_rented=days\_rented+1  
total=(days\_rented\*cost\_per\_day)\*0.5  
Hence the output will become 9.0 total.

Bottom of Form

Top of Form

Question 40: Skipped

**We are developing one school automation application. If student marks between 80 and 100,then we have to offer 'A' grade.**

**Which code block we have to use?**

* ​
  1. if marks>=80 and marks>=100:
  2. grade='A'
* ​
  1. if marks>=80 or marks<=100:
  2. grade='A'
* ​
  1. if 80<=marks<=100:
  2. grade='A'

**(Correct)**

* ​
  1. if marks>80:
  2. grade='A'

**Explanation**

Nesting of relational operators is allowed.  
a<=b<=c  
In Nesting, if all conditions are True then only result is True. If atleast one condition fails then the result is False  
if 80<=marks<=100:  
grade='A'  
If marks >=80 and marks<=100 then only grade will become 'A', which will meet our requirement.

Bottom of Form