MTA Final Document

Microsoft Python Exam Preparation
Mr. Yogesh P Murumkar
Bharat Soft Solutions
Machine Learning Data Science and Big Data

Youtube – https://www.youtube.com/c/yogeshmurumkar

1

Consider the code

```
import random
print(random.sample(range(10), 7))
```

Which of the following is valid?

- It will print list of 10 unique random numbers from 0 to 6 (Incorrect)
- It will print list of 7 unique random numbers from 0 to 9 (Correct)
- It will print list of 7 unique random numbers from 0 to 10
- It will print list of 7 unique random numbers from 1 to 10

Explanation

range(10) represents values from 0 to 9 only. Hence random.sample(range(10), 7) will print list of 7 unique random numbers from 0 to 9

2

Consider the following python code:

```
weight=62.4
zip='80098'
value=+23E4
```

The types of weight,zip and value variables respectively:

- float, str, str
- int, str, float **(Incorrect)**
- double, str, float
- float, str, float **(Correct)**

Explanation

weight=62.4 is of float type
zip='80098' is of str type
value=+23E4 is of float type

3

Consider the code:

```
start=input('How old were you at the time of joining?')
end=input('How old are you today?')
```

Which of the following code is valid to print Congratulations message?

- `print('Congratulations on '+ (int(end)-int(start))+' Years of Service!')`
- `print('Congratulations on '+ str(int(end)-int(start))+' Years of Service!')` (Correct)
- `print('Congratulations on '+ int(end-start)+' Years of Service!')`
- `print('Congratulations on '+ str(end-start)+' Years of Service!')`

Explanation

To use + operator for string types, compulsory both arguments must be str type, otherwise we will get error.
print('Congratulations on '+ (int(end)-int(start))+' Years of Service!')
TypeError: must be str, not int
print('Congratulations on '+ str(int(end)-int(start))+' Years of Service!')
print('Congratulations on '+ int(end-start)+' Years of Service!')
TypeError: unsupported operand type(s) for -: 'str' and 'str'
print('Congratulations on '+ str(end-start)+' Years of Service!')
TypeError: unsupported operand type(s) for -: 'str' and 'str'

4

You are developing a python application for your company.

A list named employees contains 600 employee names,the last 3 being company management.

You need to slice employees to display all employees excluding management.

Which two code segments we should use?

- [ ] employees[1:-2]
- [x] employees[:-3] **(Correct)**
- [ ] employees[1:-3]
- [ ] employees[0:-2]
- [ ] employees[0:-3] **(Correct)**

Explanation

list[begin:end] returns list of elements from begin index to end-1 index default value for begin is: 0

5

You are intern for XYZ Cars Company.You have to create a function that calculates the average velocity of vehicle on a 2640 foot(1/2 mile)track.

```
Consider the python code
distance=xxx(input('Enter the distance travelled in feet:')) #Line-1
distance_miles=distance/5280
time=yyy(input('Enter the time elapsed in seconds:')) #Line-2
time_hours=time/3600
velocity=distance_miles/time_hours
print('The average Velocity:',velocity,'miles/hour')
```

To generate most precise output, which modifications should be done at Line-1 and at Line-2.

- (●) xxx should be replaced with float and yyy should be replaced with float **(Correct)**
- ( ) xxx should be replaced with float and yyy should be replaced with int
- ( ) xxx should be replaced with int and yyy should be replaced with float
- ( ) xxx should be replaced with int and yyy should be replaced with int

Explanation

To get most precise output, we have to typecast into float, so that we won't miss fraction digits also.

6

You develop a Python application for your company. You required to accept input from the

user and print that information to the user screen.

Consider the code:

```
print('Enter Your FullName:')
#Line-1
print(fullname)
```

At Line-1 which code we have to write?

- `fullname=input`
- `input('fullname')` **(Incorrect)**
- `input(fullname)`
- `fullname=input()` **(Correct)**

Explanation

To get input from the keyboard, we have to use input() function. Hence the correct statement is: fullname=input()

7

The XYZ Company has hired you as an intern on the coding team that creates a e-commerce

application.You must write a script that asks the user for a value. The value must be

used as a whole number in a calculation,even if the user enters a decimal value.

Which of the following meets this requirement?

- `items=input('How many items you required?')`
- `items=float(input('How many items you required?'))` **(Incorrect)**
- `items=str(input('How many items you required?'))`
- `items=int(float(input('How many items you required?')))` **(Correct)**

Explanation

The return type of input() function is str by default. If we want to get only whole number from the given string, compulsory we have to type cast to int type. Hence the following is the correct statement we have to use.
If end user provides a float value and it is available in string form,to convert into whole number compulsory first we should convert into float and then into int.
items=int(float(input('How many items you required?')))

8

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:

```
sum=count=done=0
average=0.0
while(done != -1):
        rating=float(input('Enter Next Rating(1-5),-1 for done'))
        if rating == -1:
                break
        sum+=rating
        count+=1
average=float(sum/count)
#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))` (Incorrect)
- `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))

9

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:

```
name=input('Enter Your Name:')
score=0
count=0
sum=0
while(score != -1):
        score=int(input('Enter your scores: (-1 to end)'))
        if score==-1:
                break
        sum+=score
        count+=1
average_score=sum/count
#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))` **(Incorrect)**
- `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

Consider the following code:

```
n=[0,1,2,3,4,5,6,7,8,9]
i=0
while (i<10)#Line-1
   print(n[i])

   if n(i) = 6#Line-2
      break
   else:
          i += 1
```

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

- [ ] Line-1 should be replaced with
```
while(i<10):
```
(Correct)

- [x] Line-2 should be replaced with
```
if n[i]==6:
```
(Correct)

- [ ] Line-2 should be replaced with
```
if n[i]=6:
```

- [ ] Line-1 should be replaced with
```
while(i>0):
```

Explanation

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

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.

```
empno=input('Enter Your Employee Number(dd-ddd-dddd):')
parts=empno.split('-')
valid=False
if len(parts) == 3:
        if len(parts[0])==2 and len(parts[1])==3 and len(parts[2])==4:
                if parts[0].isdigit() and parts[1].isdigit() and parts[2].isdigit(
                        valid=True
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 **(Incorrect)**
- ○ 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.

12

You are coding a math utility by using python.

You are writing a function to compute roots

The function must meet the following requirements

If r is non-negative, return r**(1/s)

If r is negative and even, return "Result is an imaginary number"

if r is negative and odd,return -(-r)**(1/s)

Which of the following root function should be used?

```
def root(r,s):
        if r>=0:
                result=r**(1/s)
        elif r%2 == 0:
                result="Result is an imaginary number"
        else:
                result=-(-r)**(1/s)
        return result
```

(Correct)

```
def root(r,s):
        if r>=0:
                result=r**(1/s)
        elif r%2 != 0:
                result="Result is an imaginary number"
        else:
                result=-(-r)**(1/s)
        return result
```

(Incorrect)

```
def root(r,s):
        if r>=0:
                result=r**(1/s)
        if r%2 == 0:
                result="Result is an imaginary number"
        else:
                result=-(-r)**(1/s)
        return result
```

```
def root(r,s):
        if r>=0:
                result=r**(1/s)
        elif r%2 == 0:
                result=-(-r)**(1/s)
        else:
                result="Result is an imaginary number"
        return result
```

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

```
90 through 100 --> A grade
80 through 89 --> B grade
70 through 79 --> C grade
65 through 69 --> D grade
0 through 64 --> E grade
```

And developed application is :

```
# Grade Converter
marks=int(input('Enter Student Marks:'))
if marks >=90: #Line-1
        grade='A'
elif marks>=80: #Line-2
        grade='B'
elif marks>=70: #Line-3
        grade='C'
elif marks>=65:
        grade='D'
else:
        grade='E'
print('Your grade is:',grade)
```

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

- Line-1 should be replaced with
  ```
  if marks <= 90:
  ```

- Line-2 should be replaced with
  ```
  if marks>=80 and marks <= 90 :
  ```
  (Incorrect)

- Line-3 should be replaced with
  ```
  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.

14

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

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

(Correct)

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

(Incorrect)

```
products=[0,1,2,3,4,5,6,7,8,9]
index=0
while index<len(products):
        print(products[index])
        if products[index]==6:
                continue
        else:
                index+=1
```

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

15

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?

```
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
```

(Correct)

```
p=2
    is_prime=True
    while p<=200:
        for i in range(2,p):
                if p % i == 0:
                        is_prime=False
                        break

        if is_prime==True:
                print(p)
        p=p+1
```

(Incorrect)

```
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==False:
                print(p)
        p=p+1
```

```
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)
```

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

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

```
rooms={1:'Left Conference Room',2:'Right Conference Room'}
        room=input('Enter the room number:')
        if  not room in rooms:#Line-3
                print('Room does not exist')
        else:
                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) **(Incorrect)**
- Misnamed variable(s)
- None of these

Explanation

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

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.
XYZ Book Rented Amount Calculator

```
ontime=input('Was Book returned before 9 pm? y or n:').lower()
        days_rented=int(input('How many days was book rented?'))
        day_rented=input('What day the Book rented?').capitalize()
        cost_per_day=3.00
        if ontime == 'n':
                days_rented=days_rented+1
        if day_rented=='Sunday':
                total=(days_rented*cost_per_day)*0.5
        elif day_rented=='Saturday':
                total=(days_rented*cost_per_day)*0.7
        else:
                total=(days_rented*cost_per_day)
        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 **(Incorrect)**
- 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.

18

You are developing a Python application for online game.

You need to create a function that meets the following criteria:

The function is named update_score

The function receives the current score and a value.

The function adds the value to the current score.

The function returns the new score.

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

```
update_score(score,value):
        new_score=score+value
        return new_score
```

```
def update_score(score,value):
        new_score=score+value
        return new_score
```
(Correct)

```
def update_score(score,value):
        new_score=score+value
        pass new_score
```

```
def update_score():
        new_score=score+value
        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

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:

```
Line-1:
       name=input('Enter Your Name:')
       return name
 Line-2:
       calories=miles*calories_per_mile
       return calories
distance=int(input('How many miles did you bike this week:'))
burn_rate= 44
biker=get_name()
calories_burned=calc_calories(distance,burn_rate)
print(biker,", You burned about",calories_burned," calories")
```

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

- [ ]
```
Line-1 should be replaced with
  def get_name():
```
(Correct)

- [ ]
```
Line-1 should be replaced with
  def get_name(name):
```

```
Line-1 should be replaced with
  def get_name(biker):
```
(Incorrect)

```
Line-2 should be replaced with
  def calc_calories(miles,calories_per_mile):
```
(Correct)

```
Line-2 should be replaced with
  def calc_calories(miles,burn_rate):
```

```
Line-2 should be replaced with
  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

Question 20: **Incorrect**

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:

```
def get_rating(age):
        if age>=18:
                rating="A"
        elif age>=13:
                rating="T"
        else:
                rating="C"
        return rating
```

(Correct)

```
def get_rating(age):
        if age>=18:
                rating="A"
        if age>=13:
                rating="T"
        else:
                rating="C"
        return rating
```

(Incorrect)

```
def get_rating(age):
        if age>18:
                rating="A"
        elif age>13:
                rating="T"
        else:
                rating="C"
        return rating
```

```
def get_rating(age):
        if age>=18:
                rating="A"
        elif age>=13:
                rating="T"
        else:
                rating="C"
        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

Which of the following is False?

- A try statement can have one or more except clauses
- A try statement can have a finally clause without an except clause
- A try statement can have a finally clause and an except clause **(Incorrect)**
- A try statement can have one or more finally clauses **(Correct)**

Explanation

Every try statement should be associated with atmost one finally block.i.e we cannot take more than one finally block for the same try.

22

Consider the following code.

```
import os
def get_data(filename,mode):
        if os.path.isfile(filename):
                with open(filename,'r') as file:
                        return file.readline()
        else:
                return None
```

Which of the following are valid about this code?

- [ ] This function returns the first line of the file if it is available **(Correct)**
- [x] This function returns None if the file does not exist **(Correct)**
- [ ] This function returns total data present in the file
- [ ] This function returns last line of the file

Explanation

This function returns None if the file does not exist.If the file exist,then the function must return the first line.
os.path.isfile(filename) can be used to check whether the given file exists or not. If it exists returns True otherwise returns False.

We are writing a Python program for the following requirements:

Each line of the file must be read and printed

if the blank line encountered, it must be ignored

When all lines have been read,the file must be closed.

Consider the code:

```
inventory=open('inventory.txt','r')
eof=False
while eof == False:
        line=inventory.readline()
        if XXX:
                if YYY:
                        print(line,end='')

        else:
                print('End of file')
                eof=True
                inventory.close()
```

Which of the following changes are required to perform to meet the requirements

- (●)
```
XXX should be replaced with
line != ''
YYY should be replaced with
line!= '\n'
```
(Correct)

- ( )
```
XXX should be replaced with
line!= '\n'
YYY should be replaced with
line != ''
```

```
XXX should be replaced with
line != ''
YYY should be replaced with
line != ''
```

```
XXX should be replaced with
line!= '\n'
YYY should be replaced with
line!= '\n'
```

Explanation

\n represents blank line and if end of file then readline() method returns empty string. Hence
XXX should be replaced with
line != ''
YYY should be replaced with
line!= '\n'

24

You develop a python application for your school.

You need to read and write data to a text file. If the file does not exist,it must be created.

If the file has the content,the content must be removed.

Which code we have to use?

- `open('abc.txt','r')`
- `open('abc.txt','r+')` **(Incorrect)**
- `open('abc.txt','w+')` **(Correct)**
- `open('abc.txt','w')`

Explanation

If a file is available 'r+' mode will open a file in read and write mode and previouse data
will not be deleted.
If the file is not available it does not create a new file and raises 'FileNotFoundError'.

We are creating a function that reads a data file and prints each line of that file. Consider the following code:

```
import os
def read_file(filename):
        line=None
        if os.path.isfile(file_name):
                data=open(filename,'r')
        while line != '':
                line=data.readline()
                print(line)
```

The code attempts to read the file even if the file does not exist.

You need to correct the code. which lines having identation problems?

- ○ First 3 Lines inside function
- ◉ Last 3 Lines inside function (Correct)
- ○ Last 2 Lines inside function
- ○ There is no indentation problem

Explanation

The Last 3 lines having identation problem and the correct code is:
import os
def read_file(filename):
line=None
if os.path.isfile(filename):
data=open(filename,'r')
while line != '':
line=data.readline()
print(line)

Consider the code:

```
import sys
try:
        file_in=open('file1.txt','r')
        file_out=open('file2.txt','w+')
except IOError:
        print('cannot open',file_name)
else:
        i=1
        for line in file_in:
                print(line.rstrip())
                file_out.write(str(i)+":"+line)
                i=i+1
        file_in.close()
        file_out.close()
```

Assume that in.txt file is avaialble but out.txt file does not exist.

Which of the following is true about this code?

- This program will copy data from in.txt to out.txt **(Correct)**
- The code runs,but generates logical error **(Incorrect)**
- The code will generates a runtime error
- The code will generates a syntax error

Explanation

It is valid code and it is reading total data from the in.txt and writing to out.txt.

You are creating a function that manipulates a number.The function has the following requirements:

A float passed to the function

The function must take absolute value of the float

Any decimal points after the integer must be removed.

Which two math functions should be used?

- [ ] `math.frexp(x)`
- [x] `math.floor(x)` (Correct)
- [x] `math.fabs(x)` (Correct)
- [ ] `math.fmod(x)`
- [ ] `math.ceil(x)`

Explanation

fabs(x) Returns the absolute value of x
floor(x) Returns the largest integer less than or equal to x
Hence the following line will perform the required operation:
print(floor(fabs(-123.456)))

28

You are writing an application that uses the pow function. The program must reference

the function using the name power.

Which of the following import statement required to use?

- ○ import math.pow as power
- ◉ import pow from math as power **(Incorrect)**
- ○ from math import pow as power **(Correct)**
- ○ from math.pow as power

Explanation

The following is the valid syntax: from math import pow as power

29

You are writing code that generates a random integer with a minimum value of 18 and maximum value of 32. Which of the following 2 functions we required to use?

- ☐ random.randint(18,33)
- ☑ random.randint(18,32) **(Correct)**
- ☐ random.randrange(18,33,1) **(Correct)**
- ☐ random.randrange(5,11,1)

Explanation

1. randint(begin,end) generates a random int value between given 2 numbers(boundaries are inclusive)
2. randrange([start],stop,[step])
returns a random number from the range
start<= x

30

Consider the lists:

```
n=[10,20,30,40,50]
a=['a','b','c','d','e']
print( n is a)
print( n == a)
n = a
print( n is a)
print( n == a)
```

What is the result?

- False
  False
  True
  True (Correct)

- False
  True
  False
  True (Incorrect)

- True
  False
  True
  False

- False
  True
  True
  True

Explanation

'is' operator is always meant for reference comparison and == operator always meant for content comparison.

Consider the code

```
a=15
b=5
print(a/b)
```

What is the result ?

- 3
- 3.0 **(Correct)**
- 0
- 0.0

Explanation

/ always meant for floating point arithmetic

32

You are writing a python program that evaluates an arithmetic expression.

The expression is described as y is equals x multiplied by negative one,then raised to the second power,where x is the value which will be input and y is result.

```
x = eval(input('Enter a number for the expression:'))
```

Which of the following is valid expression for the given requirement?

- `y = (x-)**2`
- `y = -(x)**2` **(Incorrect)**
- `y = (-x)**2` **(Correct)**
- `y = (x)**-2`

Explanation

y = (-x)**2
y is equals x multiplied by negative one,then raised to the second power

33

Question 33: Incorrect

Consider the following expression

```
result=(2*(3+4)**2-(3**3)*3)
```

What is result value?

- 17 (Correct)
- 16 (Incorrect)
- 18
- 19

Explanation

Python Virtual Machine will give the precedence in the following order
1. Parenthesis
2. Exponent
3. Multiplication,Division,Modulo,Floor Division
4. Addition,Subtraction

etc

result=(2*(3+4)**2-(3**3)*3)=(2*(7)**2-(27)*3)=2*49-27*3=98-81=17

34

Consider the following code segments

```
#Code Segment-1
a1='10'
b1=3
c1=a1*b1
#Code Segment-2
a2=10
b2=3
c2=a2/b2
#Code Segment-3
a3=2.6
b3=1
c3=a3/b3
```

After executing Code Segments 1,2 and 3 the result types of c1,c2 and c3 are:

- c1 is of str type,c2 is of int type ,c3 is of float type
- c1 is of str type,c2 is of float type ,c3 is of float type **(Correct)**
- c1 is of str type,c2 is of int type ,c3 is of int type
- c1 is of str type,c2 is of str type ,c3 is of str type

Explanation

The value of c1 is '101010', which is str type
The value of c2 is 3.3333, which is float type
The value of c3 is 2.6,which is float type

35

Which of the following is valid python operator precedence order?

- Parenthesis
  Exponents
  Unary Positive,Negative and Not
  Addition and Subtraction
  Multiplication and Division
  And

- Exponents
  Parenthesis
  Unary Positive,Negative and Not
  Multiplication and Division
  Addition and Subtraction
  And
  (Incorrect)

- Exponents
  Unary Positive,Negative and Not
  Multiplication and Division
  Addition and Subtraction
  And
  Parenthesis

- Parenthesis
  Exponents
  Unary Positive,Negative and Not
  Multiplication and Division
  Addition and Subtraction
  And
  (Correct)

Explanation

The following is the correct order of Python Operator Precedence
Parenthesis
Exponents
Unary Positive,Negative and Not
Multiplication and Division
Addition and Subtraction
And

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 **(Incorrect)**
- 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.

37

We are creating a function to calculate the addition of two numbers by using python.

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

Consider the code(Line numbers included for reference):

```
01 # The calc_square function calculates exponents
02 # x is a first number
03 # y is a second number
04 # The value of x and y is added and returned
05 def calc_power(x, y):
06       comment="#Return the value"
07       return x + y # Adding x and 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. **(Incorrect)**
- ○ 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.

Consider the expession:

```
result=a-b*c+d
```

Which of the following are valid?

- [ ] First b*c will be evaluated followed by subtraction and addition (Correct)
- [x] First b*c will be evaluated followed by addition and subtraction (Incorrect)
- [ ] First a-b will be evaluated followed by multiplication and addition
- [ ] The above expession is equivalent to a-(b*c)+d (Correct)

Explanation

multiplication having more precedence than addition and subtraction. addition and subtraction having same precedence.

39

Which of the following is True about else block?

- ( ) else block will be executed if there is no exception in try block
- (•) Without writing except block we cannot write else block (Incorrect)
- ( ) For the same try we can write atmost one else block
- ( ) All the above (Correct)

Explanation

else block will be executed if there is no exception in try block Without writing except block we cannot write else block For the same try we can write atmost one else block.i.e more than one else block we cannot take.

40

Consider the code

```
try:
        print('try')
        print(10/0)
except:
        print('except')
else:
        print('else')
finally:
        print('finally')
```

What is the result?

- try
  except
  else
  finally

- try
  else
  finally

- try
  except **(Correct)**
  finally

- try
  finally

Explanation

except block will be executed if there is an exception in try block.
else block will be executed if there is no exception in try block.
finally block will be executed always whether exception raised or not raised and whether handled or not handled.