# Microsoft Python Certification Exam (98-381) Question Bank with Answers and Explanations


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

# INDEX

## Exam Pattern:

**Number of Questions : 40**
**Duration : 45 Minutes**
**Questions Type: Multiple Choice, Drag and Drop and Selection from drop down list**

## Syllabus:

### 1) Perform Operations using Data Types and Operators (20-25%)

- Evaluate an expression to identify the data type Python will assign to each variable
  - Identify st, int, float, and bool data types
- Perform data and data type operations
  - Convert from one data type to another type; construct data structures; perform indexing and slicing operations
- Determine the sequence of execution based on operator precedence
  - Assignment; Comparison; Logical; Arithmetic; Identity (is); Containment (in)
- Select the appropriate operator to achieve the intended result
  - Assignment; Comparison; Logical; Arithmetic; Identity (is); Containment (in)

### 2) Control Flow with Decisions and Loops (25-30%)

- Construct and analyze code segments that use branching statements
  - if; elif; else; nested and compound conditional expressions
- Construct and analyze code segments that perform iteration
  - while; for; break; continue; pass; nested loops and loops that include compound conditional expressions

### 3) Perform Input and Output Operations (20-25%)

- Construct and analyze code segments that perform file input and output operations
  - Open; close; read; write; append; check existence; delete; with statement
- Construct and analyze code segments that perform console input and output operations
  - Read input from console; print formatted text; use of command line arguments

### 4) Document and Structure Code (15-20%)

- Document code segments using comments and documentation strings
  - Use indentation, white space, comments, and documentation strings; generate documentation by using pydoc
- Construct and analyze code segments that include function definitions
  - Call signatures; default values; return; def; pass


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

## 5) Perform Troubleshooting and Error Handling (5-10%)

- Analyze, detect, and fix code segments that have errors
  - Syntax errors; logic errors; runtime errors
- Analyze and construct code segments that handle exceptions
  - Try; except; else; finally; raise

## 6) Perform Operations Using Modules and Tools (1-5%)

- Perform basic operations using built-in modules
  - Math; datetime; io; sys; os; os.path; random
- Solve complex computing problems by using built-in modules
  - Math; datetime; random


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

# Sample Score Report

Microsoft

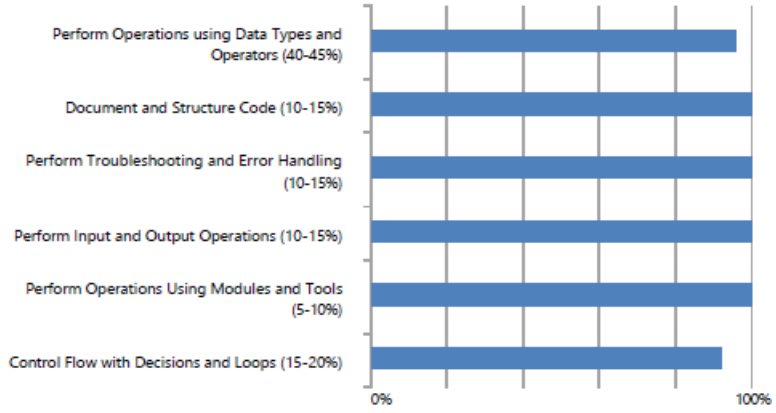
**98-381**
**Introduction to Programming Using Python**

**Candidate:** XXXXXXXXXXXXXXXXXXXXXXXXX
**Candidate ID:** M[illegible]60
**Result:** PASS
**Passing Score:** 70
**Your Score:** 97

Date: [illegible]
Site Number: [illegible]
Registration: [illegible]

**Performance by exam section**

Each section, and its corresponding percentage of the exam, appears to the left of the chart. The length of the bars represents your section-level performance. Shorter bars reflect weaker performance, and longer bars reflect stronger performance.





DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

# Sample Certificate




DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

# Topic

# Data Types and Operators


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

# Exam Objective: Perform Operations using Data Types and Operators (40-45%)

**1. Evaluate an expression to identify the data type Python will assign to each variable**
**Identify str, int, float, and bool data types**
**2. Perform data and data type operations**
**Convert from one data type to another type; construct data structures; perform indexing and slicing operations**

**3. Determine the sequence of execution based on operator precedence**
**Assignment; Comparison; Logical; Arithmetic; Identity (is); Containment (in)**
**4. Select the appropriate operator to achieve the intended result**
**Assignment; Comparison; Logical; Arithmetic; Identity (is); Containment (in)**

---

**Q1. Consider the following python code:**

```
1) age=0
2) minor=False
3) name='Durga'
```

**The types of age,minor and name variables respectively:**

**A) int, bool, str**
**B) bool, bool, str**
**C) int, bool, char**
**D) float, bool, str**

**Answer: A**

**Explanation:**

1) age=0 is of int type
2) minor=False is of bool type
3) name='Durga' is of str type

**Q2. Consider the following python code:**

```
1) weight=62.4
2) zip='80098'
3) value=+23E4
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

**The types of weight,zip and value variables respectively:**

**A)float, str, str**
**B)int, str, float**
**C)double, str, float**
**D)float, str, float**

**Answer: D**

**Explanation:**

1) **weight=62.4 is of float type**
2) **zip='80098' is of str type**
3) **value=+23E4 is of float type**

**Q3. You are writing a Python program to read two int values from the keyboard and print the sum.**

```
1) x=input('Enter First Number:')
2) y=input('Enter Second Number:')
3) #Line-1
```

**Which of the following code we have to write at Line-1 to print sum of given numbers?**

**A) print('The Result:'+(int(x)+int(y)))**
**B) print('The Result:'+(int(x+y)))**
**C) print('The Result:'+str(int(x)+int(y)))**
**D) print('The Result:'+str(int(x+y)))**

**Answer:C**

**Explanation:**
**To use + operator for string types, compulsory both arguments must be str type, otherwise we will get error.**

**A) print('The Result:'+(int(x)+int(y))) #TypeError: must be str, not int**
**B) print('The Result:'+(int(x+y))) #TypeError: unsupported operand type(s) for +: 'str' and 'str'**
**C) print('The Result:'+str(int(x)+int(y))) #valid**
**D) print('The Result:'+str(int(x+y)))**


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

**Q4. Consider the code:**

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

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

**A) print('Congratulations on '+ (int(end)-int(start))+' Years of Service!')**
**B) print('Congratulations on '+ str(int(end)-int(start))+' Years of Service!')**
**C) print('Congratulations on '+ int(end-start)+' Years of Service!')**
**D) print('Congratulations on '+ str(end-start)+' Years of Service!')**

**Answer: B**

**Explanation:**
**To use + operator for string types, compulsory both arguments must be str type, otherwise we will get error.**

**A) print('Congratulations on '+ (int(end)-int(start))+' Years of Service!')**
**TypeError: must be str, not int**
**B) print('Congratulations on '+ str(int(end)-int(start))+' Years of Service!')**
**C) print('Congratulations on '+ int(end-start)+' Years of Service!')**
**TypeError: unsupported operand type(s) for -: 'str' and 'str'**
**D) print('Congratulations on '+ str(end-start)+' Years of Service!')**
**TypeError: unsupported operand type(s) for -: 'str' and 'str'**

**Q5. You are writing a Python program. You required to handle data types properly. Consider the code segment:**

```
1) a=10+20
2) b='10'+'20'
3) c='10'*3
```

**Identify the types of a,b and c?**
**A) a is of int type,b is of str type and c is of str type**
**B) a is of int type,b is of str type and c is of int type**
**C) a is of int type,b is of int type and c is of int type**
**D) a is of int type ,b and c are invalid declarations**

**Answer: A**

**Explanation:**
**If we apply + operator between two int types the result is of int type.**
**a=10+20**
**Hence a is int type.**

If we apply + operator between two str types, then the result is of str type.
b='10'+'20'
Hence b is str type
c='10'*3
If we apply * operator for 'str' and int type then it acts as string repetition operator. Hence the result is of str type.

Q6. 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?

A) employees[1:-2]
B) employees[:-3]
C) employees[1:-3]
D) employees[0:-2]
E) employees[0:-3]

Answer: B and E

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

Q7. You are developing a python application for your company.
A list named employees contains 500 employee names,the last 3 being company management. Which of the following represents only management employees.

A) employees[497:]
B) employees[-3:]
C) employees[497:500]
D) All the above

Answer: D

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

Q8. You are developing a python application for your company.
A list named employees contains 500 employee names.
In which cases we will get IndexError while accessing employee data?

A) employees[1:1000]
B) employees[-10:10]

C) employees[0:501]
D) None of the above

Answer: D
Explanation : Slice Operator never raises IndexError

Q9. You are developing a python application for your company.
A list named employees contains 500 employee names.
In which cases we will get IndexError while accessing employee names?

A) employees[0]
B) employees[500]
C) employees[-1]
D) None of the above

Answer: B

Explanation: If we are trying to access list elements with out of range index, then we will get IndexError.
In the above example, list contains 500 names and hence valid positive index range is 0 to 499 and negative index range is -1 to -500.
employees[500]=====>IndexError

Q10. Consider the list:

list=['Apple','Banana','Carrot','Mango']
Which of the following are valid ways of accessing 'Mango':

A) list[0]
B) list[-1]
C) list[4]
D) list[3]

Answer: B and D

Explanation: Python supports both positive and negative index.
Positive index is from left to right and range is : 0 to length-1.
Negative index is from right to left and range is : -1 to -length.
In the above example we can access 'Mango' by using list[3] or list[-1]

Q11. Consider the following lists:

```
1) n1=[10,20,30,40,50]
2) n2=[10,20,30,40,50]
3) print(n1 is n2)
```

```
4)  print(n1 == n2)
```

**What is the output?**

**A)**
**True**
**True**

**B)**
**False**
**False**

**C)**
**False**
**True**

**D)**
**True**
**False**

**Answer: C**

**Explanation: 'is' operator is always meant for reference comparison and == operator always meant for content comparison.**
**In the above example n1 and n2 are different objects but with same content.**
**Hence 'is' operator returns False and '==' operator returns True**

**Q12. Consider the following lists:**

```
1)  n1=[10,20,30,40,50]
2)  n2=[10,20,30,40,50]
3)  print(n1 is n2)
4)  print(n1 == n2)
5)  n1=n2
6)  print(n1 is n2)
7)  print(n1 == n2)
```

**What is the result?**

**A)**
**False**
**False**
**True**
**True**

**B)**
**False**
**True**
**False**
**True**

**C)**
**True**
**False**
**True**
**False**

**D)**
**False**
**True**
**True**
**True**

**Answer: D**

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

**Q13. Consider the lists:**

```
1) numbers=[10,20,30,40,50]
2) alphabets=['a','b','c','d','e']
3) print( numbers is alphabets)
4) print( numbers == alphabets)
5) numbers=alphabets
6) print( numbers is alphabets)
7) print( numbers == alphabets)
```

**What is the result?**

**A)**
**False**
**False**
**True**
**True**

**B)**
**False**
**True**

False
True

C)
True
False
True
False

D)
False
True
True
True

Answer: A

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

Q14. Consider the code

```
1) a=15
2) b=5
3) print(a/b)
```

What is the result ?

A) 3
B) 3.0
C) 0
D) 0.0

Answer: B

Explanation:
/ always meant for floating point arithmetic

Q15. Consider the code

```
1) a=21
2) b=6
3) print(a/b)
4) print(a//b)
```

5) print(a%b)

What is the result?
A)
3
3
3

B)
3.5
3
3

C)
3.0
3
3
D)
3.5
3.5
3

Answer: B

Explanation: division operator in python always meant for floating point arithmetic. Hence a/b returns 3.5
But floor division(//) operator can perform both integral and floating point arithmetic. If the arguments are int type then the result is int type and if the arguments are float type then the result is float type. Hence a//b returns 3. a%b returns the remainder which is 3.

Q16. You are writing a python program that evaluates an arithmetic expression.
The expression is described as b is equals a multiplied by negative one,then raised to the second power,where a is the value which will be input and b is result.

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

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

A) b = (a-)**2
B) b = -(a)**2
C) b = (-a)**2
D) b = (a)**-2

Answer: C

**Explanation:**
**b = (-a)**2**
**b is equals a multiplied by negative one,then raised to the second power**

**Q17. Consider the following expression**
**result=(2*(3+4)**2-(3**3)*3)**
**What is result value?**
**A)17**
**B)16**
**C)18**
**D)19**

**Answer: A**

**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**

**Q18. Consider the expession:**
**result=a-b*c+d**
**Which of the following are valid?**
**A) First b*c will be evaluated followed by subtraction and addition**
**B) First b*c will be evaluated followed by addition and subtraction**
**C) First a-b will be evaluated followed by multiplication and addition**
**D) The above expession is equivalent to a-(b*c)+d**

**Answers: A and D**

**Explanation: multiplication having more precedence than addition and subtraction. addition and subtraction having same precedence.**

**Q19. Consider the following code segments**

```
1) #Code Segment-1
2) a1='10'
3) b1=3
4) c1=a1*b1
5)
```

```
#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:

A. c1 is of str type,c2 is of int type ,c3 is of float type
B. c1 is of str type,c2 is of float type ,c3 is of float type
C. c1 is of str type,c2 is of int type ,c3 is of int type
D. c1 is of str type,c2 is of str type ,c3 is of str type

Answer: B

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

Q20. Which of the following is valid python operator precedence order?

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

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

C)
Exponents
Unary Positive,Negative and Not

**Multiplication and Division**
**Addition and Subtraction**
**And**
**Parenthesis**

**D)**
**Parenthesis**
**Exponents**
**Unary Positive,Negative and Not**
**Multiplication and Division**
**Addition and Subtraction**
**And**

**Answer : D**
**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**

**Q21. You have the following code:**

```
1) a=bool([False])
2) b=bool(3)
3) c=bool("")
4) d=bool(' ')
```

**Which of the variables will represent False:**
**A) a**
**B) b**
**C) c**
**D) d**

**Answer: C**

**Explanation:**

**For Empty String, Empty List,Empty tuple,Empty set,Empty dict and range(0) arguments bool() function returns False.**
**c=bool("")**
**As the argument is empty string, it represents False.**

**Q22. Consider the following variable declarations:**

1) a= bool([])
2) b= bool(())
3) c= bool(range(0))
4) d= bool({})
5) e= bool(set())

**Which of the above variables represent True ?**

**A) c**
**B) a ,b, c, d**
**C) All Variables represent True**
**D) None of the variables represents True**

**Answer: D**

**Explanation:**

**For Empty String, Empty List,Empty tuple,Empty set,Empty dict and range(0) arguments bool() function returns False.**

**Q23.**

1) a=bool(0)
2) b=bool(3)
3) c=bool(0.5)
4) d=bool(0.0)

**Which variables represent True?**
**A) a,b**
**B) b,c**
**C) c,d**
**D) d,a**
**E) All Variables**

**Answer: B**
**Explanation:**
**In the case of integral values 0 treated as False and non-zero treated as True. In the case of float values 0.0 treated as False and all other values (non-zero values) treated as True**


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

**Q24. You have the following code:**

```
1) a=3
2) b=5
3) a += 2**3
4) a -=b//2//3
5) print(a)
```

**What is the result?**

**A)13**
**B)12**
**C)11**
**D)10**

**Answer : C**

**a+= 2**3**
**a = (a)+(2**3) =a+8=3+8=11**

**a -=b//2//3**
**a = (a) - (b//2//3) =(a) - (5//2//3) =a-(2//3)=a-0=11**

**Q25. Consider the following expression**

```
1) result=8//6%5+2**3-2
2) print(result)
```

**What is the result?**

**A) 6**
**B) 7**
**C) 8**
**D) 9**

**Answer: B**

**Explanation:**

```
1) 2**3=8
2) 8//6=1
3) result=1%5+8-2=1+8-2=9-2=7
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

**Q26. Which of the following expression will generate max value?**

**A)8%3*4**
**B)8-3*4**
**C)8//3*4**
**D)8/3*4**

**Answer: D**

**Explanation:**

1) **8%3*4--->8**
2) **8-3*4 -->-4**
3) **8//3*4--->8**
4) **8/3*4--->10.6666**

**Q27. Consider the code**

1) **a=2**
2) **a += 1**
3) **# Line-1**

**To make a value as 9,which expression required to place at Line-1**
**A) a*=2**
**B) a**=2**
**C) a+=2**
**D) a-=2**

**Answer: B**
**Explanation:**

1) **a*=2--->6**
2) **a**=2---->9**
3) **a+=2----->5**
4) **a-=2--->1**

**Q28. Consider the python code**

1) **a=1**
2) **b=3**
3) **c=5**
4) **d=7**

**In Which of the following cases the result value is 0?**
**A) result = a+b*2**

B) result = a%b-1
C) result = a-b//d
D) result = a**d-1

Answer: B and D

Explanation:

1) a+b*2--->7
2) a%b-1-->0
3) a-b//d--->1
4) a**d-1--->0

Q29. In which of the following cases we will get same result

A) 23%5
B) 3**1
C) 11/3
D) 13//4

Answer: A,B,D

Explanation:

1) 23%5-->3
2) 3**1--->3
3) 11/3--->3.6666
4) 13//4--->3

Q30. Consider the code

1) a=1
2) b=2
3) c=4
4) d=6

Which of the following expression results -4?

A) (a+b)//c%d
B) (b+c)//a%d
C) (a+b)//c*d
D) (a+b)//d-c

Answer: D

**Explanation:**

1) (a+b)//c%d--->0
2) (b+c)//a%d--->0
3) (a+b)//c*d---->0
4) (a+b)//d-c---->-4

**Q31.**

```
1) subjects=['java','python','sap']
2) more_subjects=['java','python','sap']
3) extra_subjects=more_subjects
```

**In which cases True will be printed to the console?**

**A) print(extra_subjects is more_subjects)**
**B) print(subjects is more_subjects)**
**C) print(subjects is extra_subjects)**
**D) print(subjects == extra_subjects)**

**Answer: A and D**

**Explanation:**
**We can use 'is' operator for reference comparison where as == operator for content comparison.**

```
1) print(extra_subjects is more_subjects) #True
2) print(subjects is more_subjects) #False
3) print(subjects is extra_subjects)#False
4) print(subjects == extra_subjects)#True
```

**Q32. Consider the python code**

```
1) numbers=[10,20,30,40]
2) x=0
```

**In which of the following cases 10 will be printed to the console?**

**A)**

```
1) for i in (30,40,50):
2)         if i in numbers:
3)                 x=x+5
4) print(x)
```

B)

```
1)  for i  in (30,40,50):
2)          if i not in numbers:
3)                  x=x+5
4)  print(x)
```

C)

```
1)  for i  in (30,40,50):
2)          if i not in numbers:
3)                  x=x+10
4)  print(x)
```

D)

```
1)  for i  in (30,40,50):
2)          if i in numbers:
3)                  x=x+10
4)  print(x)
```

Answer: A and C

Explanation:

---

```
1)  for i in (30,40,50):
2)          if i in numbers:
3)                  x=x+5
4)  print(x) #10
```

---

```
1)  for i  in (30,40,50):
2)          if i not in numbers:
3)                  x=x+5
4)  print(x)#5
```

---

```
1)  for i  in (30,40,50):
2)          if i not in numbers:
3)                  x=x+10
4)  print(x)#10
```

---


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

```
1) for i  in (30,40,50):
2)         if i in numbers:
3)                 x=x+10
4) print(x) #20
```

**Q33. Which of the following code snippet will produce the output:**

**Boy**
**Cat**
**Dog**

**A)**

```
1) l=['Apple','Boy','Cat','Dog']
2) for x in l:
3)         if len(x) == 3:
4)                 print(x)
```

**B)**

```
1) l=['Apple','Boy','Cat','Dog']
2) for x in l:
3)         if len(x) != 3:
4)                 print(x)
```

**C)**

```
1) l=['Apple','Boy','Cat','Dog']
2) for x in l:
3)         print(x)
```

**D)**

```
1) l=['Apple','Boy','Cat','Dog']
2) l1=l[1:]
3) for x in l1:
4)         print(x)
```

**Answer: A and D**

**Explanation:**
---

```
1) l=['Apple','Boy','Cat','Dog']
2) for x in l:
```

```
3)          if len(x) == 3:
4)                  print(x)
```

o/p:
Boy
Cat
Dog
---

```
1)  l=['Apple','Boy','Cat','Dog']
2)  for x in l:
3)          if len(x) != 3:
4)                  print(x)
```

o/p: Apple
---

```
1)  l=['Apple','Boy','Cat','Dog']
2)  for x in l:
3)          print(x)
```

o/p:
Apple
Boy
Cat
Dog
---

```
1)  l=['Apple','Boy','Cat','Dog']
2)  l1=l[1:]
3)  for x in l1:
4)          print(x)
```

o/p:
Boy
Cat
Dog

Q34. Consider the Python code:

```
1)  a=5
2)  b=10
3)  c=2
4)  d=True
5)
```

```
6)  x=a+b*c
7)  y=a+b/d
8)
9)  if(condition):
10)         print('Valid')
11) else:
12)         print('invalid')
```

**To print 'Valid' to the console, which condition we have to take for if statement?**

**A) x<y**
**B) x<=y**
**C) x>y**
**D) x==y**

**Answer: C**

**Explanation:**

```
1)  a=5
2)  b=10
3)  c=2
4)  d=True
```

```
1)  x=a+b*c=5+10*2=5+20=25
2)  y=a+b/d=5+10/1=5+10.0=15.0
```

**To print valid the condition should be True.It is possible if condition is x>y.**

```
1)  x<y===>False
2)  x<=y===>False
3)  x>y===>True
4)  x==y===>False
```

**Q35. Consider the following code**

```
1)  x= 'Durga'
2)  y= 'Durga'
3)  result=condition
4)  print(result)
```

**For which of the following condition True will be printed to the console?**

A) x is y
B) x is not y
C) x != y
D) x < y

Answer: A

Both x and y pointing to the same object.
Hence 'x is y' returns True. Except that all remaining cases returns False.

1) x is y==>True
2) x is not y===>False
3) x != y==>False
4) x < y===>False

Q36. Consider the code:

```
1) x= 8
2) y= 10
3) result= x//3*3/2+y%2**2
4) print(result)
```

What is the result?
A) 5
B) 5.0
C) 6.0
D) 7.0

Answer: B

Explanation:

1) x//3*3/2+y%2**2
2) x//3*3/2+y%4
3) 8//3*3/2+y%4
4) 2*3/2+y%4
5) 6/2+y%4
6) 3.0+10%4
7) 3.0+2
8) 5.0


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

**Q37. Consider the code**

```
1) s='AB CD'
2) list=list(s)
3) list.append('EF')
4) print(list)
```

**What is the result?**
**A)('A','B',' ','C','D','EF')**
**B)['A','B','C','D','EF']**
**C)['A','B','C','D','E','F']**
**D)['A','B',' ','C','D','EF']**
**E){'A','B',' ','C','D','EF'}**

**Answer: D**

**Explanation:**
**List elements will be printed within square brackets.**
**Whenever we are converting string to list,each character will become element of List including space also.**

**Q38. Consider the code:**

```
1) x='ACROTE'
2) y='APPLE'
3) z='TOMATO'
```

**Which of the following won't print 'CAT' to the console**

**A) print(x[1]+y[0]+z[0])**
**B) print(x[2]+y[1]+z[1])**
**C) print(x[-5]+y[0]+z[0])**
**D) print(x[-5]+y[0]+z[-2])**

**Answer: B**

**Explanation:**

```
1) print(x[1]+y[0]+z[0]) #CAT
2) print(x[2]+y[1]+z[1]) #RPO
3) print(x[-5]+y[0]+z[0]) #CAT
4) print(x[-5]+y[0]+z[-2]) #CAT
```

**Q39. Consider the code:**

```
1) s='Python is easy'
2) s1=s[-7:]
3) s2=s[-4:]
4) print(s1+s2)
```

**What is the result?**
**A)is easyeasy**
**B)easyeasy**
**C)iseasyeasy**
**D)s easyeasy**
**E)is easy easy**

**Answer: A**

**Explanation:**

1) s1 = s[-7:]===>'is easy'
2) s2 = s[-4:]===>'easy'
3) print(s1+s2)===>'is easyeasy'

**Q40. Consider the code:**

```
1) s='Python is easy'
2) s1=s[6:-4]
3) #Line-1
4) print(len(s2))
```

**To print 2 as output,which code we have to insert at Line-1**

**A)s2 = s1.lstrip()**
**B)s2 = s1.rstrip()**
**C)s2 = s1.lrstrip()**
**D)s2 = s1.strip()**

**Answer: D**

**Explanation:**

1) strip()==>It will remove spaces present at left and right sides of the string
2) lstrip()==>It will remove spaces present at only left side of the string
3) rstrip()==>It will remove spaces present at only right side of the string


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

**There is no method like lrstrip().**
**s2 = s1.lrstrip()**
**AttributeError: 'str' object has no attribute 'lrstrip'**

**Q41. Consider the code**
**t=([10,20],10,False)**

**Which line of the code assigns <class 'list'> to x**

**A) x= type(t)**
**B) x= type(t[0])**
**C) x= type(t[1])**
**D) x= type(t[0:])**

**Answer: B**

**Explanation:**

1) **x= type(t)===> <class 'tuple'>**
2) **x= type(t[0])====><class 'list'>**
3) **x= type(t[1])===><class 'int'>**
4) **x= type(t[0:])===><class 'tuple'>**

**Q42. Consider the variable declaration**
**b = 'BANANA'**

**Which of the following lines will print 'AA' to the console?**

**A) print(b[1]+b[2])**
**B) print(b[1]+b[3])**
**C) print(b[1]+b[5])**
**D) print(b[3]+b[5])**

**Answer: B,C and D**

**Explanation:**

1) **print(b[1]+b[2])==>AN**
2) **print(b[1]+b[3])==>AA**
3) **print(b[1]+b[5])==>AA**
4) **print(b[3]+b[5])==>AA**

**Q43. Consider the Variable declarations:**

```
a='5'
b='2'
```

**Which of the following expressions are of type str**

**A) a+b**
**B) a*b**
**C) a-b**
**D) a*2**

**Answer: A and D**
**a+b-->str type**
**a*b-->TypeError: can't multiply sequence by non-int of type 'str'**
**a-b-->TypeError: unsupported operand type(s) for -: 'str' and 'str'**
**a*2-->str type**

**Q44. Which expression would evaluate to 2?**

**A) 3**2**
**B) 22%5**
**C) 13//4**
**D) 11/2**

**Answer: B**

**Explanation:**

1) **3**2==>9**
2) **22%5===>2**
3) **13//4==>3**
4) **11/2===>5.5**

**Q45. Consider the code**

```
a=7
b=3
c=5
d=1
```

**Which line of the code assigns 9 to the output?**

**A) output=a%c+1**
**B) output=a+c//d**
**C) output=c*d-1**


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

**D) output=a+d*2**

**Answer: D**

**Explanation:**

1) **output=a%c+1===>3**
2) **output=a+c//d====>12**
3) **output=c*d-1====>4**
4) **output=a+d*2===>9**

**Q46. Consider the code**

1) **x=3**
2) **x +=1**
3) **#Line-1**

**Which line should be inserted at Line-1 so that x value will become 16?**

**A)x+=2**
**B)x-=2**
**C)x*=2**
**D)x**=2**

**Answer: D**

**Explanation:**

1) **x+=2===>6**
2) **x-=2===>2**
3) **x*=2===>8**
4) **x**=2==>16**

**Q47. In which of the following cases, True will be printed to the console ?**

**A)**

1) **a=45**
2) **b=45**
3) **print(a is not b)**

**B)**

1) **s1='The Python Course'**
2) **s2='The Python Course'.upper()**

```
3) print(s1 is s2)
```

C)

```
1) x=[1,2,3]
2) y=[1,2,3]
3) print(x is y)
```

D) print('r' in 'durga')

E) print('is' in 'This IS a Fake News')

Answer: D,E

Explanation:

```
1) a=45
2) b=45
3) print(a is not b) #False
```

Both a and b pointing to the same object

```
-----------------------------------------
s1='The Python Course'
s2='The Python Course'.upper()
print(s1 is s2)#False
```

s1 and s2 are not pointing to the same object

```
-----------------------------------------------
x=[1,2,3]
y=[1,2,3]
print(x is y)#False
```

x and y are not pointing to the same object

```
-----------------------------------------------
print('r' in 'durga') # True
```

character 'r' present in 'durga' hence 'in' operator returns True

```
-----------------------------------------------
print('is' in 'This IS a Fake News') #True
```

Q48) Which expression evaluates to 4?

A) 7/2*3
B) 7%2+3
C) 7//2-3

**D)7-2*3**

**Answer: B**

**Explanation:**

1) **7/2*3==>10.5**
2) **7%2+3==>4**
3) **7//2-3==>0**
4) **7-2*3==>1**

**Q49) Consider the following expression:**

**6//4%5+2**3-2//3**

**This expression results to:**
**A)9**
**B)3**
**C)-1**
**D)25**

**Answer: A**

**Explanation:**

1) **6//4%5+2**3-2//3**
2) **6//4%5+8-2//3**
3) **1%5+8-2//3**
4) **1+8-2//3**
5) **1+8-0**
6) **9**

**Q50)Consider the code**

1) **x=2**
2) **y=6**
3) **x+=2**3**
4) **x//=y//2//3**
5) **print(x)**

**What is the output?**
**A)0**
**B)9**
**C)10**
**D)7**

**Answer: C**

**Explanation:**

```
1)  x+=2**3
2)  x=(x)+(2**3)=10
3)
4)  x//=y//2//3
5)  x=(x)//(y//2//3)
6)   =10//(6//2//3)
7)   =10//(3//3)
8)   =10//1
9)   =10
```

**Q51) Consider the Code**

```
1)  x=3/3+3**3-3
2)  print(x)
```

**What is the output?**
**A)25**
**B)32**
**C)0.11**
**D)25.0**

**Answer: D**

**Explanation:**

```
1)  x=3/3+3**3-3
2)   =3/3+27-3
3)   =1.0+27-3
4)   =25.0
```

**Q52) Consider the code**

```
1)  count=input('Enter the number of customers of the bank:')
2)  #Line-1
3)  print(output)
```

**Which code inserted at Line-1 will print 20 to the console if we pass 15 as count value from the console?**

**A)output=int(count)+5**
**B)output=count+5**

C)output=str(count)+5
D)output=float(count)+5

Answer: A

Explanation:

```
1) output=int(count)+5===>20
2) output=count+5===>Error,because we can not apply + operator between str and
   int
3) output=str(count)+5===>Error,because we can not apply + operator between str
   and int
4) output=float(count)+5===>20.0
```

Q53. In which of the following cases we will get <class 'int'> as output?

A)

```
1) x=47.0
2) print(type(x))
```

B)

```
1) x='47'
2) print(type(x))
```

C)

```
1) x=10+20j
2) print(type(x))
```

D)

```
1) x=2**2**2
2) print(type(x))
```

Answer: D

Explanation:

```
1) x=47.0
2) print(type(x))#<class 'float'>
3)
4)
5) x='47'
6) print(type(x))#<class 'str'>
```

```
7)  x=10+20j
8)  print(type(x))#<class 'complex'>
9)
10) x=2**2**2
11) print(type(x))#<class 'int'>
```

**Q54. Which of the following are valid statements?**

**A) 5+False evaluates to False**
**B) True+1 evaluates to 2**
**C) True and False evaluates to False**
**D) True or False evaluates to False**
**E) type('') is <class 'bool'>**

**Answer: B,C**

**Explanation:**

1) 5+False evaluates to 5 but not False
2) True+1 evaluates to 2
3) True and False evaluates to False
4) True or False evaluates to True but not False
5) type('') is <class 'str'> but not <class 'bool'>

**Q55. Which of the following string declarations spans more than one line and considers whitespace properly when the string is printed to the console?**

**A)**

```
1) s1='durga
2) software
3) solutions'
```

**B)**

```
1)  s1="durga
2) software
3) solutions"
```

**C)**

```
1) s1='durga\n
2) software\n
3) solutions'
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

D)

```
1) s1='''durga
2) software
3) solutions'''
```

Answer: D

Multi line string literals should be enclosed within triple quotes.

Q56)
Consider the following code:
print(type(input('Enter some value:')))

if we enter 10 and 10.0 individually for every run what is the output?

A)

```
1) <class 'str'>
2) <class 'str'>
```

B)

```
1) <class 'int'>
2) <class 'float'>
```

C)

```
1) <class 'int'>
2) <class 'int'>
```

D)

```
1) <class 'float'>
2) <class 'float'>
```

Answer: A

Explanation:
input() function always returns string type only.

Q57)
Consider the following code:
print(type(eval(input('Enter some value:'))))

if we enter 10 and 10.0 individually for every run what is the output?

A)

```
1) <class 'str'>
2) <class 'str'>
```

B)

```
1) <class 'int'>
2) <class 'float'>
```

C)

```
1) <class 'int'>
2) <class 'int'>
```

D)

```
1) <class 'float'>
2) <class 'float'>
```

Answer: B
input() function always returns str type, but eval() function converts str type corresponding type automatically.

Q58) Which of the following statements are valid?

A) s="Durga Sir's Python Classes are Good"
It causes error because we cannot use double quotes and single quotes simultaneously

B) result=456+456.0
type of result is int

C) The followng expression evaluates to 12
b=False+5-True+35//4

D) The following line will print result:4.5
print('result:',(7/2)+(False or True)+(9%3))

Answer: C,D

Explanation:
s="Durga Sir's Python Classes are Good"
It won't cause any error because we can take single and double quotes simultaneously

---
result=456+456.0
type of result is float
---
The followng expression evaluates to 12
b=False+5-True+35//4=False+5-True+8=0+5-1+8=12
---
The following line will print result:4.5
print('result:',(7/2)+(False or True)+(9%3))
(7/2)+(False or True)+(9%3)
=(3.5)+(True)+(0)
=(3.5)+(1)+(0)
=4.5

Q59) Consider the code
x='10'
y='20'
The type of x+y ?

A)int
B)float
C)str
D)complex

Answer: C

Explanation:
If we use + operator between 2 string types the result is always string type

Q60) Consider the code
a=float('123.456')
Which expression evaluates to 2?

A)int(a)+False
B)bool(a)+True
C)str(a)
D)bool(a)

Answer: B

Explanation:

int(a)+False==>123
bool(a)+True==>2
str(a)===>'123.456'

bool(a)===>True

Q61)
x='TEXT'
which line of the code will assign 'TT' to the output?

A) output=x[0]+x[2]
B) output=x[1]+x[1]
C) output=x[0]+x[-1]
D) output=x[1]+x[4]

Answer: C

Explanation:
output=x[0]+x[2]===>TX
output=x[1]+x[1]===>EE
output=x[0]+x[-1]==>TT
output=x[1]+x[4]===>IndexError,because 4 is out of range index

Q62. Consider the Python code:

```
1) l1=['sunny','bunny','chinny','vinny']
2) l2=['sunny','bunny','chinny','vinny']
3) print(l1 is not l2)
4) print(l1 != l2)
5) l1=l2
6) print(l1 is not l2)
7) print(l1 != l2)
```

What is the result?

A)
True
True
False
False

B)
True
False
True
False

C)
True
False
False
True

D)
True
False
False
False

Answer: D

Explanation:
if l1 and l2 are not pointing to the same object then only 'l1 is not l2' returns True.
If l1 and l2 are not having same content then only 'l1 != l2' returns True

Q63. Consider the Python Code

```
1) l1=['sunny','bunny','chinny','vinny']
2) l2=['sunny','bunny','chinny','vinny']
3) print(l1 is l2)
4) print(l1 == l2)
5) l1=l2
6) print(l1 is l2)
7) print(l1 == l2)
```

What is the result?

A)
False
True
True
True

B)
False
False
True
True

C)
False
True

False
True

D)
False
True
True
False

Answer: A

Explanation:
== operator is always meant for content comparison
is operator is always meant for reference(address) comparison
if l1 and l2 are pointing to the same object then only 'l1 is l2' returns True.
If l1 and l2 are having same content then only 'l1 == l2' returns True

Q64. Consider the python code:

```
1) print(10==10  and 20!=20)
2) print(10==10  or 20!=20)
3) print( not 10==10)
```

What is the result?

A)
True
True
False

B)
False
True
True

C)
False
True
False

D)
False
False
False

**Answer: C**

**Explanation:**

**If both arguments are True then only 'and' returns True.**
**If atleast one argument is True then 'or' returns True**
**not x==>if x is True then it returns False and if x if False then it returns True.**

**print(10==10 and 20!=20) Here first argument is True and second argument is False. Hence and operator returns False.**
**print(10==10 or 20!=20) Here first argument is True and second argument is False. Hence or operator returns True.**
**print( not 10==10) prints False to the console.**

**Q65. Consider the code:**

```
1) print(not 0)
2) print(not 10)
3) print(not '')
4) print(not 'durga')
5) print(not None)
```

**What is the result?**

**A)**
**True**
**False**
**False**
**False**
**True**

**B)**
**True**
**False**
**True**
**False**
**True**

**C)**
**False**
**False**
**True**
**False**
**True**

**D)**
**True**
**False**
**True**
**False**
**False**

**Answer: B**
**Explanation: In boolean expressions:**
**0 is treated as False, non-zero treated as True**
**empty string is treated as False and non-empty string treated as True**
**None is always treated as False**

**Q66. Consider the code:**

```
1) lst = [7, 8, 9]
2) b = lst[:]
3) print(b is lst)
4) print(b == lst)
```

**What is the result?**

**A)**
**False**
**True**

**B)**
**False**
**False**

**C)**
**True**
**False**

**D)**
**True**
**True**

**Answer: A**
**Explanation: slice operator will create a new object.**
**== operator is always meant for content comparison**
**is operator is always meant for reference(address) comparison**

**Q67. Consider the following code:**

```
1) v1 = 1
2) v2 = 0
3) v1 = v1 ^ v2
4) v2 = v1 ^ v2
5) v1 = v1 ^ v2
6) print(v1)
```

**What is the result?**

**A) 0**
**B) 1**
**C) 2**
**D) 3**

**Answer: A**

**Explanation:**
**^ is XOR operator.**
**If both bits are same then result is 0,otherwise result is 1**

**Q68. Consider the Python code:**

```
1) a=['a','b','c','d']
2) for i in a:
3)     a.append(i.upper())
4) print(a)
```

**What is the result?**

**A) ['a','b','c','d']**
**B) ['A','B','C','D']**
**C) SyntaxError**
**D) MemoryError thrown at runtime**

**Answer: D**

**Explanation: In the above code the content will be added keep on and it won't ends. At certain point memory problem will be raised.**

**Q69. Consider the python code:**

```
1) result=str(bool(1) + float(10)/float(2))
2) print(result)
```

**What is the output?**

**A) SyntaxError**
**B) TypeError**
**C) 6**
**D) 6.0**

**Answer: D**

**Explanation:**
**/ operator has more precedence than +. Hence float(10)/float(2) will be evaluated first and its result is 5.0. bool(1) is considered as True and again will be considered as 1 whenever we are performing + operator. Hence result is 6.0.**
**str(bool(1) + float(10)/float(2))=str(bool(1) + 10.0/2.0)=str(bool(1) + 5.0)=str(True + 5.0)=str(1 + 5.0)='6.0'**

# Topic

# Console *Input* and *Output* Statements


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

# Perform Input and Output Operations (10-15%)

## Exam Objectives:

**Construct and analyze code segments that perform file input and output operations**
**Open; close; read; write; append; check existence; delete; with statement**
**Construct and analyze code segments that perform console input and output operations**
**Read input from console; print formatted text; use of command line arguments**

**Q1. 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**

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

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

**A) xxx should be replaced with float and yyy should be replaced with float**
**B) xxx should be replaced with float and yyy should be replaced with int**
**C) xxx should be replaced with int and yyy should be replaced with float**
**D) xxx should be replaced with int and yyy should be replaced with int**

**Answer: A**

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

**Q2.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:**

```
1) print('Enter Your Name:')
2) #Line-1
3) print(name)
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

**At Line-1 which code we have to write?**

**A) name=input**
**B) input('name')**
**C) input(name)**
**D) name=input()**

**Answer: D**

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

**name=input()**

**Q3. 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?**

**A)total_items=input('How many items you required?')**
**B)total_items=float(input('How many items you required?'))**
**C)total_items=str(input('How many items you required?'))**
**D)total_items=int(float(input('How many items you required?')))**

**Answer: D**

**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.**

**total_items=int(float(input('How many items you required?')))**

**Q4. If the user enters 12345 as input,Which of the following code will print 12346 to the console?**

**A)**

```
1) count=input('Enter count value:')
2) print(count+1)
```

B)

```
1) count=input('Enter count value:')
2) print(int(count)+1)
```

C)

```
1) count=eval(input('Enter count value:'))
2) print(count+1)
```

D)

```
1) count=int(input('Enter count value:'))
2) print(count+1)
```

Answer: B,C,D

Explanation: The return type of input() function is str type. We have to perform typecasting. As user providing 12345 int value, we have to typecast either by using int() or by using eval() function.

Q5. From sys module, by using which variable we can access command line arguments?

A) argv
B) args
C) argsv
D) arguments

Answer: A

Explanation: By using argv variable present in sys module,we can access command line arguments.

Q6. Given the command invocation:

py test.py Durga

Which of the following code prints 'Durga' to the console?

A)

```
1) from sys import args
2) print(args[0])
```

B)

```
1) from sys import args
2) print(args[1])
```

C)

```
1) from sys import argv
2) print(argv[0])
```

D)

```
1) from sys import argv
2) print(argv[1])
```

Answer: D

Explanation:
By using argv variable present in sys module,we can access command line arguments. argv[0] represents the name of the file. In the above case it is 'test.py'. Hence to access 'Durga', we have to use argv[1]

Q7. Consider the code:

```
1) from sys import argv
2) print(argv[0])
```

and given the command invocation:
py test.py DURGASOFT

What is the result?

A) DURGASOFT
B) test.py
C) IndexError will be thrown at runtime
D) ImportError will be thrown at runtime

Answer: B

Explanation:

By using argv variable present in sys module,we can access command line arguments. argv[0] represents the name of the file. In the above case it is 'test.py'.

**Q8. Consider the code:**

```
1) from sys import argv
2) print(argv[1]+argv[2])
```

**and given the command invocation:**
**py test.py 10 20**

**What is the result?**

**A) 30**
**B) 1020**
**C) IndexError will be thrown at runtime**
**D) ImportError will be thrown at runtime**

**Answer: B**

**Explanation:**
**By using argv variable present in sys module,we can access command line arguments.**
**argv[0] represents the name of the file.**
**The command line arguments are always considered as str type. Hence + operator meant for concatenation. In this case the output is: 1020**

**Q9. Consider the code :**

```
1) from sys import argv
2) sum=0
3) for i in range(2,len(argv)):
4)     sum += float(argv[i])
5) print("The Average for {0} is {1:.2f}".format(argv[1],sum/(len(argv)-2)))
```

**Which of the following command invocations will generate the output:**
**The Average for Durga is 20.00**

**A) py test.py Durga 10 20 30**
**B) py test.py Durga 10 20**
**C) py test.py Durga 10**
**D) py test.py 20**

**Answer: A**

**Explanation:**
**By using argv variable present in sys module,we can access command line arguments.**
**argv[0] represents the name of the file.**
**In the above code {0} will be replaced with argv[1] which is nothing but Durga.**

**{1:.2f} will be replaced with sum/(len(argv)-2)) and after decimal point 2 digits will be considered.**

**Q10. We are developing an app in which students will provide college name and city as input. If the student provides college name as DURGASOFT and city as Hyderabad, then our application has to provide the following greeting message.**

**Welcome to DURGASOFT in Hyderabad**

**Which of the following code can be used for this requirement?**

**A)**

```
1) college_name=input('Enter Your College Name:')
2) city=input('Enter Your City:')
3) print('Welcome to {} in {}'.format(college_name,city))
```

**B)**

```
1) college_name=read('Enter Your College Name:')
2) city=read('Enter Your City:')
3) print('Welcome to {} in {}'.format(college_name,city))
```

**C)**

```
1) college_name=eval('Enter Your College Name:')
2) city=eval('Enter Your City:')
3) print('Welcome to {} in {}'.format(college_name,city))
```

**D)**

```
1) college_name=str('Enter Your College Name:')
2) city=str('Enter Your City:')
3) print('Welcome to {} in {}'.format(college_name,city))
```

**Answer: A**
**Explanation: We should use input() function to read input from the keyboard.**

# Topic

# Flow Control


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

# Control Flow with Decisions and Loops (15-20%)

## Exam Objectives:

**Construct and analyze code segments that use branching statements**
**if; elif; else; nested and compound conditional expressions**
**Construct and analyze code segments that perform iteration**
**while; for; break; continue; pass; nested loops and loops that include compound conditional expressions**

**Q1. 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?**

**A) print('The average star rating for the new coffee is:{:.2f}'.format(average))**
**B) print('The average star rating for the new coffee is:{:.2d}'.format(average))**
**C) print('The average star rating for the new coffee is:{:2f}'.format(average))**
**D) print('The average star rating for the new coffee is:{:2.2d}'.format(average))**

**Answer: A**

**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))**


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

**Q2. Consider the following statements:**

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

**A) only 1 and 2**
**B) only 1 and 3**
**C) only 2 and 4**
**D) 1,2,3 and 4**

**Answer : D**

**Explanation:**
**{:.2f}==>After decimal point, the value will be rounded to 2 digits**
**{:8.2f}==>Minimum 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}===>Minimum 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.**

**Q3. 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
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

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

**A) print('%-20s,Your average score is: %4.1f' %(name,average_score))**
**B) print('%-20f,Your average score is: %4.1f' %(name,average_score))**
**C) print('%-20s,Your average score is: %1.4f' %(name,average_score))**
**D) print('%-20s,Your average score is: %4.1s' %(name,average_score))**

**Answer: A**

**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
```

**Q4.**
**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])
5)
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?**
**A) Line-1 should be replaced with**
**while(index<10):**
**B) Line-2 should be replaced with**
**if numbers[index]==6:**

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

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

Q5)
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

A) It will throw error because misuse of split() method
B) It will throw error because misuse of isDigit() method
C) There is no error but it won't fulfll our requirement.
D) No changes are required for this code and it can fulfill requirement.

Answer: D

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

Q6.
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 a is non-negative, return a**(1/b)
If a is negative and even, return "Result is an imaginary number"
if a is negative and odd,return -(-a)**(1/b)

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

**A)**

```
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
```

**B)**

```
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
```

**C)**

```
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
```

**D)**

```
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
```

**Answer: A**

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

**Q7. 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 :**

```
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?**

**A) Line-1 should be replaced with**
**if marks <= 90:**

**B) Line-2 should be replaced with**
**if marks>=80 and marks <= 90 :**

**C) Line-3 should be replaced with**
**if marks>=70 and marks <= 80 :**

**D) No Changes are required.**

**Answer: D**

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

**Q8. 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**

**A)**

```
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
```

**B)**

```
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
```

**C)**

```
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
```

D)

```
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
```

Answer: A

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

```
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
```

Q9: 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?

A)

```
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
8)
9)      if is_prime==True:
10)         print(p)
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

```
11)    p=p+1
```

**B)**

```
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
8)
9)      if is_prime==True:
10)         print(p)
11)     p=p+1
```

**C)**

```
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
8)
9)      if is_prime==False:
10)         print(p)
11)     p=p+1
```

**D)**

```
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
8)
9)      if is_prime==True:
10)         print(p)
```

**Answer: A**


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

**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**

```
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
8)
9)      if is_prime==True:
10)         print(p)
11)     p=p+1
```

**Q10.**
**You are creating a Python script to evaluate input and check for upper and lower case. Which of the following is the solution for this requirement?**

```
1)  word=input('Enter Any word:')
2)  if word.lower()==word:
3)      print('The given word in lower case')
4)  elif word.upper()==word:
5)      print('The given word in upper case')
6)  else:
7)      print('The given word in mixed case')
```

**Q.11**
**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?**


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

A) Invalid Syntax
B) Mismatched data type(s)
C) Misnamed variable(s)
D) None of these

Answer: A

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

Q12.
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?

A) The Cost of Book Rental is:$ 7.0
B) The Cost of Book Rental is:$ 8.0
C) The Cost of Book Rental is:$ 9.0
D) The Cost of Book Rental is:$ 10.0

Answer: C

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.

Q13. 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?

A)

```
1)  if marks>=80 and marks>=100:
2)      grade='A'
```

B)

```
1)  if marks>=80 or marks<=100:
2)      grade='A'
```

C)

```
1)  if 80<=marks<=100:
2)      grade='A'
```

D)

```
1)  if marks>80:
2)      grade='A'
```

Answer: C

**Explnation:**

**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**

```
1) if 80<=marks<=100:
2)     grade='A'
```

**If marks >=80 and marks<=100 then only grade will become 'A', which will meet our requirement.**

**Q14. We are developing a online shopping application. Consider the code**

```
1) d =input('Enter day of the week:')
2) discount_percentage = 3
3)
4) if d== 'Monday':
5)     discount_percentage+=5
6)
7) elif d== 'Tuesday':
8)     discount_percentage+=7
9)
10) elif d== 'Saturday':
11)     discount_percentage+=10
12)
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?**

**A) Monday**
**B) Tuesday**
**C) Thursday**
**D) Saturday**
**E) Sunday**

**Answer: C**

**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**

**Q15. 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
5)
6)  elif amount > 50000 and amount<100000:
7)      interest_rate = 12
8)
9)  elif amount >= 100000 and amount<150000:
10)     interest_rate = 16
11)
12) else:
13)     interest_rate = 22
```

**For which of the following user input interest_rate will be 12.**

**A) 50000**
**B) 50001**
**C) 100000**
**D) 100001**
**E) 150000**

**Answer: B**

**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**

**Q16. 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)
4)  if days==1:
5)      print('Leave will be approved immediately')
6)
7)  elif days>1 and days<=3:
8)      if cause=='Sick':
9)          print('Leave will be approved immediately')
10)     else:
11)         print('Needs Lead Approval')
12)
13) elif days>3 and days<5:
14)     if cause=='Sick':
15)         print('Needs Manager Approval')
16)     else:
17)         print('Needs Director Approval')
18)
19) elif days>=5 and days<=10:
20)     print('Needs Director Approval')
```

**In which of the following cases 'Needs Director Approval' will be printed to the console?**
**A) days = 2 and cause='Sick'**
**B) days = 3 and cause='Personal'**
**C) days = 4 and cause='Sick'**
**D) days = 4 and cause='Official'**

**Answer: D**

**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.**

**Q17. 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) A**
**B) B**
**C) C**
**D) D**

**Answer: C**

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

**Q18. Consider the code**
**a=12**
**b=4**
**s='He shall not be happy if he does not work'**

**In which of the following cases result value will become 9**
**A) result=3 if None else a/b**
**B) result=s.find('not') if s else None**
**C) result=s.rfind('not') if s else None**
**D) result=5 if len(s)>4 else 6**

**Answer: B**

**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

Q19. 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?

A) 350
B) 200
C) 150
D) 100

Answer:A

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.

Q20. You are developing online shopping application.
Consider the code:

```
1)  order_value=1500
2)  state='ap'
3)  delivery_charge=0
4)  if state in ['up','mp','ts']:
5)      if order_value<=1000:
6)          delivery_charge=50
7)      elif order_value>1000 and order_value<2000:
```

```
8)        delivery_charge=100
9)     else:
10)       delivery_charge=150
11) else:
12)    delivery_charge=25
13) if state in ['lp','kp','ap']:
14)    if order_value>1000:
15)       delivery_charge+=20
16)    if  order_value<2000 and state in ['kp','ap']:
17)       delivery_charge+=30
18)    else:
19)       delivery_charge+=15
20) print(delivery_charge)
```

**What is the result?**
**A) 65**
**B) 75**
**C) 85**
**D) 55**

**Answer: B**

**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.**

**Q21. 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?**

**A) 1**
**B) 2**
**C) 3**


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

D) 4

Answer : B

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

Q22. 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?

A)17
B)18
C)19
D)20

Answer: C

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

Q23. 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?
A) 12
B) 13
C) 19
D) 6

Answer: B

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)

Q24. 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?
A) 29
B) 30
C) 31
D) 32

Answer: C

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)

Q25. Consider the code:

```
1) values = [[3, 4, 5, 1], [33, 6, 1, 2]]
2)
3) v = values[0][0]
4) for lst in values:
5)     for element in lst:
6)         if v > element:
7)             v = element
8)
9) print(v)
```

What is the result?
A) 3
B) 2
C) 1
D) 4

Answer: C

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


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com



# Topic

# Functions

**Q1. Consider the code**

```
1)  def get_names():
2)      names=['Sunny','Bunny','Chinny','Vinny','Pinny']
3)      return names[2:]
4)
5)  def update_names(elements):
6)      new_names=[]
7)      for name in elements:
8)          new_names.append(name[:3].upper())
9)      return new_names
10)
11) print(update_names(get_names()))
```

**What is the result?**

**A) ['CHI', 'VIN', 'PIN']**
**B) ['VIN', 'PIN']**
**C) ['CH', 'VI', 'PI']**
**D) ['SU', 'BU']**

**Answer: A**

**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']**

**Q2. Consider the following code**

```
1)  def my_list(x):
2)      lst.append(a)
3)      return lst
4)
5)  my_list('chicken')
6)  my_list('mutton')
7)  print(my_list('fish'))
```

**to print the following to the console**
**['chicken','mutton','fish']**

**x should be replaced with**

**A)a,lst=[]**
**B)a,lst=()**


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

C)a,lst={}
D)a,lst=None

Answer: A

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

Q3. Consider the following code:

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

Which of the following method calls are valid?

A) f1()
B) f1('10','20')
C) f1(10)
D) f1('10')

Answer: A,B,C

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

Q4. Consider the following code:

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

Which of the following method calls are valid?

A) f1()
B) f1('10','20')
C) f1(10)
D) f1('10')

Answer: A,C,D

**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'**

**Q5. 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**

**A)**

```
1) def find_numbers():
2)     numbers.sort()
3)     return numbers[0],numbers[-1]
4) low,high=find_numbers()
```

**B)**

```
1) def find_numbers():
2)     numbers.sort()
3)     return numbers[0],numbers[len(numbers)]
4) low,high=find_numbers()
```

**C)**

```
1) def find_numbers():
2)     numbers.sort()
3)     return numbers[0],numbers[-1]
4) low=find_numbers()
5) high=find_numbers()
```

**D)**

```
1) def find_numbers():
2)     numbers.sort()
3)     return numbers[2],numbers[0]
4) low,high=find_numbers()
```

**Answer: A**

**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.**

**Q6)Consider the code:**

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

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

**A) print(low,high)**
**B) print(low[0],high[-1])**
**C) print(low[-1],high[0])**
**D) print(low[2],high[0])**

**Answer: B**
**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])**

**Q7. 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**

**A) calculate()**
**B) calculate(10)**
**C) calculate(5,2)**
**D) calculate(1)**

**Answer: B**

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

**Q8. 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?**

**A) 0,1,1,2,3,5,8,**
**B) 0,1,2,4,8,16,32,**
**C) 0,1,0,2,0,4,0,**
**D) None of these**

**Answer: A**

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

**Q9. 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?**

**A)**

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


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

**B)**

```
1) def update_score(score,value):
2)     new_score=score+value
3)     return new_score
```

**C)**

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

**D)**

```
1) def update_score():
2)     new_score=score+value
3)     return new_score
```

**Answer: B**

**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.**

```
1) def update_score(score,value):
2)     new_score=score+value
3)     return new_score
```

**Q10. 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
4)
5)  Line-2:
6)      calories=miles*calories_per_mile
7)      return calories
8)
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)
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

```
13) print(biker,", You burned about",calories_burned," calories")
```

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

**A)**

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

**B)**

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

**C)**

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

**D)**

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

**E)**

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

**F)**

```
1) Line-2 should be replaced with
2) def calc_calories():
```

**Answer: A and D**

**Explanation:**
**The following are valid function declarations to fulfill our requirement:**

```
1) def get_name():
2)     name=input('Enter Your Name:')
3)     return name
4)
5) def calc_calories(miles,calories_per_mile):
6)     calories=miles*calories_per_mile
7)     return calories
```

**Q11. 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:**

**A)**

```
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
```

**B)**

```
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
```

**C)**

```
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
```

D)

```
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
```

Answer: A

Explanation:
The correct function to fulfill above requirements is

```
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
```

Q12. 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?

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

**Answer: D**

**Explanation: The above program prints the number of occurrences of specified alphabet symbol in the given word_list.**

# Topic

# Exception Handling


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

**Q1. Which of the following is False?**

**A) A try statement can have one or more except clauses**
**B) A try statement can have a finally clause without an except clause**
**C) A try statement can have  a finally clause and an except clause**
**D) A try statement can have one or more finally clauses**

**Answer: D**

**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.**

**Q2. Which type of exception will be raised if we are trying to call a method on the inappropriate object?**

**A) IndexError**
**B) TypeError**
**C) AttributeError**
**D) None of these**

**Answer: C**

**Explanation: If we are trying to access a method on the object,if the corresponding class does not contain that method, then we will get AttributeError.**
**eg:**
**l=[10,20,30]**
**l.add(30)**
**AttributeError: 'list' object has no attribute 'add'**

**Q3. Consider the code**

```
f=open('abc.txt')
f.readall()
```

**Which exception will be raised?**
**A) AttributeError**
**B) EOFError**
**C) SystemError**
**D) SyntaxError**

**Answer: A**

**Explanation:**
**If we are trying to access a method on the object,if the corresponding class does not contain that method, then we will get AttributeError.**

**readall() method is not available for file object.**
**AttributeError: '_io.TextIOWrapper' object has no attribute 'readall'**

**Q4. Consider the code**

```
1)  def f1():
2)      try:
3)          return 1
4)      finally:
5)          return 2
6)
7)  x=f1()
8)  print(x)
```

**What is the result?**
**A) 1**
**B) 2**
**C) prints both 1 and 2**
**D) Error, because more than one return statement is not allowed**

**Answer: B**

**Explanation:**
**finally block return statement has more priority than try block return statement**

**Q5. Which of the following are True?**

**A) A try block can have any number of except blocks**
**B) A try block should be associated with atmost one finally block**
**C) A try block should be associated with atmost one else block**
**D) All the above**

**Answer: D**

**Explanation:**
**We can take any number of except blocks for the same try.**
**We can't take more than one finally blocks for the same try.**
**We can't take more than one else blocks for the same try.**

**Q6. The base class for all exceptions in python is:**

**A) ExceptionBase**
**B) BaseException**
**C) Exception**
**D) EOFError**

**Answer: B**

**Explanation: Every Exception class in python should be child class of BaseException either directly or indirectly.**

**Q7. Which of the following is True about else block?**

**A) else block will be executed if there is no exception in try block**
**B) Without writing except block we cannot write else block**
**C) For the same try we can write atmost one else block**
**D) All the above**

**Answer : D**

**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.**

**Q8. Consider the code:**

```
1)  try:
2)      print('try')
3)  except:
4)      print('except')
5)  else:
6)      print('else')
7)  finally:
8)      print('finally')
```

**What is the result?**

**A)**

```
try
except
else
finally
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

B)

```
try
else
finally
```

C)

```
try
except
finally
```

D)

```
try
finally
```

Answer:B

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.

Q9.
Consider the code

```
1) try:
2)     print('try')
3)     print(10/0)
4) except:
5)     print('except')
6) else:
7)     print('else')
8) finally:
9)     print('finally')
```

What is the result?

A)

```
try
except
else
finally
```

B)

```
try
else
finally
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

C)

try
except
finally

D)

try
finally

Answer:C

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.

Q10. Consider the code:

```
1)  try:
2)      print('try')
3)      print(10/0)
4)
5)  else:
6)      print('else')
7)
8)  except:
9)      print('except')
10) finally:
11)     print('finally')
```

A)

try
else
except
finally

B)

try
else
finally

C)

try
except
finally

D) SyntaxError: invalid syntax


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

**Answer:D**

**Explanation: in try-except-else-finally the order is important.**
**After except block only we have to take else block**

**Q11. Consider the code:**

```
1) f=open('abc.txt')
2) print(f.read())
3) f.close()
```

**We required to add exception handling code to handle FileNotFoundError.**
**We of the following is appropriate code for this requirement?**

**A)**

```
1) f=None
2) try:
3)     f=open('abcabcabcabc.txt')
4) except FileNotFoundError:
5)     print('File does not exist')
6) else:
7)     print(f.read())
8) finally:
9)     if f != None:
10)        f.close()
```

**B)**

```
1) f=None
2) try:
3)     f=open('abcabcabcabc.txt')
4) except FileNotFoundException:
5)     print('File does not exist')
6) else:
7)     print(f.read())
8) finally:
9)     if f != None:
10)        f.close()
```

**C)**

```
1) f=None
2) try:
3)     f=open('abcabcabcabc.txt')
```

```
4)  else:
5)      print(f.read())
6)
7)  except FileNotFoundException:
8)      print('File does not exist')
9)
10) finally:
11)     if f != None:
12)         f.close()
```

D) None of these

Answer: A

Explanation:
In Python we have FileNotFoundError but not FileNotFoundException.
In try-except-else-finally, order is important. We cannot take else block before except block.

Q12. Consider the code

```
a=10
b=20
c='30'
result=a+b+c
```

What is the result?

A) 102030
B) 3030
C) TypeError
D) ArithmeticError

Answer: C

Explanation:
We cannot apply + operator for 'int' and 'str' arguments. Othewise we will get TypeError: unsupported operand type(s) for +: 'int' and 'str'

Q13. Consider the code:

```
1)  prices=[30.5,'40.5',10.5]
2)  total=0
3)  for price in prices:
4)      total += price
```

```
5) print(total)
```

**While executing this code we are getting the following error**

```
Traceback (most recent call last):
  File "test.py", line 4, in <module>
    total += price
TypeError: unsupported operand type(s) for +=: 'float' and 'str'
```

**Which of the following code should be used to fix this error?**

**A) total += str(price)**
**B) total += int(price)**
**C) total += float(price)**
**D) total = total+price**

**Answer: C**

**Explanation:**
**If the string contains internally float value,then we should use float() function to convert into float value. If we are trying to use int() function then we will get ValueError. Hence the following line is the correct fix for the problem**

```
total += float(price)
```

**Q14. Consider the code:**

```
1) prices=[10,'20',30,'40']
2) total=0
3) for price in prices:
4)     total +=price
5) print(total)
```

**While executing this code we are getting the following error**

```
Traceback (most recent call last):
  File "test.py", line 4, in <module>
    total +=price
TypeError: unsupported operand type(s) for +=: 'int' and 'str'
```

**By using which of the following code segments we can fix this problem(Choose Two)?**

**A) total += str(price)**
**B) total += int(price)**
**C) total += float(price)**

D) total = total+price

Answer: B and C

Explanation:
We cannot apply + operator between int and str. Hence We have to type cast str to either int type or float type, then only we can apply + operator.

Q15. Consider the code

```
1) courses={1:'Java',2:'Scala',3:'Python'}
2) for i in range(1,5):
3)     print(courses[i])
```

While executing this code we are getting the following error

```
Traceback (most recent call last):
  File "test.py", line 3, in <module>
    print(courses[i])
KeyError: 4
```

By using which of the following code segments we can fix this problem ?

A)

```
1) courses={1:'Java',2:'Scala',3:'Python'}
2) for i in range(1,5):
3)     if i in courses:
4)         print(courses[i])
```

B)

```
1) courses={1:'Java',2:'Scala',3:'Python'}
2) for i in courses:
3)         print(courses[i])
```

C)

```
1) courses={1:'Java',2:'Scala',3:'Python'}
2) for i in range(1,4):
3)     print(courses[i])
```

D) All of these

Answer: D


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

**Explanataion: In the above code key 4 is not available in the dictionary. Hence we are getting KeyError. If the key is within the range from 1 to 3,then the problem will be fixed.**

**Q16. Consider the code**

```
1) def area(b,w):
2)     return B*w
3)
4) print(area(10,20))
```

**What is the result?**

**A) NameError will be raised at runtime**
**B) AttributeError will be raised at runtime**
**C) IdentationError will be raised at runtime**
**D) 200**

**Answer: A**

**Explanation: If the variable is not available, still if we are trying to access then we will get NameError. In the above code variable 'B' is not available.**

**Q17. Consider the following code:**

```
1) def get_score(total=0,valid=0):
2)     result=int(valid)/int(total)
3)     return result
```

**For which of the function calls we will get Error?**

**A) score=get_score(40,4)**
**B) score=get_score('40','4')**
**C) score=get_score(40)**
**D) score=get_score(0,10)**

**Answer: D**

**Explanation: We are performing division by zero and hence we will get ZeroDivisionError: division by zero**

**Q18. Consider the code:**

```
1) data=[]
2) def get_data():
3)     for i in range(1,5):
```

```
4)        marks=input('Enter Marks:')
5)        data.append(marks)
6)
7)  def get_avg():
8)      sum=0
9)      for mark in data:
10)         sum += mark
11)     return sum/len(data)
12) get_data()
13) print(get_avg())
```

**For the input: 10,20,30,40 what is the result?**

**A) 25**
**B) 25.0**
**C) NameError is thrown at runtime**
**D) TypeError is thrown at runtime**

**Answer: D**

**Explanation: All marks are available in string form and hence we cannot apply + operator between int and str**

**sum += mark**
**TypeError: unsupported operand type(s) for +=: 'int' and 'str'**

**Q19. Consider the code:**

```
1)  a=10
2)  b=0
3)  try:
4)      print(a/b)
```

**Which of the following except block print the name of exception which is raised,i.e exception class name?**

**A)**

```
1)  except ZeroDivisionError as e:
2)      print('Exception Type:',e.__class__.__name__)
```

**B)**

```
1)  except ZeroDivisionError as e:
2)      print('Exception Type:',type(e).__name__)
```

C)

```
except ZeroDivisionError as e:
    print('Exception Type:',e)
```

D) All of these

Answers : A,B

Explanation: We can get class name from the exception object as follows
e.__class__.__name__ or
type(e).__name__

If we use just print(e) then we will get description of the exception like: ' division by zero'

Q20. Which of the following is valid way of creating our own custom exception?

A)
```
class MyException:
    pass
```
B)
```
class MyException():
    pass
```
C)
```
 class MyException(Exception):
    pass
```
D) It is not possible to define custom exceptions in python

Answer: C

Explanation:
To define custom exceptions, compulsory we have to create child class for BaseException either directly or indirectly

Q21. Consider the code

```
x=int(input('Enter First Number:'))
y=int(input('Enter Second Number:'))
try:
    print(x/y)
```

Which of the following is valid except block that handles both ZeroDivisionError and ValueError

**A)**
```
except(ZeroDivisionError,ValueError) from e:
        print(e)
```

**B)**
```
except(ZeroDivisionError,ValueError) as e:
        print(e)
```

**C)**
```
except(ZeroDivisionError | ValueError) as e:
        print(e)
```
**D)**
```
except(ZeroDivisionError, ValueError as e):
        print(e)
```

**Answer: B**

**Explanation:**
**The following is the valid syntax:**
```
except(ZeroDivisionError,ValueError) as e:
        print(e)
```

**We should use as keyword only but not from keyword. The variable e must be outside of parenthesis because it is common for both exceptions.**

**3 and 11 changes in exams**

# Topic

# File I/O



DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

**Q1. Consider the following code.**

```
1)  import os
2)  def get_data(filename,mode):
3)      if os.path.isfile(filename):
4)          with open(filename,'r') as file:
5)              return file.readline()
6)      else:
7)          return None
```

**Which of the following are valid about this code?**

**A) This function returns the first line of the file if it is available**
**B) This function returns None if the file does not exist**
**C) This function returns total data present in the file**
**D) This function returns last line of the file**

**Answer: A and B**

**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.**

**Q2. 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:**

```
1)  inventory=open('inventory.txt','r')
2)  eof=False
3)  while eof == False:
4)      line=inventory.readline()
5)      if XXX:
6)          if YYY:
7)              print(line,end='')
8)
9)      else:
10)         print('End of file')
11)         eof=True
12)         inventory.close()
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

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

**A)**
**XXX should be replaced with**
**line != ''**
**YYY should be replaced with**
**line!= '\n'**

**B)**
**XXX should be replaced with**
**line!= '\n'**
**YYY should be replaced with**
**line != ''**

**C)**
**XXX should be replaced with**
**line != ''**
**YYY should be replaced with**
**line != ''**

**D)**
**XXX should be replaced with**
**line!= '\n'**
**YYY should be replaced with**
**line!= '\n'**

**Answer: A**

**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'**

**Q3. 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?**

**A) open('abc.txt','r')**
**B) open('abc.txt','r+')**
**C) open('abc.txt','w+')**
**D) open('abc.txt','w')**

**Answer: C**

**Explanation: in the case of 'r+' if the file has content, it won't be removed.**

**Q4. We are creating a function that reads a data file and prints each line of that file. Consider the following code:**

```
1) import os
2) def read_file(file):
3)     line=None
4)     if os.path.isfile(file):
5)         data=open(file,'r')
6)     while line != '':
7)         line=data.readline()
8)         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?**

**A) First 3 Lines inside function**
**B) Last 3 Lines inside function**
**C) Last 2 Lines inside function**
**D) There is no indentation problem**

**Answer: B**

**Explanation: The Last 3 lines having identation problem and the correct code is:**

```
1) import os
2) def read_file(file):
3)     line=None
4)     if os.path.isfile(file):
5)         data=open(file,'r')
6)         while line != '':
7)             line=data.readline()
8)             print(line)
```

**Q5. Consider the code:**

```
1) import sys
2) try:
3)     file_in=open('in.txt','r')
4)     file_out=open('out.txt','w+')
5) except IOError:
6)     print('cannot open',file_name)
```

```
7)  else:
8)      i=1
9)      for line in file_in:
10)         print(line.rstrip())
11)         file_out.write(str(i)+":"+line)
12)         i=i+1
13)     file_in.close()
14)     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?**

**A) This program will copy data from in.txt to out.txt**
**B) The code runs,but generates logical error**
**C) The code will generates a runtime error**
**D) The code will generates a syntax error**

**Answer: A**

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

**Q6. Conside the file abc.txt:**

**abc.txt:**
**--------**
**Durga:10**
**Ravi:20**
**Shiva:30**
**Pavan:40**

**Consider the python code which is present in the same location of the file**

**test.py:**
**--------**

```
1)  values=0
2)  try:
3)      f=open('abc.txt','r')
4)      content=f.readlines()
5)      for line in content:
6)          values+=float(line.split(':')[1])
7)      f.close()
8)  except Exception:
9)      print('Unable to open the file')
```

```
10) print(values)
```

**What is the result?**

**A) Unable to open the file**
**B) 100**
**C) 100.0**
**D) 10.0**

**Answer: C**

**Explanation:**
**The above program reads all the numbers prent in every line of file and converting to float value and then addding that value to value variable.**

**Q7. Consider the file abc.txt has the following content:**

**Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.**

**We have to write python code to read total data and printing to the console.**

```
1) try:
2)     f=open('abc.txt','r')
3)     //Line-1
4) except:
5)     print('Unable to open the file')
6) print(data)
```

**Which code should be inserted at Line-1 to meet the given requirement ?**

**A) data=f.read()**
**B) data=f.readline()**
**C) data=f.readlines()**
**D) data=f.load()**

**Answer: A**

**Explanation: To read total data, we have to use f.read() method.**

**Q8. To write 'Python Certificaton' to abc.txt file, which of the following is valid code?**

**A)**

```
1) f=open('abc.txt','b')
2) f.write('Python Certificaton')
3) f.close()
```

**B)**

```
1) f=open('abc.txt','r')
2) f.write('Python Certificaton')
3) f.close()
```

**C)**

```
1) f=open('abc.txt')
2) f.write('Python Certificaton')
3) f.close()
```

**D)**

```
1) f=open('abc.txt','w')
2) f.write('Python Certificaton')
3) f.close()
```

**Answer: D**

**Explanation: To write text data we have to open the file in 'w' mode**

**Q9. Consider the data present in the file: abc.txt**

**DURGASOFT,50,60,70,80,90**
**MICROSOFT,10,20,30,40,50**

**Which of the following is valid code to read total data from the file?**

**A)**

```
1) with open('abc.txt','r') f:
2)     data=f.read()
```

**B)**

```
1) with open('abc.txt') as f:
2)     data=f.read()
```

C)

```
1)  with open('abc.txt','w') as f:
2)      data=f.read()
```

D)

```
1)  with open('abc.txt') as f:
2)      data=f.readline()
```

Answer: B

Explanation: The default mode is 'r' and hence the following is valid syntax:

```
1)  with open('abc.txt') as f:
2)      data=f.read()
```

Q10. Assume that we are writing python code for some voting application.
You need to open the file voters_list.txt and add new voters info and print total data to the console?

```
1)  with open('voters_list.txt','a+') as f:
2)      f.write('New voters info')
3)      #Line-1
4)      data=f.read()
5)      print(data)
```

Which Line should be inserted at Line-1 ?
A) f.seek(0)
B) f.flush()
C) f.begin()
D) f.close()

Answer: A

Explnation: After writing the data, to read total data we have to move to beginning of the file. Hence we should use f.seek(0)

# Topic

# Built-in Modules



DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

**Q1. 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?**

**A) math.frexp(x)**
**B) math.floor(x)**
**C) math.fabs(x)**
**D) math.fmod(x)**
**E) math.ceil(x)**

**Answer: B and C**

**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)))**

**Q2.**
**You are writing an application that uses the sqrt function.The program must reference the function using the name sq.**
**Which of the following import statement required to use?**

**A) import math.sqrt as sq**
**B) import sqrt from math as sq**
**C) from math import sqrt as sq**
**D) from math.sqrt as sq**

**Answer: C**

**Explanation: The following is the valid syntax:**
**from math import sqrt as sq**

**Q3. Consider the code:**

```
1) import math
2) l =[str(round(math.pi)) for i in range (1, 6)]
3) print(l)
```

**What is the result?**


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

A) ['3', '3', '3', '3', '3']
B) ['3', '3', '3', '3', '3','3']
C) ['1', '2', '3', '4', '5']
D) ['1', '2', '3', '4', '5','6']

Answer: A

Explanation: range(1,6) means we have to consider from 1 to 5 and every time we are performing round(math.pi) and hence the output is: ['3', '3', '3', '3', '3']

Q4.
You are writing code that generates a random integer with a minimum value of 5 and maximum value of 11. Which of the following 2 functions we required to use?

A. random.randint(5,12)
B. random.randint(5,11)
C. random.randrange(5,12,1)
D. random.randrange(5,11,1)

Answer: B and C

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 <stop

Q5. Consider the following code:

```
1) import random
2) fruits=['Apple','Mango','Orange','Lemon']
```

Which of the following will print some random value from the list?

A) print(random.sample(fruits))
B) print(random.sample(fruits,3)[0])
C) print(random.choice(fruits))
D) print(random.choice(fruits)[0])

Answer: B and C

Explanation:
random.sample(population, k)
Return a k length list of unique elements chosen from the population sequence or set.

**Q6. We are developing an application for the client requirement.As the part of that we have to create a list of 7 random integers between 1 and 7 inclusive.**

**Which of the following code should be used?**

**A)**

```
1) import random
2) randints=[random.randint(1,7) for i in range(1,8)]
```

**B)**

```
1) import random
2) randints=[random.randint(1,8) for i in range(1,8)]
```

**C)**

```
1) import random
2) randints=random.randrange(1,7)
```

**D)**

```
1) import random
2) randints=random.randint(1,7)
```

**Answer: A**

**Explanation:**
**randints=[random.randint(1,8) for i in range(1,8)]**
**In this case there may be a chance of 8 also.**

**randints=random.randrange(1,7)**
**It will generate a random number from 1 to 6 but not list.**

**randints=random.randint(1,7)**
**it generates a random number from 1 to 7 but not list**

**Q7. Consider the code:**

```
1) import random
2) fruits=['Apple','Mango','Orange','Lemon']
3) random_list=[random.choice(fruits)[:2] for i in range(3)]
4) print(''.join(random_list))
```

**Which of the following are possible outputs?**

A) ApApAp
B) ApMaOr
C) LeMaOr
D) OrOraM

Answer: A,B,C

Explanation:
The above code selects 3 random words from the list and from every word first 2 characters will be collected and joined into a single string.
Hence the following outputs are possible: ApApAp, ApMaOr, LeMaOr

Q8. We are developing a mathematical function to find area for the given circle. if r is the radius then area is : pi*r**2
Which of the following is valid function for this requirement?

A)

```
1) import math
2) def find_area(r):
3)     return math.pi*math.fmod(r,2)
```

B)

```
1) import math
2) def find_area(r):
3)     return math.pi*math.fabs(r)
```

C)

```
1) import math
2) def find_area(r):
3)     return math.pi*math.pow(r,2)
```

D) None of these

Answer: C

Explanation:
fmod(x, y) Returns the remainder when x is divided by y
fabs(x) Returns the absolute value of x
pow(x, y) Returns x raised to the power y

**Q9. Consider the python code:**

```
1) import random
2) print(int(random.random()*5))
```

**Which of the following is true?**
**A) It will print a random int value from 0 to 5**
**B) It will print a random int value from 1 to 5**
**C) It will print a random int value from 0 to 5**
**D) It will print a random int value from 0 to 4**
**E) It will print 5**

**Answer: D**

**Explanation:**
**random() function will generates a random float value which is >=0 and <1.**
**random()*5 generates a float value which is >=0 and <5**
**Hence int(random.random()*5) will print a random int value from 0 to 4**

**Q10. Consider the code**

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

**Which of the following is valid?**
**A) It will print list of 10 unique random numbers from 0 to 6**
**B) It will print list of 7 unique random numbers from 0 to 9**
**C) It will print list of 7 unique random numbers from 0 to 10**
**D) It will print list of 7 unique random numbers from 1 to 10**

**Answer: B**

**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**

# Topic

# Document And Structure Code


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com

**Q1.**
**You want to add comments to your code so that other team members can understand it. What should you do?**
**A. Place the comments after the #sign on any line**
**B. Place the comments after the last line of the code separated by a blank line**
**C. Place the comments before the first line of code separated by a blank line**
**D. Place the comments inside parentheses anywhere**

**Answer: A**
**Explanation: If any line starts with #sign then it acts as python single line comment.**

**Q2.**
**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):**

```
01 # The calc_power function calculates exponents
02 # x is the base
03 # y is the exponent
04 # The value of x raided to the y power is returned
05 def calc_power(x, y):
06       comment="#Return the value"
07       return x**y #raise x to the power y
```

**Which of the following statements are true?**
**A) Lines 01 through 04 will be ignored for syntax checking**
**B) The hash sign(#) is optional for lines 01 and 03.**
**C) The String in line 06 will be interpreted as a comment**

**Answer: A**

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