SYED AMMAL ENGINEERING COLLEGE

(An ISO 9001: 2008 Certified Institution)
Dr. E.M. Abdullah Campus, Ramanathapuram – 623 502

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(Accredited by NBA)

PYTHON PROGRAMMING

TECHNICAL QUESTIONS AND ANSWERS

1. What is the output of the following code :

```
print 9//2
```

A) 4.5
B) 4.0
C) 4
D) Error

Answer: (C)

Explanation: The '//' operator in Python returns the integer part of the floating number.

2. Which function overloads the >> operator?

(A) more()
(B) gt()
(C) ge()
(D) rshift()
Answer: (D)
Explanation: rshift() overloads the >> operator

3. What is the output of the following program :

```
i = 0
while i < 3:
    print i
    print i+1
```

(A) 0 2 1 3 2 4
(B) 0 1 2 3 4 5
(C) 0 1 1 2 2 3
(D) 1 0 2 4 3 5

Answer: (C)

4. What is the output of the following program:

```
print "Hello World"[::-1]
```

(A) dlroW olleH
(B) Hello Worl
(C) d
(D) Error

Answer: (A)
Explanation: [::] depicts extended slicing in Python and [::-1] returns the reverse of the string.

5. Which module in Python supports regular expressions?
(A) re
(B) regex
(C) pyregex
(D) None of the above

Answer: (A)
Explanation: re is a part of the standard library and can be imported using: import re.

6. What is the output of the following program :

```
print 0.1 + 0.2 == 0.3
```

(A) True
(B) False
(C) Machine dependent
(D) Error

Answer: (B)

Explanation: Neither of 0.1, 0.2 and 0.3 can be represented accurately in binary. The round off errors from 0.1 and 0.2 accumulate and hence there is a difference of 5.5511e-17 between (0.1 + 0.2) and 0.3.

7. Which of these is not a core data type?
   (A) Lists
   (B) Dictionary
   (C) Tuples
   (D) Class

   Answer: (D)
   Explanation: Class is a user defined data type

8. What data type is the object below?
   L = [1, 23, 'hello', 1]
   (A) List
   (B) Dictionary
   (C) Tuple
   (D) Array

   Answer: (A)
   Explanation: [ ] defines a list

9. What is the output of the following program :

```
 def myfunc(a):
   a = a + 2
       a = a * 2
   return a

print myfunc(2)
```

   (A) 8
   (B) 16
   (C) Indentation Error
   (D) Runtime Error

   Answer: (C)
   Explanation: Python creates blocks of code based on the indentation of the code. Thus, new indent defines a new scope.

10. What is the output of the expression : 3*1**3

(A) 27
(B) 9
(C) 3
(D) 1

Answer: (C)
Explanation: Precedence of ** is higher than that of 3, thus first 1**3 will be executed and the result will be multiplied by 3.

11. What is the output of the following program :

```
print '{0:.2}'.format(1.0 / 3)
```

(A) 0.333333
(B) 0.33
(C) 0.333333:-2
(D) Error

Answer: (B)
Explanation: .2 defines the precision of the floating point number.

12. What is the output of the following program :

```
print '{0:-2%}'.format(1.0 / 3)
```

(A) 0.33
(B) 0.33%
(C) 33.33%
(D) 33%

Answer: (C)
Explanation: The % converts the 0.33 to percentage with respect to 1.0

13. What is the output of the following program :

```
i = 0
 while i < 3:
    print i
    i += 1
 else:
```

```
            print 0
```

(A) 0 1 2 3 0
(B) 0 1 2 0
(C) 0 1 2
(D) Error

Answer: (B)
Explanation: The else part is executed when the condition in the while statement is false.

14. What is the output of the following program :

```
i = 0
while i < 5:
   print(i)
   i += 1
   if i == 3:
      break
else:
   print(0)
```

(A) 0 1 2 0
(B) 0 1 2
(C) Error
(D) None of the above

Answer: (B)
Explanation: The else part is not executed if control breaks out of the loop.

15. What is the output of the following program :

```
print 'cd'.partition('cd')
```

(A) (‘cd’)
(B) (”)
(C) (‘cd’, ”, ”)
(D) (”, ‘cd’, ”)

Answer: (D)
Explanation: The entire string has been passed as the separator hence the first and the last item of the tuple returned are null strings.

16. What is the output of the following program :

```
print 'abcefd'.replace('cd', '12')
```

(A) ab1ef2
(B) abcefd
(C) ab1efd
(D) ab12ed2

Answer: (B)
Explanation: The first substring is not present in the given string and hence nothing is replaced.

17. What will be displayed by the following code?

```
def f(value, values):
    v = 1
    values[0] = 44
t = 3
v = [1, 2, 3]
f(t, v)
print(t, v[0])
```

(A) 1 1
(B) 1 44
(C) 3 1
(D) 3 44

Answer: (D)

Explanation: The value of t=3 is passed in funcion f(value,values) , v [list] is passed as values in the same function. The v is stored in values and values[0]=44 , changes the value at index[‘0’] in the list hence v=[44,2,3].

18. Predict the output of following python program:

```
 r = lambda q: q * 2
s = lambda q: q * 3
x = 2
x = r(x)
x = s(x)
x = r(x)
print x
```

Output: 24

Explanation: In the above program r and s are lambda functions or anonymous functions and q is the argument to both of the functions. In first step we have initialized x to 2. In second step we have passed x as argument to the lambda function r, this will return x*2 which is stored in x. That is, x = 4 now. Similarly in third step we have passed x to lambda function s, So x = 4*3. i.e, x = 12 now. Again in the last step, x is multiplied by 2 by passing it to function r. Therefore, x = 24.

19. Predict the output of following python programs

```
a = 4.5
b = 2
print a//b
```

Output: 2

Explanation : This type of division is called truncating division where the remainder is truncated or dropped.

20. Predict the output of following python programs

```
a = True
b = False
c = False

if a or b and c:
    print "SAEC"
else:
    print "saec"
```

Output: saec

Explanation : In Python, AND operator has higher precedence than OR operator. So, it is evaluated first. i.e, (b and c) evaluates to false.Now OR operator is evaluated. Here, (True or False) evaluates to True. So the if condition becomes True and SAEC is printed as output.

21. Predict the output of following python programs

```
 count = 1

 def doThis():

global count
    for i in (1, 2, 3):
      count += 1
 doThis()
 print count
```

Output: 4

Explanation: The variable count declared outside the function is global variable and also the count variable being referenced in the function is the same global variable defined outside of the function. So, the changes made to variable in the function are reflected to the original variable. So, the output of the program is 4.

22. Predict the output of following python programs

```
dictionary = {1:'1', 2:'2', 3:'3'}
del dictionary[1]
dictionary[1] = '10'
del dictionary[2]
print len(dictionary)
```

Output: 2

Explanation : The task of the 'del' function is to remove key-value pairs from a dictionary. Initially the size of the given dictionary was 3. Then the key value pair for key 1 is first removed and then added back with a new value. Then the key value pair for key 2 is removed. So, finally the size of the dictionary is 2.

23. Predict the output of following python programs

```
nameList = ['Harsh', 'Pratik', 'Bob', 'Dhruv']
print nameList[1][-1]
```

Output: k

Explanation:
The index position -1 represents either the last element in a list or the last character in a String. In the above given list of names "nameList", the index 1 represents the second element i.e, the second string "Pratik" and the index -1 represen jts the last character in the string "Pratik". So, the output is "k".

24. Predict the output of following python programs

```
geekCodes = [1, 2, 3, 4]
geekCodes.append([5,6,7,8])
print len(geekCodes)
```

Output: 5

Explanation:
The task of append() method is to append a passed *obj* into an existing list. But instead of passing a list to the append method will not merge the two lists, the entire list which is passed is added as an element of the list. So the output is 5.

25. Predict the output of following python programs

```
def addToList(listcontainer):
    listcontainer += [10]

mylistContainer = [10, 20, 30, 40]
addToList(mylistContainer)
print len(mylistContainer)
```

Output: 5

Explanation:
In Python, everything is a reference and references are passed by value. Parameter passing in Python is same as reference passing in Java. As a consequence, the function can modify the value referred by passed argument, i.e. the value of the variable in the caller's scope can be changed. Here the task of the function "addToList" is to add an element 10 in the list, So this will increase the length of list by 1. So the output of program is 5.

26.

```
def gfgFunction():
   "Syed Ammal Engineering College"
    return 1

print gfgFunction.__doc__[18:21]
```

Output: ring

Explanation:
There is a docstring defined for this method, by putting a string on the first line after the start of the function definition. The docstring can be referenced using the __doc__ attribute of the function. And hence it prints the indexed string.

27. Predict the output of following python programs

```
class A(object):
   val = 1

class B(A):
   pass

class C(A):
   pass

print A.val, B.val, C.val
B.val = 2
print A.val, B.val, C.val
A.val = 3
print A.val, B.val, C.val
```

Output:

```
  1 1 1
  1 2 1
  3 2 3
```

Explanation:
In Python, class variables are internally handled as dictionaries. If a variable name is not found in the dictionary of the current class, the class hierarchy (i.e., its parent classes) are searched until the referenced variable name is found, if the variable is not found error is being thrown.
So, in the above program the first call to print() prints the initialized value i.e, 1.
In the second call since B. val is set to 2, the output is 1 2 1.
The last output 3 2 3 may be surprising. Instead of 3 3 3, here B.val reflects 2 instead of 3 since it is overridden earlier.

28. Predict the output of following python programs

```
check1 = ['Learn', 'Quiz', 'Practice', 'Contribute']
check2 = check1
check3 = check1[:]
check2[0] = 'Code'
check3[1] = 'Mcq'
count = 0
for c in (check1, check2, check3):
  if c[0] == 'Code':
     count += 1
  if c[1] == 'Mcq':
     count += 10
print count
```

Output: 12

Explanation:
When assigning check1 to check2, we create a second reference to the same list. Changes to check2 affects check1. When assigning the slice of all elements in check1 to check3, we are creating a full copy of check1 which can be modified independently (i.e, any change in check3 will not affect check1).
So, while checking check1 ‘Code’ gets matched and count increases to 1, but Mcq doest gets matched since its available only in check3.
Now checking check2 here also ‘Code’ gets matched resulting in count value to 2. Finally while checking check3 which is separate than both check1 and check2 here only Mcq gets matched and count becomes 12.

29. Predict the output of following python programs

```
list1 = ['physics', 'chemistry', 1997, 2000]

list2 = [1, 2, 3, 4, 5, 6, 7 ]
print "list1[0]: ", list1[0]       #statement 1
print "list1[0]: ", list1[-2]      #statement 2
print "list1[-2]: ", list1[1:]     #statement 3
print "list2[1:5]: ", list2[1:5]   #statement 4
```

Output:

```
list1[0]:  physics
list1[0]:  1997
list1[-2]:  ['chemistry', 1997, 2000]
```

```
   list2[1:5]:  [2, 3, 4, 5]
```

Explanation:
To access values in lists, we use the square brackets for slicing along with the index or indices to obtain required value available at that index.For N items in a List MAX value of index will be N-1.
Statement 1 : This will print item located at index 0 in Output.
Statement 2 : This will print item located at index -2 i.e.second last element in Output.
Statement 3 : This will print items located from index 1 to end of the list.
Statement 4 : This will print items located from index 1 to 4 of the list.

30. Predict the output of following python programs

```
list1 = ['physics', 'chemistry', 1997, 2000]
print "list1[1][1]: ", list1[1][1] #statement 1
print "list1[1][-1]: ", list1[1][-1] #statement 2
```

Output:

```
   list1[1][1]:  h
   list1[1][-1]:  y
```

Explanation:
In python we can slice a list but we can also slice a element within list if it is a string. The declaration list[x][y] will mean that ‘x’ is the index of element within a list and ‘y’ is the index of entity within that string.

31. Predict the output of following python programs

```
list1 = range(100, 110) #statement 1
print "index of element 105 is : ", list1.index(105)  #statement 2
```

Output:

```
   index of element 105 is :  5
```

Explanation:
Statement 1 : will genetrate numbers from 100 to 110 and appent all these numbers in the list.
Statement 2 : will give the index value of 105 in the list list1.

32. Predict the output of following python programs

```
list1 = [1, 2, 3, 4, 5]
list2 = list1
list2[0] = 0;
print "list1= : ", list1 #statement 2
```

Output:

```
list1= :  [0, 2, 3, 4, 5]
```

Explanation:
In this problem, we have provided a reference to the list1 with another name list2 but these two lists are same which have two references(list1 and list2). So any alteration with list2 will affect the original list.

33. Predict the output of following python programs

```
list = ['python', 'learning', 'for', 'technical', 'skill', 'training']

print list[::]
print list[0:6:2]
print list[ :6: ]
print list[ :6:2]
print list[ ::3]
print list[ ::-2]
```

Output:

```
['python', 'learning', 'for', 'technical', 'skill', 'training']
['python', 'for', 'skill']
['python', 'learning', 'for', 'technical', 'skill', 'training']
['python', 'for', 'skill']
['python', 'technical']
['training', 'technical', 'learning']
```

Explanation:
In python list slicing can also be done by using the syntax listName[x:y:z] where x means the initial index, y-1 defines the final index value and z specifies the step size. If anyone of the values among x, y and z is missing the interpreter takes default value.

34. Predict the output of following python programs

```
dictionary1 = {'Google' : 1,
               'Facebook' : 2,
               'Microsoft' : 3
               }
dictionary2 = {'GFG' : 1,
               'Microsoft' : 2,
               'Youtube' : 3
               }
dictionary1.update(dictionary2);
for key, values in dictionary1.items():
    print(key, values)
```

a) Compilation error
b) Runtime error
c) (‘Google’, 1)
(‘Facebook’, 2)
(‘Youtube’, 3)
(‘Microsoft’, 2)
(‘GFG’, 1)
d) None of these

Ans. (c)
Explanation: dictionary1.update(dictionary2) is used to update the entries of dictionary1 with entries of dictionary2. If there are same keys in two dictionaries, then the value in second dictionary is used.

35. What is the output of the following program?

```
dictionary1 = {'GFG' : 1,
              'Google' : 2,
              'GFG' : 3
              }
print(dictionary1['GFG']);
```

a) Compilation error due to duplicate keys
b) Runtime time error due to duplicate keys
c) 3
d) 1

Ans. (c)
Explanation: Here, GFG is the duplicate key. Duplicate keys are not allowed in python. If there are same keys in a dictionay, then the **value assigned mostly recently** is assigned to the that key.

36. What is the output of the following program?

```
temp = dict()
temp['key1'] = {'key1' : 44, 'key2' : 566}
temp['key2'] = [1, 2, 3, 4]
for (key, values) in temp.items():
    print(values, end = "")
```

a) Compilation error
b) {‘key1’: 44, ‘key2’: 566}[1, 2, 3, 4]
c) Runtime error
d) None of the above

Ans. (b)
Explanation: A dictionary can hold any value such as an integer, string, list or even another dictionary holding key value pairs.
Note: This code can be executed only in python versions above 3

37. What is the output of the following program?

```
data = [2, 3, 9]
temp = [[x for x in[data]] for x in range(3)]
print (temp)
```

a) [[[2, 3, 9]], [[2, 3, 9]], [[2, 3, 9]]]
b) [[2, 3, 9], [2, 3, 9], [2, 3, 9]]
c) [[[2, 3, 9]], [[2, 3, 9]]]
d) None of these

Ans. (a)
Explanation: [x for x in[data] returns a new list copying the values in the list data and the outer for statement prints the newly created list 3 times.

38. What is the output of the following program?

```
data = [x for x in range(5)]
temp = [x for x in range(7) if x in data and x%2==0]
print(temp)
```

a) [0, 2, 4, 6]
b) [0, 2, 4]
c) [0, 1, 2, 3, 4, 5]
d) Runtime error

Ans. (b)
Explanation: The is statement checks whether the value lies in list data and if it does whether it’s divisible by 2. It does so for x in (0, 7).

39. What is the output of the following program?

```
L1 = list()
L1.append([1, [2, 3], 4])
L1.extend([7, 8, 9])
print(L1[0][1][1] + L1[2])
```

a) TypeError: can only concatenate list (not “int”) to list
b) 12
c) 11
d) 38
Ans. (c)
Explanation: In the print(), indexing is used. L1[0] denotes [1, [2, 3], 4], L1[0][1] denotes [2, 3],
L1[0][1][1] = 3 and L1[2] = 8. Thus, the two integers are added, 3 + 8 = 11 and output comes as 11.

40. What is the output of the following program?

```
L1 = [1, 2, 3, 4]
L2 = L1
L3 = L1.copy()
L4 = list(L1)
L1[0] = [5]
print(L1, L2, L3, L4)
```

a) [5, 2, 3, 4] [5, 2, 3, 4] [1, 2, 3, 4] [1, 2, 3, 4]
b) [[5], 2, 3, 4] [[5], 2, 3, 4] [[5], 2, 3, 4] [1, 2, 3, 4]
c) [5, 2, 3, 4] [5, 2, 3, 4] [5, 2, 3, 4] [1, 2, 3, 4]
d) [[5], 2, 3, 4] [[5], 2, 3, 4] [1, 2, 3, 4] [1, 2, 3, 4]
Ans. (d)
Explanation: List L2 is the Shallow copy of L1, while L3 and L4 are Deep Copy(True Copy) of List L1. L1[0] = [5], implies that at index 0, list [5] will be present and not integer value 5.

41. What is the output of the following program?

```
import sys
L1 = tuple()
```

```
print(sys.getsizeof(L1), end = " ")
L1 = (1, 2)
print(sys.getsizeof(L1), end = " ")
L1 = (1, 3, (4, 5))
print(sys.getsizeof(L1), end = " ")
L1 = (1, 2, 3, 4, 5, [3, 4], 'p', '8', 9.777, (1, 3))
print(sys.getsizeof(L1))
```

a) 0 2 3 10
b) 32 34 35 42
c) 48 64 72 128
d) 48 144 192 480
Ans. (c)
Explanation: An Empty Tuple has 48 Bytes as Overhead size and each additional element requires 8 Bytes.
(1, 2) Size: 48 + 2 * 8 = 64
(1, 3, (4, 5)) Size: 48 + 3 * 8 = 72
(1, 2, 3, 4, 5, [3, 4], ‘p’, ‘8’, 9.777, (1, 3)) Size: 48 + 10 * 8 = 128

42. What is the output of the following program?

```
List = [True, 50, 10]
List.insert(2, 5)
print(List, "Sum is: ", sum(List))
```

a) [True, 50, 10, 5] Sum is: 66
b) [True, 50, 5, 10] Sum is: 65
c) TypeError: unsupported operand type(s) for +: ‘int’ and ‘str’
d) [True, 50, 5, 10] Sum is: 66
Ans. (d)
Explanation: The List is initially has 3 elements. The insert() adds element 5 at index 2, moving element 10 at index 3 and the List becomes [True, 50, 5, 10]. Boolean has an integer value of 1, thus sum becomes 1 + 50 + 5 + 10 = 66.

43. What is the output of the following program?

```
T = (1, 2, 3, 4, 5, 6, 7, 8)
print(T[T.index(5)], end = " ")
print(T[T[T[6]-3]-6])
```

a) 4 0
b) 5 8
c) 5 IndexError

d) 4 1

Ans. (b)

Explanation: The inbuilt function index() returns the index of the element. T.index(5) = 4 and T[4] = 5. The other print statement has indexing of tuples, similar to that of Lists. T[6] = 7, T[6]-3 = 4, T[T[6]-3] = 5, T[T[6]-3]-6 = -1 and T[T[T[6]-3]-6], i.e. T[-1] = 8.

44. What is the output of the following program?

```
L = [1, 3, 5, 7, 9]
print(L.pop(-3), end = '  ')
print(L.remove(L[0]), end = '  ')
print(L)
```

a) 5 None [3, 7, 9]
b) 5 1 [3, 7, 9]
c) 5 1 [3, 7, 9]
d) 5 None [1, 3, 7, 9]

Ans. (a)

Explanation: pop() will delete and return the element whose index was passed as parameter. L.pop(-3) will delete 5 and return 5, which is printed by print(). remove() does return any value, it’s a void function. L[0] = 1, L.remove(1) will delete 1 from the list and the list remains to be [3, 7, 9].

45. What is the output of the following program?

```
def REVERSE(L):
    L.reverse()
    return(L)
def YKNJS(L):
    List = list()
    List.extend(REVERSE(L))
   print(List)

L = [1, 3.1, 5.31, 7.531]
YKNJS(L)
```

a) [1, 3.1, 5.31, 7.531]
b) [7.531, 5.31, 3.1, 1]
c) IndexError
d) AttributeError: ‘NoneType’ object has no attribute ‘REVERSE’

Ans. (b)

Explanation: REVERSE() reverses the list and returns it. YKNJS() adds reverse of a list L to the empty list List. L = [1, 3.1, 5.31, 7.531], gets reversed and becomes [7.531, 5.31, 3.1, 1].

46. What is the output of the following program?

```
from math import sqrt
L1 = [x**2 for x in range(10)].pop()
L1 + = 19
print(sqrt(L1), end = " ")
L1 = [x**2 for x in reversed(range(10))].pop()
L1 + = 16
print(int(sqrt(L1)))
```

a) 10.0 4.0
b) 4.3588 4
c) 10 .0 4
d) 10.0 0
Ans. (c)
Explanation: The first list comphrension will create list as [0, 1, 4, 9, 16, 25, 36, 49, 64, 81] and pop() will return 81. 81 + 19 = 100, whose square root is 10.0 and similarly in 2nd case pop() will return 0 due to the reversed range and integer value of square root of 16 is 4.

47. What is the output of the following program?

```
D = dict()
for x in enumerate(range(2)):
    D[x[0]] = x[1]
    D[x[1]+7] = x[0]
print(D)
```

a) KeyError
b) {0: 1, 7: 0, 1: 1, 8: 0}
c) {0: 0, 7: 0, 1: 1, 8: 1}
d) {1: 1, 7: 2, 0: 1, 8: 1}
Ans. (c)
Explanation: enumerate() will return a tuple, the loop will have x = (0, 0), (1, 1). Thus D[0] = 0, D[1] = 1, D[0 + 7] = D[7] = 0 and D[1 + 7] = D[8] = 1.
Note: Dictionary is unordered, so the sequence of the key-value pair may differ in each output.

48. What is the output of the following program?

```
D = {1 : 1, 2 : '2', '1' : 1, '2' : 3}
D['1'] = 2
print(D[D[D[str(D[1])]]])
```

a) 2
b) 3
c) ‘2’
d) KeyError
Ans. (b)
Explanation: Simple key-value pair is used recursively, D[1] = 1, str(1) = ‘1’. So, D[str(D[1])] = D[‘1’] = 2, D[2] = ‘2’ and D[‘2’] = 3.

49. What is the output of the following program?

```
D = dict()
for i in range (3):
    for j in range(2):
        D[i] = j
print(D)
```

a) {0: 0, 1: 0, 2: 0}
b) {0: 1, 1: 1, 2: 1}
c) {0: 0, 1: 0, 2: 0, 0: 1, 1: 1, 2: 1}
d) TypeError: Immutable object
Ans. (b)
Explanation: 1st loop will give 3 values to i 0, 1 and 2. In the empty dictionary, valued are added and overwrited in j loop, for eg. D[0] = [0] becomes D[0] = 1, due to overwriting.

50. What is the output of the following program?

```
from math import *
a = 2.13
b = 3.7777
c = -3.12
print(int(a), floor(b), ceil(c), fabs(c))
```

a) 2 3 -4 3
b) 2 3 -3 3.12
c) 2 4 -3 3
d) 2 3 -4 3.12
Ans. (b)

Explanation: int() returns the integer value of a number, int(2.13) = 2. floor() returns the largest integer lesser or equal to the number, floor(3.777) = 3. ceil() returns smallest integer greater or equal to the number, ceil(-3.12) = -3. fabs() return the modulus of the number, thus fabs(-3.12) = 3.12.

51. What is the output of the following program?

```
x = ['ab', 'cd']
for i in x:
    i.upper()
print(x)
```

Output:

```
['ab', 'cd']
```

Explanation:
The function upper() does not modify a string in place, but it returns a new string which here isn’t being stored anywhere. So we will get our original list as output.

52. What is the output of the following program?

```
x = ['ab', 'cd']
for i in x:
    x.append(i.upper())
print(x)
```

Output:

```
No Output
```

Explanation:
The loop does not terminate as new elements are being added to the list in each iteration. So our program will stuck in infinite loop.

53. What is the output of the following program?

```
for i in [1, 2, 3, 4][::-1]:
    print (i)
```

Output:

```
4
```

```
3
2
1
```

Explanation:
Adding [::-1] beside your list reverses the list. So output will be the elements of orignal list but in reverse order.

54. What is the output of the following program?

```
i = 0
while i < 3:
    print(i)
    i += 1
else:
    print(0)
```

1. 0 1 2 3 0
2. 0 1 2 0
3. 0 1 2
4. Error

Output:

2. 0 1 2 0

Explanation: The else part is executed when the condition in the while statement is false.

55. What is the output of the following program?

```
my_string = 'geeksforgeeks'
for i in range(my_string):
    print(i)
```

1. 0 1 2 3 … 12
2. saec
3. None
4. Error

Output: 4. Error

Explanation: ***range(str)*** is not allowed

56. What is the output of the following program?

```
my_string = 'geeksforgeeks'
for i in range(len(my_string)):
    my_string[i].upper()
print (my_string)
```

1. SAEC
2. saec
3. Error
4. None

Output:

2.saec

Explanation: Changes do not happen in-place, rather it will return a new instance of the string.

57. What is the output of the following program?

```
my_string = 'geeksforgeeks'
for i in range(len(my_string)):
    print (my_string)
    my_string = 'a'
```

1. gaaaaaaaaaaaa
2. saec a a a a a a a a a a a a
3. Error
4. None

Output:

2. saec a a a a a a a a a a a a

Explanation: String is modified only after ‘geeksforgeeks’ has been printed once.

58. What is the output of the following program?

```
numberGames = {}
numberGames[(1,2,4)] = 8
numberGames[(4,2,1)] = 10
numberGames[(1,2)] = 12
sum = 0
for k in numberGames:
  sum += numberGames[k]
print len(numberGames) + sum
```

Output: 33

Explanation:
Tuples can be used for keys into dictionary. The tuples can have mixed length and the order of the items in the tuple is considered when comparing the equality of the keys.

59. What is the output of the following program?

```
t = (1, 2)
print 2 * t
```

Output: (1, 2, 1, 2)

Explanation:
Asterick Operatr (*) operator concatenates tuple.

60. What is the output of the following program?

```
d1 = {"john":40, "peter":45}
d2 = {"john":466, "peter":45}
print d1 > d2
```

Output: False

Explanation:
Arithmetic operator less than (<) or greater than (>) can be used with dictionaries and each corresponding key with its values are compared

61. What is the output of the following program?

```
my_tuple = (6, 9, 0, 0)
my_tuple1 = (5, 2, 3, 4)
print my_tuple > my_tuple1
```

Output: True

Explanation: Each elements of the tuples are compared one by one and if maximum number of elements are there in tuple1 which are greater of equal to corresponding element of tuple2 then tuple1 is said to be grater than tuple2.

62. What is the output of the following program?

```
L = list('123456')
L[0] = L[5] = 0
L[3] = L[-2]
print(L)
```

a) [0, ‘2’, ‘3’, ‘4’, ‘5’, 0]
b) [‘6’, ‘2’, ‘3’, ‘5’, ‘5’, ‘6’]
c) [‘0’, ‘2’, ‘3’, ‘5’, ‘5’, ‘0’]
d) [0, ‘2’, ‘3’, ‘5’, ‘5’, 0]
Ans. (d)
Explanation: L[0] is ‘1’ and L[5] is ‘6’, both of these elements will be replaced by 0 in the List. L[3], which is 4 will be replaced L[-2] i.e. 5.

63. What is the output of the following program?

```
T = tuple('geeks')
a, b, c, d, e = T
b = c = '*'
T = (a, b, c, d, e)
print(T)
```

a) (‘g’, ‘*’, ‘*’, ‘k’, ‘s’)
b) (‘g’, ‘e’, ‘e’, ‘k’, ‘s’)
c) (‘geeks’, ‘*’, ‘*’)
d) KeyError
Ans. (a)
Explanation: A tuple is created as T = (‘g’, ‘e’, ‘e’, ‘k’, ‘s’), then it is unpacked into a, b, c, d and e, mapping from ‘g’ to a and ‘s’ to e. b and c which are both ‘e’ are equal to ‘*’ and then the existing tuple is replaced by packing a, b, c, d and e into a tupple T

64. What is the output of the following program?

```
L = [2e-04, 'a', False, 87]
T = (6.22, 'boy', True, 554)
for i in range(len(L)):
    if L[i]:
        L[i] = L[i] + T[i]
    else:
        T[i] = L[i] + T[i]
        break
```

a) [6.222e-04, ‘aboy’, True, 641]
b) [6.2202, ‘aboy’, 1, 641]
c) [6.2202, ‘aboy’, True, 87]
d) [6.2202, ‘aboy’, False, 87]

Ans. (d)
Explanation: The for loop will run for i = 0 to i = 3, i.e. 4 times(len(L) = 4). 2e-04 is same as 0.0002, thus L[i] = 6.22 + 0.0002 = 6.2202. String addition will result in concatenation, ‘a’ + ‘boy’ = ‘aboy’. False + True is True, it’ll return the integer value of 1. As tuples are immutable, the code will end with TypeError, but elements of L will be updated.

65. What is the output of the following program?

```
T = (2e-04, True, False, 8, 1.001, True)
val = 0
for x in T:
    val += int(x)
print(val)
```

a) 12
b) 11
c) 11.001199999999999
d) TypeError

Ans. (b)
Explanation: Integer value of 2e-04(0.0002) is 0, True holds a value 1 and False a 0, integer value of 1.001 is 1. Thus total 0 + 1 + 0 + 8 + 1 + 1 = 11.

66. What is the output of the following program?

```
L = [3, 1, 2, 4]
```

```
T = ('A', 'b', 'c', 'd')
L.sort()
counter = 0
for x in T:
    L[counter] += int(x)
    counter += 1
    break
print(L)
```

a) [66, 97, 99, 101]
b) [66, 68, 70, 72]
c) [66, 67, 68, 69]
d) ValueError

Ans. (d)
Explanation: After sort(L), L will be = [1, 2, 3, 4]. Counter = 0, L[0] i.e. 1, x = ‘A’, but Type Conversion of char ‘A’ to integer will throw error and the value cannot be stored in L[0], thus a ValueError.

67. What is the output of the following program?

```
str1 = '{2}, {1} and {0}'.format('a', 'b', 'c')
str2 = '{0}{1}{0}'.format('abra', 'cad')
print(str1, str2)
```

a) c, b and a abracad0
b) a, b and c abracadabra
c) a, b and c abracadcad
d) c, b and a abracadabra

Ans. (d)
Explanation: String function format takes a format string and an arbitrary set of positional and keyword arguments. For str1 ‘a’ has index 2, ‘b’ index 1 and ‘c’ index 0. str2 has only two indices 0 and 1. Index 0 is used twice at 1st and 3rd time.

68. What is the output of the following program?

```
a = 2
b = '3.77'
c = -8
str1 = '{0:.4f} {0:3d} {2} {1}'.format(a, b, c)
print(str1)
```

a) 2.0000 2 -8 3.77
b) 2 3.77 -8 3.77
c) 2.000 3 -8 3.77
d) 2.000 2 8 3.77

Ans. (a)
Explanation: At Index 0, integer a is formatted into a float with 4 decimal points, thus 2.0000. At Index 0, a = 2 is formatted into a integer, thus it remains to 2. Index 2 and 1 values are picked next, which are -8 and 3.77 respectively.

69. What is the output of the following program?

```
import string
import string
Line1 = "And Then There Were None"
Line2 = "Famous In Love"
Line3 = "Famous Were The Kol And Klaus"
Line4 = Line1 + Line2 + Line3
print(string.find(Line1, 'Were'), string.count((Line4), 'And'))
```

a) True 1
b) 15 2
c) (15, 2)
d) True 2

Ans. (c)
Explanation: ‘Were’ is at Index 15 in Line1, find() returns the index of substring if found in the string Line1. count() returns the total number of occurences of the substring. Line4 is concatenated string from Line1, Line2 and Line3. This code works well with Python v2.x, as some string functions are deprecated in Python v3.x.

70. What is the output of the following program?

```
line =  "What will have so will"
L = line.split('a')
for i in L:
    print(i, end=' ')
```

a)[‘What’, ‘will’, ‘have’, ‘so’, ‘will’]
b) Wh t will h ve so will
c) What will have so will
d) [‘Wh’, ‘t will h’, ‘ve so will’]

Ans. (b)
Explanation: split() will use ‘a’ as the delimiter. It’ll create parition at ‘a’, thus split() return an array L, which is in [‘Wh’, ‘t will h’, ‘ve so will’]. For loop will print the elements of the list.

71. What is the output of the following program?

```
tuple = (1, 2, 3, 4)
tuple.append( (5, 6, 7) )
print(len(my_tuple))
```

Options:

1. 1
2. 2
3. 5
4. Error

Output:

4. Error

Explanation: In this case an exception will be thrown as tuples are immutable and don’t have an append method.

72. What is the output of the following program?

```
tuple = {}
tuple[(1,2,4)] = 8
tuple[(4,2,1)] = 10
tuple[(1,2)] = 12
_sum = 0
for k in tuple:
   _sum += tuple[k]
print(len(tuple) + _sum)
```

Options:

1. 34
2. 12
3. 31
4. 33

Output: 33

Explanation: Tuples can be used for keys into dictionary. The tuples can have mixed length and the order of the items in the tuple is considered when comparing the equality of the keys.

73. What is the output of the following program?

```
tuple1 = (1, 2, 4, 3)
tuple2 = (1, 2, 3, 4)
print(tuple1 < tuple2)
```

Options:

1. Error
2. True
3. False
4. Unexpected

Output: False

Explanation: In this case elements will be compared one by one. So, when it compare 4 with 3 it will return False.

74. What is the output of the following program?

```
tuple=("Check")*3
print(tuple)
```

Options:

1. Unexpected
2. (3Check)
3. (CheckCheckCheck)
4. Syntax Error

Output:

3. (CheckCheckCheck)

Explanation: Here (“Check”) will be treated as is a string not a tuple as there is no comma after the element.

75. What is the output of the following program?

```
print(bool('False'))
print(bool())
```

1. False, True
2. None, None
3. True, True
4. True, False

Output:

4. True, False

Explanation: If the argument passed to the bool function does not amount to zero then the Boolean function returns true else it always returns false. In the above code, in first line ‘False’ is passed to the function which is not amount to 0. Therefore output is true. In the second line, an empty list is passed to the function bool. Hence the output is false.

76. What is the output of the following program?

```
print(not(4>3))
print(not(5&5))
```

1. False, False
2. None, None
3. True, True
4. Ture, False

Output:

1. False, False

Explanation: The **not** function returns true if the argument is false, and false if the argument is true. Hence the first line of above code returns false, and the second line will also returns false.

77. What is the output of the following program?

```
print(['love', 'python'][bool('gfg')])
```

1. love
2. python
3. gfg
4. None

Output: python

Explanation: We can read the above code as print ‘love’ if the argument passed to the Boolean function is zero else print ‘python’. The argument passed to the Boolean function in the above code is ‘gfg’, which does not amount to zero and hence the output is: **“python”**.

78. What is the output of the following program?

```
mylist =[0, 5, 2, 0, 'gfg', '', []]
print(list(filter(bool, mylist)))
```

1. [0, 0,]
2. [0, 5, 2, 0, ‘gfg’, ”, []]
3. Error
4. [5, 2, ‘gfg’]

Output:

4. [5, 2, 'gfg']

Explanation: The code above returns a new list containing only those elements of the list mylist which are not equal to zero. Hence the output is: **[5, 2, ‘gfg’].**

79. What is the output of the following program?

```
if (7 < 0) and (0 < -7):
    print("abhi")
elif (7 > 0) or False:
    print("love")
else:
    print("flowers")
```

1. flowers

2. love
3. abhi
4. Error

Output:

2. love

Explanation: The code shown above prints the appropriate option depending on the conditions given. The condition which matches is (7>0), and hence the output is: **"love".**

80. What is the output of the following program?

```
mylist = ['saec', 'cse']
for i in mylist:
    i.upper()
print(mylist)
```

1. ['SAEC', 'CSE'].
2. ['saec', 'cse'].
3. [None, None].
4. Unexpected

Output: ['saec', 'cse']

Explanation: The function ***upper()*** does not modify a string in place, it returns a new string which isn't being stored anywhere.

81. What is the output of the following program?

```
sets = {1, 2, 3, 4, 4}
print(sets)
```

Options:

1. {1, 2, 3}
2. {1, 2, 3, 4}
3. {1, 2, 3, 4, 4}
4. Error

Output: {1, 2, 3, 4}

Explanation : Duplicate values are not allowed in sets. Hence, the output of the code shown above will be a set containing the duplicate value only once. Hence output will be {1, 2, 3, 4}.

82. What is the output of the following program?

```
sets = {3, 4, 5}
sets.update([1, 2, 3])
print(sets)
```

Options:

1. {1, 2, 3, 4, 5}
2. {3, 4, 5, 1, 2, 3}
3. {1, 2, 3, 3, 4, 5}
4. Error

Output: {1, 2, 3, 4, 5}

Explanation: The method update adds elements to a set.

83. What is the output of the following program?

```
set1 = {1, 2, 3}
set2 = set1.copy()
set2.add(4)
print(set1)
```

Options:

1. {1, 2, 3, 4}
2. {1, 2, 3}
3. Invalid Syntax
4. Error

Output: {1, 2, 3}

Explanation: In the above piece of code, set2 is barely a copy and not an alias of set1. Hence any change made in set2 isn’t reflected in set1.

84. What is the output of the following program?

```
set1 = {1, 2, 3}
```

```
set2 = set1.add(4)
print(set2)
```

Options:

1. {1, 2, 3, 4}
2. {1, 2, 3}
3. Invalid Syntax
4. None

Output: None

Explanation: add method doesn’t return anything. Hence there will be no output.

85. What is the output of the following program?

```
set1 = {1, 2, 3}
set2 = {4, 5, 6}
print(len(set1 + set2))
```

Options:

1. 3
2. 6
3. Unexpected
4. Error

Output:

4. Error

Explanation: unsupported operand type(s) for +: ‘set’ and ‘set’.

86. What is the type of each element in sys.argv?

a) set
b) list
c) tuple
d) string

Answer: d
Explanation: It is a list of strings.

87. What is the length of sys.argv?

a) number of arguments

b) number of arguments + 1
c) number of arguments – 1
d) none of the mentioned

Answer: b
Explanation: The first argument is the name of the program itself. Therefore the length of sys.argv is one more than the number arguments.

88. What is the output of the following code?

```
def foo(k):
    k[0] = 1
q = [0]
foo(q)
print(q)
```

a) [0].
b) [1].
c) [1, 0].
d) [0, 1].

Answer: b
Explanation: Lists are passed by reference

89. What is the output of the following code?

```
def foo(fname, val):
    print(fname(val))
foo(max, [1, 2, 3])
foo(min, [1, 2, 3])
```

a) 3 1
b) 1 3
c) error
d) none of the mentioned

Answer: a
Explanation: It is possible to pass function names as arguments to other functions.

90. What is the output of the following?

```
elements = [0, 1, 2]
def incr(x):
        return x+1
print(list(map(elements, incr)))
```

a) [1, 2, 3].
b) [0, 1, 2].
c) error
d) none of the mentioned

Answer: c
Explanation: The list should be the second parameter to the mapping function.

91. What is the output of the following?

```
elements = [0, 1, 2]
def incr(x):
        return x+1
print(list(map(incr, elements)))
```

a) [1, 2, 3].
b) [0, 1, 2].
c) error
d) none of the mentioned

Answer: a
Explanation: Each element of the list is incremented.

92. What is the output of the following?

```
def to_upper(k):
 return k.upper()
x = ['ab', 'cd']
print(list(map(to_upper, x)))
```

a) [‘AB’, ‘CD’].
b) [‘ab’, ‘cd’].
c) none of the mentioned
d) error

Answer: a
Explanation: Each element of the list is converted to uppercase.

93. What is the output of the following?

```
x = ['ab', 'cd']
print(len(list(map(list, x))))
```

a) 2
b) 4
c) error
d) none of the mentioned

Answer: a
Explanation: The outer list has two lists in it. So it’s length is 2.

94. Program code making use of a given module is called a ______ of the module.

a) Client
b) Docstring
c) Interface
d) Modularity

Answer: a
Explanation: Program code making use of a given module is called the client of the module. There may be multiple clients for a module.

95. What is the output of the following piece of code?

```
#mod1
def change(a):
        b=[x*2 for x in a]
        print(b)
#mod2
def change(a):
        b=[x*x for x in a]
        print(b)
from mod1 import change
from mod2 import change
#main
s=[1,2,3]
change(s)
```

a) [2,4,6].
b) [1,4,9].
c) [2,4,6].
d) There is a name clash

Answer: d
Explanation: A name clash is when two different entities with the same identifier become part of the same scope. Since both the modules have the same function name, there is a name clash.

96. What is the output of the following program?

```
tday=datetime.date.today()
print(tday.month())
```

a) August
b) Aug
c) 08
d) 8

Answer: d
Explanation: The code shown above prints the month number from the system date. Therefor the output will be 8 if the system date is 18th August, 2016.

97. Which of the following formatting options can be used in order to add 'n' blank spaces after a given string 'S'?

a) print("-ns"%S)
b) print("-ns"%S)
c) print("%ns"%S)
d) print("%-ns"%S)

Answer: d
Explanation: In order to add 'n' blank spaces after a given string 'S', we use the formatting option:("%-ns"%S).

98. What is the output of the following program?

```
f = None
for i in range (5):
    with open("data.txt", "w") as f:
        if i > 2:
      break
print(f.closed)
```

a) True
b) False
c) None
d) Error

Answer: a
Explanation: The WITH statement when used with open file guarantees that the file object is closed when the with block exits.

99. To read the entire remaining contents of the file as a string from a file object infile, we use

a) infile.read(2)
b) infile.read()
c) infile.readline()
d) infile.readlines()

Answer: b
Explanation: read function is used to read all the lines in a file.

100. Suppose t = (1, 2, 4, 3), which of the following is incorrect?

a) print(t[3])
b) t[3] = 45
c) print(max(t))
d) print(len(t))

Answer: b
Explanation: Values cannot be modified in the case of tuple, that is, tuple is immutable.