Python MCQ's

1. Who developed Python Programming Language?
a) Wick van Rossum
b) Rasmus Lerdorf
c) Guido van Rossum
d) Niene Stom

Explanation: Python language is designed by a Dutch programmer Guido van Rossum in the Netherlands.

---

2. Which type of Programming does Python support?
a) object-oriented programming
b) structured programming
c) functional programming
d) all of the mentioned

Explanation: Python is an interpreted programming language, which supports object-oriented, structured, and functional programming.

---

3. Is Python case sensitive when dealing with identifiers?
a) no
b) yes
c) machine dependent
d) none of the mentioned

Explanation: Case is always significant while dealing with identifiers in python.

---

4. Which of the following is the correct extension of the Python file?
a) .python
b) .pl
c) .py
d) .p

Explanation: '.py' is the correct extension of the Python file. Python programs can be written in any text editor. To save these programs we need to save in files with file extension '.py'.

5. Is Python code compiled or interpreted?
a) Python code is both compiled and interpreted
b) Python code is neither compiled nor interpreted
c) Python code is only compiled
d) Python code is only interpreted

Explanation: Many languages have been implemented using both compilers and interpreters, including C, Pascal, and Python.

---

6. All keywords in Python are in _____
a) Capitalized
b) lower case
c) UPPER CASE
d) None of the mentioned

Explanation: True, False and None are capitalized while the others are in lower case.

---

7. What will be the value of the following Python expression?

```
4 + 3 % 5
```

a) 7
b) 2
c) 4
d) 1

Explanation: The order of precedence is: %, +. Hence the expression above, on simplification results in 4 + 3 = 7. Hence the result is 7.

---

8. Which of the following is used to define a block of code in Python language?
a) Indentation
b) Key
c) Brackets
d) All of the mentioned

Explanation: In Python, to define a block of code we use indentation. Indentation refers to whitespaces at the beginning of the line.

9. Which keyword is used for function in Python language?
a) Function
b) def
c) Fun
d) Define

Explanation: The def keyword is used to create, (or define) a function in python.

10. Which of the following character is used to give single-line comments in Python?
a) //
b) #
c) !
d) /*

Explanation: To write single-line comments in Python use the Hash character (#) at the beginning of the line. It is also called number sign or pound sign. To write multi-line comments, close the text between triple quotes.
Example: “”” comment
text “””

11. What will be the output of the following Python code?

```
i = 1
while True:
    if i%3 == 0:
        break
    print(i)

    i + = 1
```

a) 1 2 3
b) error
c) 1 2
d) none of the mentioned

Explanation: SyntaxError, there shouldn’t be a space between + and = in +=.

12. Which of the following functions can help us to find the version of python that we are currently working on?
a) sys.version(1)

b) sys.version(0)
c) sys.version()
d) sys.version

Explanation: The function sys.version can help us to find the version of python that we are currently working on. It also contains information on the build number and compiler used. For example, 3.5.2, 2.7.3 etc. this function also returns the current date, time, bits etc along with the version.

13. Python supports the creation of anonymous functions at runtime, using a construct called _____
a) pi
b) anonymous
c) lambda
d) none of the mentioned

Explanation: Python supports the creation of anonymous functions (i.e. functions that are not bound to a name) at runtime, using a construct called lambda. Lambda functions are restricted to a single expression. They can be used wherever normal functions can be used.

14. What is the order of precedence in python?
a) Exponential, Parentheses, Multiplication, Division, Addition, Subtraction
b) Exponential, Parentheses, Division, Multiplication, Addition, Subtraction
c) Parentheses, Exponential, Multiplication, Division, Subtraction, Addition
d) Parentheses, Exponential, Multiplication, Division, Addition, Subtraction

Explanation: For order of precedence, just remember this PEMDAS (similar to BODMAS).

15. What will be the output of the following Python code snippet if x=1?

```
x<<2
```

a) 4
b) 2
c) 1
d) 8

Explanation: The binary form of 1 is 0001. The expression x<<2 implies we are performing bitwise left shift on x. This shift yields the value: 0100, which is the binary form of the number 4.

16. What does pip stand for python?
a) Pip Installs Python
b) Pip Installs Packages
c) Preferred Installer Program
d) All of the mentioned

Explanation: pip is a package manager for python. Which is also called Preferred Installer Program.

17. Which of the following is true for variable names in Python?
a) underscore and ampersand are the only two special characters allowed
b) unlimited length
c) all private members must have leading and trailing underscores
d) none of the mentioned

Explanation: Variable names can be of any length.

18. What are the values of the following Python expressions?

```
2**(3**2)
 (2**3)**2
 2**3**2
```

a) 512, 64, 512
b) 512, 512, 512
c) 64, 512, 64
d) 64, 64, 64

Explanation: Expression 1 is evaluated as: 2**9, which is equal to 512. Expression 2 is evaluated as 8**2, which is equal to 64. The last expression is evaluated as 2**(3**2). This is because the associativity of ** operator is from right to left. Hence the result of the third expression is 512.

19. Which of the following is the truncation division operator in Python?
a) |
b) //
c) /
d) %

Explanation: // is the operator for truncation division. It is called so because it returns only the integer part of the quotient, truncating the decimal part. For example: 20//3 = 6.

20. What will be the output of the following Python code?

```
l=[1, 0, 2, 0, 'hello', '', []]
list(filter(bool, l))
```

a) [1, 0, 2, ‘hello’, ”, []]
b) Error
c) [1, 2, ‘hello’]
d) [1, 0, 2, 0, ‘hello’, ”, []]

Explanation: The code shown above returns a new list containing only those elements of the list l which do not amount to zero. Hence the output is: [1, 2, ‘hello’].

21. Which of the following functions is a built-in function in python?
a) factorial()
b) print()
c) seed()
d) sqrt()

Explanation: The function seed is a function which is present in the random module. The functions sqrt and factorial are a part of the math module. The print function is a built-in function which prints a value directly to the system output.

22. Which of the following is the use of id() function in python?
a) Every object doesn’t have a unique id
b) Id returns the identity of the object
c) All of the mentioned
d) None of the mentioned

Explanation: Each object in Python has a unique id. The id() function returns the object’s id.

23. The following python program can work with ____ parameters.

```
  return f1def f(x):
    def f1(*args, **kwargs):
           print("Sanfoundry")
           return x(*args, **kwargs)
    return f1
```

a) any number of
b) 0
c) 1
d) 2

Explanation: The code shown above shows a general decorator which can work with any number of arguments.

24. What will be the output of the following Python function?

```
min(max(False,-3,-4), 2,7)
```

a) -4
b) -3
c) 2
d) False

Explanation: The function max() is being used to find the maximum value from among -3, -4 and false. Since false amounts to the value zero, hence we are left with min(0, 2, 7) Hence the output is 0 (false).

25. Which of the following is not a core data type in Python programming?
a) Tuples
b) Lists
c) Class
d) Dictionary

Explanation: Class is a user-defined data type.

26. What will be the output of the following Python expression if x=56.236?

```
print("%.2f"%x)
```

a) 56.236
b) 56.23
c) 56.0000
d) 56.24

Explanation: The expression shown above rounds off the given number to the number of decimal places specified. Since the expression given specifies rounding off to two decimal places, the output of this expression will be 56.24. Had the value been x=56.234 (last digit being any number less than 5), the output would have been 56.23.

27. Which of these is the definition for packages in Python?
a) A set of main modules
b) A folder of python modules
c) A number of files containing Python definitions and statements
d) A set of programs making use of Python modules

Explanation: A folder of python programs is called as a package of modules.

28. What will be the output of the following Python function?

```
len(["hello",2, 4, 6])
```

a) Error
b) 6
c) 4
d) 3

Explanation: The function len() returns the length of the number of elements in the iterable. Therefore the output of the function shown above is 4.

29. What will be the output of the following Python code?

```
x = 'abcd'
for i in x:
    print(i.upper())
```

a)a
B
C
D

b) a b c d

c) error

d)A
B
C
D

Explanation: The instance of the string returned by upper() is being printed.

30. What is the order of namespaces in which Python looks for an identifier?
a) Python first searches the built-in namespace, then the global namespace and finally the local namespace
b) Python first searches the built-in namespace, then the local namespace and finally the global namespace

c) Python first searches the local namespace, then the global namespace and finally the built-in namespace
d) Python first searches the global namespace, then the local namespace and finally the built-in namespace

Explanation: Python first searches for the local, then the global and finally the built-in namespace.

31. What will be the output of the following Python code snippet?

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

a) 4 3 2 1
b) error
c) 1 2 3 4
d) none of the mentioned

Explanation: [::-1] reverses the list.

32. What will be the output of the following Python statement?

```
1.>>>"a"+"bc"
```

a) bc
b) abc
c) a
d) bca

Explanation: + operator is concatenation operator.

33. Which function is called when the following Python program is executed?

```
f = foo()
format(f)
```

a) str()
b) format()
c) __str__()
d) __format__()

Explanation: Both str(f) and format(f) call f.__str__().

34. Which one of the following is not a keyword in Python language?
a) pass

b) eval
c) assert
d) nonlocal

Explanation: eval can be used as a variable.

35. What will be the output of the following Python code?

```
1.class tester:
2.    def __init__(self, id):
3.        self.id = str(id)
4.        id="224"
5.
6.>>>temp = tester(12)
7.>>>print(temp.id)
```

a) 12
b) 224
c) None
d) Error

Explanation: Id in this case will be the attribute of the instance.

36. What will be the output of the following Python program?

```
def foo(x):
    x[0] = ['def']
    x[1] = ['abc']
    return id(x)
q = ['abc', 'def']
print(id(q) == foo(q))
```

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

Explanation: The same object is modified in the function.

37. Which module in the python standard library parses options received from the command line?
a) getarg
b) getopt
c) main

d) os

Explanation: getopt parses options received from the command line.

38. What will be the output of the following Python program?

```
z=set('abc')
z.add('san')
z.update(set(['p', 'q']))
z
```

a) {‘a’, ‘c’, ‘c’, ‘p’, ‘q’, ‘s’, ‘a’, ‘n’}
b) {‘abc’, ‘p’, ‘q’, ‘san’}
c) {‘a’, ‘b’, ‘c’, ‘p’, ‘q’, ‘san’}
d) {‘a’, ‘b’, ‘c’, [‘p’, ‘q’], ‘san}

Explanation: The code shown first adds the element ‘san’ to the set z. The set z is then updated and two more elements, namely, ‘p’ and ‘q’ are added to it. Hence the output is: {‘a’, ‘b’, ‘c’, ‘p’, ‘q’, ‘san’}

39. What arithmetic operators cannot be used with strings in Python?
a) *
b) –
c) +
d) All of the mentioned

Explanation: + is used to concatenate and * is used to multiply strings.

40. What will be the output of the following Python code?

```
print("abc. DEF".capitalize())
```

a) Abc. def
b) abc. def
c) Abc. Def
d) ABC. DEF

Explanation: The first letter of the string is converted to uppercase and the others are converted to lowercase.

41. Which of the following statements is used to create an empty set in Python?
a) ( )

b) [ ]
c) { }
d) set()

Explanation: { } creates a dictionary not a set. Only set() creates an empty set.

42. What will be the value of 'result' in following Python program?

```
list1 = [1,2,3,4]
list2 = [2,4,5,6]
list3 = [2,6,7,8]
result = list()
result.extend(i for i in list1 if i not in (list2+list3) and i not in result)
result.extend(i for i in list2 if i not in (list1+list3) and i not in result)
result.extend(i for i in list3 if i not in (list1+list2) and i not in result)
```

a) [1, 3, 5, 7, 8]
b) [1, 7, 8]
c) [1, 2, 4, 7, 8]
d) error

Explanation: Here, 'result' is a list which is extending three times. When first time 'extend' function is called for 'result', the inner code generates a generator object, which is further used in 'extend' function. This generator object contains the values which are in 'list1' only (not in 'list2' and 'list3').
Same is happening in second and third call of 'extend' function in these generator object contains values only in 'list2' and 'list3' respectively.
So, 'result' variable will contain elements which are only in one list (not more than 1 list).

43. To add a new element to a list we use which Python command?
a) list1.addEnd(5)
b) list1.addLast(5)
c) list1.append(5)
d) list1.add(5)

Explanation: We use the function append to add an element to the list.

44. What will be the output of the following Python code?

```
print('*', "abcde".center(6), '*', sep='')
```

a) * abcde *
b) *abcde *
c) * abcde*

d) * abcde *

Explanation: Padding is done towards the right-hand-side first when the final string is of even length.

45. What will be the output of the following Python code?

```
1.>>>list1 = [1, 3]
2.>>>list2 = list1
3.>>>list1[0] = 4
4.>>>print(list2)
```

a) [1, 4]
b) [1, 3, 4]
c) [4, 3]
d) [1, 3]

Explanation: Lists should be copied by executing [:] operation.

46. Which one of the following is the use of function in python?
a) Functions don't provide better modularity for your application
b) you can't also create your own functions
c) Functions are reusable pieces of programs
d) All of the mentioned

Explanation: Functions are reusable pieces of programs. They allow you to give a name to a block of statements, allowing you to run that block using the specified name anywhere in your program and any number of times.

47. Which of the following Python statements will result in the output: 6?

```
A = [[1, 2, 3],
     [4, 5, 6],
     [7, 8, 9]]
```

a) A[2][1]
b) A[1][2]
c) A[3][2]
d) A[2][3]

Explanation: The output that is required is 6, that is, row 2, item 3. This position is represented by the statement: A[1][2].

48. What is the maximum possible length of an identifier in Python?
a) 79 characters
b) 31 characters
c) 63 characters
d) none of the mentioned

Explanation: Identifiers can be of any length.

49. What will be the output of the following Python program?

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

a) error
b) 0 1 2 0
c) 0 1 2
d) none of the mentioned

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

50. What will be the output of the following Python code?

```
x = 'abcd'
for i in range(len(x)):
    print(i)
```

a) error
b) 1 2 3 4
c) a b c d
d) 0 1 2 3

Explanation: i takes values 0, 1, 2 and 3.

51. What are the two main types of functions in Python?
a) System function
b) Custom function
c) Built-in function & User defined function
d) User function

Explanation: Built-in functions and user defined ones. The built-in functions are part of the Python language. Examples are: dir(), len() or abs(). The user defined functions are functions created with the def keyword.

52. What will be the output of the following Python program?

```
1.def addItem(listParam):
2.    listParam += [1]
3.
4.mylist = [1, 2, 3, 4]
5.addItem(mylist)
6.print(len(mylist))
```

a) 5
b) 8
c) 2
d) 1

Explanation: + will append the element to the list.

53. Which of the following is a Python tuple?
a) {1, 2, 3}
b) {}
c) [1, 2, 3]
d) (1, 2, 3)

Explanation: Tuples are represented with round brackets.

54. What will be the output of the following Python code snippet?

```
z=set('abc$de')
'a' in z
```

a) Error
b) True
c) False
d) No output

Explanation: The code shown above is used to check whether a particular item is a part of a given set or not. Since ‘a’ is a part of the set z, the output is true. Note that this code would result in an error in the absence of the quotes.

55. What will be the output of the following Python expression?

```
round(4.576)
```

a) 4
b) 4.6
c) 5
d) 4.5

Explanation: This is a built-in function which rounds a number to give precision in decimal digits. In the above case, since the number of decimal places has not been specified, the decimal number is rounded off to a whole number. Hence the output will be 5.

56. Which of the following is a feature of Python DocString?
a) In Python all functions should have a docstring
b) Docstrings can be accessed by the __doc__ attribute on objects
c) It provides a convenient way of associating documentation with Python modules, functions, classes, and methods
d) All of the mentioned

Explanation: Python has a nifty feature called documentation strings, usually referred to by its shorter name docstrings. DocStrings are an important tool that you should make use of since it helps to document the program better and makes it easier to understand.

57. What will be the output of the following Python code?

```
print("Hello {0[0]} and {0[1]}".format(('foo', 'bin')))
```

a) Hello (‘foo’, ‘bin’) and (‘foo’, ‘bin’)
b) Error
c) Hello foo and bin
d) None of the mentioned

Explanation: The elements of the tuple are accessed by their indices.

58. What is output of print(math.pow(3, 2))?
a) 9.0
b) None
c) 9
d) None of the mentioned

Explanation: math.pow() returns a floating point number.

59. Which of the following is the use of id() function in python?
a) Every object in Python doesn’t have a unique id

b) In Python Id function returns the identity of the object
c) None of the mentioned
d) All of the mentioned

Explanation: Each object in Python has a unique id. The id() function returns the object's id.

60. What will be the output of the following Python code?

```
x = [[0], [1]]
print((' '.join(list(map(str, x))),))
```

a) 01
b) [0] [1]
c) ('01')
d) ('[0] [1]',)

Explanation: (element,) is not the same as element. It is a tuple with one item.

61. The process of pickling in Python includes _____
a) conversion of a Python object hierarchy into byte stream
b) conversion of a datatable into a list
c) conversion of a byte stream into Python object hierarchy
d) conversion of a list into a datatable

Explanation: Pickling is the process of serializing a Python object, that is, conversion of a Python object hierarchy into a byte stream. The reverse of this process is known as unpickling.

62. What will be the output of the following Python code?

```
def foo():
    try:
        return 1
    finally:
        return 2
k = foo()
print(k)
```

a) error, there is more than one return statement in a single try-finally block
b) 3
c) 2
d) 1

Explanation: The finally block is executed even there is a return statement in the try block.

Python MCQ's – Variable Names

1. Is Python case sensitive when dealing with identifiers?
a) yes
b) no
c) machine dependent
d) none of the mentioned

Explanation: Case is always significant while dealing with identifiers in python.

2. What is the maximum possible length of an identifier?
a) 31 characters
b) 63 characters
c) 79 characters
d) none of the mentioned

Explanation: Identifiers can be of any length.

3. Which of the following is invalid?
a) _a = 1
b) __a = 1
c) __str__ = 1
d) none of the mentioned

Explanation: All the statements will execute successfully but at the cost of reduced readability.

4. Which of the following is an invalid variable?
a) my_string_1
b) 1st_string
c) foo
d) _

Explanation: Variable names should not start with a number.

5. Why are local variable names beginning with an underscore discouraged?
a) they are used to indicate a private variables of a class
b) they confuse the interpreter
c) they are used to indicate global variables

d) they slow down execution

Explanation: As Python has no concept of private variables, leading underscores are used to indicate variables that must not be accessed from outside the class.

6. Which of the following is not a keyword?
a) eval
b) assert
c) nonlocal
d) pass

Explanation: eval can be used as a variable.

7. All keywords in Python are in _____
a) lower case
b) UPPER CASE
c) Capitalized
d) None of the mentioned

Explanation: True, False and None are capitalized while the others are in lower case.

8. Which of the following is true for variable names in Python?
a) unlimited length
b) all private members must have leading and trailing underscores
c) underscore and ampersand are the only two special characters allowed
d) none of the mentioned

Explanation: Variable names can be of any length.

9. Which of the following is an invalid statement?
a) abc = 1,000,000
b) a b c = 1000 2000 3000
c) a,b,c = 1000, 2000, 3000
d) a_b_c = 1,000,000

Explanation: Spaces are not allowed in variable names.

10. Which of the following cannot be a variable?
a) __init__
b) in
c) it

d) on

Explanation: in is a keyword.

Python MCQ's – Basic Operators

1. Which is the correct operator for power(x^y)?
a) X^y
b) Xy**
c) X^^y
d) None of the mentioned

Explanation: In python, power operator is x**y i.e. 2**3=8.

2. Which one of these is floor division?
a) /
b) //
c) %
d) None of the mentioned

Explanation: When both of the operands are integer then python chops out the fraction part and gives you the round off value, to get the accurate answer use floor division. This is floor division. For ex, 5/2 = 2.5 but both of the operands are integer so answer of this expression in python is 2. To get the 2.5 answer, use floor division.

3. What is the order of precedence in python?
i) Parentheses
ii) Exponential
iii) Multiplication
iv) Division
v) Addition
vi) Subtraction

a) i,ii,iii,iv,v,vi
b) ii,i,iii,iv,v,vi
c) ii,i,iv,iii,v,vi
d) i,ii,iii,iv,vi,v

Explanation: For order of precedence, just remember this PEMDAS (similar to BODMAS).

4. What is the answer to this expression, 22 % 3 is?
a) 7

b) 1
c) 0
d) 5

Explanation: Modulus operator gives the remainder. So, 22%3 gives the remainder, that is, 1.

5. Mathematical operations can be performed on a string.
a) True
b) False

Explanation: You can't perform mathematical operation on string even if the string is in the form: '1234…'.

6. Operators with the same precedence are evaluated in which manner?
a) Left to Right
b) Right to Left
c) Can't say
d) None of the mentioned

Explanation: None.

7. What is the output of this expression, 3*13?**
a) 27
b) 9
c) 3
d) 1

Explanation: First this expression will solve 1**3 because exponential has higher precedence than multiplication, so 1**3 = 1 and 3*1 = 3. Final answer is 3.

8. Which one of the following has the same precedence level?
a) Addition and Subtraction
b) Multiplication, Division and Addition
c) Multiplication, Division, Addition and Subtraction
d) Addition and Multiplication

Explanation: "Addition and Subtraction" are at the same precedence level. Similarly, "Multiplication and Division" are at the same precedence level. However, Multiplication and Division operators are at a higher precedence level than Addition and Subtraction operators.

9. The expression Int(x) implies that the variable x is converted to integer.
a) True
b) False

Explanation: Int(x) converts the datatype of the variable to integer and is the example of explicit data conversion.

10. Which one of the following has the highest precedence in the expression?
a) Exponential
b) Addition
c) Multiplication
d) Parentheses

Explanation: Just remember: PEMDAS, that is, Parenthesis, Exponentiation, Division, Multiplication, Addition, Subtraction. Note that the precedence order of Division and Multiplication is the same. Likewise, the order of Addition and Subtraction is also the same.

Python MCQ's – Core Data types

1. Which of these in not a core data type?
a) Lists
b) Dictionary
c) Tuples
d) Class

Explanation: Class is a user defined data type.

2. Given a function that does not return any value, What value is thrown by default when executed in shell.
a) int
b) bool
c) void
d) None

Explanation: Python shell throws a NoneType object back.

3. What will be the output of the following Python code?

```
1.>>>str="hello"
2.>>>str[:2]
3.>>>
```

a) he
b) lo

c) olleh
d) hello

Explanation: We are printing only the 1st two bytes of string and hence the answer is “he”.

4. Which of the following will run without errors?
a) round(45.8)
b) round(6352.898,2,5)
c) round()
d) round(7463.123,2,1)

Explanation: Execute help(round) in the shell to get details of the parameters that are passed into the round function.

5. What is the return type of function id?
a) int
b) float
c) bool
d) dict

Explanation: Execute help(id) to find out details in python shell.id returns a integer value that is unique.

6. In python we do not specify types, it is directly interpreted by the compiler, so consider the following operation to be performed.

```
1.>>>x = 13 ? 2
```

objective is to make sure x has a integer value, select all that apply (python 3.xx)
a) x = 13 // 2
b) x = int(13 / 2)
c) x = 13 % 2
d) All of the mentioned

Explanation: // is integer operation in python 3.0 and int(..) is a type cast operator.

7. What error occurs when you execute the following Python code snippet?

```
apple = mango
```

a) SyntaxError
b) NameError
c) ValueError
d) TypeError

Explanation: Mango is not defined hence name error.

8. What will be the output of the following Python code snippet?

```
1.def example(a):
2.     a = a + '2'
3.      a = a*2
4.     return a
5.>>>example("hello")
```

a) indentation Error
b) cannot perform mathematical operation on strings
c) hello2
d) hello2hello2

Explanation: Python codes have to be indented properly.

9. What data type is the object below?

L = [1, 23, 'hello', 1]

a) list
b) dictionary
c) array
d) tuple

Explanation: List data type can store any values within it.

10. In order to store values in terms of key and value we use what core data type.
a) list
b) tuple
c) class
d) dictionary

Explanation: Dictionary stores values in terms of keys and values.

11. Which of the following results in a SyntaxError?
a) '"Once upon a time…", she said.'
b) "He said, 'Yes!'"
c) '3\'
d) "'That's okay"'

Explanation: Carefully look at the colons.

12. The following is displayed by a print function call. Select all of the function calls that result in this output.

```
1.tom
2.dick
3.harry
```

a) print('"tom
\ndick
\nharry"')

b) print("'tomdickharry"')

c) print('tom\ndick\nharry')

d) print('tom
dick
harry')

Explanation: The \n adds a new line.

13. What is the average value of the following Python code snippet?

```
1.>>>grade1 = 80
2.>>>grade2 = 90
3.>>>average = (grade1 + grade2) / 2
```

a) 85.0
b) 85.1
c) 95.0
d) 95.1

Explanation: Cause a decimal value of 0 to appear as output.

14. Select all options that print.

hello-how-are-you

a) print(‘hello’, ‘how’, ‘are’, ‘you’)
b) print(‘hello’, ‘how’, ‘are’, ‘you’ + ‘-‘ * 4)
c) print(‘hello-‘ + ‘how-are-you’)
d) print(‘hello’ + ‘-‘ + ‘how’ + ‘-‘ + ‘are’ + ‘you’)

Explanation: Execute in the shell.

15. What is the return value of trunc()?
a) int
b) bool
c) float
d) None

Explanation: Execute help(math.trunc) to get details.

Python MCQ’s – Numeric Types

1. What is the output of print 0.1 + 0.2 == 0.3?
a) True
b) False
c) Machine dependent
d) Error

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.

2. Which of the following is not a complex number?
a) k = 2 + 3j
b) k = complex(2, 3)
c) k = 2 + 3l
d) k = 2 + 3J

Explanation: l (or L) stands for long.

3. What is the type of inf?
a) Boolean

b) Integer
c) Float
d) Complex

Explanation: Infinity is a special case of floating point numbers. It can be obtained by float('inf').

4. What does ~4 evaluate to?
a) -5
b) -4
c) -3
d) +3

Explanation: ~x is equivalent to -(x+1).

5. What does ~~~~~~5 evaluate to?
a) +5
b) -11
c) +11
d) -5

Explanation: ~x is equivalent to -(x+1).
~~x = – (-(x+1) + 1) = (x+1) – 1 = x
~~x is equivalent to x
Extrapolating further ~~~~~~x would be same as x in the final result.
In the question, x value is given as 5 and "~" is repeated 6 times. So, the correct answer for "~~~~~~5" is 5.

6. Which of the following is incorrect?
a) x = 0b101
b) x = 0x4f5
c) x = 19023
d) x = 03964

Explanation: Numbers starting with a 0 are octal numbers but 9 isn't allowed in octal numbers.

7. What is the result of cmp(3, 1)?
a) 1
b) 0
c) True
d) False

Explanation: cmp(x, y) returns 1 if x > y, 0 if x == y and -1 if x < y.

8. Which of the following is incorrect?
a) float('inf')
b) float('nan')
c) float('56'+'78')
d) float('12+34')

Explanation: '+' cannot be converted to a float.

9. What is the result of round(0.5) – round(-0.5)?
a) 1.0
b) 2.0
c) 0.0
d) Value depends on Python version

Explanation: The behavior of the **round()** function is different in Python 2 and Python 3. In Python 2, it rounds off numbers away from 0 when the number to be rounded off is exactly halfway through. round(0.5) is 1 and round(-0.5) is -1 whereas in Python 3, it rounds off numbers towards nearest even number when the number to be rounded off is exactly halfway through. See the below output.

Here's the runtime output for Python version 2.7 interpreter.

```
$ python
Python 2.7.17 (default, Nov  7 2019, 10:07:09)
>>> round(0.5)
1.0
>>> round(-0.5)
-1.0
>>>
```

In the above output, you can see that the round() functions on 0.5 and -0.5 are moving away from 0 and hence "**round(0.5) – (round(-0.5)) = 1 – (-1) = 2**"
Here's the runtime output for Python version 3.6 interpreter.

```
$ python3
Python 3.6.8 (default, Oct  7 2019, 12:59:55)
>>> round(0.5)
0
>>> round(-0.5)
0
>>> round(2.5)
2
>>> round(3.5)
4
>>>
```

In the above output, you can see that the round() functions on 0.5 and -0.5 are moving towards 0 and hence "**round(0.5) – (round(-0.5)) = 0 – 0 = 0**". Also note that the round(2.5) is 2 (which is an even number) whereas round(3.5) is 4 (which is an even number).

10. What does 3 ^ 4 evaluate to?
a) 81
b) 12
c) 0.75
d) 7

Explanation: ^ is the Binary XOR operator.

Python MCQ's – Precedence and Associativity – 1

1. The value of the expressions 4/(3*(2-1)) and 4/3*(2-1) is the same.
a) True
b) False

Explanation: Although the presence of parenthesis does affect the order of precedence, in the case shown above, it is not making a difference. The result of both of these expressions is 1.333333333. Hence the statement is true.

2. What will be the value of the following Python expression?

```
4 + 3 % 5
```

a) 4
b) 7
c) 2
d) 0

Explanation: The order of precedence is: %, +. Hence the expression above, on simplification results in 4 + 3 = 7. Hence the result is 7.

3. Evaluate the expression given below if A = 16 and B = 15.

```
A % B // A
```

a) 0.0
b) 0

c) 1.0
d) 1

Explanation: The above expression is evaluated as 16%15//16, which is equal to 1//16, which results in 0.

4. Which of the following operators has its associativity from right to left?
a) +
b) //
c) %
d) **

Explanation: All of the operators shown above have associativity from left to right, except exponentiation operator (**) which has its associativity from right to left.

5. What will be the value of x in the following Python expression?

```
x = int(43.55+2/2)
```

a) 43
b) 44
c) 22
d) 23

Explanation: The expression shown above is an example of explicit conversion. It is evaluated as int(43.55+1) = int(44.55) = 44. Hence the result of this expression is 44.

6. What is the value of the following expression?

```
2+4.00, 2**4.0
```

a) (6.0, 16.0)
b) (6.00, 16.00)
c) (6, 16)
d) (6.00, 16.0)

Explanation: The result of the expression shown above is (6.0, 16.0). This is because the result is automatically rounded off to one decimal place.

7. Which of the following is the truncation division operator?
a) /
b) %
c) //

d) |

Explanation: // is the operator for truncation division. It is called so because it returns only the integer part of the quotient, truncating the decimal part. For example: 20//3 = 6.

8. What are the values of the following Python expressions?

```
2**(3**2)
(2**3)**2
2**3**2
```

a) 64, 512, 64
b) 64, 64, 64
c) 512, 512, 512
d) 512, 64, 512

Explanation: Expression 1 is evaluated as 2**9, which is equal to 512. Expression 2 is evaluated as 8**2, which is equal to 64. The last expression is evaluated as 2**(3**2). This is because the associativity of ** operator is from right to left. Hence the result of the third expression is 512.

9. What is the value of the following expression?

```
8/4/2, 8/(4/2)
```

a) (1.0, 4.0)
b) (1.0, 1.0)
c) (4.0. 1.0)
d) (4.0, 4.0)

Explanation: The above expressions are evaluated as: 2/2, 8/2, which is equal to (1.0, 4.0).

10. What is the value of the following expression?

```
float(22//3+3/3)
```

a) 8
b) 8.0
c) 8.3
d) 8.33

Explanation: The expression shown above is evaluated as: float(7+1) = float(8) = 8.0. Hence the result of this expression is 8.0.

Python MCQ's – Precedence and Associativity – 2

1. What will be the output of the following Python expression?

```
print(4.00/(2.0+2.0))
```

a) Error
b) 1.0
c) 1.00
d) 1

Explanation: The result of the expression shown above is 1.0 because print rounds off digits.

2. What will be the value of X in the following Python expression?

```
X = 2+9*((3*12)-8)/10
```

a) 30.0
b) 30.8
c) 28.4
d) 27.2

Explanation: The expression shown above is evaluated as: 2+9*(36-8)/10, which simplifies to give 2+9*(2.8), which is equal to 2+25.2 = 27.2. Hence the result of this expression is 27.2.

3. Which of the following expressions involves coercion when evaluated in Python?
a) 4.7 – 1.5
b) 7.9 * 6.3
c) 1.7 % 2
d) 3.4 + 4.6

Explanation: Coercion is the implicit (automatic) conversion of operands to a common type. Coercion is automatically performed on mixed-type expressions. The expression 1.7 % 2 is evaluated as 1.7 % 2.0 (that is, automatic conversion of int to float).

4. What will be the output of the following Python expression?

```
24//6%3, 24//4//2
```

a) (1,3)
b) (0,3)
c) (1,0)

d) (3,1)

Explanation: The expressions are evaluated as: 4%3 and 6//2 respectively. This results in the answer (1,3). This is because the associativity of both of the expressions shown above is left to right.

5. Which among the following list of operators has the highest precedence?

```
+, -, **, %, /, <<, >>, |
```

a) <<, >>
b) **
c) |
d) %

Explanation: The highest precedence is that of the exponentiation operator, that is of **.

6. What will be the value of the following Python expression?

```
float(4+int(2.39)%2)
```

a) 5.0
b) 5
c) 4.0
d) 4

Explanation: The above expression is an example of explicit conversion. It is evaluated as: float(4+int(2.39)%2) = float(4+2%2) = float(4+0) = 4.0. Hence the result of this expression is 4.0.

7. Which of the following expressions is an example of type conversion?
a) 4.0 + float(3)
b) 5.3 + 6.3
c) 5.0 + 3
d) 3 + 7

Explanation: Type conversion is nothing but explicit conversion of operands to a specific type. Options 5.3 + 6.3 and 5.0 + 3 are examples of implicit conversion whereas option 4.0 + float(3) is an example of explicit conversion or type conversion.

8. Which of the following expressions results in an error?
a) float(‘10’)
b) int(‘10’)
c) float(’10.8’)

d) int('10.8')

Explanation: All of the above examples show explicit conversion. However the expression int('10.8') results in an error.

9. What will be the value of the following Python expression?

```
4+2**5//10
```

a) 3
b) 7
c) 77
d) 0

Explanation: The order of precedence is: **, //, +. The expression 4+2**5//10 is evaluated as 4+32//10, which is equal to 4+3 = 7. Hence the result of the expression shown above is 7.

10. The expression 22**3 is evaluates as: (2**2)**3.**
a) True
b) False

Explanation: The value of the expression (2**2)**3 = 4**3 = 64. When the expression 2**2**3 is evaluated in python, we get the result as 256, because this expression is evaluated as 2**(2**3). This is because the associativity of exponentiation operator (**) is from right to left and not from left to right.

Python MCQ's – Bitwise – 1

1. What will be the output of the following Python code snippet if x=1?

```
x<<2
```

a) 8
b) 1
c) 2
d) 4

Explanation: The binary form of 1 is 0001. The expression x<<2 implies we are performing bitwise left shift on x. This shift yields the value: 0100, which is the binary form of the number 4.

2. What will be the output of the following Python expression?

```
bin(29)
```

a) ‘0b10111’
b) ‘0b11101’
c) ‘0b11111’
d) ‘0b11011’

Explanation: The binary form of the number 29 is 11101. Hence the output of this expression is ‘0b11101’.

3. What will be the value of x in the following Python expression, if the result of that expression is 2?

```
x>>2
```

a) 8
b) 4
c) 2
d) 1

Explanation: When the value of x is equal to 8 (1000), then x>>2 (bitwise right shift) yields the value 0010, which is equal to 2. Hence the value of x is 8.

4. What will be the output of the following Python expression?

```
int(1011)?
```

a) 1011
b) 11
c) 13
d) 1101

Explanation: The result of the expression shown will be 1011. This is because we have not specified the base in this expression. Hence it automatically takes the base as 10.

5. To find the decimal value of 1111, that is 15, we can use the function:
a) int(1111,10)
b) int(‘1111’,10)
c) int(1111,2)
d) int(‘1111’,2)

Explanation: The expression int(‘1111’,2) gives the result 15. The expression int(‘1111’, 10) will give the result 1111.

6. What will be the output of the following Python expression if x=15 and y=12?

```
x & y
```

a) b1101
b) 0b1101
c) 12
d) 1101

Explanation: The symbol '&' represents bitwise AND. This gives 1 if both the bits are equal to 1, else it gives 0. The binary form of 15 is 1111 and that of 12 is 1100. Hence on performing the bitwise AND operation, we get 1100, which is equal to 12.

7. Which of the following expressions results in an error?
a) int(1011)
b) int('1011',23)
c) int(1011,2)
d) int('1011')

Explanation: The expression int(1011,2) results in an error. Had we written this expression as int('1011',2), then there would not be an error.

8. Which of the following represents the bitwise XOR operator?
a) &
b) ^
c) |
d) !

Explanation: The ^ operator represent bitwise XOR operation. &: bitwise AND, | : bitwise OR and ! represents bitwise NOT.

9. What is the value of the following Python expression?

```
bin(0x8)
```

a) '0bx1000'
b) 8
c) 1000
d) '0b1000'

Explanation: The prefix 0x specifies that the value is hexadecimal in nature. When we convert this hexadecimal value to binary form, we get the result as: '0b1000'.

10. What will be the output of the following Python expression?

```
0x35 | 0x75
```

a) 115
b) 116
c) 117
d) 118

Explanation: The binary value of 0x35 is 110101 and that of 0x75 is 1110101. On OR-ing these two values we get the output as: 1110101, which is equal to 117. Hence the result of the above expression is 117.

Python MCQ's – Bitwise – 2

1. It is not possible for the two's complement value to be equal to the original value in any case.
a) True
b) False

Explanation: In most cases the value of two's complement is different from the original value. However, there are cases in which the two's complement value may be equal to the original value. For example, the two's complement of 10000000 is also equal to 10000000. Hence the statement is false.

2. The one's complement of 110010101 is:
a) 001101010
b) 110010101
c) 001101011
d) 110010100

Explanation: The one's complement of a value is obtained by simply changing all the 1's to 0's and all the 0's to 1's. Hence the one's complement of 110010101 is 001101010.

3. Bitwise _____ gives 1 if either of the bits is 1 and 0 when both of the bits are 1.
a) OR
b) AND
c) XOR
d) NOT

Explanation: Bitwise XOR gives 1 if either of the bits is 1 and 0 when both of the bits are 1.

4. What will be the output of the following Python expression?

```
4^12
```

a) 2
b) 4
c) 8
d) 12

Explanation: ^ is the XOR operator. The binary form of 4 is 0100 and that of 12 is 1100. Therefore, 0100^1100 is 1000, which is equal to 8.

5. Any odd number on being AND-ed with _____ always gives 1. Hint: Any even number on being AND-ed with this value always gives 0.
a) 10
b) 2
c) 1
d) 0

Explanation: Any odd number on being AND-ed with 1 always gives 1. Any even number on being AND-ed with this value always gives 0.

6. What will be the value of the following Python expression?

```
bin(10-2)+bin(12^4)
```

a) 0b10000
b) 0b10001000
c) 0b1000b1000
d) 0b10000b1000

Explanation: The output of bin(10-2) = 0b1000 and that of bin(12^4) is ob1000. Hence the output of the above expression is: 0b10000b1000.

7. Which of the following expressions can be used to multiply a given number ‘a’ by 4?
a) a<<2
b) a<<4
c) a>>2
d) a>>4

Explanation: Let us consider an example wherein a=2. The binary form of 2 is 0010. When we left shift this value by 2, we get 1000, the value of which is 8. Hence if we want to multiply a given number ‘a’ by 4, we can use the expression: a<<2.

8. What will be the output of the following Python code if a=10 and b =20?

```
a=10
b=20
a=a^b
b=a^b
a=a^b
print(a,b)
```

a) 10 20
b) 10 10
c) 20 10
d) 20 20

Explanation: The code shown above is used to swap the contents of two memory locations using bitwise X0R operator. Hence the output of the code shown above is: 20 10.

9. What is the two’s complement of -44?
a) 1011011
b) 11010100
c) 11101011
d) 10110011

Explanation: The binary form of -44 is 00101100. The one’s complement of this value is 11010011. On adding one to this we get: 11010100 (two’s complement).

10. What will be the output of the following Python expression?

```
~100?
```

a) 101
b) -101
c) 100
d) -100

Explanation: Suppose we have an expression ~A. This is evaluated as: -A – 1. Therefore, the expression ~100 is evaluated as -100 – 1, which is equal to -101.

Python MCQ's– Boolean

1. What will be the output of the following Python code snippet?

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

a) True
   True

b)False
  True

c)False
  False

d)True
   False

Explanation: The Boolean function returns true if the argument passed to the bool function does not amount to zero. In the first example, the string 'False' is passed to the function bool. This does not amount to zero and hence the output is true. In the second function, an empty list is passed to the function bool. Hence the output is false.

2. What will be the output of the following Python code snippet?

```
['hello', 'morning'][bool('')]
```

a) error
b) no output
c) hello
d) morning

Explanation: The line of code shown above can be simplified to state that 'hello' should be printed if the argument passed to the Boolean function amounts to zero, else 'morning' will be printed.

3. What will be the output of the following Python code snippet?

```
not(3>4)
not(1&1)
```

a)True

True

b)True
False

c)False
True

d)False
False

Explanation: The function not returns true if the argument amounts to false, and false if the argument amounts to true. Hence the first function returns false, and the second function returns false.

4. What will be the output of the following Python code?

```
['f', 't'][bool('spam')]
```

a) t
b) f
c) No output
d) Error

Explanation: The line of code can be translated to state that ‘f’ is printed if the argument passed to the Boolean function amount to zero. Else ‘t’ is printed. The argument given to the Boolean function in the above case is ‘spam’, which does not amount to zero. Hence the output is t.

5. What will be the output of the following Python code?

```
l=[1, 0, 2, 0, 'hello', '', []]
list(filter(bool, l))
```

a) Error
b) [1, 0, 2, 0, ‘hello’, ‘’, []]
c) [1, 0, 2, ‘hello’, ‘’, []]
d) [1, 2, ‘hello’]

Explanation: The code shown above returns a new list containing only those elements of the list l which do not amount to zero. Hence the output is: [1, 2, ‘hello’].

6. What will be the output of the following Python code if the system date is 21st June, 2017 (Wednesday)?

```
[] or {}
{} or []
```

a)[]
  {}

b)[]
  []

c){}
  []

d){}
  {}

Explanation: The code shown above shows two functions. In both the cases the right operand is returned. This is because each function is evaluated from left to right. Since the left operand is false, it is assumed that the right operand must be true and hence the right operand is returned in each of the above case.

7. What will be the output of the following Python code?

```
class Truth:
        pass
x=Truth()
bool(x)
```

a) pass
b) true
c) false
d) error

Explanation: If the truth method is not defined, the object is considered true. Hence the output of the code shown above is true.

8. What will be the output of the following Python code?

```
if (9 < 0) and (0 < -9):
    print("hello")
```

```
elif (9 > 0) or False:
    print("good")
else:
    print("bad")
```

a) error
b) hello
c) good
d) bad

Explanation: The code shown above prints the appropriate option depending on the conditions given. The condition which matches is (9>0), and hence the output is: good.

9. Which of the following Boolean expressions is not logically equivalent to the other three?
a) not(-6<0 or-6>10)
b) -6>=0 and -6<=10
c) not(-6<10 or-6==10)
d) not(-6>10 or-6==10)

Explanation: The expression not(-6<0 or -6>10) returns the output False.
The expression -6>=0 and -6<=10 returns the output False.
The expression not(-6<10 or -6==10) returns the output False.
The expression not(-6>10 or -6==10) returns the output True.

10. What will be the output of the following Python code snippet?

```
not(10<20) and not(10>30)
```

a) True
b) False
c) Error
d) No output

Explanation: The expression not(10<20) returns false. The expression not(10>30) returns true. The and operation between false and true returns false. Hence the output is false.

Python MCQ's – While and For Loops – 1

1. What will be the output of the following Python code?

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

```
print(x)
```

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

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

2. What will be the output of the following Python code?

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

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

Explanation: The loop does not terminate as new elements are being added to the list in each iteration.

3. What will be the output of the following Python code?

```
i = 1
while True:
    if i%3 == 0:
        break
    print(i)

    i + = 1
```

a) 1 2
b) 1 2 3
c) error
d) none of the mentioned

Explanation: SyntaxError, there shouldn’t be a space between + and = in +=.

4. What will be the output of the following Python code?

```
i = 1
while True:
    if i%0O7 == 0:
        break
    print(i)
    i += 1
```

a) 1 2 3 4 5 6
b) 1 2 3 4 5 6 7
c) error
d) none of the mentioned

Explanation: Control exits the loop when i becomes 7.

5. What will be the output of the following Python code?

```
i = 5
while True:
    if i%0O11 == 0:
        break
    print(i)
    i += 1
```

a) 5 6 7 8 9 10
b) 5 6 7 8
c) 5 6
d) error

Explanation: 0O11 is an octal number.

6. What will be the output of the following Python code?

```
i = 5
while True:
    if i%0O9 == 0:
        break
    print(i)
    i += 1
```

a) 5 6 7 8
b) 5 6 7 8 9
c) 5 6 7 8 9 10 11 12 13 14 15 ….
d) error

Explanation: 9 isn’t allowed in an octal number.

7. What will be the output of the following Python code?

```
i = 1
while True:
    if i%2 == 0:
        break
    print(i)
    i += 2
```

a) 1
b) 1 2
c) 1 2 3 4 5 6 …
d) 1 3 5 7 9 11 …

Explanation: The loop does not terminate since i is never an even number.

8. What will be the output of the following Python code?

```
i = 2
while True:
    if i%3 == 0:
        break
    print(i)
    i += 2
```

a) 2 4 6 8 10 …
b) 2 4
c) 2 3
d) error

Explanation: The numbers 2 and 4 are printed. The next value of i is 6 which is divisible by 3 and hence control exits the loop.

9. What will be the output of the following Python code?

```
i = 1
while False:
    if i%2 == 0:
        break
    print(i)
    i += 2
```

a) 1
b) 1 3 5 7 …
c) 1 2 3 4 …
d) none of the mentioned

Explanation: Control does not enter the loop because of False.

10. What will be the output of the following Python code?

```
True = False
while True:
    print(True)
    break
```

a) True
b) False
c) None
d) none of the mentioned

Explanation: SyntaxError, True is a keyword and it's value cannot be changed.

Python MCQ's – While and For Loops – 2

1. What will be the output of the following Python code?

```
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 mentioned

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

2. What will be the output of the following Python code?

```
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

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

3. What will be the output of the following Python code?

```
x = "abcdef"
while i in x:
    print(i, end=" ")
```

a) a b c d e f
b) abcdef
c) i i i i i i …
d) error

Explanation: NameError, i is not defined.

4. What will be the output of the following Python code?

```
x = "abcdef"
i = "i"
while i in x:
    print(i, end=" ")
```

a) no output
b) i i i i i i …
c) a b c d e f
d) abcdef

Explanation: “i” is not in “abcdef”.

5. What will be the output of the following Python code?

```
x = "abcdef"
i = "a"
while i in x:
    print(i, end = " ")
```

a) no output
b) i i i i i i …
c) a a a a a a …
d) a b c d e f

Explanation: As the value of i or x isn’t changing, the condition will always evaluate to True.

6. What will be the output of the following Python code?

```
x = "abcdef"
i = "a"
while i in x:
    print('i', end = " ")
```

a) no output
b) i i i i i i …
c) a a a a a a …
d) a b c d e f

Explanation: Here i i i i i … printed continuously because as the value of i or x isn't changing, the condition will always evaluate to True. But also here we use a citation marks on "i", so, here i treated as a string, not like a variable.

7. What will be the output of the following Python code?

```
x = "abcdef"
i = "a"
while i in x:
    x = x[:-1]
    print(i, end = " ")
```

a) i i i i i i
b) a a a a a a
c) a a a a a
d) none of the mentioned

Explanation: The string x is being shortened by one character in each iteration.

8. What will be the output of the following Python code?

```
x = "abcdef"
i = "a"
while i in x[:-1]:
    print(i, end = " ")
```

a) a a a a a
b) a a a a a a
c) a a a a a a …
d) a

Explanation: String x is not being altered and i is in x[:-1].

9. What will be the output of the following Python code?

```
x = "abcdef"
i = "a"
while i in x:
    x = x[1:]
    print(i, end = " ")
```

a) a a a a a a
b) a
c) no output
d) error

Explanation: The string x is being shortened by one character in each iteration.

10. What will be the output of the following Python code?

```
x = "abcdef"
i = "a"
while i in x[1:]:
    print(i, end = " ")
```

a) a a a a a a
b) a
c) no output
d) error

Explanation: i is not in x[1:].

Python MCQ's – While and For Loops – 3

1. What will be the output of the following Python code?

```
x = 'abcd'
for i in x:
    print(i)
    x.upper()
```

a) a B C D
b) a b c d
c) A B C D
d) error

Explanation: Changes do not happen in-place, rather a new instance of the string is returned.

2. What will be the output of the following Python code?

```
x = 'abcd'
for i in x:
    print(i.upper())
```

a) a b c d
b) A B C D
c) a B C D
d) error

Explanation: The instance of the string returned by upper() is being printed.

3. What will be the output of the following Python code?

```
x = 'abcd'
for i in range(x):
    print(i)
```

a) a b c d
b) 0 1 2 3
c) error
d) none of the mentioned

Explanation: range(str) is not allowed.

4. What will be the output of the following Python code?

```
x = 'abcd'
for i in range(len(x)):
    print(i)
```

a) a b c d
b) 0 1 2 3
c) error
d) 1 2 3 4

Explanation: i takes values 0, 1, 2 and 3.

5. What will be the output of the following Python code?

```
x = 'abcd'
for i in range(len(x)):
    print(i.upper())
```

a) a b c d
b) 0 1 2 3
c) error
d) 1 2 3 4

Explanation: Objects of type int have no attribute upper().

6. What will be the output of the following Python code snippet?

```
x = 'abcd'
for i in range(len(x)):
    i.upper()
print (x)
```

a) a b c d
b) 0 1 2 3
c) error
d) none of the mentioned

Explanation: Objects of type int have no attribute upper().

7. What will be the output of the following Python code snippet?

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

a) abcd
b) ABCD
c) error
d) none of the mentioned

Explanation: Changes do not happen in-place, rather a new instance of the string is returned.

8. What will be the output of the following Python code snippet?

```
x = 'abcd'
for i in range(len(x)):
```

```
    i[x].upper()
print (x)
```

a) abcd
b) ABCD
c) error
d) none of the mentioned

Explanation: Objects of type int aren't subscriptable. However, if the statement was x[i], an error would not have been thrown.

9. What will be the output of the following Python code snippet?

```
x = 'abcd'
for i in range(len(x)):
    x = 'a'
    print(x)
```

a) a
b) abcd abcd abcd
c) a a a a
d) none of the mentioned

Explanation: range() is computed only at the time of entering the loop.

10. What will be the output of the following Python code snippet?

```
x = 'abcd'
for i in range(len(x)):
    print(x)
    x = 'a'
```

a) a
b) abcd abcd abcd abcd
c) a a a a
d) none of the mentioned

Explanation: abcd a a a is the output as x is modified only after 'abcd' has been printed once.

Python MCQ's – While and For Loops – 4

1. What will be the output of the following Python code?

```
x = 123
for i in x:
    print(i)
```

a) 1 2 3
b) 123
c) error
d) none of the mentioned

Explanation: Objects of type int are not iterable.

2. What will be the output of the following Python code?

```
d = {0: 'a', 1: 'b', 2: 'c'}
for i in d:
    print(i)
```

a) 0 1 2
b) a b c
c) 0 a 1 b 2 c
d) none of the mentioned

Explanation: Loops over the keys of the dictionary.

3. What will be the output of the following Python code?

```
d = {0: 'a', 1: 'b', 2: 'c'}
for x, y in d:
    print(x, y)
```

a) 0 1 2
b) a b c
c) 0 a 1 b 2 c
d) none of the mentioned

Explanation: Error, objects of type int aren’t iterable.

4. What will be the output of the following Python code?

```
d = {0: 'a', 1: 'b', 2: 'c'}
for x, y in d.items():
    print(x, y)
```

a) 0 1 2
b) a b c
c) 0 a 1 b 2 c
d) none of the mentioned

Explanation: Loops over key, value pairs.

5. What will be the output of the following Python code?

```
d = {0: 'a', 1: 'b', 2: 'c'}
for x in d.keys():
    print(d[x])
```

a) 0 1 2
b) a b c
c) 0 a 1 b 2 c
d) none of the mentioned

Explanation: Loops over the keys and prints the values.

6. What will be the output of the following Python code?

```
d = {0: 'a', 1: 'b', 2: 'c'}
for x in d.values():
    print(x)
```

a) 0 1 2
b) a b c
c) 0 a 1 b 2 c
d) none of the mentioned

Explanation: Loops over the values.

7. What will be the output of the following Python code?

```
d = {0: 'a', 1: 'b', 2: 'c'}
for x in d.values():
    print(d[x])
```

a) 0 1 2
b) a b c
c) 0 a 1 b 2 c
d) none of the mentioned

Explanation: Causes a KeyError.

8. What will be the output of the following Python code?

```
d = {0, 1, 2}
for x in d.values():
    print(x)
```

a) 0 1 2
b) None None None
c) error
d) none of the mentioned

Explanation: Objects of type set have no attribute values.

9. What will be the output of the following Python code?

```
d = {0, 1, 2}
for x in d:
    print(x)
```

a) 0 1 2
b) {0, 1, 2} {0, 1, 2} {0, 1, 2}
c) error
d) none of the mentioned

Explanation: Loops over the elements of the set and prints them.

10. What will be the output of the following Python code?

```
d = {0, 1, 2}
for x in d:
    print(d.add(x))
```

a) 0 1 2
b) 0 1 2 0 1 2 0 1 2 …
c) None None None
d) None of the mentioned

Explanation: Variable x takes the values 0, 1 and 2. set.add() returns None which is printed.

11. What will be the output of the following Python code?

```
for i in range(0):
    print(i)
```

a) 0
b) no output
c) error
d) none of the mentioned

Explanation: range(0) is empty.

Python MCQ's – While and For Loops – 5

1. What will be the output of the following Python code?

```
for i in range(2.0):
    print(i)
```

a) 0.0 1.0
b) 0 1
c) error
d) none of the mentioned

Explanation: Object of type float cannot be interpreted as an integer.

2. What will be the output of the following Python code?

```
for i in range(int(2.0)):
    print(i)
```

a) 0.0 1.0
b) 0 1
c) error
d) none of the mentioned

Explanation: range(int(2.0)) is the same as range(2).

3. What will be the output of the following Python code?

```
for i in range(float('inf')):
    print (i)
```

a) 0.0 0.1 0.2 0.3 …
b) 0 1 2 3 …
c) 0.0 1.0 2.0 3.0 …
d) none of the mentioned

Explanation: Error, objects of type float cannot be interpreted as an integer.

4. What will be the output of the following Python code?

```
for i in range(int(float('inf'))):
    print (i)
```

a) 0.0 0.1 0.2 0.3 …
b) 0 1 2 3 …
c) 0.0 1.0 2.0 3.0 …
d) none of the mentioned

Explanation: OverflowError, cannot convert float infinity to integer.

5. What will be the output of the following Python code snippet?

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

a) 1 2 3 4
b) 4 3 2 1
c) error
d) none of the mentioned

Explanation: [::-1] reverses the list.

6. What will be the output of the following Python code snippet?

```
for i in ''.join(reversed(list('abcd'))):
    print (i)
```

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

Explanation: ‘ ‘.join(reversed(list(‘abcd’))) reverses a string.

7. What will be the output of the following Python code snippet?

```
for i in 'abcd'[::-1]:
    print (i)
```

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

Explanation: [::-1] reverses the string.

8. What will be the output of the following Python code snippet?

```
for i in '':
    print (i)
```

a) None
b) (nothing is printed)
c) error
d) none of the mentioned

Explanation: The string does not have any character to loop over.

9. What will be the output of the following Python code snippet?

```
x = 2
for i in range(x):
    x += 1
    print (x)
```

a) 0 1 2 3 4 …
b) 0 1
c) 3 4
d) 0 1 2 3

Explanation: Variable x is incremented and printed twice.

10. What will be the output of the following Python code snippet?

```
x = 2
for i in range(x):
    x -= 2
    print (x)
```

a) 0 1 2 3 4 …
b) 0 -2
c) 0
d) error

Explanation: The loop is entered twice.

Python MCQ's – While and For Loops – 6

1. What will be the output of the following Python code?

```
for i in range(10):
    if i == 5:
        break
    else:
        print(i)
else:
    print("Here")
```

a) 0 1 2 3 4 Here
b) 0 1 2 3 4 5 Here
c) 0 1 2 3 4
d) 1 2 3 4 5

Explanation: The else part is executed if control doesn't break out of the loop.

2. What will be the output of the following Python code?

```
for i in range(5):
    if i == 5:
        break
    else:
        print(i)
else:
    print("Here")
```

a) 0 1 2 3 4 Here
b) 0 1 2 3 4 5 Here
c) 0 1 2 3 4
d) 1 2 3 4 5

Explanation: The else part is executed if control doesn't break out of the loop.

3. What will be the output of the following Python code?

```
x = (i for i in range(3))
for i in x:
    print(i)
```

a) 0 1 2
b) error
c) 0 1 2 0 1 2
d) none of the mentioned

Explanation: The first statement creates a generator object.

4. What will be the output of the following Python code?

```
x = (i for i in range(3))
for i in x:
    print(i)
for i in x:
    print(i)
```

a) 0 1 2
b) error
c) 0 1 2 0 1 2
d) none of the mentioned

Explanation: We can loop over a generator object only once.

5. What will be the output of the following Python code?

```
string = "my name is x"
for i in string:
    print (i, end=", ")
```

a) m, y, , n, a, m, e, , i, s, , x,
b) m, y, , n, a, m, e, , i, s, , x
c) my, name, is, x,
d) error

Explanation: Variable i takes the value of one character at a time.

6. What will be the output of the following Python code?

```
string = "my name is x"
```

```
for i in string.split():
    print (i, end=", ")
```

a) m, y, , n, a, m, e, , i, s, , x,
b) m, y, , n, a, m, e, , i, s, , x
c) my, name, is, x,
d) error

Explanation: Variable i takes the value of one word at a time.

7. What will be the output of the following Python code snippet?

```
a = [0, 1, 2, 3]
for a[-1] in a:
    print(a[-1])
```

a) 0 1 2 3
b) 0 1 2 2
c) 3 3 3 3
d) error

Explanation: The value of a[-1] changes in each iteration.

8. What will be the output of the following Python code snippet?

```
a = [0, 1, 2, 3]
for a[0] in a:
    print(a[0])
```

a) 0 1 2 3
b) 0 1 2 2
c) 3 3 3 3
d) error

Explanation: The value of a[0] changes in each iteration. Since the first value that it takes is itself, there is no visible error in the current example.

9. What will be the output of the following Python code snippet?

```
a = [0, 1, 2, 3]
i = -2
for i not in a:
    print(i)
    i += 1
```

a) -2 -1
b) 0
c) error
d) none of the mentioned

Explanation: SyntaxError, not in isn't allowed in for loops.

10. What will be the output of the following Python code snippet?

```
string = "my name is x"
for i in ' '.join(string.split()):
    print (i, end=", ")
```

a) m, y, , n, a, m, e, , i, s, , x,
b) m, y, , n, a, m, e, , i, s, , x
c) my, name, is, x,
d) error

Explanation: Variable i takes the value of one character at a time.

Python MCQ's – Strings – 1

1. What will be the output of the following Python statement?

```
1.>>>"a"+"bc"
```

a) a
b) bc
c) bca
d) abc

Explanation: + operator is concatenation operator.

2. What will be the output of the following Python statement?

```
1.>>>"abcd"[2:]
```

a) a
b) ab
c) cd
d) dc

Explanation: Slice operation is performed on string.

3. The output of executing string.ascii_letters can also be achieved by:
a) string.ascii_lowercase_string.digits
b) string.ascii_lowercase+string.ascii_uppercase
c) string.letters
d) string.lowercase_string.uppercase

Explanation: Execute in shell and check.

4. What will be the output of the following Python code?

```
1.>>> str1 = 'hello'
2.>>> str2 = ','
3.>>> str3 = 'world'
4.>>> str1[-1:]
```

a) olleh
b) hello
c) h
d) o

Explanation: -1 corresponds to the last index.

5. What arithmetic operators cannot be used with strings?
a) +
b) *
c) –
d) All of the mentioned

Explanation: + is used to concatenate and * is used to multiply strings.

6. What will be the output of the following Python code?

```
1.>>>print (r"\nhello")
```

a) a new line and hello
b) \nhello
c) the letter r and then hello
d) error

Explanation: When prefixed with the letter ‘r’ or ‘R’ a string literal becomes a raw string and the escape sequences such as \n are not converted.

7. What will be the output of the following Python statement?

```
1.>>>print('new' 'line')
```

a) Error
b) Output equivalent to print ‘new\nline’
c) newline
d) new line

Explanation: String literal separated by whitespace are allowed. They are concatenated.

8. What will be the output of the following Python statement?

```
1.>>> print('x\97\x98')
```

a) Error

b)97
98

c) x\97

d) \x97\x98

Explanation: \x is an escape sequence that means the following 2 digits are a hexadecimal number encoding a character.

9. What will be the output of the following Python code?

```
1.>>>str1="helloworld"
2.>>>str1[::-1]
```

a) dlrowolleh
b) hello
c) world
d) helloworld

Explanation: Execute in shell to verify.

10. What will be the output of the following Python code?

```
print(0xA + 0xB + 0xC)
```

a) 0xA0xB0xC
b) Error
c) 0x22

d) 33

Explanation: 0xA and 0xB and 0xC are hexadecimal integer literals representing the decimal values 10, 11 and 12 respectively. There sum is 33.

Python MCQ's – Strings – 2

1. What will be the output of the following Python code?

```
1.class father:
2.    def __init__(self, param):
3.        self.o1 = param
4.
5.class child(father):
6.    def __init__(self, param):
7.        self.o2 = param
8.
9.>>>obj = child(22)
10.       >>>print "%d %d" % (obj.o1, obj.o2)
```

a) None None
b) None 22
c) 22 None
d) Error is generated

Explanation: self.o1 was never created.

2. What will be the output of the following Python code?

```
1.class tester:
2.    def __init__(self, id):
3.        self.id = str(id)
4.        id="224"
5.
6.>>>temp = tester(12)
7.>>>print(temp.id)
```

a) 224
b) Error
c) 12
d) None

Explanation: Id in this case will be the attribute of the class.

3. What will be the output of the following Python code?

```
1.>>>example = "snow world"
2.>>>print("%s" % example[4:7])
```

a) wo
b) world
c) sn
d) rl

Explanation: Execute in the shell and verify.

4. What will be the output of the following Python code?

```
1.>>>example = "snow world"
2.>>>example[3] = 's'
3.>>>print example
```

a) snow
b) snow world
c) Error
d) snos world

Explanation: Strings cannot be modified.

5. What will be the output of the following Python code?

```
1.>>>max("what are you")
```

a) error
b) u
c) t
d) y

Explanation: Max returns the character with the highest ascii value.

6. Given a string example=”hello” what is the output of example.count(‘l’)?
a) 2
b) 1
c) None
d) 0

Explanation: l occurs twice in hello.

7. What will be the output of the following Python code?

```
1.>>>example = "helle"
```

```
2.>>>example.find("e")
```

a) Error
b) -1
c) 1
d) 0

Explanation: Returns lowest index.

8. What will be the output of the following Python code?

```
1.>>>example = "helle"
2.>>>example.rfind("e")
```

a) -1
b) 4
c) 3
d) 1

Explanation: Returns highest index.

9. What will be the output of the following Python code?

```
1.>>>example="helloworld"
2.>>>example[::-1].startswith("d")
```

a) dlrowolleh
b) True
c) -1
d) None

Explanation: Starts with checks if the given string starts with the parameter that is passed.

10. To concatenate two strings to a third what statements are applicable?
a) s3 = s1 . s2
b) s3 = s1.add(s2)
c) s3 = s1.__add__(s2)
d) s3 = s1 * s2

Explanation: __add__ is another method that can be used for concatenation.

Python MCQ's – Strings – 3

1. What will be the output of the following Python statement?

```
1.>>>chr(ord('A'))
```

a) A
b) B
c) a
d) Error

Explanation: Execute in shell to verify.

2. What will be the output of the following Python statement?

```
1.>>>print(chr(ord('b')+1))
```

a) a
b) b
c) c
d) A

Explanation: Execute in the shell to verify.

3. Which of the following statement prints hello\example\test.txt?
a) print(“hello\example\test.txt”)
b) print(“hello\\example\\test.txt”)
c) print(“hello\”example\”test.txt”)
d) print(“hello”\example”\test.txt”)

Explanation: \is used to indicate that the next \ is not an escape sequence.

4. Suppose s is “\t\tWorld\n”, what is s.strip()?
a) \t\tWorld\n
b) \t\tWorld\n
c) \t\tWORLD\n
d) World

Explanation: Execute help(string.strip) to find details.

5. The format function, when applied on a string returns _____
a) Error
b) int
c) bool
d) str

Explanation: Format function returns a string.

6. What will be the output of the “hello” +1+2+3?
a) hello123
b) hello
c) Error
d) hello6

Explanation: Cannot concatenate str and int objects.

7. What will be the output of the following Python code?

```
1.>>>print("D", end = ' ')
2.>>>print("C", end = ' ')
3.>>>print("B", end = ' ')
4.>>>print("A", end = ' ')
```

a) DCBA
b) A, B, C, D
c) D C B A
d) D, C, B, A will be displayed on four lines

Explanation: Execute in the shell.

8. What will be the output of the following Python statement?(python 3.xx)

```
1.>>>print(format("Welcome", "10s"), end = '#')
2.>>>print(format(111, "4d"), end = '#')
3.>>>print(format(924.656, "3.2f"))
```

a) Welcome# 111#924.66
b) Welcome#111#924.66
c) Welcome#111#.66
d) Welcome # 111#924.66

Explanation: Execute in the shell to verify.

9. What will be displayed by print(ord(‘b’) – ord(‘a’))?
a) 0
b) 1
c) -1
d) 2

Explanation: ASCII value of b is one more than a. Hence the output of this code is 98-97, which is equal to 1.

10. Say s=”hello” what will be the return value of type(s)?
a) int
b) bool
c) str
d) String

Explanation: str is used to represent strings in python.

Python MCQ’s – Strings – 4

1. What is “Hello”.replace(“l”, “e”)?
a) Heeeo
b) Heelo
c) Heleo
d) None

Explanation: Execute in shell to verify.

2. To retrieve the character at index 3 from string s=”Hello” what command do we execute (multiple answers allowed)?
a) s[]
b) s.getitem(3)
c) s.__getitem__(3)
d) s.getItem(3)

Explanation: __getitem(..) can be used to get character at index specified as parameter.

3. To return the length of string s what command do we execute?
a) s.__len__()
b) len(s)
c) size(s)
d) s.size()

Explanation: Execute in shell to verify.

4. If a class defines the __str__(self) method, for an object obj for the class, you can use which command to invoke the __str__ method.
a) obj.__str__()
b) str(obj)
c) print obj
d) all of the mentioned

Explanation: Execute in shell to verify.

5. To check whether string s1 contains another string s2, use _____
a) s1.__contains__(s2)
b) s2 in s1
c) s1.contains(s2)
d) si.in(s2)

Explanation: s2 in s1 works in the same way as calling the special function __contains__ .

6. Suppose i is 5 and j is 4, i + j is same as _____
a) i.__add(j)
b) i.__add__(j)
c) i.__Add(j)
d) i.__ADD(j)

Explanation: Execute in shell to verify.

7. What will be the output of the following Python code?

```
1.class Count:
2.    def __init__(self, count = 0):
3.       self.__count = count
4.
5.c1 = Count(2)
6.c2 = Count(2)
7.print(id(c1) == id(c2), end = " ")
8.
9.s1 = "Good"
10.       s2 = "Good"
11.       print(id(s1) == id(s2))
```

a) True False
b) True True
c) False True
d) False False

Explanation: Execute in the shell objects cannot have same id, however in the case of strings its different.

8. What will be the output of the following Python code?

```
1.class Name:
2.    def __init__(self, firstName, mi, lastName):
3.        self.firstName = firstName
4.        self.mi = mi
```

```
5.         self.lastName = lastName
6.
7.firstName = "John"
8.name = Name(firstName, 'F', "Smith")
9.firstName = "Peter"
10.       name.lastName = "Pan"
11.       print(name.firstName, name.lastName)
```

a) Peter Pan
b) John Pan
c) Peter Smith
d) John Smith

Explanation: Execute in the shell to verify.

9. What function do you use to read a string?
a) input("Enter a string")
b) eval(input("Enter a string"))
c) enter("Enter a string")
d) eval(enter("Enter a string"))

Explanation: Execute in shell to verify.

10. Suppose x is 345.3546, what is format(x, "10.3f") (_ indicates space).
a) __345.355
b) ___345.355
c) ____345.355
d) _____345.354

Explanation: Execute in the shell to verify.

Python MCQ's – Strings – 5

1. What will be the output of the following Python code?

```
print("abc DEF".capitalize())
```

a) abc def
b) ABC DEF
c) Abc def
d) Abc Def

Explanation: The first letter of the string is converted to uppercase and the others are converted to lowercase.

2. What will be the output of the following Python code?

```
print("abc. DEF".capitalize())
```

a) abc. def
b) ABC. DEF
c) Abc. def
d) Abc. Def

Explanation: The first letter of the string is converted to uppercase and the others are converted to lowercase.

3. What will be the output of the following Python code?

```
print("abcdef".center())
```

a) cd
b) abcdef
c) error
d) none of the mentioned

Explanation: The function center() takes at least one parameter.

4. What will be the output of the following Python code?

```
print("abcdef".center(0))
```

a) cd
b) abcdef
c) error
d) none of the mentioned

Explanation: The entire string is printed when the argument passed to center() is less than the length of the string.

5. What will be the output of the following Python code?

```
print('*', "abcdef".center(7), '*')
```

a) * abcdef *
b) * abcdef *
c) *abcdef *
d) * abcdef*

Explanation: Padding is done towards the left-hand-side first when the final string is of odd length. Extra spaces are present since we haven't overridden the value of sep.

6. What will be the output of the following Python code?

```
print('*', "abcdef".center(7), '*', sep='')
```

a) * abcdef *
b) * abcdef *
c) *abcdef *
d) * abcdef*

Explanation: Padding is done towards the left-hand-side first when the final string is of odd length.

7. What will be the output of the following Python code?

```
print('*', "abcde".center(6), '*', sep='')
```

a) * abcde *
b) * abcde *
c) *abcde *
d) * abcde*

Explanation: Padding is done towards the right-hand-side first when the final string is of even length.

8. What will be the output of the following Python code?

```
print("abcdef".center(7, 1))
```

a) 1abcdef
b) abcdef1
c) abcdef
d) error

Explanation: TypeError, the fill character must be a character, not an int.

9. What will be the output of the following Python code?

```
print("abcdef".center(7, '1'))
```

a) 1abcdef
b) abcdef1
c) abcdef
d) error

Explanation: The character ‘1’ is used for padding instead of a space.

10. What will be the output of the following Python code?

```
print("abcdef".center(10, '12'))
```

a) 12abcdef12
b) abcdef1212
c) 1212abcdef
d) error

Explanation: The fill character must be exactly one character long.

Python MCQ's – Strings – 6

1. What will be the output of the following Python code?

```
print("xyyzxyzxzxyy".count('yy'))
```

a) 2
b) 0
c) error
d) none of the mentioned

Explanation: Counts the number of times the substring 'yy' is present in the given string.

2. What will be the output of the following Python code?

```
print("xyyzxyzxzxyy".count('yy', 1))
```

a) 2
b) 0
c) 1
d) none of the mentioned

Explanation: Counts the number of times the substring 'yy' is present in the given string, starting from position 1.

3. What will be the output of the following Python code?

```
print("xyyzxyzxzxyy".count('yy', 2))
```

a) 2
b) 0
c) 1

d) none of the mentioned

Explanation: Counts the number of times the substring ‘yy’ is present in the given string, starting from position 2.

4. What will be the output of the following Python code?

```
print("xyyzxyzxzxyy".count('xyy', 0, 100))
```

a) 2
b) 0
c) 1
d) error

Explanation: An error will not occur if the end value is greater than the length of the string itself.

5. What will be the output of the following Python code?

```
print("xyyzxyzxzxyy".count('xyy', 2, 11))
```

a) 2
b) 0
c) 1
d) error

Explanation: Counts the number of times the substring ‘xyy’ is present in the given string, starting from position 2 and ending at position 11.

6. What will be the output of the following Python code?

```
print("xyyzxyzxzxyy".count('xyy', -10, -1))
```

a) 2
b) 0
c) 1
d) error

Explanation: Counts the number of times the substring ‘xyy’ is present in the given string, starting from position 2 and ending at position 11.

7. What will be the output of the following Python code?

```
print('abc'.encode())
```

a) abc
b) ‘abc’
c) b’abc’
d) h’abc’

Explanation: A bytes object is returned by encode.

8. What is the default value of encoding in encode()?
a) ascii
b) qwerty
c) utf-8
d) utf-16

Explanation: The default value of encoding is utf-8.

9. What will be the output of the following Python code?

```
print("xyyzxyzxzxyy".endswith("xyy"))
```

a) 1
b) True
c) 3
d) 2

Explanation: The function returns True if the given string ends with the specified substring.

10. What will be the output of the following Python code?

```
print("xyyzxyzxzxyy".endswith("xyy", 0, 2))
```

a) 0
b) 1
c) True
d) False

Explanation: The function returns False if the given string does not end with the specified substring.

Python MCQ’s – Strings – 7

1. What will be the output of the following Python code?

```
print("ab\tcd\tef".expandtabs())
```

a) ab cd ef
b) abcdef
c) ab\tcd\tef
d) ab cd ef

Explanation: Each \t is converted to 8 blank spaces by default.

2. What will be the output of the following Python code?

```
print("ab\tcd\tef".expandtabs(4))
```

a) ab cd ef
b) abcdef
c) ab\tcd\tef
d) ab cd ef

Explanation: Each \t is converted to 4 blank spaces.

3. What will be the output of the following Python code?

```
print("ab\tcd\tef".expandtabs('+'))
```

a) ab+cd+ef
b) ab++++++++cd++++++++ef
c) ab cd ef
d) none of the mentioned

Explanation: TypeError, an integer should be passed as an argument.

4. What will be the output of the following Python code?

```
print("abcdef".find("cd") == "cd" in "abcdef")
```

a) True
b) False
c) Error
d) None of the mentioned

Explanation: The function find() returns the position of the sunstring in the given string whereas the in keyword returns a value of Boolean type.

5. What will be the output of the following Python code?

```
print("abcdef".find("cd"))
```

a) True
b) 2
c) 3
d) None of the mentioned

Explanation: The first position in the given string at which the substring can be found is returned.

6. What will be the output of the following Python code?

```
print("ccdcddcd".find("c"))
```

a) 4
b) 0
c) Error
d) True

Explanation: The first position in the given string at which the substring can be found is returned.

7. What will be the output of the following Python code?

```
print("Hello {0} and {1}".format('foo', 'bin'))
```

a) Hello foo and bin
b) Hello {0} and {1} foo bin
c) Error
d) Hello 0 and 1

Explanation: The numbers 0 and 1 represent the position at which the strings are present.

8. What will be the output of the following Python code?

```
print("Hello {1} and {0}".format('bin', 'foo'))
```

a) Hello foo and bin
b) Hello bin and foo
c) Error
d) None of the mentioned

Explanation: The numbers 0 and 1 represent the position at which the strings are present.

9. What will be the output of the following Python code?

```
print("Hello {} and {}".format('foo', 'bin'))
```

a) Hello foo and bin
b) Hello {} and {}
c) Error
d) Hello and

Explanation: It is the same as Hello {0} and {1}.

10. What will be the output of the following Python code?

```
print("Hello {name1} and {name2}".format('foo', 'bin'))
```

a) Hello foo and bin
b) Hello {name1} and {name2}
c) Error
d) Hello and

Explanation: The arguments passed to the function format aren't keyword arguments.

Python MCQ's – Strings – 8

1. What will be the output of the following Python code?

```
print("Hello {name1} and {name2}".format(name1='foo', name2='bin'))
```

a) Hello foo and bin
b) Hello {name1} and {name2}
c) Error
d) Hello and

Explanation: The arguments are accessed by their names.

2. What will be the output of the following Python code?

```
print("Hello {0!r} and {0!s}".format('foo', 'bin'))
```

a) Hello foo and foo
b) Hello 'foo' and foo
c) Hello foo and 'bin'
d) Error

Explanation: !r causes the characters ' or " to be printed as well.

3. What will be the output of the following Python code?

```
print("Hello {0} and {1}".format(('foo', 'bin')))
```

a) Hello foo and bin
b) Hello (‘foo’, ‘bin’) and (‘foo’, ‘bin’)
c) Error
d) None of the mentioned

Explanation: IndexError, the tuple index is out of range.

4. What will be the output of the following Python code?

```
print("Hello {0[0]} and {0[1]}".format(('foo', 'bin')))
```

a) Hello foo and bin
b) Hello (‘foo’, ‘bin’) and (‘foo’, ‘bin’)
c) Error
d) None of the mentioned

Explanation: The elements of the tuple are accessed by their indices.

5. What will be the output of the following Python code snippet?

```
print('The sum of {0} and {1} is {2}'.format(2, 10, 12))
```

a) The sum of 2 and 10 is 12
b) Error
c) The sum of 0 and 1 is 2
d) None of the mentioned

Explanation: The arguments passed to the function format can be integers also.

6. What will be the output of the following Python code snippet?

```
print('The sum of {0:b} and {1:x} is {2:o}'.format(2, 10, 12))
```

a) The sum of 2 and 10 is 12
b) The sum of 10 and a is 14
c) The sum of 10 and a is c
d) Error

Explanation: 2 is converted to binary, 10 to hexadecimal and 12 to octal.

7. What will be the output of the following Python code snippet?

```
print('{:,}'.format(1112223334))
```

a) 1,112,223,334
b) 111,222,333,4
c) 1112223334
d) Error

Explanation: A comma is added after every third digit from the right.

8. What will be the output of the following Python code snippet?

```
print('{:,}'.format('1112223334'))
```

a) 1,112,223,334
b) 111,222,333,4
c) 1112223334
d) Error

Explanation: An integer is expected.

9. What will be the output of the following Python code snippet?

```
print('{:$}'.format(1112223334))
```

a) 1,112,223,334
b) 111,222,333,4
c) 1112223334
d) Error

Explanation: $ is an invalid format code.

10. What will be the output of the following Python code snippet?

```
print('{:#}'.format(1112223334))
```

a) 1,112,223,334
b) 111,222,333,4
c) 1112223334
d) Error

Explanation: The number is printed as it is.

Python MCQ's – Strings – 9

1. What will be the output of the following Python code?

```
print('{0:.2}'.format(1/3))
```

a) 0.333333
b) 0.33
c) 0.333333:.2
d) Error

Explanation: .2 specifies the precision.

2. What will be the output of the following Python code?

```
print('{0:.2%}'.format(1/3))
```

a) 0.33
b) 0.33%
c) 33.33%
d) 33%

Explanation: The symbol % is used to represent the result of an expression as a percentage.

3. What will be the output of the following Python code?

```
print('ab12'.isalnum())
```

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

Explanation: The string has only letters and digits.

4. What will be the output of the following Python code?

```
print('ab,12'.isalnum())
```

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

Explanation: The character , is not a letter or a digit.

5. What will be the output of the following Python code?

```
print('ab'.isalpha())
```

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

Explanation: The string has only letters.

6. What will be the output of the following Python code?

```
print('a B'.isalpha())
```

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

Explanation: Space is not a letter.

7. What will be the output of the following Python code snippet?

```
print('0xa'.isdigit())
```

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

Explanation: Hexadecimal digits aren’t considered as digits (a-f).

8. What will be the output of the following Python code snippet?

```
print(''.isdigit())
```

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

Explanation: If there are no characters then False is returned.

9.What will be the output of the following Python code snippet?

```
print('my_string'.isidentifier())
```

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

Explanation: It is a valid identifier.

10. What will be the output of the following Python code snippet?

```
print('__foo__'.isidentifier())
```

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

Explanation: It is a valid identifier.

Python MCQ's – Strings – 10

1. What will be the output of the following Python code snippet?

```
print('for'.isidentifier())
```

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

Explanation: Even keywords are considered as valid identifiers.

2. What will be the output of the following Python code snippet?

```
print('abc'.islower())
```

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

Explanation: There are no uppercase letters.

3. What will be the output of the following Python code snippet?

```
print('a@ 1,'.islower())
```

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

Explanation: There are no uppercase letters.

4. What will be the output of the following Python code snippet?

```
print('11'.isnumeric())
```

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

Explanation: All the character are numeric.

5. What will be the output of the following Python code snippet?

```
print('1.1'.isnumeric())
```

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

Explanation: The character . is not a numeric character.

6. What will be the output of the following Python code snippet?

```
print('1@ a'.isprintable())
```

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

Explanation: All those characters are printable.

7. What will be the output of the following Python code snippet?

```
print(''''''.isspace())
```

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

Explanation: None.

8. What will be the output of the following Python code snippet?

```
print('\t'.isspace())
```

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

Explanation: Tab Spaces are considered as spaces.

9. What will be the output of the following Python code snippet?

```
print('HelloWorld'.istitle())
```

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

Explanation: The letter W is uppercased.

10. What will be the output of the following Python code snippet?

```
print('Hello World'.istitle())
```

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

Explanation: It is in title form.

Python MCQ's – Strings – 11

1. What will be the output of the following Python code?

```
print('Hello!2@#World'.istitle())
```

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

Explanation: It is in the form of a title.

2. What will be the output of the following Python code?

```
print('1Rn@'.lower())
```

a) n
b) 1rn@
c) rn
d) r

Explanation: Uppercase letters are converted to lowercase. The other characters are left unchanged.

3. What will be the output of the following Python code?

```
print('''
 \tfoo'''.lstrip())
```

a) \tfoo
b) foo
c) foo
d) none of the mentioned

Explanation: All leading whitespace is removed.

4. What will be the output of the following Python code?

```
print('xyyzxxyxyy'.lstrip('xyy'))
```

a) error
b) zxxyxyy
c) z
d) zxxy

Explanation: The leading characters containing xyy are removed.

5. What will be the output of the following Python code?

```
print('xyxxyyzxxy'.lstrip('xyy'))
```

a) zxxy
b) xyxxyyzxxy
c) xyxzxxy
d) none of the mentioned

Explanation: All combinations of the characters passed as an argument are removed from the left hand side.

6. What will be the output of the following Python code?

```
print('cba'.maketrans('abc', '123'))
```

a) {97: 49, 98: 50, 99: 51}
b) {65: 49, 66: 50, 67: 51}
c) 321
d) 123

Explanation: A translation table is returned by maketrans.

7. What will be the output of the following Python code?

```
print('a'.maketrans('ABC', '123'))
```

a) {97: 49, 98: 50, 99: 51}
b) {65: 49, 66: 50, 67: 51}
c) {97: 49}
d) 1

Explanation: maketrans() is a static method so it's behaviour does not depend on the object from which it is being called.

8. What will be the output of the following Python code?

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

a) ('ab', 'ef')
b) ('abef')
c) ('ab', 'cd', 'ef')
d) 2

Explanation: The string is split into three parts by partition.

9. What will be the output of the following Python code?

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

a) (‘ab’, ‘cd’, ‘ef’, ‘cd’, ‘gh’)
b) (‘ab’, ‘cd’, ‘efcdgh’)
c) (‘abcdef’, ‘cd’, ‘gh’)
d) error

Explanation: The string is partitioned at the point where the separator first appears.

10. What will be the output of the following Python code?

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

a) (‘ab’, ‘cd’, ”)
b) (‘ab’, ‘cd’)
c) error
d) none of the mentioned

Explanation: The last item is a null string.

Python MCQ’s – Strings – 12

1. What will be the output of the following Python code snippet?

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

a) (‘cd’)
b) (”)
c) (‘cd’, ”, ”)
d) (”, ‘cd’, ”)

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

2. What will be the output of the following Python code snippet?

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

a) (‘abef’)
b) (‘abef’, ‘cd’, ”)
c) (‘abef’, ”, ”)
d) error

Explanation: The separator is not present in the string hence the second and the third elements of the tuple are null strings.

3. What will be the output of the following Python code snippet?

```
print('abcdef12'.replace('cd', '12'))
```

a) ab12ef12
b) abcdef12
c) ab12efcd
d) none of the mentioned

Explanation: All occurrences of the first substring are replaced by the second substring.

4. What will be the output of the following Python code snippet?

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

a) abef
b) 12
c) error
d) none of the mentioned

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

5. What will be the output of the following Python code snippet?

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

a) ab1ef2
b) abcefd
c) ab1efd
d) ab12ed2

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

6. What will be the output of the following Python code snippet?

```
print('xyyxyyxyxyxxy'.replace('xy', '12', 0))
```

a) xyyxyyxyxyxxy
b) 12y12y1212x12
c) 12yxyyxyxyxxy

d) xyyxyyxyxyx12

Explanation: The first 0 occurrences of the given substring are replaced.

7. What will be the output of the following Python code snippet?

```
print('xyyxyyxyxyxxy'.replace('xy', '12', 100))
```

a) xyyxyyxyxyxxy
b) 12y12y1212x12
c) none of the mentioned
d) error

Explanation: The first 100 occurrences of the given substring are replaced.

8. What will be the output of the following Python code snippet?

```
print('abcdefcdghcd'.split('cd'))
```

a) [‘ab’, ‘ef’, ‘gh’]
b) [‘ab’, ‘ef’, ‘gh’, ”]
c) (‘ab’, ‘ef’, ‘gh’)
d) (‘ab’, ‘ef’, ‘gh’, ”)

Explanation: The given string is split and a list of substrings is returned.

9. What will be the output of the following Python code snippet?

```
print('abcdefcdghcd'.split('cd', 0))
```

a) [‘abcdefcdghcd’]
b) ‘abcdefcdghcd’
c) error
d) none of the mentioned

Explanation: The given string is split at 0 occurances of the specified substring.

10. What will be the output of the following Python code snippet?

```
print('abcdefcdghcd'.split('cd', -1))
```

a) [‘ab’, ‘ef’, ‘gh’]
b) [‘ab’, ‘ef’, ‘gh’, ”]
c) (‘ab’, ‘ef’, ‘gh’)

d) (‘ab’, ‘ef’, ‘gh’, ”)

Explanation: Calling the function with a negative value for maxsplit is the same as calling it without any maxsplit specified. The string will be split into as many substring s as possible.

Python MCQ’s – Strings – 13

1. What will be the output of the following Python code snippet?

```
print('abcdefcdghcd'.split('cd', 2))
```

a) [‘ab’, ‘ef’, ‘ghcd’]
b) [‘ab’, ‘efcdghcd’]
c) [‘abcdef’, ‘ghcd’]
d) none of the mentioned

Explanation: The string is split into a maximum of maxsplit+1 substrings.

2. What will be the output of the following Python code snippet?

```
print('ab\ncd\nef'.splitlines())
```

a) [‘ab’, ‘cd’, ‘ef’]
b) [‘ab\n’, ‘cd\n’, ‘ef\n’]
c) [‘ab\n’, ‘cd\n’, ‘ef’]
d) [‘ab’, ‘cd’, ‘ef\n’]

Explanation: It is similar to calling split(‘\n’).

3. What will be the output of the following Python code snippet?

```
print('Ab!2'.swapcase())
```

a) AB!@
b) ab12
c) aB!2
d) aB1@

Explanation: Lowercase letters are converted to uppercase and vice-versa.

4. What will be the output of the following Python code snippet?

```
print('ab cd ef'.title())
```

a) Ab cd ef
b) Ab cd eF
c) Ab Cd Ef
d) None of the mentioned

Explanation: The first letter of every word is capitalized.

5. What will be the output of the following Python code snippet?

```
print('ab cd-ef'.title())
```

a) Ab cd-ef
b) Ab Cd-ef
c) Ab Cd-Ef
d) None of the mentioned

Explanation: The first letter of every word is capitalized. Special symbols terminate a word.

6. What will be the output of the following Python code snippet?

```
print('abcd'.translate('a'.maketrans('abc', 'bcd')))
```

a) bcde
b) abcd
c) error
d) bcdd

Explanation: The output is bcdd since no translation is provided for d.

7. What will be the output of the following Python code snippet?

```
print('abcd'.translate({97: 98, 98: 99, 99: 100}))
```

a) bcde
b) abcd
c) error
d) none of the mentioned

Explanation: The output is bcdd since no translation is provided for d.

8. What will be the output of the following Python code snippet?

```
print('abcd'.translate({'a': '1', 'b': '2', 'c': '3', 'd': '4'}))
```

a) abcd
b) 1234
c) error
d) none of the mentioned

Explanation: The function translate expects a dictionary of integers. Use maketrans() instead of doing the above.

9. What will be the output of the following Python code snippet?

```
print('ab'.zfill(5))
```

a) 000ab
b) 00ab0
c) 0ab00
d) ab000

Explanation: The string is padded with zeros on the left hand side. It is useful for formatting numbers.

10. What will be the output of the following Python code snippet?

```
print('+99'.zfill(5))
```

a) 00+99
b) 00099
c) +0099
d) +++99

Explanation: zeros are filled in between the first sign and the rest of the string.

Python MCQ's – Lists – 1

1. Which of the following commands will create a list?
a) list1 = list()
b) list1 = []
c) list1 = list([1, 2, 3])
d) all of the mentioned

Explanation: Execute in the shell to verify

2. What is the output when we execute list("hello")?
a) ['h', 'e', 'l', 'l', 'o']

b) [‘hello’]
c) [‘llo’]
d) [‘olleh’]

Explanation: Execute in the shell to verify.

3. Suppose listExample is [‘h’,’e’,’l’,’l’,’o’], what is len(listExample)?
a) 5
b) 4
c) None
d) Error

Explanation: Execute in the shell and verify.

4. Suppose list1 is [2445,133,12454,123], what is max(list1)?
a) 2445
b) 133
c) 12454
d) 123

Explanation: Max returns the maximum element in the list.

5. Suppose list1 is [3, 5, 25, 1, 3], what is min(list1)?
a) 3
b) 5
c) 25
d) 1

Explanation: Min returns the minimum element in the list.

6. Suppose list1 is [1, 5, 9], what is sum(list1)?
a) 1
b) 9
c) 15
d) Error

Explanation: Sum returns the sum of all elements in the list.

7. To shuffle the list(say list1) what function do we use?
a) list1.shuffle()
b) shuffle(list1)
c) random.shuffle(list1)

d) random.shuffleList(list1)

Explanation: Execute in the shell to verify.

8. Suppose list1 is [4, 2, 2, 4, 5, 2, 1, 0], Which of the following is correct syntax for slicing operation?
a) print(list1[2:])
b) print(list1[:2])
c) print(list1[:-2])
d) all of the mentioned

Explanation: Slicing is allowed in lists just as in the case of strings.

9. Suppose list1 is [2, 33, 222, 14, 25], What is list1[-1]?
a) Error
b) None
c) 25
d) 2

Explanation: -1 corresponds to the last index in the list.

10. Suppose list1 is [2, 33, 222, 14, 25], What is list1[:-1]?
a) [2, 33, 222, 14]
b) Error
c) 25
d) [25, 14, 222, 33, 2]

Explanation: Execute in the shell to verify.

Python MCQ's – Lists – 2

1. What will be the output of the following Python code?

```
1.>>>names = ['Amir', 'Bear', 'Charlton', 'Daman']
2.>>>print(names[-1][-1])
```

a) A
b) Daman
c) Error
d) n

Explanation: Execute in the shell to verify.

2. What will be the output of the following Python code?

```
1.names1 = ['Amir', 'Bear', 'Charlton', 'Daman']
2.names2 = names1
3.names3 = names1[:]
4.
5.names2[0] = 'Alice'
6.names3[1] = 'Bob'
7.
8.sum = 0
9.for ls in (names1, names2, names3):
10.           if ls[0] == 'Alice':
11.               sum += 1
12.           if ls[1] == 'Bob':
13.               sum += 10
14.
15.       print sum
```

a) 11
b) 12
c) 21
d) 22

Explanation: When assigning names1 to names2, we create a second reference to the same list. Changes to names2 affect names1. When assigning the slice of all elements in names1 to names3, we are creating a full copy of names1 which can be modified independently.

3. Suppose list1 is [1, 3, 2], What is list1 * 2?
a) [2, 6, 4]
b) [1, 3, 2, 1, 3]
c) [1, 3, 2, 1, 3, 2]
d) [1, 3, 2, 3, 2, 1]

Explanation: Execute in the shell and verify.

4. Suppose list1 = [0.5 * x for x in range(0, 4)], list1 is:
a) [0, 1, 2, 3]
b) [0, 1, 2, 3, 4]
c) [0.0, 0.5, 1.0, 1.5]
d) [0.0, 0.5, 1.0, 1.5, 2.0]

Explanation: Execute in the shell to verify.

5. What will be the output of the following Python code?

```
1.>>>list1 = [11, 2, 23]
2.>>>list2 = [11, 2, 2]
3.>>>list1 < list2
```

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

Explanation: Elements are compared one by one.

6. To add a new element to a list we use which command?
a) list1.add(5)
b) list1.append(5)
c) list1.addLast(5)
d) list1.addEnd(5)

Explanation: We use the function append to add an element to the list.

7. To insert 5 to the third position in list1, we use which command?
a) list1.insert(3, 5)
b) list1.insert(2, 5)
c) list1.add(3, 5)
d) list1.append(3, 5)

Explanation: Execute in the shell to verify.

8. To remove string “hello” from list1, we use which command?
a) list1.remove(“hello”)
b) list1.remove(hello)
c) list1.removeAll(“hello”)
d) list1.removeOne(“hello”)

Explanation: Execute in the shell to verify.

9. Suppose list1 is [3, 4, 5, 20, 5], what is list1.index(5)?
a) 0
b) 1
c) 4
d) 2

Explanation: Execute help(list.index) to get details.

10. Suppose list1 is [3, 4, 5, 20, 5, 25, 1, 3], what is list1.count(5)?
a) 0
b) 4
c) 1
d) 2

Explanation: Execute in the shell to verify.

Python MCQ's – Lists – 3

1. Suppose list1 is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after list1.reverse()?
a) [3, 4, 5, 20, 5, 25, 1, 3]
b) [1, 3, 3, 4, 5, 5, 20, 25]
c) [25, 20, 5, 5, 4, 3, 3, 1]
d) [3, 1, 25, 5, 20, 5, 4, 3]

Explanation: Execute in the shell to verify.

2. Suppose listExample is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after listExample.extend([34, 5])?

a) [3, 4, 5, 20, 5, 25, 1, 3, 34, 5]
b) [1, 3, 3, 4, 5, 5, 20, 25, 34, 5]
c) [25, 20, 5, 5, 4, 3, 3, 1, 34, 5]
d) [1, 3, 4, 5, 20, 5, 25, 3, 34, 5]

Explanation: Execute in the shell to verify.

3. Suppose listExample is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after listExample.pop(1)?

a) [3, 4, 5, 20, 5, 25, 1, 3]
b) [1, 3, 3, 4, 5, 5, 20, 25]
c) [3, 5, 20, 5, 25, 1, 3]
d) [1, 3, 4, 5, 20, 5, 25]

Explanation: pop() removes the element at the position specified in the parameter.

4. Suppose listExample is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after listExample.pop()?
a) [3, 4, 5, 20, 5, 25, 1]
b) [1, 3, 3, 4, 5, 5, 20, 25]
c) [3, 5, 20, 5, 25, 1, 3]

d) [1, 3, 4, 5, 20, 5, 25]

Explanation: pop() by default will remove the last element.

5. What will be the output of the following Python code?

```
1.>>>"Welcome to Python".split()
```

a) [“Welcome”, “to”, “Python”]
b) (“Welcome”, “to”, “Python”)
c) {“Welcome”, “to”, “Python”}
d) “Welcome”, “to”, “Python”

Explanation: split() function returns the elements in a list.

6. What will be the output of the following Python code?

```
1.>>>list("a#b#c#d".split('#'))
```

a) [‘a’, ‘b’, ‘c’, ‘d’]
b) [‘a b c d’]
c) [‘a#b#c#d’]
d) [‘abcd’]

Explanation: Execute in the shell to verify.

7. What will be the output of the following Python code?

```
1.myList = [1, 5, 5, 5, 5, 1]
2.max = myList[0]
3.indexOfMax = 0
4.for i in range(1, len(myList)):
5.    if myList[i] > max:
6.        max = myList[i]
7.        indexOfMax = i
8.
9.>>>print(indexOfMax)
```

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

Explanation: First time the highest number is encountered is at index 1.

8. What will be the output of the following Python code?

```
1.myList = [1, 2, 3, 4, 5, 6]
2.for i in range(1, 6):
3.    myList[i - 1] = myList[i]
4.
5.for i in range(0, 6):
6.    print(myList[i], end = " ")
```

a) 2 3 4 5 6 1
b) 6 1 2 3 4 5
c) 2 3 4 5 6 6
d) 1 1 2 3 4 5

Explanation: Execute in the shell to verify.

9. What will be the output of the following Python code?

```
1.>>>list1 = [1, 3]
2.>>>list2 = list1
3.>>>list1[0] = 4
4.>>>print(list2)
```

a) [1, 3]
b) [4, 3]
c) [1, 4]
d) [1, 3, 4]

Explanation: Lists should be copied by executing [:] operation.

10. What will be the output of the following Python code?

```
1.def f(values):
2.    values[0] = 44
3.
4.v = [1, 2, 3]
5.f(v)
6.print(v)
```

a) [1, 44]
b) [1, 2, 3, 44]
c) [44, 2, 3]
d) [1, 2, 3]

Explanation: Execute in the shell to verify.

Python MCQ's – Lists – 4

1. What will be the output of the following Python code?

```
1.def f(i, values = []):
2.    values.append(i)
3.    return values
4.
5.f(1)
6.f(2)
7.v = f(3)
8.print(v)
```

a) [1] [2] [3]
b) [1] [1, 2] [1, 2, 3]
c) [1, 2, 3]
d) 1 2 3

Explanation: Execute in the shell to verify

2. What will be the output of the following Python code?

```
1.names1 = ['Amir', 'Bala', 'Chales']
2.
3.if 'amir' in names1:
4.    print(1)
5.else:
6.    print(2)
```

a) None
b) 1
c) 2
d) Error

Explanation: Execute in the shell to verify.

3. What will be the output of the following Python code?

```
1.names1 = ['Amir', 'Bala', 'Charlie']
2.names2 = [name.lower() for name in names1]
3.
4.print(names2[2][0])
```

a) None
b) a
c) b
d) c

Explanation: List Comprehension are a shorthand for creating new lists.

4. What will be the output of the following Python code?

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

a) 4
b) 5
c) 8
d) 12

Explanation: A list is passed in append so the length is 5.

5. To which of the following the "in" operator can be used to check if an item is in it?
a) Lists
b) Dictionary
c) Set
d) All of the mentioned

Explanation: In can be used in all data structures.

6. What will be the output of the following Python code?

```
1.list1 = [1, 2, 3, 4]
2.list2 = [5, 6, 7, 8]
3.
4.print(len(list1 + list2))
```

a) 2
b) 4
c) 5
d) 8

Explanation: + appends all the elements individually into a new list.

7. What will be the output of the following Python code?

```
1.def addItem(listParam):
2.    listParam += [1]
3.
4.mylist = [1, 2, 3, 4]
5.addItem(mylist)
6.print(len(mylist))
```

a) 1
b) 4
c) 5
d) 8

Explanation: + will append the element to the list.

8. What will be the output of the following Python code?

```
1.def increment_items(L, increment):
2.    i = 0
3.    while i < len(L):
4.        L[i] = L[i] + increment
5.        i = i + 1
6.
7.values = [1, 2, 3]
8.print(increment_items(values, 2))
9.print(values)
```

a)None
[3, 4, 5]

b)None
[1, 2, 3]

c)[3, 4, 5]
[1, 2, 3]

d)[3, 4, 5]
None

Explanation: Execute in the shell to verify.

9. What will be the output of the following Python code?

```
1.def example(L):
2.    ''' (list) -> list
3.    '''
4.    i = 0
5.    result = []
6.    while i < len(L):
```

```
7.          result.append(L[i])
8.          i = i + 3
9.      return result
```

a) Return a list containing every third item from L starting at index 0
b) Return an empty list
c) Return a list containing every third index from L starting at index 0
d) Return a list containing the items from L starting from index 0, omitting every third item

Explanation: Run the code to get a better understanding with many arguments.

10. What will be the output of the following Python code?

```
1.veggies = ['carrot', 'broccoli', 'potato', 'asparagus']
2.veggies.insert(veggies.index('broccoli'), 'celery')
3.print(veggies)
```

a) [‘carrot’, ‘celery’, ‘broccoli’, ‘potato’, ‘asparagus’] Correct 1.00

b) [‘carrot’, ‘celery’, ‘potato’, ‘asparagus’]

c) [‘carrot’, ‘broccoli’, ‘celery’, ‘potato’, ‘asparagus’]

d) [‘celery’, ‘carrot’, ‘broccoli’, ‘potato’, ‘asparagus’]

Explanation: Execute in the shell to verify.

Python MCQ’s – Lists – 5

1. What will be the output of the following Python code?

```
1.>>>m = [[x, x + 1, x + 2] for x in range(0, 3)]
```

a) [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
b) [[0, 1, 2], [1, 2, 3], [2, 3, 4]]
c) [1, 2, 3, 4, 5, 6, 7, 8, 9]
d) [0, 1, 2, 1, 2, 3, 2, 3, 4]

Explanation: Execute in the shell to verify.

2. How many elements are in m?

```
1.m = [[x, y] for x in range(0, 4) for y in range(0, 4)]
```

a) 8
b) 12
c) 16
d) 32

Explanation: Execute in the shell to verify.

3. What will be the output of the following Python code?

```
values = [[3, 4, 5, 1], [33, 6, 1, 2]]

v = values[0][0]
for row in range(0, len(values)):
    for column in range(0, len(values[row])):
        if v < values[row][column]:
            v = values[row][column]

print(v)
```

a) 3
b) 5
c) 6
d) 33

Explanation: Execute in the shell to verify.

4. What will be the output of the following Python code?

```
values = [[3, 4, 5, 1], [33, 6, 1, 2]]

v = values[0][0]
for lst in values:
    for element in lst:
        if v > element:
            v = element

print(v)
```

a) 1
b) 3
c) 5
d) 6

Explanation: Execute in the shell to verify.

5. What will be the output of the following Python code?

```
values = [[3, 4, 5, 1 ], [33, 6, 1, 2]]

for row in values:
    row.sort()
    for element in row:
        print(element, end = " ")
    print()
```

a) The program prints two rows 3 4 5 1 followed by 33 6 1 2
b) The program prints on row 3 4 5 1 33 6 1 2
c) The program prints two rows 3 4 5 1 followed by 33 6 1 2
d) The program prints two rows 1 3 4 5 followed by 1 2 6 33

Explanation: Execute in the shell to verify.

6. What will be the output of the following Python code?

```
matrix = [[1, 2, 3, 4],
       [4, 5, 6, 7],
       [8, 9, 10, 11],
       [12, 13, 14, 15]]

for i in range(0, 4):
    print(matrix[i][1], end = " ")
```

a) 1 2 3 4
b) 4 5 6 7
c) 1 3 8 12
d) 2 5 9 13

Explanation: Execute in the shell to verify.

7. What will be the output of the following Python code?

```
def m(list):
    v = list[0]
    for e in list:
      if v < e: v = e
    return v

values = [[3, 4, 5, 1], [33, 6, 1, 2]]

for row in values:
          print(m(row), end = " ")
```

a) 3 33
b) 1 1
c) 5 6

d) 5 33

Explanation: Execute in the shell to verify.

8. What will be the output of the following Python code?

```
1.data = [[[1, 2], [3, 4]], [[5, 6], [7, 8]]]
2.
3.print(data[1][0][0])
```

a) 1
b) 2
c) 4
d) 5

Explanation: Execute in the shell to verify.

9. What will be the output of the following Python code?

```
1.data = [[[1, 2], [3, 4]], [[5, 6], [7, 8]]]
2.
3.def ttt(m):
4.    v = m[0][0]
5.
6.    for row in m:
7.        for element in row:
8.           if v < element: v = element
9.
10.           return v
11.
12.       print(ttt(data[0]))
```

a) 1
b) 2
c) 4
d) 5

Explanation: Execute in the shell to verify.

10. What will be the output of the following Python code?

```
1.points = [[1, 2], [3, 1.5], [0.5, 0.5]]
2.points.sort()
3.print(points)
```

a) [[1, 2], [3, 1.5], [0.5, 0.5]]
b) [[3, 1.5], [1, 2], [0.5, 0.5]]

c) [[0.5, 0.5], [1, 2], [3, 1.5]]
d) [[0.5, 0.5], [3, 1.5], [1, 2]]

Explanation: Execute in the shell to verify.

Python MCQ's – Lists – 6

1. What will be the output of the following Python code?

```
a=[10,23,56,[78]]
b=list(a)
a[3][0]=95
a[1]=34
print(b)
```

a) [10,34,56,[95]]
b) [10,23,56,[78]]
c) [10,23,56,[95]]
d) [10,34,56,[78]]

Explanation: The above copy is a type of shallow copy and only changes made in sublist is reflected in the copied list.

2. What will be the output of the following Python code?

```
print(list(zip((1,2,3),('a'),('xxx','yyy'))))
print(list(zip((2,4),('b','c'),('yy','xx'))))
```

a) [(1,2,3),('a'),('xxx','yyy')]
[(2,4),('b','c'),('yy','xx')]

b) [(1, 'a', 'xxx'),(2,' ','yyy'),(3,' ',' ')]
[(2, 'b', 'yy'), (4, 'c', 'xx')]

c) Syntax error

d) [(1, 'a', 'xxx')]
[(2, 'b', 'yy'), (4, 'c', 'xx')]

Explanation: The zip function combines the individual attributes of the lists into a list of tuples.

3. What will be the output of the following Python code?

```
import copy
a=[10,23,56,[78]]
b=copy.deepcopy(a)
a[3][0]=95
a[1]=34
print(b)
```

a) [10,34,56,[95]]
b) [10,23,56,[78]]
c) [10,23,56,[95]]
d) [10,34,56,[78]]

Explanation: The above copy is deepcopy. Any change made in the original list isn't reflected.

4. What will be the output of the following Python code?

```
s="a@b@c@d"
a=list(s.partition("@"))
print(a)
b=list(s.split("@",3))
print(b)
```

a) ['a','b','c','d']
['a','b','c','d']

b) ['a','@','b','@','c','@','d']
['a','b','c','d']

c) ['a','@','b@c@d']
['a','b','c','d']

d) ['a','@','b@c@d']
['a','@','b','@','c','@','d']

Explanation: The partition function only splits for the first parameter along with the separator while split function splits for the number of times given in the second argument but without the separator.

5. What will be the output of the following Python code?

```
a=[1,2,3,4]
b=[sum(a[0:x+1]) for x in range(0,len(a))]
print(b)
```

a) 10
b) [1,3,5,7]
c) 4
d) [1,3,6,10]

Explanation: The above code returns the cumulative sum of elements in a list.

6. What will be the output of the following Python code?

```
a="hello"
b=list((x.upper(),len(x)) for x in a)
print(b)
```

a) [('H', 1), ('E', 1), ('L', 1), ('L', 1), ('O', 1)]
b) [('HELLO', 5)]
c) [('H', 5), ('E', 5), ('L', 5), ('L', 5), ('O', 5)]
d) Syntax error

Explanation: Variable x iterates over each letter in string a hence the length of each letter is 1.

7. What will be the output of the following Python code?

```
a=[1,2,3,4]
b=[sum(a[0:x+1]) for x in range(0,len(a))]
print(b)
```

a) 10
b) [1,3,5,7]
c) 4
d) [1,3,6,10]

Explanation: The above code returns the cumulative sum of elements in a list.

8. What will be the output of the following Python code?

```
a=[[]]*3
a[1].append(7)
print(a)
```

a) Syntax error
b) [[7], [7], [7]]
c) [[7], [], []]
d) [[],7, [], []]

Explanation: The first line of the code creates multiple reference copies of sublist. Hence when 7 is appended, it gets appended to all the sublists.

9. What will be the output of the following Python code?

```
b=[2,3,4,5]
a=list(filter(lambda x:x%2,b))
print(a)
```

a) [2,4]
b) [ ]
c) [3,5]
d) Invalid arguments for filter function

Explanation: The filter function gives value from the list b for which the condition is true, that is, x%2==1.

10. What will be the output of the following Python code?

```
lst=[3,4,6,1,2]
lst[1:2]=[7,8]
print(lst)
```

a) [3, 7, 8, 6, 1, 2]
b) Syntax error
c) [3,[7,8],6,1,2]
d) [3,4,6,7,8]

Explanation: In the piece of code, slice assignment has been implemented. The sliced list is replaced by the assigned elements in the list. Type in python shell to verify.

Python MCQ's – Lists – 7

1. What will be the output of the following Python code?

```
a=[1,2,3]
b=a.append(4)
print(a)
print(b)
```

a) [1,2,3,4]
[1,2,3,4]

b) [1, 2, 3, 4]
None

c) Syntax error

d) [1,2,3]
[1,2,3,4]

Explanation: Append function on lists doesn't return anything. Thus the value of b is None.

2. What will be the output of the following Python code?

```
>>> a=[14,52,7]
>>>> b=a.copy()
>>> b is a
```

a) True
b) False

Explanation: List b is just a copy of the original list. Any copy made in list b will not be reflected in list a.

3. What will be the output of the following Python code?

```
a=[13,56,17]
a.append([87])
a.extend([45,67])
print(a)
```

a) [13, 56, 17, [87], 45, 67]
b) [13, 56, 17, 87, 45, 67]
c) [13, 56, 17, 87,[45, 67]]
d) [13, 56, 17, [87], [45, 67]]

Explanation: The append function simply adds its arguments to the list as it is while extend function extends its arguments and later appends it.

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

```
a=list((45,)*4)
print((45)*4)
print(a)
```

a) 180
[(45),(45),(45),(45)]

b) (45,45,45,45)
[45,45,45,45]

c) 180
[45,45,45,45]

d) Syntax error

Explanation: (45) is an int while (45,) is a tuple of one element. Thus when a tuple is multiplied, it created references of itself which is later converted to a list.

5. What will be the output of the following Python code?

```
lst=[[1,2],[3,4]]
print(sum(lst,[]))
```

a) [[3],[7]]
b) [1,2,3,4]
c) Error
d) [10]

Explanation: The above piece of code is used for flattening lists.

6. What will be the output of the following Python code?

```
word1="Apple"
word2="Apple"
list1=[1,2,3]
list2=[1,2,3]
print(word1 is word2)
print(list1 is list2)
```

a) True
True
b) False
True
c) False
False
d) True

False

Explanation: In the above case, both the lists are equivalent but not identical as they have different objects.

7. What will be the output of the following Python code?

```
def unpack(a,b,c,d):
    print(a+d)
x = [1,2,3,4]
unpack(*x)
```

a) Error
b) [1,4]
c) [5]
d) 5

Explanation: unpack(*x) unpacks the list into the separate variables. Now, a=1 and d=4. Thus 5 gets printed.

8. What will be the output of the following Python code?

```
places = ['Bangalore', 'Mumbai', 'Delhi']
<br class="blank" />places1 = places
places2 = places[:]
<br class="blank" />places1[1]="Pune"
places2[2]="Hyderabad"
print(places)
```

a) [‘Bangalore’, ‘Pune’, ‘Hyderabad’]
b) [‘Bangalore’, ‘Pune’, ‘Delhi’]
c) [‘Bangalore’, ‘Mumbai’, ‘Delhi’]
d) [‘Bangalore’, ‘Mumbai’, ‘Hyderabad’]

Explanation: places1 is an alias of the list places. Hence, any change made to places1 is reflected in places. places2 is a copy of the list places. Thus, any change made to places2 isn’t reflected in places.

9. What will be the output of the following Python code?

```
x=[[1],[2]]
print(" ".join(list(map(str,x))))
```

a) [1] [2]
b) [49] [50]
c) Syntax error
d) [[1]] [[2]]

Explanation: The elements 1 and 2 are first put into separate lists and then combined with a space in between using the join attribute.

10. What will be the output of the following Python code?

```
a=165
b=sum(list(map(int,str(a))))
print(b)
```

a) 561
b) 5
c) 12
d) Syntax error

Explanation: First, map converts the number to string and then places the individual digits in a list. Then, sum finds the sum of the digits in the list. The code basically finds the sum of digits in the number.

11. What will be the output of the following Python code?

```
a= [1, 2, 3, 4, 5]
for i in range(1, 5):
    a[i-1] = a[i]
for i in range(0, 5):
    print(a[i],end = " ")
```

a) 5 5 1 2 3
b) 5 1 2 3 4
c) 2 3 4 5 1
d) 2 3 4 5 5

Explanation: The items having indexes from 1 to 4 are shifted forward by one index due to the first for-loop and the item of index four is printed again because of the second for-loop.

12. What will be the output of the following Python code?

```
def change(var, lst):
    var = 1
    lst[0] = 44
k = 3
a = [1, 2, 3]
```

```
change(k, a)
print(k)
print(a)
```

a) 3
[44, 2, 3]

b) 1
[1,2,3]

c) 3
[1,2,3]

d) 1
[44,2,3]

Explanation: A list is mutable, hence it's value changes after function call. However, integer isn't mutable. Thus its value doesn't change.

13. What will be the output of the following Python code?

```
a = [1, 5, 7, 9, 9, 1]
<br class="blank" />b=a[0]
<br class="blank" />x= 0
for x in range(1, len(a)):
    if a[x] > b:
        b = a[x]
        b= x
print(b)
```

a) 5
b) 3
c) 4
d) 0

Explanation: The above piece of code basically prints the index of the largest element in the list.

14. What will be the output of the following Python code?

```
a=["Apple","Ball","Cobra"]
<br class="blank" />a.sort(key=len)
```

```
print(a)
```

a) [‘Apple’, ‘Ball’, ‘Cobra’]
b) [‘Ball’, ‘Apple’, ‘Cobra’]
c) [‘Cobra’, ‘Apple’, ‘Ball’]
d) Invalid syntax for sort()

Explanation: The syntax isn’t invalid and the list is sorted according to the length of the strings in the list since key is given as len.

15. What will be the output of the following Python code?

```
num = ['One', 'Two', 'Three']
for i, x in enumerate(num):
    print('{}: {}'.format(i, x),end=" ")
```

a) 1: 2: 3:
b) Exception is thrown
c) One Two Three
d) 0: One 1: Two 2: Three

Explanation: enumerate(iterator,start=0) is a built-in function which returns (0,lst[0]),(1,lst[1]) and so on where lst is a list(iterator).

Python MCQ’s – List Comprehension

1. What will be the output of the following Python code snippet?

```
k = [print(i) for i in my_string if i not in "aeiou"]
```

a) prints all the vowels in my_string
b) prints all the consonants in my_string
c) prints all characters of my_string that aren’t vowels
d) prints only on executing print(k)

Explanation: print(i) is executed if the given character is not a vowel.

2. What is the output of print(k) in the following Python code snippet?

```
k = [print(i) for i in my_string if i not in "aeiou"]
print(k)
```

a) all characters of my_string that aren't vowels
b) a list of Nones
c) list of Trues
d) list of Falses

Explanation: print() returns None.

3. What will be the output of the following Python code snippet?

```
my_string = "hello world"
k = [(i.upper(), len(i)) for i in my_string]
print(k)
```

a) [('HELLO', 5), ('WORLD', 5)]
b) [('H', 1), ('E', 1), ('L', 1), ('L', 1), ('O', 1), (' ', 1), ('W', 1), ('O', 1), ('R', 1), ('L', 1), ('D', 1)]
c) [('HELLO WORLD', 11)]
d) none of the mentioned

Explanation: We are iterating over each letter in the string.

4. Which of the following is the correct expansion of list_1 = [expr(i) for i in list_0 if func(i)]?
a)

```
list_1 = []
for i in list_0:
    if func(i):
        list_1.append(i)
```

b)

```
for i in list_0:
    if func(i):
        list_1.append(expr(i))
```

c)

```
list_1 = []
for i in list_0:
    if func(i):
        list_1.append(expr(i))
```

d) none of the mentioned

Explanation: We have to create an empty list, loop over the contents of the existing list and check if a condition is satisfied before performing some operation and adding it to the new list.

5. What will be the output of the following Python code snippet?

```
x = [i**+1 for i in range(3)]; print(x);
```

a) [0, 1, 2]
b) [1, 2, 5]
c) error, **+ is not a valid operator
d) error, ';' is not allowed

Explanation: i**+1 is evaluated as (i)**(+1).

6. What will be the output of the following Python code snippet?

```
print([i.lower() for i in "HELLO"])
```

a) ['h', 'e', 'l', 'l', 'o']
b) 'hello'
c) ['hello']
d) hello

Explanation: We are iterating over each letter in the string.

7. What will be the output of the following Python code snippet?

```
print([i+j for i in "abc" for j in "def"])
```

a) ['da', 'ea', 'fa', 'db', 'eb', 'fb', 'dc', 'ec', 'fc']
b) [['ad', 'bd', 'cd'], ['ae', 'be', 'ce'], ['af', 'bf', 'cf']]
c) [['da', 'db', 'dc'], ['ea', 'eb', 'ec'], ['fa', 'fb', 'fc']]
d) ['ad', 'ae', 'af', 'bd', 'be', 'bf', 'cd', 'ce', 'cf']

Explanation: If it were to be executed as a nested for loop, i would be the outer loop and j the inner loop.

8. What will be the output of the following Python code snippet?

```
print([[i+j for i in "abc"] for j in "def"])
```

a) [‘da’, ‘ea’, ‘fa’, ‘db’, ‘eb’, ‘fb’, ‘dc’, ‘ec’, ‘fc’]
b) [[‘ad’, ‘bd’, ‘cd’], [‘ae’, ‘be’, ‘ce’], [‘af’, ‘bf’, ‘cf’]]
c) [[‘da’, ‘db’, ‘dc’], [‘ea’, ‘eb’, ‘ec’], [‘fa’, ‘fb’, ‘fc’]]
d) [‘ad’, ‘ae’, ‘af’, ‘bd’, ‘be’, ‘bf’, ‘cd’, ‘ce’, ‘cf’]

Explanation: The inner list is generated once for each value of j.

9. What will be the output of the following Python code snippet?

```
print([if i%2==0: i; else: i+1; for i in range(4)])
```

a) [0, 2, 2, 4]
b) [1, 1, 3, 3]
c) error
d) none of the mentioned

Explanation: Syntax error.

10. Which of the following is the same as list(map(lambda x: x-1, [1, 2, 3]))?**

a) [x**-1 for x in [(1, 2, 3)]]
b) [1/x for x in [(1, 2, 3)]]
c) [1/x for x in (1, 2, 3)]
d) error

Explanation: x**-1 is evaluated as (x)**(-1).

Python MCQ’s – List Comprehension – 1

1. What will be the output of the following Python code?

```
l=[1,2,3,4,5]
[x&1 for x in l]
```

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

Explanation: In the code shown above, each of the numbers of the list, that is, 1, 2, 3, 4 and 5 are AND-ed with 1 and the result is printed in the form of a list. Hence the output is [1, 0, 1, 0, 1].

2. What will be the output of the following Python code?

```
l1=[1,2,3]
l2=[4,5,6]
[x*y for x in l1 for y in l2]
```

a) [4, 8, 12, 5, 10, 15, 6, 12, 18]
b) [4, 10, 18]
c) [4, 5, 6, 8, 10, 12, 12, 15, 18]
d) [18, 12, 6, 15, 10, 5, 12, 8, 4]

Explanation: The code shown above returns x*y, where x belongs to the list l1 and y belongs to the list l2. Therefore, the output is: [4, 5, 6, 8, 10, 12, 12, 15, 18].

3. Write the list comprehension to pick out only negative integers from a given list 'l'.

a) [x<0 in l]
b) [x for x<0 in l]
c) [x in l for x<0]
d) [x for x in l if x<0]

Explanation: To pick out only the negative numbers from a given list 'l', the correct list comprehension statement would be: [x for x in l if x<0].

For example if we have a list l=[-65, 2, 7, -99, -4, 3]
>>> [x for x in l if x<0]
The output would be: [-65, -99, -4].

4. What will be the output of the following Python code?

```
s=["pune", "mumbai", "delhi"]
[(w.upper(), len(w)) for w in s]
```

a) Error
b) ['PUNE', 4, 'MUMBAI', 6, 'DELHI', 5]
c) [PUNE, 4, MUMBAI, 6, DELHI, 5]
d) [('PUNE', 4), ('MUMBAI', 6), ('DELHI', 5)]

Explanation: If we need to generate two results, we need to put it in the form of a tuple. The code shown above returns each word of list in uppercase, along with the length of the word. Hence the output of the code is: [('PUNE', 4), ('MUMBAI', 6), ('DELHI', 5)].

5. What will be the output of the following Python code?

```
l1=[2,4,6]
l2=[-2,-4,-6]
for i in zip(l1, l2):
        print(i)
```

a) 2, -2
   4, -4
   6, -6

b) [(2, -2), (4, -4), (6, -6)]

c) (2, -2)
(4, -4)
(6, -6)

d) [-4, -16, -36]

Explanation: The output of the code shown will be:
(2, -2)
(4, -4)
(6, -6)
This format is due to the statement print(i).

6. What will be the output of the following Python code?

```
l1=[10, 20, 30]
l2=[-10, -20, -30]
l3=[x+y for x, y in zip(l1, l2)]
l3
```

a) Error
b) 0
c) [-20, -60, -80]
d) [0, 0, 0]

Explanation: The code shown above returns x+y, for x belonging to the list l1 and y belonging to the list l2. That is, l3=[10-10, 20-20, 30-20], which is, [0, 0, 0].

7. Write a list comprehension for number and its cube for l=[1, 2, 3, 4, 5, 6, 7, 8, 9].

a) [x3 for x in l]**
b) [x^3 for x in l]

c) [x**3 in l]
d) [x^3 in l]

Explanation: The list comprehension to print a list of cube of the numbers for the given list is: [x**3 for x in l].

8. What will be the output of the following Python code?

```
l=[[1 ,2, 3], [4, 5, 6], [7, 8, 9]]
[[row[i] for row in l] for i in range(3)]
```

a) Error
b) [[1, 4, 7], [2, 5, 8], [3, 6, 9]]
c) 1 4 7
2 5 8
3 6 9
d) (1 4 7)
(2 5 8)
(3 6 9)

Explanation: In the code shown above, '3' is the index of the list. Had we used a number greater than 3, it would result in an error. The output of this code is: [[1, 4, 7], [2, 5, 8], [3, 6, 9]].

9. What will be the output of the following Python code?

```
import math
[str(round(math.pi)) for i in range (1, 6)]
```

a) ['3', '3', '3', '3', '3', '3']
b) ['3.1', '3.14', '3.142', '3.1416', '3.14159', '3.141582']
c) ['3', '3', '3', '3', '3']
d) ['3.1', '3.14', '3.142', '3.1416', '3.14159']

Explanation: The list comprehension shown above rounds off pi(3.141) and returns its value, that is 3. This is done 5 times. Hence the output is: ['3', '3', '3', '3', '3'].

10. What will be the output of the following Python code?

```
l1=[1,2,3]
l2=[4,5,6]
l3=[7,8,9]
for x, y, z in zip(l1, l2, l3):
        print(x, y, z)
```

a) 1 4 7
2 5 8
3 6 9
b) (1 4 7)
(2 5 8)
(3 6 9)
c) [(1, 4, 7), (2, 5, 8), (3, 6, 9)]
d) Error

Explanation: The output of the code shown above is:
1 4 7
2 5 8
3 6 9
This is due to the statement: print(x, y,z).

Python MCQ's – List Comprehension – 2

1. Read the information given below carefully and write a list comprehension such that the output is: ['e', 'o']

```
w="hello"
v=('a', 'e', 'i', 'o', 'u')
```

a) [x for w in v if x in v]
b) [x for x in w if x in v]
c) [x for x in v if w in v]
d) [x for v in w for x in w]

Explanation: The tuple 'v' is used to generate a list containing only vowels in the string 'w'. The result is a list containing only vowels present in the string "hello". Hence the required list comprehension is: [x for x in w if x in v].

2. What will be the output of the following Python code?

```
[ord(ch) for ch in 'abc']
```

a) [97, 98, 99]
b) ['97', '98', '99']
c) [65, 66, 67]
d) Error

Explanation: The list comprehension shown above returns the ASCII value of each alphabet of the string 'abc'. Hence the output is: [97, 98, 99]. Had the string been 'ABC', the output would be: [65, 66, 67].

3. What will be the output of the following Python code?

```
t=32.00
[round((x-32)*5/9) for x in t]
```

a) [0]
b) 0
c) [0.00]
d) Error

Explanation: The value of t in the code shown above is equal to 32.00, which is a floating point value. 'Float' objects are not iterable. Hence the code results in an error.

4. Write a list comprehension for producing a list of numbers between 1 and 1000 that are divisible by 3.

a) [x in range(1, 1000) if x%3==0]
b) [x for x in range(1000) if x%3==0]
c) [x%3 for x in range(1, 1000)]
d) [x%3=0 for x in range(1, 1000)]

Explanation: The list comprehension [x for x in range(1000) if x%3==0] produces a list of numbers between 1 and 1000 that are divisible by 3.

5. Write a list comprehension equivalent for the Python code shown below.

```
for i in range(1, 101):
        if int(i*0.5)==i*0.5:
                print(i)
```

a) [i for i in range(1, 100) if int(i*0.5)==(i*0.5)]
b) [i for i in range(1, 101) if int(i*0.5)==(i*0.5)]
c) [i for i in range(1, 101) if int(i*0.5)=(i*0.5)]
d) [i for i in range(1, 100) if int(i*0.5)=(i*0.5)]

Explanation: The code shown above prints the value 'i' only if it satisfies the condition: int(i*0.5) is equal to (i*0.5). Hence the required list comprehension is: [i for i in range(1, 101) if int(i*0.5)==(i*0.5)].

6. What is the list comprehension equivalent for: list(map(lambda x:x-1, [1, 2, 3]))?**

a) [1|x for x in [1, 2, 3]]
b) [-1**x for x in [1, 2, 3]]
c) [x-1 for x in [1, 2, 3]]**

d) [x^-1 for x in range(4)]

Explanation: The output of the function list(map(lambda x:x**-1, [1, 2, 3])) is [1.0, 0.5, 0.3333333333333333] and that of the list comprehension [x**-1 for x in [1, 2, 3]] is [1.0, 0.5, 0.3333333333333333]. Hence the answer is: [x**-1 for x in [1, 2, 3]].

7. Write a list comprehension to produce the list: [1, 2, 4, 8, 16……212].

a) [(2x) for x in range(0, 13)]**
b) [(x**2) for x in range(1, 13)]
c) [(2**x) for x in range(1, 13)]
d) [(x**2) for x in range(0, 13)]

Explanation: The required list comprehension will print the numbers from 1 to 12, each raised to 2. The required answer is thus, [(2**x) for x in range(0, 13)].

8. What is the list comprehension equivalent for?

{x : x is a whole number less than 20, x is even} (including zero)

a) [x for x in range(1, 20) if (x%2==0)]
b) [x for x in range(0, 20) if (x//2==0)]
c) [x for x in range(1, 20) if (x//2==0)]
d) [x for x in range(0, 20) if (x%2==0)]

Explanation: The required list comprehension will print a whole number, less than 20, provided that the number is even. Since the output list should contain zero as well, the answer to this question is: [x for x in range(0, 20) if (x%2==0)].

9. What will be the output of the following Python list comprehension?

```
[j for i in range(2,8) for j in range(i*2, 50, i)]
```

a) A list of prime numbers up to 50
b) A list of numbers divisible by 2, up to 50
c) A list of non prime numbers, up to 50
d) Error

Explanation: The list comprehension shown above returns a list of non-prime numbers up to 50. The logic behind this is that the square root of 50 is almost equal to 7. Hence all the multiples of 2-7 are not prime in this range.

10. What will be the output of the following Python code?

```
l=["good", "oh!", "excellent!", "#450"]
[n for n in l if n.isalpha() or n.isdigit()]
```

a) [‘good’, ‘oh’, ‘excellent’, ‘450’]
b) [‘good’]
c) [‘good’, ‘#450’]
d) [‘oh!’, ‘excellent!’, ‘#450’]

Explanation: The code shown above returns a new list containing only strings which do not have any punctuation in them. The only string from the list which does not contain any punctuation is ‘good’. Hence the output of the code shown above is [‘good’].

Python MCQ’s – Matrix List Comprehension

1. Which of the following matrices will throw an error in Python?
a) A = [[1, 2, 3],
[4, 5, 6],
[7, 8, 9]]

b) B = [[3, 3, 3]
[4, 4, 4]
[5, 5, 5]]

c) C = [(1, 2, 4),
(5, 6, 7),
(8, 9, 10)]

d) D = [2, 3, 4,
3, 3, 3,
4, 5, 6]

Explanation: In matrix B will result in an error because in the absence of a comma at the end of each row, it behaves like three separate lists. The error thrown states that the list integers must be integers or slices, not tuples.

2. What will be the output of the following Python code?

```
A = [[1, 2, 3],
     [4, 5, 6],
     [7, 8, 9]]
A[1]
```

a) [4, 5, 6]
b) [3, 6, 9]

c) [1, 4, 7]
d) [1, 2, 3]

Explanation: We can index the rows and columns using normal index operations. The statement A[1] represents the second row, that is, the middle row. Hence the output of the code will be: [4, 5, 6].

3. Which of the following Python statements will result in the output: 6?

```
A = [[1, 2, 3],
     [4, 5, 6],
     [7, 8, 9]]
```

a) A[2][3]
b) A[2][1]
c) A[1][2]
d) A[3][2]

Explanation: The output that is required is 6, that is, row 2, item 3. This position is represented by the statement: A[1][2].

4. What will be the output of the following Python code?

```
A = [[1, 2, 3],
      [4, 5, 6],
      [7, 8, 9]]
[A[row][1] for row in (0, 1, 2)]
```

a) [7, 8, 9]
b) [4, 5, 6]
c) [2, 5, 8]
d) [1, 4, 7]

Explanation: To get a particular column as output, we can simple iterate across the rows and pull out the desired column, or iterate through positions in rows and index as we go. Hence the output of the code shown above is: [2, 5, 8].

5. What will be the output of the following Python code?

```
A = [[1, 2, 3],
     [4, 5, 6],
     [7, 8, 9]]
[A[i][i] for i in range(len(A))]
```

a) [1, 5, 9]
b) [3, 5, 7]
c) [4, 5, 6]
d) [2, 5, 8]

Explanation: We can also perform tasks like pulling out a diagonal. The expression shown above uses range to generate the list of offsets and the indices with the row and column the same, picking out A[0][0], then A[1][1] and so on. Hence the output of the code is: [1, 5, 9].

6. What will be the output of the following Python code?

```
l=[[1, 2, 3], [4, 5, 6]]
for i in range(len(l)):
        for j in range(len(l[i])):
                l[i][j]+=10
l
```

a) No output
b) Error
c) [[1, 2, 3], [4, 5, 6]]
d) [[11, 12, 13], [14, 15, 16]]

Explanation: We use range twice if the shapes differ. Each element of list l is increased by 10. Hence the output is: [[11, 12, 13], [14, 15, 16]]

7. What will be the output of the following Python code?

```
A = [[1, 2, 3],
     [4, 5, 6],
     [7, 8, 9]]

[[col + 10 for col in row] for row in A]
```

a) [[11, 12, 13], [14, 15, 16], [17, 18, 19]]
b) Error
c) [11, 12, 13], [14, 15, 16], [17, 18, 19]
d) [11, 12, 13, 14, 15, 16, 17, 18, 19]

Explanation: The code shown above shows a list comprehension which adds 10 to each element of the matrix A and prints it row-wise. Hence the output of the code is: [[11, 12, 13], [14, 15, 16], [17, 18, 19]]

8. What will be the output of the following Python code?

```
A = [[1, 2, 3],
```

```
      [4, 5, 6],
      [7, 8, 9]]
[A[i][len(A)-1-i] for i in range(len(A))]
```

a) [1, 5, 9]
b) [4, 5, 6]
c) [3, 5, 7]
d) [2, 5, 8]

Explanation: This expression scales the common index to fetch A[0][2], A[1][1], etc. We assume the matrix has the same number of rows and columns.

9. What will be the output of the following Python code?

```
A = [[1, 2, 3],
      [4, 5, 6],
      [7, 8, 9]]
B = [[3, 3, 3],
      [4, 4, 4],
      [5, 5, 5]]
[B[row][col]*A[row][col] for row in range(3) for col in range(3)]
```

a) [3, 6, 9, 16, 20, 24, 35, 40, 45]
b) Error
c) [0, 30, 60, 120, 160, 200, 300, 350, 400]
d) 0

Explanation: In the code shown above, we have used list comprehension to combine values of multiple matrices. We have multiplied the elements of the matrix B with that of the matrix A, in the range(3). Hence the output of this code is: [3, 6, 9, 16, 20, 24, 35, 40, 45].

10. What will be the output of the following Python code?

```
r = [11, 12, 13, 14, 15, 16, 17, 18, 19]
A = [[0, 10, 20],
               [30, 40, 50],
               [60, 70, 80]]
for row in A:
        for col in row:
               r.append(col+10)
r
```

a) [11, 12, 13, 14, 15, 16, 17, 18, 19, 10, 20, 30, 40, 50, 60, 70, 80, 90]
b) [10, 20, 30, 40, 50, 60, 70, 80, 90]
c) [11, 12, 13, 14, 15, 16, 17, 18, 19]

d) [0, 10, 20, 30, 40, 50, 60, 70, 80]

Explanation: The code shown above adds 10 to each element of the matrix and prints the output row-wise. Since the list l already contains some elements, the new elements are appended to it. Hence the output of this code is: [11, 12, 13, 14, 15, 16, 17, 18, 19, 10, 20, 30, 40, 50, 60, 70, 80, 90].

11. What will be the output of the following Python code?

```
A = [[1, 2, 3],
     [4, 5, 6],
     [7, 8, 9]]
B = [[3, 3, 3],
     [4, 4, 4],
     [5, 5, 5]]
[[col1 * col2 for (col1, col2) in zip(row1, row2)] for (row1, row2) in zip(A,
B)]
```

a) [0, 30, 60, 120, 160, 200, 300, 350, 400]
b) [[3, 6, 9], [16, 20, 24], [35, 40, 45]]
c) No output
d) Error

Explanation: The list comprehension shown above results in the output: [[3, 6, 9], [16, 20, 24], [35, 40, 45]].

12. What will be the output of the following Python code?

```
A = [[1, 2, 3],
     [4, 5, 6],
     [7, 8, 9]]
B = [[3, 3, 3],
     [4, 4, 4],
     [5, 5, 5]]
zip(A, B)
```

a) Address of the zip object
b) Address of the matrices A and B
c) No output
d) [3, 6, 9, 16, 20, 24, 35, 40, 45]

Explanation: The output of the code shown above returns the address of the zip object. If we print it in the form of a list, we get:
>>> list(zip(A, B))
[([1, 2, 3], [3, 3, 3]), ([4, 5, 6], [4, 4, 4]), ([7, 8, 9], [5, 5, 5])]

Python MCQ's – Tuples – 1

1. Which of the following is a Python tuple?
a) [1, 2, 3]
b) (1, 2, 3)
c) {1, 2, 3}
d) {}

Explanation: Tuples are represented with round brackets.

2. 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))

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

3. What will be the output of the following Python code?

```
1.>>>t=(1,2,4,3)
2.>>>t[1:3]
```

a) (1, 2)
b) (1, 2, 4)
c) (2, 4)
d) (2, 4, 3)

Explanation: Slicing in tuples takes place just as it does in strings.

4. What will be the output of the following Python code?

```
1.>>>t=(1,2,4,3)
2.>>>t[1:-1]
```

a) (1, 2)
b) (1, 2, 4)
c) (2, 4)
d) (2, 4, 3)

Explanation: Slicing in tuples takes place just as it does in strings.

5. What will be the output of the following Python code?

```
1.>>>t = (1, 2, 4, 3, 8, 9)
2.>>>[t[i] for i in range(0, len(t), 2)]
```

a) [2, 3, 9]
b) [1, 2, 4, 3, 8, 9]
c) [1, 4, 8]
d) (1, 4, 8)

Explanation: Execute in the shell to verify.

6. What will be the output of the following Python code?

```
1.d = {"john":40, "peter":45}
2.d["john"]
```

a) 40
b) 45
c) “john”
d) “peter”

Explanation: Execute in the shell to verify.

7. What will be the output of the following Python code?

```
1.>>>t = (1, 2)
2.>>>2 * t
```

a) (1, 2, 1, 2)
b) [1, 2, 1, 2]
c) (1, 1, 2, 2)
d) [1, 1, 2, 2]

Explanation: * operator concatenates tuple.

8. What will be the output of the following Python code?

```
1.>>>t1 = (1, 2, 4, 3)
2.>>>t2 = (1, 2, 3, 4)
3.>>>t1 < t2
```

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

Explanation: Elements are compared one by one in this case.

9. What will be the output of the following Python code?

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

a) 1
b) 2
c) 5
d) Error

Explanation: Tuples are immutable and don't have an append method. An exception is thrown in this case.

10. What will be the output of the following Python code?

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

a) 30
b) 24
c) 33
d) 12

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.

Python MCQ's – Tuples – 2

1. What is the data type of (1)?
a) Tuple
b) Integer
c) List
d) Both tuple and integer

Explanation: A tuple of one element must be created as (1,).

2. If a=(1,2,3,4), a[1:-1] is _____
a) Error, tuple slicing doesn't exist
b) [2,3]
c) (2,3,4)
d) (2,3)

Explanation: Tuple slicing exists and a[1:-1] returns (2,3).

3. What will be the output of the following Python code?

```
>>> a=(1,2,(4,5))
>>> b=(1,2,(3,4))
>>> a<b
```

a) False
b) True
c) Error, < operator is not valid for tuples
d) Error, < operator is valid for tuples but not if there are sub-tuples

Explanation: Since the first element in the sub-tuple of a is larger that the first element in the subtuple of b, False is printed.

4. What will be the output of the following Python code?

```
>>> a=("Check")*3
>>> a
```

a) ('Check','Check','Check')
b) * Operator not valid for tuples
c) ('CheckCheckCheck')
d) Syntax error

Explanation: Here ("Check") is a string not a tuple because there is no comma after the element.

5. What will be the output of the following Python code?

```
>>> a=(1,2,3,4)
>>> del(a[2])
```

a) Now, a=(1,2,4)
b) Now, a=(1,3,4)
c) Now a=(3,4)

d) Error as tuple is immutable

Explanation: ‘tuple’ object doesn’t support item deletion.

6. What will be the output of the following Python code?

```
>>> a=(2,3,4)
>>> sum(a,3)
```

a) Too many arguments for sum() method
b) The method sum() doesn’t exist for tuples
c) 12
d) 9

Explanation: In the above case, 3 is the starting value to which the sum of the tuple is added to.

7. Is the following Python code valid?

```
>>> a=(1,2,3,4)
>>> del a
```

a) No because tuple is immutable
b) Yes, first element in the tuple is deleted
c) Yes, the entire tuple is deleted
d) No, invalid syntax for del method

Explanation: The command del a deletes the entire tuple.

8. What type of data is: a=[(1,1),(2,4),(3,9)]?
a) Array of tuples
b) List of tuples
c) Tuples of lists
d) Invalid type

Explanation: The variable a has tuples enclosed in a list making it a list of tuples.

9. What will be the output of the following Python code?

```
>>> a=(0,1,2,3,4)
>>> b=slice(0,2)
>>> a[b]
```

a) Invalid syntax for slicing
b) [0,2]
c) (0,1)
d) (0,2)

Explanation: The method illustrated in the above piece of code is that of naming of slices.

10. Is the following Python code valid?

```
>>> a=(1,2,3)
>>> b=('A','B','C')
>>> c=tuple(zip(a,b))
```

a) Yes, c will be ((1, ‘A’), (2, ‘B’), (3, ‘C’))
b) Yes, c will be ((1,2,3),(‘A’,’B’,’C’))
c) No because tuples are immutable
d) No because the syntax for zip function isn’t valid

Explanation: Zip function combines individual elements of two iterables into tuples. Execute in Python shell to verify.

Python MCQ’s – Tuples-3

1. Is the following Python code valid?

```
>>> a,b,c=1,2,3
>>> a,b,c
```

a) Yes, [1,2,3] is printed
b) No, invalid syntax
c) Yes, (1,2,3) is printed
d) 1 is printed

Explanation: A tuple needn’t be enclosed in parenthesis.

2. What will be the output of the following Python code?

```
a = ('check',)
n = 2
for i in range(int(n)):
    a = (a,)
    print(a)
```

a) Error, tuples are immutable

b) (('check',),)
(((('check',),),),)

c) (('check',)'check',)

d) (('check',)'check',)
(((('check',)'check',)'check',)

Explanation: The loop runs two times and each time the loop runs an extra parenthesis along with a comma is added to the tuple (as a=(a')).

3. Is the following Python code valid?

```
>>> a,b=1,2,3
```

a) Yes, this is an example of tuple unpacking. a=1 and b=2
b) Yes, this is an example of tuple unpacking. a=(1,2) and b=3
c) No, too many values to unpack
d) Yes, this is an example of tuple unpacking. a=1 and b=(2,3)

Explanation: For unpacking to happen, the number of values of the right hand side must be equal to the number of variables on the left hand side.

4. What will be the output of the following Python code?

```
>>> a=(1,2)
>>> b=(3,4)
>>> c=a+b
>>> c
```

a) (4,6)
b) (1,2,3,4)
c) Error as tuples are immutable
d) None

Explanation: In the above piece of code, the values of the tuples aren't being changed. Both the tuples are simply concatenated.

5. What will be the output of the following Python code?

```
>>> a,b=6,7
```

```
>>> a,b=b,a
>>> a,b
```

a) (6,7)
b) Invalid syntax
c) (7,6)
d) Nothing is printed

Explanation: The above piece of code illustrates the unpacking of variables.

6. What will be the output of the following Python code?

```
>>> import collections
>>> a=collections.namedtuple('a',['i','j'])
>>> obj=a(i=4,j=7)
>>> obj
```

a) a(i=4, j=7)
b) obj(i=4, j=7)
c) (4,7)
d) An exception is thrown

Explanation: The above piece of code illustrates the concept of named tuples.

7. Tuples can’t be made keys of a dictionary.
a) True
b) False

Explanation: Tuples can be made keys of a dictionary because they are hashable.

8. Is the following Python code valid?

```
>>> a=2,3,4,5
>>> a
```

a) Yes, 2 is printed
b) Yes, [2,3,4,5] is printed
c) No, too many values to unpack
d) Yes, (2,3,4,5) is printed

Explanation: A tuple needn’t be enclosed in parenthesis.

9. What will be the output of the following Python code?

```
>>> a=(2,3,1,5)
>>> a.sort()
>>> a
```

a) (1,2,3,5)
b) (2,3,1,5)
c) None
d) Error, tuple has no attribute sort

Explanation: A tuple is immutable thus it doesn't have a sort attribute.

10. Is the following Python code valid?

```
>>> a=(1,2,3)
>>> b=a.update(4,)
```

a) Yes, a=(1,2,3,4) and b=(1,2,3,4)
b) Yes, a=(1,2,3) and b=(1,2,3,4)
c) No because tuples are immutable
d) No because wrong syntax for update() method

Explanation: Tuple doesn't have any update() attribute because it is immutable.

11. What will be the output of the following Python code?

```
>>> a=[(2,4),(1,2),(3,9)]
>>> a.sort()
>>> a
```

a) [(1, 2), (2, 4), (3, 9)]
b) [(2,4),(1,2),(3,9)]
c) Error because tuples are immutable
d) Error, tuple has no sort attribute

Explanation: A list of tuples is a list itself. Hence items of a list can be sorted.

Python MCQ's – Sets – 1

1. Which of these about a set is not true?
a) Mutable data type

b) Allows duplicate values
c) Data type with unordered values
d) Immutable data type

Explanation: A set is a mutable data type with non-duplicate, unordered values, providing the usual mathematical set operations.

2. Which of the following is not the correct syntax for creating a set?
a) set([[1,2],[3,4]])
b) set([1,2,2,3,4])
c) set((1,2,3,4))
d) {1,2,3,4}

Explanation: The argument given for the set must be an iterable.

3. What will be the output of the following Python code?

```
nums = set([1,1,2,3,3,3,4,4])
print(len(nums))
```

a) 7
b) Error, invalid syntax for formation of set
c) 4
d) 8

Explanation: A set doesn't have duplicate items.

4. What will be the output of the following Python code?

```
a = [5,5,6,7,7,7]
b = set(a)
def test(lst):
    if lst in b:
        return 1
    else:
        return 0
for i in  filter(test, a):
    print(i,end=" ")
```

a) 5 5 6
b) 5 6 7
c) 5 5 6 7 7 7
d) 5 6 7 7 7

Explanation: The filter function will return all the values from list a which are true when passed to function test. Since all the members of the set are non-duplicate members of the list, all of the values will return true. Hence all the values in the list are printed.

5. Which of the following statements is used to create an empty set?
a) { }
b) set()
c) [ ]
d) ( )

Explanation: { } creates a dictionary not a set. Only set() creates an empty set.

6. What will be the output of the following Python code?

```
>>> a={5,4}
>>> b={1,2,4,5}
>>> a<b
```

a) {1,2}
b) True
c) False
d) Invalid operation

Explanation: a<b returns True if a is a proper subset of b.

7. If a={5,6,7,8}, which of the following statements is false?
a) print(len(a))
b) print(min(a))
c) a.remove(5)
d) a[2]=45

Explanation: The members of a set can be accessed by their index values since the elements of the set are unordered.

8. If a={5,6,7}, what happens when a.add(5) is executed?
a) a={5,5,6,7}
b) a={5,6,7}
c) Error as there is no add function for set data type
d) Error as 5 already exists in the set

Explanation: There exists add method for set data type. However 5 isn't added again as set consists of only non-duplicate elements and 5 already exists in the set. Execute in python shell to verify.

9. What will be the output of the following Python code?

```
>>> a={4,5,6}
>>> b={2,8,6}
>>> a+b
```

a) {4,5,6,2,8}
b) {4,5,6,2,8,6}
c) Error as unsupported operand type for sets
d) Error as the duplicate item 6 is present in both sets

Explanation: Execute in python shell to verify.

10. What will be the output of the following Python code?

```
>>> a={4,5,6}
>>> b={2,8,6}
>>> a-b
```

a) {4,5}
b) {6}
c) Error as unsupported operand type for set data type
d) Error as the duplicate item 6 is present in both sets

Explanation: – operator gives the set of elements in set a but not in set b.

11. What will be the output of the following Python code?

```
>>> a={5,6,7,8}
>>> b={7,8,10,11}
>>> a^b
```

a) {5,6,7,8,10,11}
b) {7,8}
c) Error as unsupported operand type of set data type
d) {5,6,10,11}

Explanation: ^ operator returns a set of elements in set A or set B, but not in both (symmetric difference).

12. What will be the output of the following Python code?

```
>>> s={5,6}
```

```
>>> s*3
```

a) Error as unsupported operand type for set data type
b) {5,6,5,6,5,6}
c) {5,6}
d) Error as multiplication creates duplicate elements which isn't allowed

Explanation: The multiplication operator isn't valid for the set data type.

13. What will be the output of the following Python code?

```
>>> a={5,6,7,8}
>>> b={7,5,6,8}
>>> a==b
```

a) True
b) False

Explanation: It is possible to compare two sets and the order of elements in both the sets doesn't matter if the values of the elements are the same.

14. What will be the output of the following Python code?

```
>>> a={3,4,5}
>>> b={5,6,7}
>>> a|b
```

a) Invalid operation
b) {3, 4, 5, 6, 7}
c) {5}
d) {3,4,6,7}

Explanation: The operation in the above piece of code is union operation. This operation produces a set of elements in both set a and set b.

15. Is the following Python code valid?

```
a={3,4,{7,5}}
print(a[2][0])
```

a) Yes, 7 is printed
b) Error, elements of a set can't be printed
c) Error, subsets aren't allowed

d) Yes, {7,5} is printed

Explanation: In python, elements of a set must not be mutable and sets are mutable. Thus, subsets can't exist.

Python MCQ's – Sets – 2

1. Which of these about a frozenset is not true?
a) Mutable data type
b) Allows duplicate values
c) Data type with unordered values
d) Immutable data type

Explanation: A frozenset is an immutable data type.

2. What is the syntax of the following Python code?

```
>>> a=frozenset(set([5,6,7]))
>>> a
```

a) {5,6,7}
b) frozenset({5,6,7})
c) Error, not possible to convert set into frozenset
d) Syntax error

Explanation: The above piece of code is the correct syntax for creating a frozenset.

3. Is the following Python code valid?

```
>>> a=frozenset([5,6,7])
>>> a
>>> a.add(5)
```

a) Yes, now a is {5,5,6,7}
b) No, frozen set is immutable
c) No, invalid syntax for add method
d) Yes, now a is {5,6,7}

Explanation: Since a frozen set is immutable, add method doesn't exist for frozen method.

4. Set members must not be hashable.
a) True
b) False

Explanation: Set members must always be hashable.

5. What will be the output of the following Python code?

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

a) Error, no method called update for set data type
b) {1, 2, 3, 4, 5}
c) Error, list can't be added to set
d) Error, duplicate item present in list

Explanation: The method update adds elements to a set.

6. What will be the output of the following Python code?

```
>>> a={1,2,3}
>>> a.intersection_update({2,3,4,5})
>>> a
```

a) {2,3}
b) Error, duplicate item present in list
c) Error, no method called intersection_update for set data type
d) {1,4,5}

Explanation: The method intersection_update returns a set which is an intersection of both the sets.

7. What will be the output of the following Python code?

```
>>> a={1,2,3}
>>> b=a
>>> b.remove(3)
>>> a
```

a) {1,2,3}
b) Error, copying of sets isn't allowed
c) {1,2}

d) Error, invalid syntax for remove

Explanation: Any change made in b is reflected in a because b is an alias of a.

8. What will be the output of the following Python code?

```
>>> a={1,2,3}
>>> b=a.copy()
>>> b.add(4)
>>> a
```

a) {1,2,3}
b) Error, invalid syntax for add
c) {1,2,3,4}
d) Error, copying of sets isn't allowed

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

9. What will be the output of the following Python code?

```
>>> a={1,2,3}
>>> b=a.add(4)
>>> b
```

a) 0
b) {1,2,3,4}
c) {1,2,3}
d) Nothing is printed

Explanation: The method add returns nothing, hence nothing is printed.

10. What will be the output of the following Python code?

```
>>> a={1,2,3}
>>> b=frozenset([3,4,5])
>>> a-b
```

a) {1,2}
b) Error as difference between a set and frozenset can't be found out
c) Error as unsupported operand type for set data type
d) frozenset({1,2})

Explanation: – operator gives the set of elements in set a but not in set b.

11. What will be the output of the following Python code?

```
>>> a={5,6,7}
>>> sum(a,5)
```

a) 5
b) 23
c) 18
d) Invalid syntax for sum method, too many arguments

Explanation: The second parameter is the start value for the sum of elements in set a. Thus, sum(a,5) = 5+(5+6+7)=23.

12. What will be the output of the following Python code?

```
>>> a={1,2,3}
>>> {x*2 for x in a|{4,5}}
```

a) {2,4,6}
b) Error, set comprehensions aren't allowed
c) {8, 2, 10, 4, 6}
d) {8,10}

Explanation: Set comprehensions are allowed.

13. What will be the output of the following Python code?

```
>>> a={5,6,7,8}
>>> b={7,8,9,10}
>>> len(a+b)
```

a) 8
b) Error, unsupported operand '+' for sets
c) 6
d) Nothing is displayed

Explanation: Duplicate elements in a+b is eliminated and the length of a+b is computed.

14. What will be the output of the following Python code?

```
a={1,2,3}
b={1,2,3}
c=a.issubset(b)
```

```
print(c)
```

a) True
b) Error, no method called issubset() exists
c) Syntax error for issubset() method
d) False

Explanation: The method issubset() returns True if b is a proper subset of a.

15. Is the following Python code valid?

```
a={1,2,3}
b={1,2,3,4}
c=a.issuperset(b)
print(c)
```

a) False
b) True
c) Syntax error for issuperset() method
d) Error, no method called issuperset() exists

Explanation: The method issubset() returns True if b is a proper subset of a.

Python MCQ's – Sets – 3

1. What will be the output of the following Python code?

```
s=set()
type(s)
```

a) <'set'>
b) <class 'set'>
c) set
d) class set

Explanation: When we find the type of a set, the output returned is: .

2. The following Python code results in an error.

```
s={2, 3, 4, [5, 6]}
```

a) True
b) False

Explanation: The set data type makes use of a principle known as hashing. This means that each item in the set should be hashable. Hashable in this context means immutable. List is mutable and hence the line of code shown above will result in an error.

3. Set makes use of _____
Dictionary makes use of _____
a) keys, keys
b) key values, keys
c) keys, key values
d) key values, key values

Explanation: Set makes use of keys.
Dictionary makes use of key values.

4. Which of the following lines of code will result in an error?
a) s={abs}
b) s={4, 'abc', (1,2)}
c) s={2, 2.2, 3, 'xyz'}
d) s={san}

Explanation: The line: s={san} will result in an error because 'san' is not defined. The line s={abs} does not result in an error because abs is a built-in function. The other sets shown do not result in an error because all the items are hashable.

5. What will be the output of the following Python code?

```
s={2, 5, 6, 6, 7}
s
```

a) {2, 5, 7}
b) {2, 5, 6, 7}
c) {2, 5, 6, 6, 7}
d) Error

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. Therefore the output is: {2, 5, 6, 7}

6. Input order is preserved in sets.
a) True

b) False

Explanation: The input order in sets is not maintained. This is demonstrated by the code shown below:

```
>>> s={2, 6, 8, 1, 5}
>>> s
{8, 1, 2, 5, 6}
```

7. Write a list comprehension for number and its cube for:

```
l=[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

a) [x3 for x in l]**
b) [x^3 for x in l]
c) [x**3 in l]
d) [x^3 in l]

Explanation: The list comprehension to print a list of cube of the numbers for the given list is: [x**3 for x in l].

8. What will be the output of the following Python code?

```
s={1, 2, 3}
s.update(4)
s
```

a) {1, 2, 3, 4}
b) {1, 2, 4, 3}
c) {4, 1, 2, 3}
d) Error

Explanation: The code shown above will result in an error because the argument given to the function update should necessarily be an iterable. Hence if we write this function as: s.update([4]), there will be no error.

9. Which of the following functions cannot be used on heterogeneous sets?
a) pop
b) remove
c) update
d) sum

Explanation: The functions sum, min and max cannot be used on mixed type (heterogeneous) sets. The functions pop, remove, update etc can be used on homogenous as well as heterogeneous sets. An example of heterogeneous sets is: {'abc', 4, (1, 2)}

10. What will be the output of the following Python code?

```
s={4>3, 0, 3-3}
all(s)
any(s)
```

a) True
False

b) False
True

c) True
True

d) False
False

Explanation: The function all returns true only if all the conditions given are true. But in the example shown above, we have 0 as a value. Hence false is returned. Similarly, any returns true if any one condition is true. Since the condition 4>3 is true, true is returned.

Python MCQ's – Sets – 4

1. Which of the following functions will return the symmetric difference between two sets, x and y?
a) x | y
b) x ^ y
c) x & y
d) x – y

Explanation: The function x ^ y returns the symmetric difference between the two sets x and y. This is basically an XOR operation being performed on the two sets.

2. What will be the output of the following Python code snippet?

```
z=set('abc$de')
'a' in z
```

a) True
b) False
c) No output
d) Error

Explanation: The code shown above is used to check whether a particular item is a part of a given set or not. Since ‘a’ is a part of the set z, the output is true. Note that this code would result in an error in the absence of the quotes.

3. What will be the output of the following Python code snippet?

```
z=set('abc')
z.add('san')
z.update(set(['p', 'q']))
z
```

a) {‘abc’, ‘p’, ‘q’, ‘san’}
b) {‘a’, ‘b’, ‘c’, [‘p’, ‘q’], ‘san}
c) {‘a’, ‘c’, ‘c’, ‘p’, ‘q’, ‘s’, ‘a’, ‘n’}
d) {‘a’, ‘b’, ‘c’, ‘p’, ‘q’, ‘san’}

Explanation: The code shown first adds the element ‘san’ to the set z. The set z is then updated and two more elements, namely, ‘p’ and ‘q’ are added to it. Hence the output is: {‘a’, ‘b’, ‘c’, ‘p’, ‘q’, ‘san’}

4. What will be the output of the following Python code snippet?

```
s=set([1, 2, 3])
s.union([4, 5])
s|([4, 5])
```

a) {1, 2, 3, 4, 5}
{1, 2, 3, 4, 5}

b) Error
{1, 2, 3, 4, 5}

c) {1, 2, 3, 4, 5}
Error

d) Error
Error

Explanation: The first function in the code shown above returns the set {1, 2, 3, 4, 5}. This is because the method of the function union allows any iterable. However the second function results in an error because of unsupported data type, that is list and set.

5. What will be the output of the following Python code snippet?

```
for x in set('pqr'):
        print(x*2)
```

a) pp
qq
rr

b) pqr
pqr

c) ppqqrr

d) pqrpqr

Explanation: The code shown above prints each element of the set twice separately. Hence the output of this code is:
pp
qq
rr

6. What will be the output of the following Python code snippet?

```
{a**2 for a in range(4)}
```

a) {1, 4, 9, 16}
b) {0, 1, 4, 9, 16}
c) Error
d) {0, 1, 4, 9}

Explanation: The code shown above returns a set containing the square of values in the range 0-3, that is 0, 1, 2 and 3. Hence the output of this line of code is: {0, 1, 4, 9}.

7. What will be the output of the following Python function?

```
{x for x in 'abc'}
{x*3 for x in 'abc'}
```

a) {abc}
aaa
bbb
ccc

b) abc
abc abc abc

c) {‘a’, ‘b’, ‘c’}
{‘aaa’, ‘bbb’, ‘ccc’}

d) {‘a’, ‘b’, ‘c’}
abc
abc
abc

Explanation: The first function prints each element of the set separately, hence the output is: {‘a’, ‘b’, ‘c’}. The second function prints each element of the set thrice, contained in a new set. Hence the output of the second function is: {‘aaa’, ‘bbb’, ‘ccc’}. (Note that the order may not be the same)

8. The output of the following code is: class<’set’>.

```
type({})
```

a) True
b) False

Explanation: The output of the line of code shown above is: class<’dict’>. This is because {} represents an empty dictionary, whereas set() initializes an empty set. Hence the statement is false.

9. What will be the output of the following Python code snippet?

```
a=[1, 4, 3, 5, 2]
b=[3, 1, 5, 2, 4]
a==b
set(a)==set(b)
```

a) True
False

b) False
False
c) False
True
d) True
True

Explanation: In the code shown above, when we check the equality of the two lists, a and b, we get the output false. This is because of the difference in the order of elements of the two lists. However, when these lists are converted to sets and checked for equality, the output is true. This is known as order-neutral equality. Two sets are said to be equal if and only if they contain exactly the same elements, regardless of order.

10. What will be the output of the following Python code snippet?

```
l=[1, 2, 4, 5, 2, 'xy', 4]
set(l)
l
```

a) {1, 2, 4, 5, 2, ‘xy’, 4}
[1, 2, 4, 5, 2, ‘xy’, 4]

b) {1, 2, 4, 5, ‘xy’}
[1, 2, 4, 5, 2, ‘xy’, 4]

c) {1, 5, ‘xy’}
[1, 5, ‘xy’]

d) {1, 2, 4, 5, ‘xy’}
[1, 2, 4, 5, ‘xy’]

Explanation: In the code shown above, the function set(l) converts the given list into a set. When this happens, all the duplicates are automatically removed. Hence the output is: {1, 2, 4, 5, ‘xy’}. On the other hand, the list l remains unchanged. Therefore the output is: [1, 2, 4, 5, 2, ‘xy’, 4].
Note that the order of the elements may not be the same.

Python MCQ’s – Sets – 5

1. What will be the output of the following Python code?

```
s1={3, 4}
s2={1, 2}
s3=set()
i=0
j=0
for i in s1:
    for j in s2:
        s3.add((i,j))
        i+=1
        j+=1
print(s3)
```

a) {(3, 4), (1, 2)}
b) Error
c) {(4, 2), (3, 1), (4, 1), (5, 2)}
d) {(3, 1), (4, 2)}

Explanation: The code shown above finds the Cartesian product of the two sets, s1 and s2. The Cartesian product of these two sets is stored in a third set, that is, s3. Hence the output of this code is: {(4, 2), (3, 1), (4, 1), (5, 2)}.

2. The _____ function removes the first element of a set and the last element of a list.
a) remove
b) pop
c) discard
d) dispose

Explanation: The function pop removes the first element when used on a set and the last element when used to a list.

3. The difference between the functions discard and remove is that:
a) Discard removes the last element of the set whereas remove removes the first element of the set
b) Discard throws an error if the specified element is not present in the set whereas remove does not throw an error in case of absence of the specified element
c) Remove removes the last element of the set whereas discard removes the first element of the set
d) Remove throws an error if the specified element is not present in the set whereas discard does not throw an error in case of absence of the specified element

Explanation: The function remove removes the element if it is present in the set. If the element is not present, it throws an error. The function discard removes the element if it is present in the set. If the element is not present, no action is performed (Error is not thrown).

4. What will be the output of the following Python code?

```
s1={1, 2, 3}
s2={3, 4, 5, 6}
s1.difference(s2)
s2.difference(s1)
```

a) {1, 2}
{4, 5, 6}

b) {1, 2}
{1, 2}

c) {4, 5, 6}
{1, 2}

d) {4, 5, 6}
{4, 5, 6}

Explanation: The function s1.difference(s2) returns a set containing the elements which are present in the set s1 but not in the set s2. Similarly, the function s2.difference(s1) returns a set containing elements which are present in the set s2 but not in the set s1. Hence the output of the code shown above will be:
{1, 2}
{4, 5, 6}.

5. What will be the output of the following Python code?

```
s1={1, 2, 3}
s2={4, 5, 6}
s1.isdisjoint(s2)
s2.isdisjoint(s1)
```

a) True
False

b) False
True

c) True
True

d) False
False

Explanation: The function isdisjoint returns true the two sets in question are disjoint, that is if they do not have even a single element in common. The two sets s1 and s2 do not have any elements in common, hence true is returned in both the cases.

6. If we have two sets, s1 and s2, and we want to check if all the elements of s1 are present in s2 or not, we can use the function:

a) s2.issubset(s1)
b) s2.issuperset(s1)
c) s1.issuperset(s2)
d) s1.isset(s2)

Explanation: Since we are checking whether all the elements present in the set s1 are present in the set s2. This means that s1 is the subset and s1 is the superset. Hence the function to be used is: s2.issuperset(s1). This operation can also be performed by the function: s1.issubset(s2).

7. What will be the output of the following Python code?

```
s1={1, 2, 3, 8}
s2={3, 4, 5, 6}
s1|s2
s1.union(s2)
```

a) {3}
{1, 2, 3, 4, 5, 6, 8}

b) {1, 2, 4, 5, 6, 8}
{1, 2, 4, 5, 6, 8}

c) {3}
{3}

d) {1, 2, 3, 4, 5, 6, 8}
{1, 2, 3, 4, 5, 6, 8}

Explanation: The function s1|s2 as well as the function s1.union(s2) returns a union of the two sets s1 and s2. Hence the output of both of these functions is: {1, 2, 3, 4, 5, 6, 8}.

8. What will be the output of the following Python code?

```
a=set('abc')
b=set('def')
```

```
b.intersection_update(a)
a
b
```

a) set()
‘e’, ‘d’, ‘f’}

b) {}
{}

c) {‘b’, ‘c’, ‘a’}
set()

d) set()
set()

Explanation: The function b.intersection_update(a) puts those elements in the set b which are common to both the sets a and b. The set a remains as it is. Since there are no common elements between the sets a and b, the output is:
‘b’, ‘c’, ‘a’}
set().

9. What will be the output of the following Python code, if s1= {1, 2, 3}?

```
s1.issubset(s1)
```

a) True
b) Error
c) No output
d) False

Explanation: Every set is a subset of itself and hence the output of this line of code is true.

10. What will be the output of the following Python code?

```
x=set('abcde')
y=set('xyzbd')
x.difference_update(y)
x
y
```

a) {‘a’, ‘b’, ‘c’, ‘d’, ‘e’}
{‘x’, ‘y’, ‘z’}

b) {‘a’, ‘c’, ‘e’}
{‘x’, ‘y’, ‘z’, ‘b’, ‘d’}

c) {‘b’, ‘d’}
{‘b’, ‘d’}

d) {‘a’, ‘c’, ‘e’}
{‘x’, ‘y’, ‘z’}

Explanation: The function x.difference_update(y) removes all the elements of the set y from the set x. Hence the output of the code is:
{‘a’, ‘c’, ‘e’}
{‘x’, ‘y’, ‘z’, ‘b’, ‘d’}.

Python MCQ’s – Dictionary – 1

1. Which of the following statements create a dictionary?
a) d = {}
b) d = {“john”:40, “peter”:45}
c) d = {40:”john”, 45:”peter”}
d) All of the mentioned

Explanation: Dictionaries are created by specifying keys and values.

2. What will be the output of the following Python code snippet?

```
1.d = {"john":40, "peter":45}
```

a) “john”, 40, 45, and “peter”
b) “john” and “peter”
c) 40 and 45
d) d = (40:”john”, 45:”peter”)

Explanation: Dictionaries appear in the form of keys and values.

3. What will be the output of the following Python code snippet?

```
1.d = {"john":40, "peter":45}
2."john" in d
```

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

Explanation: In can be used to check if the key is int dictionary.

4. What will be the output of the following Python code snippet?

```
1.d1 = {"john":40, "peter":45}
2.d2 = {"john":466, "peter":45}
3.d1 == d2
```

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

Explanation: If d2 was initialized as d2 = d1 the answer would be true.

5. What will be the output of the following Python code snippet?

```
1.d1 = {"john":40, "peter":45}
2.d2 = {"john":466, "peter":45}
3.d1 > d2
```

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

Explanation: Arithmetic > operator cannot be used with dictionaries.

6. What will be the output of the following Python code snippet?

```
1.d = {"john":40, "peter":45}
2.d["john"]
```

a) 40
b) 45
c) “john”

d) “peter”

Explanation: Execute in the shell to verify.

7. Suppose d = {“john”:40, “peter”:45}, to delete the entry for “john” what command do we use?
a) d.delete(“john”:40)
b) d.delete(“john”)
c) del d[“john”]
d) del d(“john”:40)

Explanation: Execute in the shell to verify.

8. Suppose d = {“john”:40, “peter”:45}. To obtain the number of entries in dictionary which command do we use?
a) d.size()
b) len(d)
c) size(d)
d) d.len()

Explanation: Execute in the shell to verify.

9. What will be the output of the following Python code snippet?

```
1.d = {"john":40, "peter":45}
2.print(list(d.keys()))
```

a) [“john”, “peter”]
b) [“john”:40, “peter”:45]
c) (“john”, “peter”)
d) (“john”:40, “peter”:45)

Explanation: The output of the code shown above is a list containing only keys of the dictionary d, in the form of a list.

10. Suppose d = {“john”:40, “peter”:45}, what happens when we try to retrieve a value using the expression d[“susan”]?

a) Since “susan” is not a value in the set, Python raises a KeyError exception
b) It is executed fine and no exception is raised, and it returns None
c) Since “susan” is not a key in the set, Python raises a KeyError exception
d) Since “susan” is not a key in the set, Python raises a syntax error

Explanation: Execute in the shell to verify.

Python MCQ's – Dictionary – 2

1. Which of these about a dictionary is false?
a) The values of a dictionary can be accessed using keys
b) The keys of a dictionary can be accessed using values
c) Dictionaries aren't ordered
d) Dictionaries are mutable

Explanation: The values of a dictionary can be accessed using keys but the keys of a dictionary can't be accessed using values.

2. Which of the following is not a declaration of the dictionary?
a) {1: 'A', 2: 'B'}
b) dict([[1,"A"],[2,"B"]])
c) {1,"A",2"B"}
d) { }

Explanation: Option c is a set, not a dictionary.

3. What will be the output of the following Python code snippet?

```
a={1:"A",2:"B",3:"C"}
for i,j in a.items():
    print(i,j,end=" ")
```

a) 1 A 2 B 3 C
b) 1 2 3
c) A B C
d) 1:"A" 2:"B" 3:"C"

Explanation: In the above code, variables i and j iterate over the keys and values of the dictionary respectively.

4. What will be the output of the following Python code snippet?

```
a={1:"A",2:"B",3:"C"}
print(a.get(1,4))
```

a) 1
b) A
c) 4
d) Invalid syntax for get method

Explanation: The get() method returns the value of the key if the key is present in the dictionary and the default value(second parameter) if the key isn't present in the dictionary.

5. What will be the output of the following Python code snippet?

```
a={1:"A",2:"B",3:"C"}
print(a.get(5,4))
```

a) Error, invalid syntax
b) A
c) 5
d) 4

Explanation: The get() method returns the default value(second parameter) if the key isn't present in the dictionary.

6. What will be the output of the following Python code snippet?

```
a={1:"A",2:"B",3:"C"}
print(a.setdefault(3))
```

a) {1: 'A', 2: 'B', 3: 'C'}
b) C
c) {1: 3, 2: 3, 3: 3}
d) No method called setdefault() exists for dictionary

Explanation: setdefault() is similar to get() but will set dict[key]=default if key is not already in the dictionary.

7. What will be the output of the following Python code snippet?

```
a={1:"A",2:"B",3:"C"}
a.setdefault(4,"D")
print(a)
```

a) {1: 'A', 2: 'B', 3: 'C', 4: 'D'}
b) None
c) Error
d) [1,3,6,10]

Explanation: setdefault() will set dict[key]=default if key is not already in the dictionary.

8. What will be the output of the following Python code?

```
a={1:"A",2:"B",3:"C"}
b={4:"D",5:"E"}
a.update(b)
print(a)
```

a) {1: ‘A’, 2: ‘B’, 3: ‘C’}
b) Method update() doesn’t exist for dictionaries
c) {1: ‘A’, 2: ‘B’, 3: ‘C’, 4: ‘D’, 5: ‘E’}
d) {4: ‘D’, 5: ‘E’}

Explanation: update() method adds dictionary b’s key-value pairs to dictionary a. Execute in python shell to verify.

9. What will be the output of the following Python code?

```
a={1:"A",2:"B",3:"C"}
b=a.copy()
b[2]="D"
print(a)
```

a) Error, copy() method doesn’t exist for dictionaries
b) {1: ‘A’, 2: ‘B’, 3: ‘C’}
c) {1: ‘A’, 2: ‘D’, 3: ‘C’}
d) “None” is printed

Explanation: Changes made in the copy of the dictionary isn’t reflected in the original one.

10. What will be the output of the following Python code?

```
a={1:"A",2:"B",3:"C"}
a.clear()
print(a)
```

a) None
b) { None:None, None:None, None:None}
c) {1:None, 2:None, 3:None}
d) { }

Explanation: The clear() method clears all the key-value pairs in the dictionary.

11. Which of the following isn't true about dictionary keys?
a) More than one key isn't allowed
b) Keys must be immutable
c) Keys must be integers
d) When duplicate keys encountered, the last assignment wins

Explanation: Keys of a dictionary may be any data type that is immutable.

12. What will be the output of the following Python code?

```
a={1:5,2:3,3:4}
a.pop(3)
print(a)
```

a) {1: 5}
b) {1: 5, 2: 3}
c) Error, syntax error for pop() method
d) {1: 5, 3: 4}

Explanation: pop() method removes the key-value pair for the key mentioned in the pop() method.

13. What will be the output of the following Python code?

```
a={1:5,2:3,3:4}
print(a.pop(4,9))
```

a) 9
b) 3
c) Too many arguments for pop() method
d) 4

Explanation: pop() method returns the value when the key is passed as an argument and otherwise returns the default value(second argument) if the key isn't present in the dictionary.

14. What will be the output of the following Python code?

```
a={1:"A",2:"B",3:"C"}
for i in a:
    print(i,end=" ")
```

a) 1 2 3
b) ‘A’ ‘B’ ‘C’
c) 1 ‘A’ 2 ‘B’ 3 ‘C’
d) Error, it should be: for i in a.items():

Explanation: The variable i iterates over the keys of the dictionary and hence the keys are printed.

15. What will be the output of the following Python code?

```
>>> a={1:"A",2:"B",3:"C"}
>>> a.items()
```

a) Syntax error
b) dict_items([(‘A’), (‘B’), (‘C’)])
c) dict_items([(1,2,3)])
d) dict_items([(1, ‘A’), (2, ‘B’), (3, ‘C’)])

Explanation: The method items() returns list of tuples with each tuple having a key-value pair.

Python MCQ’s – Dictionary – 3

1. Which of the statements about dictionary values if false?
a) More than one key can have the same value
b) The values of the dictionary can be accessed as dict[key]
c) Values of a dictionary must be unique
d) Values of a dictionary can be a mixture of letters and numbers

Explanation: More than one key can have the same value.

2. What will be the output of the following Python code snippet?

```
>>> a={1:"A",2:"B",3:"C"}
>>> del a
```

a) method del doesn’t exist for the dictionary
b) del deletes the values in the dictionary
c) del deletes the entire dictionary
d) del deletes the keys in the dictionary

Explanation: del deletes the entire dictionary and any further attempt to access it will throw an error.

3. If a is a dictionary with some key-value pairs, what does a.popitem() do?
a) Removes an arbitrary element
b) Removes all the key-value pairs
c) Removes the key-value pair for the key given as an argument
d) Invalid method for dictionary

Explanation: The method popitem() removes a random key-value pair.

4. What will be the output of the following Python code snippet?

```
total={}
def insert(items):
    if items in total:
        total[items] += 1
    else:
        total[items] = 1
insert('Apple')
insert('Ball')
insert('Apple')
print (len(total))
```

a) 3
b) 1
c) 2
d) 0

Explanation: The insert() function counts the number of occurrences of the item being inserted into the dictionary. There are only 2 keys present since the key ‘Apple’ is repeated. Thus, the length of the dictionary is 2.

5. What will be the output of the following Python code snippet?

```
a = {}
a[1] = 1
a['1'] = 2
a[1]=a[1]+1
count = 0
for i in a:
    count += a[i]
print(count)
```

a) 1
b) 2
c) 4
d) Error, the keys can’t be a mixture of letters and numbers

Explanation: The above piece of code basically finds the sum of the values of keys.

6. What will be the output of the following Python code snippet?

```
numbers = {}
letters = {}
comb = {}
numbers[1] = 56
numbers[3] = 7
letters[4] = 'B'
comb['Numbers'] = numbers
comb['Letters'] = letters
print(comb)
```

a) Error, dictionary in a dictionary can't exist
b) 'Numbers': {1: 56, 3: 7}
c) {'Numbers': {1: 56}, 'Letters': {4: 'B'}}
d) {'Numbers': {1: 56, 3: 7}, 'Letters': {4: 'B'}}

Explanation: Dictionary in a dictionary can exist.

7. What will be the output of the following Python code snippet?

```
test = {1:'A', 2:'B', 3:'C'}
test = {}
print(len(test))
```

a) 0
b) None
c) 3
d) An exception is thrown

Explanation: In the second line of code, the dictionary becomes an empty dictionary. Thus, length=0.

8. What will be the output of the following Python code snippet?

```
test = {1:'A', 2:'B', 3:'C'}
del test[1]
test[1] = 'D'
del test[2]
print(len(test))
```

a) 0
b) 2
c) Error as the key-value pair of 1:'A' is already deleted
d) 1

Explanation: After the key-value pair of 1:'A' is deleted, the key-value pair of 1:'D' is added.

9. What will be the output of the following Python code snippet?

```
a = {}
a[1] = 1
a['1'] = 2
a[1.0]=4
count = 0
for i in a:
    count += a[i]
print(count)
```

a) An exception is thrown
b) 3
c) 6
d) 2

Explanation: The value of key 1 is 4 since 1 and 1.0 are the same. Then, the function count() gives the sum of all the values of the keys (2+4).

10. What will be the output of the following Python code snippet?

```
a={}
a['a']=1
a['b']=[2,3,4]
print(a)
```

a) Exception is thrown
b) {'b': [2], 'a': 1}
c) {'b': [2], 'a': [3]}
d) {'b': [2, 3, 4], 'a': 1}

Explanation: Mutable members can be used as the values of the dictionary but they cannot be used as the keys of the dictionary.

11. What will be the output of the following Python code snippet?

```
>>>import collections
>>> a=collections.Counter([1,1,2,3,3,4,4,4])
>>> a
```

a) {1,2,3,4}
b) Counter({4, 1, 3, 2})
c) Counter({4: 3, 1: 2, 3: 2, 2: 1})
d) {4: 3, 1: 2, 3: 2, 2: 1}

Explanation: The statement a=collections.OrderedDict() generates a dictionary with the number as the key and the count of times the number appears as the value.

12. What will be the output of the following Python code snippet?

```
>>>import collections
>>> b=collections.Counter([2,2,3,4,4,4])
>>> b.most_common(1)
```

a) Counter({4: 3, 2: 2, 3: 1})
b) {3:1}
c) {4:3}
d) [(4, 3)]

Explanation: The most_common() method returns the n number key-value pairs where the value is the most recurring.

13. What will be the output of the following Python code snippet?

```
>>>import collections
>>> b=collections.Counter([2,2,3,4,4,4])
>>> b.most_common(1)
```

a) Counter({4: 3, 2: 2, 3: 1})
b) {3:1}
c) {4:3}
d) [(4, 3)]

Explanation: The most common() method returns the n number key-value pairs where the value is the most recurring.

14. What will be the output of the following Python code snippet?

```
>>> import collections
>>> a=collections.Counter([2,2,3,3,3,4])
>>> b=collections.Counter([2,2,3,4,4])
>>> a|b
```

a) Counter({3: 3, 2: 2, 4: 2})
b) Counter({2: 2, 3: 1, 4: 1})
c) Counter({3: 2})
d) Counter({4: 1})

Explanation: a|b returns the pair of keys and the highest recurring value.

15. What will be the output of the following Python code snippet?

```
>>> import collections
>>> a=collections.Counter([3,3,4,5])
>>> b=collections.Counter([3,4,4,5,5,5])
>>> a&b
```

a) Counter({3: 12, 4: 1, 5: 1})
b) Counter({3: 1, 4: 1, 5: 1})
c) Counter({4: 2})
d) Counter({5: 1})

Explanation: a&b returns the pair of keys and the lowest recurring value.

Python MCQ's – Dictionary – 4

1. The following Python code is invalid.

```
class demo(dict):
  def   test _(self,key):
    return []
a = demo()
a['test'] = 7
print(a)
```

a) True
b) False

Explanation: The output of the code is: {'test':7}.

2. What will be the output of the following Python code?

```
count={}
count[(1,2,4)] = 5
count[(4,2,1)] = 7
count[(1,2)] = 6
count[(4,2,1)] = 2
tot = 0
for i in count:
    tot=tot+count[i]
print(len(count)+tot)
```

a) 25
b) 17

c) 16
d) Tuples can't be made keys of a dictionary

Explanation: Tuples can be made keys of a dictionary. Length of the dictionary is 3 as the value of the key (4,2,1) is modified to 2. The value of the variable tot is 5+6+2=13.

3. What will be the output of the following Python code?

```
a={}
a[2]=1
a[1]=[2,3,4]
print(a[1][1])
```

a) [2,3,4]
b) 3
c) 2
d) An exception is thrown

Explanation: Now, a={1:[2,3,4],2:1} . a[1][1] refers to second element having key 1.

4. What will be the output of the following Python code?

```
>>> a={'B':5,'A':9,'C':7}
>>> sorted(a)
```

a) ['A','B','C']
b) ['B','C','A']
c) [5,7,9]
d) [9,5,7]

Explanation: Return a new sorted list of keys in the dictionary.

5. What will be the output of the following Python code?

```
>>> a={i: i*i for i in range(6)}
>>> a
```

a) Dictionary comprehension doesn't exist
b) {0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6:36}
c) {0: 0, 1: 1, 4: 4, 9: 9, 16: 16, 25: 25}
d) {0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

Explanation: Dictionary comprehension is implemented in the above piece of code.

6. What will be the output of the following Python code?

```
>>> a={}
>>> a.fromkeys([1,2,3],"check")
```

a) Syntax error
b) {1:"check",2:"check",3:"check"}
c) "check"
d) {1:None,2:None,3:None}

Explanation: The dictionary takes values of keys from the list and initializes it to the default value (value given in the second parameter). Execute in Python shell to verify.

7. What will be the output of the following Python code?

```
>>> b={}
>>> all(b)
```

a) { }
b) False
c) True
d) An exception is thrown

Explanation: Function all() returns True if all keys of the dictionary are true or if the dictionary is empty.

8. If b is a dictionary, what does any(b) do?
a) Returns True if any key of the dictionary is true
b) Returns False if dictionary is empty
c) Returns True if all keys of the dictionary are true
d) Method any() doesn't exist for dictionary

Explanation: Method any() returns True if any key of the dictionary is true and False if the dictionary is empty.

9. What will be the output of the following Python code?

```
>>> a={"a":1,"b":2,"c":3}
>>> b=dict(zip(a.values(),a.keys()))
>>> b
```

a) {‘a’: 1, ‘b’: 2, ‘c’: 3}
b) An exception is thrown
c) {‘a’: ‘b’: ‘c’: }
d) {1: ‘a’, 2: ‘b’, 3: ‘c’}

Explanation: The above piece of code inverts the key-value pairs in the dictionary.

10. What will be the output of the following Python code?

```
>>> a={i: 'A' + str(i) for i in range(5)}
>>> a
```

a) An exception is thrown
b) {0: ‘A0’, 1: ‘A1’, 2: ‘A2’, 3: ‘A3’, 4: ‘A4’}
c) {0: ‘A’, 1: ‘A’, 2: ‘A’, 3: ‘A’, 4: ‘A’}
d) {0: ‘0’, 1: ‘1’, 2: ‘2’, 3: ‘3’, 4: ‘4’}

Explanation: Dictionary comprehension and string concatenation is implemented in the above piece of code.

11. What will be the output of the following Python code?

```
>>> a=dict()
>>> a[1]
```

a) An exception is thrown since the dictionary is empty
b) ‘ ‘
c) 1
d) 0

Explanation: The values of a dictionary can be accessed through the keys only if the keys exist in the dictionary.

12. What will be the output of the following Python code?

```
>>> import collections
>>> a=dict()
>>> a=collections.defaultdict(int)
>>> a[1]
```

a) 1
b) 0
c) An exception is thrown

d) ‘ ‘

Explanation: The statement a=collections.defaultdict(int) gives the default value of 0 (since int data type is given within the parenthesis) even if the keys don’t exist in the dictionary.

13. What will be the output of the following Python code?

```
>>> import collections
>>> a=dict()
>>> a=collections.defaultdict(str)
>>> a['A']
```

a) An exception is thrown since the dictionary is empty
b) ‘ ‘
c) ‘A’
d) 0

Explanation: The statement a=collections.defaultdict(str) gives the default value of ‘ ‘ even if the keys don’t exist in the dictionary.

14. What will be the output of the following Python code?

```
>>> import collections
>>> b=dict()
>>> b=collections.defaultdict(lambda: 7)
>>> b[4]
```

a) 4
b) 0
c) An exception is thrown
d) 7

Explanation: The statement a=collections.defaultdict(lambda: x) gives the default value of x even if the keys don’t exist in the dictionary.

15. What will be the output of the following Python code?

```
>>> import collections
>>> a=collections.OrderedDict((str(x),x) for x in range(3))
>>> a
```

a) {‘2’:2, ‘0’:0, ‘1’:1}
b) OrderedDict([(‘0’, 0), (‘1’, 1), (‘2’, 2)])

c) An exception is thrown
d) ‘ ‘

Explanation: The line of code a=collections.OrderedDict() generates a dictionary satisfying the conditions given within the parenthesis and in an ascending order of the keys.