**PYTHON MCQ-all chapters**

**Introduction to Python(Variables,Expressions,Statements)**

A)10 marks (1 mark each)

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

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

3. Which of the following is invalid?  
a) \_a = 1  
b) \_\_a = 1  
c) \_\_str\_\_ = 1  
d) none of the mentioned

4. Which of the following is an invalid variable?  
a) my\_string\_1  
b) 1st\_string  
c) foo  
d) \_

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

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

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

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

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

10. Which of the following cannot be a variable?  
a) \_\_init\_\_  
b) in  
c) it  
d) on

B)Write answers in short(1 mark each))

11)What are the advantages of Python Programming Language?

12)What are two modes of working in IDLE?

13)What is a variable?

14)How to assign a value to a variable?

15)Write data types in Python.

16)What is difference between mutable data type n immutable data types?

17)What are the different types of operators in Python

18)What is an expression?

19)Explain the print () function in Python

20)Explain the input () function in Python

**Functions,Conditional and Looping Construct**

1.What will be the output of the following code :

print type(type(int))

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

* [A.](javascript:%20void(0)) type ‘int’
* [B.](javascript:%20void(0)) type ‘type’
* [C.](javascript:%20void(0)) error
* [D.](javascript:%20void(0)) 0

|  |
| --- |
|  |

2.What is the output of the following code :

|  |
| --- |
| L = ['a','b','c','d']  print "".join(L) |

* [A.](javascript:%20void(0)) error
* [B.](javascript:%20void(0)) none
* [C.](javascript:%20void(0)) abcd
* [D.](javascript:%20void(0)) [‘a’,’b’,’c’,’c’]

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

* for i in range(0,2,-1):
* print("Hello")
* A. Hello  
  B. Hello Hello  
  C. No Output  
  D. Error

4.What is called when a function is defined inside a class?

* [A.](javascript:%20void(0)) module
* [B.](javascript:%20void(0)) class
* [C.](javascript:%20void(0)) another function
* [D.](javascript:%20void(0)) [‘a’,’b’,’c’,’c’]

5.A while loop in Python is used for what type of iteration?

* A. indefinite  
  B. discriminant  
  C. definite  
  D. indeterminate

6.When does the else statement written after loop executes?

* A. When break statement is executed in the loop  
  B. When loop condition becomes false  
  C. Else statement is always executed  
  D. None of the above

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

* x = "abcdef"
* i = "i"
* while i in x:
* print(i, end=" ")
* A. a b c d e f  
  B. abcdef  
  C. i i i i i.....  
  D. No Output

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

* x = "abcd"
* for i in range(len(x)):
* print(i)
* A. abcd  
  B. 0 1 2 3  
  C. 1 2 3 4  
  D. a b c d

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

* x = 12
* for i in x:
* print(i)
* A. 12  
  B. 1 2  
  C. Error  
  D. None of the above

10.Which of the following sequences would be generated bt the given line of code?

range (5, 0, -2)

A. 5 4 3 2 1 0 -1  
B. 5 4 3 2 1 0  
C. 5 3 1  
D. None of the above

11.List out the conditional construct

12.Explain the if…else construct

13.Explain the nested if…else construct

14.Explain the while loop

15.Explain the for statement

16.Explain loop control statement break,continue and pass

17.What is the use of compound expression in conditional construct?

18.Explain the random module with example

19.Explain the math module with example

20.Explain the date and time module with example

21.What is mean by user defined function

22.Write the difference between call by value n call by reference

24.Explain the void function?

25.Explain the local and global scope of variables?

**Chapter 3:Srings**

1. What is the output when following code is executed ?

print r"\nhello"

The output is

* [A.](javascript:%20void(0)) a new line and hello
* [B.](javascript:%20void(0)) \nhello
* [C.](javascript:%20void(0)) the letter r and then hello
* [D.](javascript:%20void(0)) Error

2. What is the output of the following code ?

example = "snow world"

example[3] = 's'

print example

* [A.](javascript:%20void(0)) snow
* [B.](javascript:%20void(0)) snow world
* [C.](javascript:%20void(0)) Error
* [D.](javascript:%20void(0)) snos world

3. What is the output of “hello”+1+2+3 ?

* [A.](javascript:%20void(0)) hello123
* [B.](javascript:%20void(0)) hello
* [C.](javascript:%20void(0)) Error
* [D.](javascript:%20void(0)) hello6

4. Suppose i is 5 and j is 4, i + j is same as

* [A.](javascript:%20void(0)) i.\_\_add(j)
* [B.](javascript:%20void(0)) i.\_\_add\_\_(j)
* [C.](javascript:%20void(0)) i.\_\_Add(j)
* [D.](javascript:%20void(0)) i.\_\_ADD(j)

5. What is the output of the following?

print('\*', "abcdef".center(7), '\*')-B

* [A.](javascript:%20void(0)) \* abcdef \*
* [B.](javascript:%20void(0)) \* abcdef \*
* [C.](javascript:%20void(0)) \*abcdef \*
* [D.](javascript:%20void(0)) \* abcdef\*

6. What is the output of the following?

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

* [A.](javascript:%20void(0)) 2
* [B.](javascript:%20void(0)) 0
* [C.](javascript:%20void(0)) 1
* [D.](javascript:%20void(0)) Error

7. What is the output of the following?

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

* [A.](javascript:%20void(0)) Hello foo and bin
* [B.](javascript:%20void(0)) Hello bin and foo
* [C.](javascript:%20void(0)) Error
* [D.](javascript:%20void(0)) None of the mentioned

8. What is the output of the following?

print('The sum of {0} and {1} is {2}'.format(2, 10, 12))

* [A.](javascript:%20void(0)) The sum of 2 and 10 is 12
* [B.](javascript:%20void(0)) Error
* [C.](javascript:%20void(0)) The sum of 0 and 1 is 2
* [D.](javascript:%20void(0)) None of the mentioned

9. What is the output of the following?

print('ab'.isalpha())

* [A.](javascript:%20void(0)) True
* [B.](javascript:%20void(0)) False
* [C.](javascript:%20void(0)) None
* [D.](javascript:%20void(0)) Error

10. What is the output of the following?

print('1.1'.isnumeric())

* [A.](javascript:%20void(0)) True
* [B.](javascript:%20void(0)) False
* [C.](javascript:%20void(0)) None
* [D.](javascript:%20void(0)) Error

11. What is the output of the following?

print('a'.maketrans('ABC', '123'))

* [A.](javascript:%20void(0)) {97: 49, 98: 50, 99: 51}
* [B.](javascript:%20void(0)) {65: 49, 66: 50, 67: 51}
* [C.](javascript:%20void(0)) {97: 49}
* [D.](javascript:%20void(0)) 1

12. What is the output of the following?

print('xyyxyyxyxyxxy'.replace('xy', '12', 100))

* [A.](javascript:%20void(0)) xyyxyyxyxyxxy
* [B.](javascript:%20void(0)) 12y12y1212x12
* [C.](javascript:%20void(0)) none of the mentioned
* [D.](javascript:%20void(0)) error

13. What is the output of the following?

print('abcd'.translate({'a': '1', 'b': '2', 'c': '3', 'd': '4'}))

* [A.](javascript:%20void(0)) abcd
* [B.](javascript:%20void(0)) 1234
* [C.](javascript:%20void(0)) error
* [D.](javascript:%20void(0)) none of the mentioned

14.How to create the string and initialize the string?

15.How to access the elements of the string?

16.Explain the string operators.

17.Explain the built-in functions and methods of string with example.

18.Explain the string constants defined in string module

19.What is regular expression?

20.What are the types of regular expression?

21.Explain the regular expression patterns

22.Explain the match() method with example

23.Explain the search() method with example

**Chapter 4.Lists,Tuples Sets,Dictionaries**

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

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

print(a[::2])

A. [1,2]  
B. [8,9]  
C. [1,3,5,7,9]  
D. [1,2,3]

#### Q-2. What will be the output of the following code snippet?

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

a[::2]=10,20,30,40,50,60

print(a)

**A.** ValueError: attempt to assign sequence of size 6 to extended slice of size 5  
**B.** [10, 2, 20, 4, 30, 6, 40, 8, 50, 60]  
**C.** [1, 2, 10, 20, 30, 40, 50, 60]  
**D.** [1, 10, 3, 20, 5, 30, 7, 40, 9, 50, 60]

#### Q-3. What will be the output of the following code snippet?

a=[1,2,3,4,5]

print(a[3:0:-1])

**A.** Syntax error  
**B.** [4, 3, 2]  
**C.** [4, 3]  
**D.** [4, 3, 2, 1]

#### Q-4. What is the correct command to shuffle the following list?

fruit=['apple', 'banana', 'papaya', 'cherry']

**A.** fruit.shuffle()  
**B.** shuffle(fruit)  
**C.** random.shuffle(fruit)  
**D.** random.shuffleList(fruit)

#### Q-5. What will be the output of the following code snippet?

arr = [[1, 2, 3, 4],

[4, 5, 6, 7],

[8, 9, 10, 11],

[12, 13, 14, 15]]

for i in range(0, 4):

print(arr[i].pop())

**A.** 1 2 3 4  
**B.** 1 4 8 12  
**C.** 4 7 11 15  
**D.** 12,13,14,15

#### Q-6. What will be the output of the following code snippet?

arr = [1, 2, 3, 4, 5, 6]

for i in range(1, 6):

arr[i - 1] = arr[i]

for i in range(0, 6):

print(arr[i], end = " ")

**A.** 1 2 3 4 5 6  
**B.** 2 3 4 5 6 1  
**C.** 1 1 2 3 4 5  
**D.** 2 3 4 5 6 6

#### Q-7. What will be the output of the following code snippet?

init\_tuple = ()

print (init\_tuple.\_\_len\_\_())

**A.** None  
**B.**  1  
**C.** 0  
**D.** Exception

#### Q-8. What will be the output of the following code snippet?

init\_tuple\_a = 'a', 'b'

init\_tuple\_b = ('a', 'b')

print (init\_tuple\_a == init\_tuple\_b)

**A.** 0  
**B.**  1  
**C.** False  
**D.** True

#### Q-9. Which of the following statements given below is/are true?

**A.** Tuples have structure, lists have an order.  
**B.** Tuples are homogeneous, lists are heterogeneous.  
**C.** Tuples are immutable, lists are mutable.  
**D.** All of them.

#### Q-10. What will be the output of the following code snippet?-

init\_tuple = ('Python') \* 3

print(type(init\_tuple))

**A.** <class ‘tuple’>  
**B.** <class ‘str’>  
**C.** <class ‘list’>  
**D.** <class ‘function’>

#### Q-11. What will be the output of the following code snippet?-

a = {(1,2):1,(2,3):2}

print(a[1,2])

**A.** Key Error  
**B.**  1  
**C.** {(2,3):2}  
**D.** {(1,2):1}

#### Q-12. What will be the output of the following code snippet?-

a = {'a':1,'b':2,'c':3}

print (a['a','b'])

**A.** Key Error  
**B.** [1,2]  
**C.** {‘a’:1,’b’:2}  
**D.** (1,2)

#### Q-13. What will be the output of the following code snippet?-

dict = {'c': 97, 'a': 96, 'b': 98}

for \_ in sorted(dict):

print (dict[\_])

**A.** 96 98 97  
**B.** 96 97 98  
**C.** 98 97 96  
**D.** NameError

#### Q-14. What will be the output of the following code snippet?-

rec = {"Name" : "Python", "Age":"20"}

r = rec.copy()

print(id(r) == id(rec))

**A.** True  
**B.** False  
**C.** 0  
**D.** 1

15.Explain the different list operators with example

16.What are list comprehensions?

17.What is tuple?

18.Explain the tuple function with example

19.What are the different tuple operations?

20.What is set? Write different operations performed on the sets

21.What is the difference between list and tuple?

22.What is the difference between mutable and immutable?

**Chapter 5.Modules**

1. Which of these definitions correctly describes a module?-  
a) Denoted by triple quotes for providing the specification of certain program elements  
b) Design and implementation of specific functionality to be incorporated into a program  
c) Defines the specification of how it is to be used  
d) Any program that reuses code

2. Which of the following is not an advantage of using modules?-  
a) Provides a means of reuse of program code  
b) Provides a means of dividing up tasks  
c) Provides a means of reducing the size of the program  
d) Provides a means of testing individual parts of the program

3. Program code making use of a given module is called a \_\_\_\_\_\_ of the module.-  
a) Client  
b) Docstring  
c) Interface  
d) Modularity

4. \_\_\_\_\_\_ is a string literal denoted by triple quotes for providing the specifications of certain program elements.-  
a) Interface  
b) Modularity  
c) Client  
d) Docstring

5. Which of the following is not a valid namespace?-  
a) Global namespace  
b) Public namespace  
c) Built-in namespace  
d) Local namespace

6. Which of the following is false about “import modulename” form of import?-  
a) The namespace of imported module becomes part of importing module  
b) This form of import prevents name clash  
c) The namespace of imported module becomes available to importing module  
d) The identifiers in module are accessed as: modulename.identifier

7. Which of the statements about modules is false?-  
a) In the “from-import” form of import, identifiers beginning with two underscores are private and aren’t imported  
b) dir() built-in function monitors the items in the namespace of the main module  
c) In the “from-import” form of import, all identifiers regardless of whether they are private or public are imported  
d) When a module is loaded, a compiled version of the module with file extension .pyc is automatically produced

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

from math import factorial

print(math.factorial(5))

a) 120  
b) Nothing is printed  
c) Error, method factorial doesn’t exist in math module  
d) Error, the statement should be: print(factorial(5))

8. What is module?Why we need module?

9.What are the advantages of modules?

10.How to execute module as a script?

11.Explain the dir() function

12.What is package?

13.How to import from the package?

**Chapter 6.I/O and file handling**

### Q-1.  Which of the following command is used to open a file “c:\temp.txt” in read-mode only?-

**A.** infile = open(“c:\temp.txt”, “r”)  
**B.** infile = open(“c:\\temp.txt”, “r”)  
**C.** infile = open(file = “c:\temp.txt”, “r+”)  
**D.** infile = open(file = “c:\\temp.txt”, “r+”)

### Q-2.  Which of the following command is used to open a file “c:\temp.txt” in write-mode -only?-

**A.** outfile = open(“c:\temp.txt”, “w”)  
**B.** outfile = open(“c:\\temp.txt”, “w”)  
**C.** outfile = open(file = “c:\temp.txt”, “w+”)  
**D.** outfile = open(file = “c:\\temp.txt”, “w+”)

### Q-3.  Which of the following command is used to open a file “c:\temp.txt” in append-mode?

**A.** outfile = open(“c:/temp.txt”, “a”)  
**B.** outfile = open(“c:\\temp.txt”, “rw”)  
**C.** outfile = open(“c:\temp.txt”, “w+”)  
**D.** outfile = open(“c:\\temp.txt”, “r+”)  
**E.** outfile = open(“c:\\temp.txt”, “a”)

### Q-4.  Which of the following statements are true regarding the opening modes of a file?

**A.** When you open a file for reading, if the file does not exist, an error occurs.  
**B.** When you open a file for writing, if the file does not exist, an error occurs.  
**C.** When you open a file for reading, if the file does not exist, the program will open an empty file.  
**D.** When you open a file for writing, if the file does not exist, a new file is created.  
**E.** When you open a file for writing, if the file exists, the existing file is overwritten with the new file.

### Q-5. Which of the following commands can be used to read “n” number of characters from a file using the file object <file>?

**A.** file.read(n)  
**B.** n = file.read()  
**C.** file.readline(n)  
**D.** file.readlines()

### Q-6. Which of the following commands can be used to read the entire contents of a file as a string using the file object <tmpfile>?

**A.** tmpfile.read(n)  
**B.** tmpfile.read()  
**C.** tmpfile.readline()  
**D.** tmpfile.readlines()

### Q-7. Which of the following commands can be used to read the next line in a file using the file object <tmpfile>?-

**A.** tmpfile.read(n)  
**B.** tmpfile.read()  
**C.** tmpfile.readline()  
**D.** tmpfile.readlines()

### Q-8. Which of the following commands can be used to read the remaining lines in a file using the file object <tmpfile>?-

**A.** tmpfile.read(n)  
**B.** tmpfile.read()  
**C.** tmpfile.readline()  
**D.** tmpfile.readlines()

### Q-9. What does the <readlines()> method returns?-

**A.** str  
**B.** a list of lines  
**C.** list of single characters  
**D.** list of integers

### Q-10. Which of the following functions can be used to check if a file “logo” exists?-

**A.** os.path.isFile(logo)  
**B.** os.path.exists(logo)  
**C.** os.path.isfile(logo)  
**D.** os.isFile(logo)

### Q-11. Which of the following functions displays a file dialog for opening an existing file?-

**A.** tmpfile = askopenfilename()  
**B.** tmpfile = asksaveasfilename()  
**C.** tmpfile = openfilename()  
**D.** tmpfile = saveasfilename()

### Q-12. Which of the following functions displays a file dialog for saving a file?-

**A.** tmpfile = askopenfilename()  
**B.** tmpfile = openfilename()  
**C.** tmpfile = asksaveasfilename()  
**D.** tmpfile = saveasfilename()

### Q-15. Which of the following functions do you use to write data in the binary format?-

**A.** write  
**B.** output  
**C.** dump  
**D.** send

Q.16. Explain the format() method

Q.17.Explain the open() file function

Q.18.Explain the close() file function

Q.19. Explain the write() file function

Q.20. Explain the read() file function

**Chapter 7.Errors and Exceptions**

1. How many except statements can a try-except block have?-  
a) zero  
b) one  
c) more than one  
d) more than zero

2. When will the else part of try-except-else be executed?-  
a) always  
b) when an exception occurs  
c) when no exception occurs  
d) when an exception occurs in to except block

3. Is the following Python code valid?-

try:

*# Do something*

except:

*# Do something*

finally:

*# Do something*

a) no, there is no such thing as finally  
b) no, finally cannot be used with except  
c) no, finally must come before except  
d) yes

4. Can one block of except statements handle multiple exception?-  
a) yes, like except TypeError, SyntaxError [,…]  
b) yes, like except [TypeError, SyntaxError]  
c) no  
d) none of the mentioned

5. When is the finally block executed?-d  
a) when there is no exception  
b) when there is an exception  
c) only if some condition that has been specified is satisfied  
d) always

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

def foo():

try:

return 1

finally:

return 2

k = foo()

print(k)

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

7. What happens when ‘1’ == 1 is executed?  
a) we get a True  
b) we get a False  
c) an TypeError occurs  
d) a ValueError occurs

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

def foo():

try:

print(1)

finally:

print(2)

foo()

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

9.What do you mean by syntax errors

10.What is exception?Explain with an example

11.Explain standard exceptions

12.How to raise an exceptions

**Chapter 8.Introduction to OOP in Python**

1. \_\_\_\_\_ represents an entity in the real world with its identity and behaviour.-  
a) A method  
b) An object  
c) A class  
d) An operator

2. \_\_\_\_\_ is used to create an object.-b  
a) class  
b) constructor  
c) User-defined functions  
d) In-built functions

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

class test:

def \_\_init\_\_(self,a="Hello World"):

self.a=a

def display(self):

print(self.a)

obj=test()

obj.display()

a) The program has an error because constructor can’t have default arguments  
b) Nothing is displayed  
c) “Hello World” is displayed  
d) The program has an error display function doesn’t have parameters

4. What is setattr() used for?-b  
a) To access the attribute of the object  
b) To set an attribute  
c) To check if an attribute exists or not  
d) To delete an attribute

5. What is getattr() used for?-a  
a) To access the attribute of the object  
b) To delete an attribute  
c) To check if an attribute exists or not  
d) To set an attribute

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

class test:

def \_\_init\_\_(self,a):

self.a=a

def display(self):

print(self.a)

obj=test()

obj.display()

a) Runs normally, doesn’t display anything  
b) Displays 0, which is the automatic default value  
c) Error as one argument is required while creating the object  
d) Error as display function requires additional argument

7. What is Instantiation in terms of OOP terminology?-  
a) Deleting an instance of class  
b) Modifying an instance of class  
c) Copying an instance of class  
d) Creating an instance of class

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

class Demo:

def \_\_init\_\_(self):

pass

def test(self):

print(\_\_name\_\_)

obj = Demo()

obj.test()

a) Exception is thrown  
b) \_\_main\_\_  
c) Demo  
d) test

9. What is object?

10.What is namespaces?