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

2. Which keyword is used for function?  
a) Fun  
b) Define  
c) Def  
d) Function

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

1. def sayHello():
2. print('Hello World!')
3. sayHello()
4. sayHello()

a)

Hello World!

Hello World!

**Take Python Programming Practice Tests - Chapterwise!  
Start the Test Now:**[**Chapter 1,**](https://rank.sanfoundry.com/python-programming-practice-test-1/)[**2,**](https://rank.sanfoundry.com/python-programming-practice-test-2/)[**3,**](https://rank.sanfoundry.com/python-programming-practice-test-3/)[**4,**](https://rank.sanfoundry.com/python-programming-practice-test-4/)[**5,**](https://rank.sanfoundry.com/python-programming-practice-test-5/)[**6,**](https://rank.sanfoundry.com/python-programming-practice-test-6/)[**7,**](https://rank.sanfoundry.com/python-programming-practice-test-7/)[**8,**](https://rank.sanfoundry.com/python-programming-practice-test-8/)[**9,**](https://rank.sanfoundry.com/python-programming-practice-test-9/)[**10**](https://rank.sanfoundry.com/python-programming-practice-test-10/)

b)

'Hello World!'

'Hello World!'

c)

Hello

Hello

d) None of the mentioned

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

1. def printMax(a, b):
2. if a > b:
3. print(a, 'is maximum')
4. elif a == b:
5. print(a, 'is equal to', b)
6. else:
7. print(b, 'is maximum')
8. printMax(3, 4)

a) 3  
b) 4  
c) 4 is maximum  
d) None of the mentioned

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

1. x = 50
2. def func(x):
3. print('x is', x)
4. x = 2
5. print('Changed local x to', x)
6. func(x)
7. print('x is now', x)

a)

x is 50

Changed local x to 2

x is now 50

b)

x is 50

Changed local x to 2

x is now 2

c)

x is 50

Changed local x to 2

x is now 100

d) None of the mentioned

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

1. x = 50
2. **def** func():
3. **global** x
4. **print**('x is', x)
5. x = 2
6. **print**('Changed global x to', x)
7. func()
8. **print**('Value of x is', x)

a)

x is 50

Changed global x to 2

Value of x is 50

b)

x is 50

Changed global x to 2

Value of x is 2

c)

x is 50

Changed global x to 50

Value of x is 50

d) None of the mentioned

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

1. **def** say(message, times = 1):
2. **print**(message \* times)
3. say('Hello')
4. say('World', 5)

a)

Hello

WorldWorldWorldWorldWorld

b)

Hello

World 5

c)

Hello

World,World,World,World,World

d)

Hello

HelloHelloHelloHelloHello

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

1. def func(a, b=5, c=10):
2. print('a is', a, 'and b is', b, 'and c is', c)
4. func(3, 7)
5. func(25, c = 24)
6. func(c = 50, a = 100)

a)

a is 7 and b is 3 and c is 10

a is 25 and b is 5 and c is 24

a is 5 and b is 100 and c is 50

b)

a is 3 and b is 7 and c is 10

a is 5 and b is 25 and c is 24

a is 50 and b is 100 and c is 5

c)

a is 3 and b is 7 and c is 10

a is 25 and b is 5 and c is 24

a is 100 and b is 5 and c is 50

d) None of the mentioned

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

1. **def** maximum(x, y):
2. **if** x > y:
3. **return** x
4. **elif** x == y:
5. **return** 'The numbers are equal'
6. **else**:
7. **return** y
9. **print**(maximum(2, 3))

a) 2  
b) 3  
c) The numbers are equal  
d) None of the mentioned

10. Which of the following is a feature of DocString?  
a) Provide a convenient way of associating documentation with Python modules, functions, classes, and methods  
b) All functions should have a docstring  
c) Docstrings can be accessed by the \_\_doc\_\_ attribute on objects  
d) All of the mentioned

This set of Python Questions for entrance examinations focuses on “Functions”.

1. Which are the advantages of functions in python?  
a) Reducing duplication of code  
b) Decomposing complex problems into simpler pieces  
c) Improving clarity of the code  
d) All of the mentioned

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

3. Where is function defined?  
a) Module  
b) Class  
c) Another function  
d) All of the mentioned

4. What is called when a function is defined inside a class?  
a) Module  
b) Class  
c) Another function  
d) Method

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

6. Which of the following refers to mathematical function?  
a) sqrt  
b) rhombus  
c) add  
d) rhombus

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

1. **def** cube(x):
2. **return** x \* x \* x
3. x = cube(3)
4. **print** x

a) 9  
b) 3  
c) 27  
d) 30

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

1. **def** C2F(c):
2. **return** c \* 9/5 + 32
3. **print** C2F(100)
4. **print** C2F(0)

a)

212

32

b)

314

24

c)

567

98

d) None of the mentioned

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

1. **def** power(x, y=2):
2. r = 1
3. **for** i **in** range(y):
4. r = r \* x
5. **return** r
6. **print** power(3)
7. **print** power(3, 3)

a)

212

32

b)

9

27

c)

567

98

d) None of the mentioned

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

1. **def** sum(\*args):
2. '''Function returns the sum
3. of all values'''
4. r = 0
5. **for** i **in** args:
6. r += i
7. **return** r
8. **print** sum.\_\_doc\_\_
9. **print** sum(1, 2, 3)
10. **print** sum(1, 2, 3, 4, 5)

a)

6

15

b)

6

100

c)

123

12345

d) None of the mentioned

#-------------------------------------------------------------------------------------------------------------------------

1. What type of inheritance is illustrated in the following Python code?

**class** A():

**pass**

**class** B(A):

**pass**

**class** C(B):

**pass**

a) Multi-level inheritance  
b) Multiple inheritance  
c) Hierarchical inheritance  
d) Single-level inheritance

2. What does single-level inheritance mean?  
a) A subclass derives from a class which in turn derives from another class  
b) A single superclass inherits from multiple subclasses  
c) A single subclass derives from a single superclass  
d) Multiple base classes inherit a single derived class

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

**class** A:

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

self.\_\_i = 1

self.j = 5

**def** display(self):

**print**(self.\_\_i, self.j)

**class** B(A):

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

super().\_\_init\_\_()

self.\_\_i = 2

self.j = 7

c = B()

c.display()

a) 2 7  
b) 1 5  
c) 1 7  
d) 2 5

4. Which of the following statements isn’t true?  
a) A non-private method in a superclass can be overridden  
b) A derived class is a subset of superclass  
c) The value of a private variable in the superclass can be changed in the subclass  
d) When invoking the constructor from a subclass, the constructor of superclass is automatically invoked

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

**class** A:

**def** \_\_init\_\_(self,x):

self.x = x

**def** count(self,x):

self.x = self.x+1

**class** B(A):

**def** \_\_init\_\_(self, y=0):

A.\_\_init\_\_(self, 3)

self.y = y

**def** count(self):

self.y += 1

**def** main():

obj = B()

obj.count()

**print**(obj.x, obj.y)

main()

a) 3 0  
b) 3 1  
c) 0 1  
d) An exception in thrown

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

>>> **class** A:

**pass**

>>> **class** B(A):

**pass**

>>> obj=B()

>>> isinstance(obj,A)

a) True  
b) False  
c) Wrong syntax for isinstance() method  
d) Invalid method for classes

7. Which of the following statements is true?  
a) The \_\_new\_\_() method automatically invokes the \_\_init\_\_ method  
b) The \_\_init\_\_ method is defined in the object class  
c) The \_\_eq(other) method is defined in the object class  
d) The \_\_repr\_\_() method is defined in the object class

8. Method issubclass() checks if a class is a subclass of another class.  
a) True  
b) False

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

**class** A:

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

self.\_\_x = 1

**class** B(A):

**def** display(self):

**print**(self.\_\_x)

**def** main():

obj = B()

obj.display()

main()

a) 1  
b) 0  
c) Error, invalid syntax for object declaration  
d) Error, private class member can’t be accessed in a subclass

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

**class** A:

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

self.\_x = 5

**class** B(A):

**def** display(self):

**print**(self.\_x)

**def** main():

obj = B()

obj.display()

main()

a) Error, invalid syntax for object declaration  
b) Nothing is printed  
c) 5  
d) Error, private class member can’t be accessed in a subclass

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

**class** A:

**def** \_\_init\_\_(self,x=3):

self.\_x = x

**class** B(A):

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

super().\_\_init\_\_(5)

**def** display(self):

**print**(self.\_x)

**def** main():

obj = B()

obj.display()

main()

a) 5  
b) Error, class member x has two values  
c) 3  
d) Error, protected class member can’t be accessed in a subclass

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

**class** A:

**def** test1(self):

**print**(" test of A called ")

**class** B(A):

**def** test(self):

**print**(" test of B called ")

**class** C(A):

**def** test(self):

**print**(" test of C called ")

**class** D(B,C):

**def** test2(self):

**print**(" test of D called ")

obj=D()

obj.test()

a)

test of B called

test of C called

b)

test of C called

test of B called

c) test of B called  
d) Error, both the classes from which D derives has same method test()

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

**class** A:

**def** test(self):

**print**("test of A called")

**class** B(A):

**def** test(self):

**print**("test of B called")

super().test()

**class** C(A):

**def** test(self):

**print**("test of C called")

super().test()

**class** D(B,C):

**def** test2(self):

**print**("test of D called")

obj=D()

obj.test()

a)

test of B called

test of C called

test of A called

b)

test of C called

test of B called

c)

test of B called

test of C called

d) Error, all the three classes from which D derives has same method test()

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.  
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 displayed  
c) “Hello World” is displayedd) The program has an error display function doesn’t have parameters

4. What is setattr() used for?  
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) 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** change:

**def** \_\_init\_\_(self, x, y, z):

self.a = x + y + z

x = change(1,2,3)

y = getattr(x, 'a')

setattr(x, 'a', y+1)

**print**(x.a)

a) 6  
b) 7  
c) Error  
d) 0

7. 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

8. Is the following Python code correct?

>>> **class** A:

**def** \_\_init\_\_(self,b):

self.b=b

**def** display(self):

**print**(self.b)

>>> obj=A("Hello")

>>> **del** obj

a) True  
b) False

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

**class** test:

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

self.variable = 'Old'

self.Change(self.variable)

**def** Change(self, var):

var = 'New'

obj=test()

**print**(obj.variable)

a) Error because function change can’t be called in the \_\_init\_\_ function  
b) ‘New’ is printed  
c) ‘Old’ is printed  
d) Nothing is printed

10. 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

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

**class** fruits:

**def** \_\_init\_\_(self, price):

self.price = price

obj=fruits(50)

obj.quantity=10

obj.bags=2

**print**(obj.quantity+len(obj.\_\_dict\_\_))

a) 12  
b) 52  
c) 13  
d) 60

12. 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