A. Multiple Choice Questions

1. What is the relation between object and class?
   a. A class is an instance of an object
   b. An object is an instance of an object
   c. An object is an attribute of a class
   d. None of the above
2. Which method should be used to create default values in a class constructor?
   a. __doc__
   b. __new__
   c. __init__
   d. __del__
3. Instantiation is a process of:
   a. Destroying an object
   b. Initialising an object with a default value
   c. Creating a new object with a default value
   d. None of the above
4. What method is called when an object is created?
   a. Self
   b. obj.self
   c. init
   d. __int__
5. We have an object instance obj and want to call its method calc_area(). Which is the correct way of calling the function calc_area()?
   a. obj.calc_area(self)
   b. calc_area.obj()
   c. obj.calc_area()
   d. calc_area.obj(self)
6. Method overriding is _____.
   a. A method with different name,
   b. A method in a subclass which has the same name and same header as that of the super class
   c. Both a and b
   d. None of the above
7. What is used to create an object?
   a. Constructor
   b. Class
   c. Method
   d. None of the above
8. Which of the following statements are true?
   a. Objects of the same type have the same id.
   b. Each object has a unique id.
   c. Both a and b
   d. None of the above

9. _____ represents an entity in the real world which can be distinctly identified?
   a. Object
   b. Class
   c. Method
   d. None of the above
10. Analyze the code given below to find the reason behind the error in the program.

```
class A:
    def __init__(self):
        self.P = 10
        self.__Q = 20

    def getY(self):
        return self.__Q
a = A()
print(a.__Q)
```

   a. Q is private and cannot be access outside of the class.
   b. P is private and cannot be access outside of the class.
   c. Both a and b
   d. None of the above
11. Analyze the code given below to find the reason behind the error in the program.

```
class Base:
    def __init__(self, X):
        self.X = X
    def print(self):
        print(self.X)

Ob1 = Base()
Ob1.print()
```

   a. The class Base does not have a constructor.
   b. X is not defined in print.
   c. The constructor is invoked without arguments.
   d. None of the above
12. What will be the output of the following program?

```
class A:
    def __init__(self, s):
        self.s = s
    def display():
        print(s)
a = A("Welcome")
a.display()
```

   a. Welcome
   b. Error: The self is missing in method display()
   c. Cannot access method display()
   d. None of the above

13. Which statement is correct about self?
    a. Self refers to the previous object.
    b. Self refers to the next object.
    c. Self refers to the current object.
    d. None of the above
14. Which method runs as soon as an object of a class is instantiated?
    a. __init__
    b. __del__
    c. self
    d. None of the above
15. Which of the following is not a type of inheritance?
    a. Single
    b. Multilevel
    c. Distributive
    d. Multiple
16. Look at the following definition of class and determine the type of inheritance the class is using.

```
class  A:
  Pass
class  B:
  Pass
class C(A , B):
    Pass
```

    a. Single
    b. Multilevel
    c. Multiple
    d. None of the above
17. Which method is used to display the attributes of a class?
    a. __init__
    b. __dict__
    c. __del__
    d. None of the above
18. The __del__ is executed only if all_____.
    a. The references to a current instance object have been removed
    b. The references to a previous object have been removed
    c. The references to an instance object have been removed
    d. None of the above
19. Suppose B is a subclass of A. Which syntax will be used to invoke the __init__ method defined in class A from class B?
    a. super()
    b. super().__init(self)__
    c. super().__init()__
    d. None of the above
20. If Ob1 is an instance of class A. Which statement can be used to check whether the object Ob1 is an instance of class A?
    a. Ob1.isinstance(A)
    b. A.isinstance(Ob1)
    c. isinstance(Ob1, A)
    d. isinstance(A,Ob1)