## Python OOP Assignment

Q1. What is the purpose of Python&#39;s OOP?

**To make the software in such a way that it can be scalable ,secure ,less maintainable .**

Q2. Where does an inheritance search look for an attribute?

**It search first in instance object then in the class instance created from then its super class**

Q3. How do you distinguish between a class object and an instance object?

**Class object can be called from class name where an instance object can be called using instance**

Q4. What makes the first argument in a class’s method function special?

**First argument is cls .This convert into class method**

Q5. What is the purpose of the \_\_init\_\_ method?

**To initialize instance value . It acts as constructor**

Q6. What is the process for creating a class instance?

**Class Animal:**

**animal=Animal();**

Q7. What is the process for creating a class?

**Class Animal :**

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

**Def instancemethod(self)**

**@classmethod**

**Def classmethod(cls):**

Q8. How would you define the superclasses of a class?

**Class A : //superclass**

**Class B(A): //subclass**

Q9. What is the relationship between classes and modules?

**Modules Is way to organize the classes. Module can have 0 or more classes.**

Q10. How do you make instances and classes?

**Class Car:**

**C=Car()**

Q11. Where and how should be class attributes created?

**Class attribute created inside the class and outside of the method .It can invoke using class name**

**Class Car:**

**carName=”Maruti”**

**print(Car.carName)**

Q12. Where and how are instance attributes created?

Instance attribute which can be invoked using instance and its attribute not shared by objects

**Class Car :**

**Def \_\_init\_\_(self,color):**

**Self.carColor=color**

**Def showColor():**

**Print(self.carColor)**

**myCar=Car(“Blue”)**

**myCar.showColor()**

Q13. What does the term &quot;self&quot; in a Python class mean?

**Self is used to reference the class object**

Q14. How does a Python class handle operator overloading?

**Def \_\_add\_\_(obj1 ,obj2):**

**X=obj1.x+obj2.x**

**Y=obj11.y+obj2.y**

**Print (x,y)**

**Z=Obj1+obj2**

Q15. When do you consider allowing operator overloading of your classes?

**When you want to add/multiply/ etc operations on objects or strings**

Q16. What is the most popular form of operator overloading?

**Addition. Same addition operator used to add two string**

Q17. What are the two most important concepts to grasp in order to comprehend Python OOP code?

**Overloading,Encapsulation,Inheritance,Abstraction**

Q18. Describe three applications for exception processing.

**1-During Database connection**

**2- During Input from user**

**3- During mathematics operation like -division with 0**

Q19. What happens if you don&#39;t do something extra to treat an exception?

**It will halt the program and wont give user freidnly message**

Q20. What are your options for recovering from an exception in your script?

**Need to write the logic to gracefully handle it or mentioned the proper exception in except block**

Q21. Describe two methods for triggering exceptions in your script.

Def method1():

def reciprocal( num1 ):   
 try:   
 reci = 1 / num1   
 except ZeroDivisionError:   
 print( "We cannot divide by zero" )   
 else:   
 print ( reci )   
reciprocal( 4 )   
reciprocal( 0 )

def divide(x, y):  
 try:  
 result = x // y  
 print(result)  
 except ZeroDivisionError:  
 print("dividing by zero ")  
  
  
divide(3, 2)

Q22. Identify two methods for specifying actions to be executed at termination time, regardless of

whether or not an exception exists

**def method1():**

**try:**

**catch:**

**final:**

**def method2():**

**try:**

**catch:**

**else:**

**final:**

Q23. What is the purpose of the try statement?

**If any statement found exception they try statement ask cursor to move into exception bloc**

Q24. What are the two most popular try statement variations?

**Try :**

**Except:**

**Final:**

Q25. What is the purpose of the raise statement?

**It is used to raise an exception explicitly**

Q26. What does the assert statement do, and what other statement is it like?

**Assert statement is used to check assertion**

Q27. What is the purpose of the with/as argument, and what other statement is it like?

With argument used in filehandling and as keyword used for alias

**With (file.txt) as f:**

**f.read()**

Q28. What are \*args, \*\*kwargs?

**It’s a type of argument which helps to give n no of argument in function**

**Args is list argument where as kwargs is a dictonary type of argument**

**Def myfunc(\*args,\*\*kwargs):**

**Print (args)**

**Print (kwargs)**

Q29. How can I pass optional or keyword parameters from one function to another?

**Def func(x,y=1,z=2)**

**Print(x,y,z)**

**Func(1)**

**Func(1,2,3)**

**Func(1,2)**

Q30. What are Lambda Functions?

**Lambda func are function who does not has any name**

**lambdaFunc=Lambda x,y:x+y**

**print(lambdaFunc(1,2))**

Q31. Explain Inheritance in Python with an example?

Inheritance is mechanism to reuse the method written in parent class instead of writing in current class . It is one of OOPS concept

**Class A:**

**Def func():**

**Print(“inside class A”)**

**Class B(A):**

**Pass**

**BObj=B()**

**BObj.func()**

Q32. Suppose class C inherits from classes A and B as class C(A,B).Classes A and B both have their own versions of method func(). If we call func() from an object of class C, which version gets invoked?

**Class A function would get invoked**

Q33. Which methods/functions do we use to determine the type of instance and inheritance?

**X=[1,2,,3]**

**Print(type(x))**

**Inheritance—**

**Issubclass(<childclass>,<parentclass>) –return true if the subclass is child of parent otherwise return false**

**Isinstance(obj,classname)—return true if the object is type of class otherwise return false**

Q34.Explain the use of the 'nonlocal' keyword in Python.

**It is used to reference the variable which is outside of nested func. It used mostly in nested func.Instead of creating new variable with id it change the value in original id**

Q35. What is the global keyword?

**If any variable is defined outside of the method or outside of its scope and you would change its value then you need to defined that variable inside method using global and then you can change the value**

**Ex-**

**X=10**

**Def changeMyValue():**

**Global x;**

**X=5**

**Print(x)**