**Python OOPs Assignment**

Q1. What is the purpose of Python's OOP?

Ans => OOP concept uses class and object to achieve inheritance, abstraction, polymorphism and encapsulation.

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

Ans => inheritance search look for an attribute in the instance of an object.

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

Ans =>Class object is a blueprint of the class and instance object is a virtual class.

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

Ans =>Because it represents the class object itself.

Q5. What is the purpose of the init method?

Ans =>Lets the class initialize the object’s attributes.

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

Ans =>Call the class using the class name and add the arguments

Q7. What is the process for creating a class?

Ans =>Use class keyword to create a class.

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

Ans =>A superclass is a class from which the subclasses can be created.

Q9. What is the relationship between classes and modules?

Ans =>A module in python is to organise the code where class and function are present.

Q10. How do you make instances and classes?

Ans => Call the class using the class name and add the arguments

Q11. Where and how should be class attributes created?

Ans =>Class attributes should be created in class itself.

Q12. Where and how are instance attributes created?

Ans => instance attributes created in the contructor of a class.

Q13. What does the term "self" in a Python class mean?

Ans => self means the own class object itself.

Q14. How does a Python class handle operator overloading?

Ans => Python class handle operator overloading by overriding the method with specific operator.

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

Ans =>when you are making the new class that falls into the abstract class.

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

Ans => (+)Operator

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

Ans =>Inheritance and polymorphism.

Q18. Describe three applications for exception processing.

Ans =>Syntax, Exception and logical

Q19. What happens if you don't do something extra to treat an exception?

Ans =>The code could crash.

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

Ans =>Using Try and Except.

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

Ans =>Try -This catches the exception raised by program

Raise- Triggers an exception manually using custom exception.

Q22. Identify two methods for specifying actions to be executed at termination time, regardless of  
whether or not an exception exists.

Ans => Finally and Raise

Q23. What is the purpose of the try statement?

Ans => To the catch the exception raised by the program

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

Ans =>Else and Finally

Q25. What is the purpose of the raise statement?

Ans =>Used to raise the custom exception.

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

Ans =>Lets you test if a condition in your program if true, if not the program will raise an assertion error.

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

Ans => With statement is a replacement of Try-Finally error handling.

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

Ans => \*args – Used for non-keyword arguments.

\*\*kwargs – Used for keyword arguments.

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

Ans =>Using assignment operator and assign a value in the parameter.

Q30. What are Lambda Functions?

Ans => It is an anonymous function that can take any number of arguments and result only one expression.

Q31. Explain Inheritance in Python with an example?

Ans =>Inheritance is using a class attributes or method in their subclass.

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?

Ans => fun() in Class A

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

Ans =>isinstance() function.

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

Ans => Used to work with the variables inside a nested functions, where the variable should not work with the inner function

Q35. What is the global keyword?

Ans => Global keyword allows to modify the variable outside the scope.