AKURUT JOSELINE

S23B13/008

B24300

NO.1)

Object Oriented Programming is a programming pattern that models real world entities as objects. These objects have attributes and methods. The following shows the main pillars of object oriented programming.

Encapsulation this principle involves bundling data and its associated methods into a single unit while hiding the internal implementation details thus protecting data integrity and modularity.

Inheritance allows classes to inherit properties and methods from other classes and creating a relationship thus promotes code reuse and enables the creation of more complex classes.

Polymorphism refers to the ability of objects of different classes to be treated as if they were of the same type. Polymorphism can be achieved through method overriding and overloading making code more flexible and adaptable.

Abstraction focuses on the essential characteristics of an object and ignoring unnecessary details. This helps in creating simplified models of complex real world entities hence making code easier to understand and manage.

These pillars work together to create well-structured and maintainable software. By encapsulating data, using inheritance for code reuse, employing polymorphism for flexibility and applying abstraction for clarity, OOP provides a powerful framework for building robust and efficient applications.

NO.2

In Python, a constructor is a special method that is automatically called when an object of a class is created hence initializes the attributes of the object with appropriate values and ensures that the object is in a valid state when it comes into existence.

The init method is the default constructor in Python. It's a special method that begins with double underscores and ends with double underscores. When a new object of a class is created, the init method is called on automatically passing the necessary arguments to initialize the object's attributes.

Constructors in Python are essential for creating and initializing objects. The init method provides a convenient way to set the initial values of an object's attributes ensuring that it is ready for use

NO.3)

Class variables are variables that are defined inside a class but outside any init method. All objects created from the same class share the same class variable while Instance variables are variables that are defined inside an init method.

For example a car factory, the factory has a blueprint that outlines the common features of all cars produced such as number of wheels, engine type. This blueprint contains class variables that are shared by all cars the blueprint might specify that all cars should have 4 wheels.

Therefore each car that rolls off the assembly line is an instance of the car class and each car has its own unique instance variables that differentiate it from other cars for example one car might have a red color while another might be blue so the color would be an instance variable specific to that particular car. In conclusion, class variables are shared among all instances of a class while instance variables are unique to each object.

NO.4)

A class method is a method that is bound to a class rather than an instance of the class. Class methods are typically used to implement functionality that is related to the class as a whole rather than to individual instances.

A static method is a method that belongs to a class but it does not have access to the instance variables. Static methods are essentially functions that are defined inside a class but they do not depend on the class.

Class methods have access to the class variables, while instance methods have access to the instance variables hence Static methods do not have access to either.

NO.5)

In a real-world scenario, using class variables would be more appropriate than using instance variables when modeling a university's student enrollment system.

A university has multiple departments each offering various courses and the university wants to keep track of the total number of students enrolled in each course across all departments. Here, they can create a Course class with a class variable total enrolled to keep track of the total number of students enrolled in each course.

Class Variables are Suitable because there is efficient Updates when a new student enrolls in a course, they can simply add the total enrolled class variable. This ensures that the total enrollment count is updated efficiently without having to iterate over all instances of the Course class.

There is easy access of the total enrolled class variable that can be accessed from any instance of the Course class making it easy to retrieve the total enrollment count for a particular course.