**RAJAT NAIR - ACE11670**

**OOPs stands for OBJECT ORIENTED PROGRAMMING LANGUAGE**

The main aim of object-oriented programming is to implement real-world entities, for example, object, classes, abstraction, inheritance, polymorphism, etc.

**Why do we need OOPs?**

* Building programs from ready-made modules saves time and makes coding easier.
* In OOPs, we solve a big problem by dividing it into smaller parts, handling one part (object) at a time.
* Code reusability.

**What do you mean by Object?**

Any entity that has state and behavior is known as an object.

**Example:** A cat is an object because it has states like color, name, breed, etc. as well as behaviors like wagging the tail, meowing, eating, etc.

**What do you mean by Class?**

A class in Java is a set of objects which shares common characteristics/ behavior and common properties/ attributes. It is a user-defined blueprint or prototype from which objects are created.

Pillars of OOPs

* Abstraction
* Encapsulation
* Inheritance
* Polymorphism

**What is Abstraction?**

Using simple things to represent complexity. We all know how to turn the PC on, but we don’t need to know how it works. Abstraction means simple things like objects, classes and variables represent more complex underlying code and data. This is important because it lets you avoid repeating the same work multiple times.

**What is Encapsulation?**

The practice of keeping fields within a class private, then providing access to those fields via public methods. Encapsulation is a protective barrier that keeps the data and code safe within the class itself. We can then reuse objects like code components or variables without allowing open access to the data system-wide.

**What is Inheritance?**

Inheritance is a mechanism of acquiring the features and behaviors of a class by another class. The class whose members are inherited is called the base class, and the class that inherits those members is called the derived class.

**What is Polymorphism?**

Polymorphism describes a situation in which something occurs in several different forms. Polymorphism describes the concept that you can access objects of different types through the same interface. Each type can provide its own independent implementation of this interface.

**What are Access Modifiers?**

**1. Private** - The methods or data members declared as private are accessible only within the class in which they are declared.

**2. Protected** - The methods or data members declared as protected are accessible within the same package or subclasses in different packages.

**3. Public** - Classes, methods, or data members that are declared as public are accessible from everywhere in the program. There is no restriction on the scope of public data members.