**Saikumar G**

**D and a batch**

**Object-Oriented Programming (OOP)**

The main aim of object-oriented programming (OOP) is to design software that is more modular, reusable, and easier to maintain by organizing software into objects that represent real-world entities and their interactions.

why do we need oops

* Code reusability
* Abstraction
* Encapsulation
* Inheritance
* Polymorphism

**Object:**

An object is an instance of a class in object-oriented programming, embodying both data and the methods that operate on that data.

**Class:**

A class is a blueprint or template for creating objects in object-oriented programming, defining their attributes and behaviours. It serves as a logical entity encapsulating data and functions that operate on that data.

**Main concepts of OOPS:**

* Encapsulation
* Inheritance
* Polymorphism
* Abstraction

**Encapsulation:**

Encapsulation is a fundamental principle in object-oriented programming, involving bundling data and the methods that operate on that data within a single unit, called an object. This concept restricts access to the internal details of the object, allowing the implementation to be hidden.

**Inheritance:**

Inheritance is a core concept in object-oriented programming where a new class, called a subclass or derived class, can inherit attributes and behaviours from an existing class, known as the superclass or base class. This promotes code reuse by allowing the subclass to access and extend the functionalities of the superclass.

**Polymorphism:**

Polymorphism in object-oriented programming allows objects of different types to be treated as objects of a common base type. This enables a single interface to represent multiple underlying data types, promoting flexibility and adaptability in code. Polymorphism is achieved through method overloading and method overriding.

**Abstraction:**

Abstraction in programming involves simplifying complex systems by modelling classes based on their essential features while hiding unnecessary details. It allows developers to focus on relevant aspects of an object or system, creating a high-level representation. Abstraction is achieved through the creation of abstract classes and interfaces, defining a common structure without specifying the implementation.