Day 8 – Object-Oriented Programming Concepts – Part 1

Objective:

To understand the basic principles of Object-Oriented Programming (OOP) in Java, focusing on classes, objects, and constructors.

Content:

Today, I learned about the foundation of Object-Oriented Programming in Java.
OOP allows developers to model real-world entities as objects, making programs more modular, reusable, and easier to maintain.

1. Classes and Objects

- A **class** is a blueprint or template that defines attributes (variables) and behaviors (methods).
- An **object** is an instance of a class that has its own values for the class's attributes.

Example:

```
class Student {
    String name;
    int age;

    void display() {
        System.out.println("Name: " + name + ", Age: " + age);
    }
}
public class Main {
    public static void main(String[] args) {
        Student s1 = new Student();  // object creation
        s1.name = "Husanpreet";
        s1.age = 20;
        s1.display();
    }}
```

Output:

```
Name: Husanpreet, Age: 20
```

---

2. Constructors

Constructors are special methods used to initialize objects.
They have the same name as the class and are automatically called when an object is created.

Types of Constructors:

- **Default Constructor:** No parameters, assigns default values.
- **Parameterized Constructor:** Accepts arguments to initialize variables.
- **Copy Constructor:** Initializes an object using another object's data.

Example:

```
class Student {
    String name;
    int age;
    Student(String n, int a) {   // parameterized constructor
        name = n;
        age = a;
    }
}
```

---

3. Access Modifiers

Access modifiers define the visibility of class members.

- **public:** Accessible from anywhere
- **private:** Accessible only within the class
- **protected:** Accessible within the package and subclasses
- **default:** Accessible only within the same package

Learning Outcome:

Understood the concept of classes, objects, and constructors in Java.
Learned how to initialize and access class members using objects.
Gained practical knowledge of object creation, access modifiers, and the role of constructors in OOP.