OOP Lecture Notes: Class, Variables, Objects, Methods, Constructors

Class

- A class is a template or blueprint used to create objects.
- It defines properties (variables) and behaviors (methods) common to all objects of that type.
- Think of it as a design plan—like a house blueprint. You can build many houses (objects) using that same blueprint (class).
- In Java, everything must be inside a class.
- Example:

```
class Car {
   String color;
   void drive() {
      System.out.println("Driving...");
   }
}
```

Variables

- A variable is a named container that holds a value which can change during execution.
- Every variable has:
  - A data type (e.g., int, String)
  - A name (e.g., age)
  - A value (e.g., 25)
- Follows Java naming rules—camelCase is recommended.
- Example:

```
int age = 25;
String name = "Nimesha";
```

Objects

- An object is an instance of a class—it's what you get when you build something from the class blueprint.
- An object has:
  - State – defined by variables (e.g., color of a car)
  - Behavior – defined by methods (e.g., drive, stop)
  - Identity – its unique name or reference
- Example:

```
Car myCar = new Car();
myCar.color = "Red";
myCar.drive();
```

Methods

- A method is a block of code that performs a task or behavior.
- They help us reuse code and improve program organization.
- Types:
  - Predefined methods – come with Java libraries (System.out.println())
  - User-defined methods – created by the programmer
- Method structure:

```
public void greet() {
   System.out.println("Hello!");
}
```

- Calling the method:

```
greet();
```

Constructors

- A constructor is a special method that is automatically called when an object is created.
- Used to initialize object values.
- Has the same name as the class and no return type.
- Types of constructors:
  - Default constructor – no parameters
  - Parameterized constructor – accepts arguments
  - Copy constructor – copies values from another object
- Example:

```
class Car {
   String color;

   Car(String c) {
      color = c;
   }
}
```