

## OOP'S CH 2 : Objects and Classes

- Java is an object-oriented programming language. Objects and classes are the basic building blocks of OOP.
- An object is a basic unit of OOP and represents the real life entities like a house, a tree
- Classes create objects and objects use methods to communicate between them.

### **Object in Java:**

- **An object is** A real-world entity that has state and behaviour.

### **Class in Java:**

- A class is a user-defined type which groups data members.
- In Java language the data members are called fields and the functions are called methods.

### **CONSTRUCTORS :**

- A constructor is a special method of a class in Java programming that initializes an object.
- Constructors have the same name as the class itself. A constructor is automatically called when an object is created.
- Constructors can be classified into two types, default constructors and parameterized constructors.

#### Default Constructor:

- The constructor which does not accept any argument is called default constructor.
- In other word, when the object is created Java creates a no-argument constructor automatically known as default constructor.
- It does not contain any parameters nor does it contain any statements in its body.

#### Parameterized Constructor:

- A constructor that has parameters is known as parameterized constructor.
- Parameterized constructor is used to provide different values to the distinct objects.

### **CONSTRUCTOR OVERLOADING :**

- Constructor having the same name with different parameter list is called as constructor overloading.

### **USE OF 'this' KEYWORD :**

- 'this' keyword can be used to refer current class instance variable.
- 'this' keyword can be used to invoke current class constructor.
- The 'this' is used inside the method or constructor to refer its own object

### **STATIC BLOCK, STATIC FIELDS AND METHODS :**

- In Java basically, class contains variables called **instance variables** and methods called **instance method**.

#### **Static Block :**

- A class can contain code in a static block that does not exist within a method body. Static code block executes only once when the class is loaded.
- A static block is used to initialize static attributes. Static blocks are also called static initialization blocks.
- A static initialization block is a normal block of code enclosed in braces, { }, and preceded by the static keyword. static {

//whatever code is needed for initialization goes here, ...

}

- A class can have any number of static initialization blocks, and they can appear anywhere in the class body.

#### **Static Field :**

- A static field of a class is referred as a class variable .
- A static field gets memory only once for the whole class.
- To declare a static field It's syntax is, **static datatype fieldName;**

#### Static Methods :

- If you apply static keyword with any method, it is known as static method.
- A static method belongs to the class.
- Static method can access static data member and can change the value of it.
- A static method is also called class method as it is associated with a class and not with individual instance of the class.
- A static method cannot access non-static method.

### Object Class :

- The object class is the parent class of all the classes in Java by default. In other words, it is the topmost class of java.
- The Object class provides some common behaviours to all the objects such as object can be compared, object can be cloned, object can be notified etc.
- Some **methods** of the **object** class are:  
Boolean equals(Object obj) , protected Object clone() , int hashCode() , void notify() . void notifyAll() .

### String Class :

- The strings in Java are treated as objects of type 'String' class. This class is present in the package java.lang.
- This package contains two string classes String class and StringBuffer class.
- The string class is used when we work with the string which cannot change whereas StringBuffer class is used when we want to manipulate the contents of the string.
- When we create object of String Class they are designed to be immutable.
- We can use + operator to overload for string objects. Only two operators i.e. '+' & '+=' are overloaded for string classes.

**Example:** String str = "Kal" + "pa" + "na";

**Output:** Kalpana

### StringBuffer Class :

- It is a peer class which provides the functionality of strings. The string generally represents fixed length, immutable character sequence whereas StringBuffer represents growable and writeable character sequences.
- StringBuffer may have some characters
- Java generally manipulate the strings using + as overloaded operator. StringBuffer class in Java is used to created mutable (modifiable) string.
- The StringBuffer class in Java is same as String class except it is mutable i.e., it can be changed.

**Advantages of StringBuffer Class:** 1. Alternative to String class. 2. Can be used wherever a string is used. 3. More flexible than String.

### Difference between String and StringBuffer:

- String objects are constants and immutable whereas StringBuffer objects are not constants and immutable.
- StringBuffer Class supports growable and modifiable string whereas String class supports constant strings.
- Strings once created we cannot modify them Whereas StingBuffer objects after creation also can be able to delete or append any characteres to it.
- String values are resolved at **run time** whereas StringBuffer values are resolved at **compile time**.

### WRAPPER CLASSES :

- A data type is to be converted into an object and then added to a Stack. For this conversion, the introduced wrapper classes.

### PACKAGES :

- A Java package is a mechanism for organizing Java classes.
- A package can be defined as "a group of similar types of classes, interface, enumeration and sub-packages".
- Packages are collection of classes and interface or packages act as container for classes.
- There are two types of packages:
  1. **Built-in Package:** Existing built-in Java packages like java.lang, java.util, java.sql etc.
  2. **User-define Package:** Java package created by user to categorized classes, interface and enumeration.

### Creating Packages :

- simply include a package command as the first statement in a Java source file.
- package pkg\_name;
- There are three ways to **access the package** from outside the package.
- **Using packagename:**
  - If we use package.\* then all the classes and interfaces of this package will be accessible but not subpackages.
  - The 'import' keyword is used to make the classes and interface of another package accessible to the current package.
- **Using packagename.classname:**
  - If we import package.classname then only declared class of this package will be accessible.

### User Defined Packages :

- The packages credited by user are called as user defined package.
- User defined packages generally represent programs data.

