**Object Oriented Programming Using - JAVA**

# PRACTICAL – 4

**Aim:** Write programs in Java to use Wrapper class of each primitive data Types.

**Theory:** Wrapper classes in Java are used to convert primitive data types into objects. Java provides a separate class for each primitive type, such as Integer for int, Double for double, Character for char, and so on. These classes are part of the **java.lang** package and come with useful methods for converting, comparing, and manipulating values.
In this practical, we used wrapper classes to demonstrate **boxing** (conversion from primitive to object). Wrapper classes are especially useful when working with data structures like collections, which only store objects.

**Code:**

```
public class WrapperDemo {
   public static void main(String[] args)
   {
      int intValue = 10;
      Integer wrappedInt = Integer.valueOf(intValue);

      double doubleValue = 20.5;
      Double wrappedDouble = Double.valueOf(doubleValue);

      boolean booleanValue = true;
      Boolean wrappedBoolean = Boolean.valueOf(booleanValue);

      char charValue = 'A';
      Character wrappedChar = Character.valueOf(charValue);

      System.out.println("Wrapped Integer    : " + wrappedInt);
      System.out.println("Wrapped Double     : " + wrappedDouble);
      System.out.println("Wrapped Boolean    : " + wrappedBoolean);
      System.out.println("Wrapped Character: " + wrappedChar);
   }

}
```

**Output:**

```
Wrapped Integer   : 10
Wrapped Double    : 20.5
Wrapped Boolean   : true
Wrapped Character: A
```

**Conclusion:** Through this practical, we understood how wrapper classes work and how they help in converting primitive data types to objects. We also learned about boxing, which make Java code more flexible and object-oriented.