Date. ________
Page No. ________

Assignment - 01

Q1. The most important features of Java programming language are:

• Simple →
It is a simple programming language as its syntax is based on C++.

Eg.
```
public class name {
        public static void main (String args[]) {
            System.out.println("Hi!");
        }
}
```

• Object - Oriented →
OOPs in Java is to improve code readability, reusability by defining the code efficiently.

• Portable →
Because it can be executed on several platforms for e.g. Windows, Linux, Mac/OS, etc. The code is compiled by the compiler & converted into bytecode and since, bytecode is platform independent, it can be executed on several platforms, i.e. Write Once, Run Anywhere (WORA).

• Platform Independent
• Architectural Neutral
• Interpreted
• High Performance → Java bytecode is close to native code.

Date.
Page No.

Q2. Data Types:
They specify the different sizes & values that can be stored in the variable. There are 2 types:

• Primitive Datatypes →

	Boolean → true/false
(16 Bit)	Char → To store characters
(32 Bit)	Integer → For integral values
(32 Bit)	Float → For floating point no.
(64 Bit)	Long → For integers with range > int
(64 Bit)	Double → Floating point no. with m ra
(16 Bit)	Short →
(8 Bit)	Byte

• Non - Primitive Data types →
Classes
Interfaces
Arrays

Q3. Autoboxing is the automatic conversion of a primitive value (an int, for e.g.) into an object of corresponding wrapper class (Integer) ~~converting a primitive values (an int into~~
The Java compiler applies autoboxing when a primitive value is:
• Passed as a parameter to a method that expects an object of the corresponding wrapper class.
! Assigned to a variable of the corresponding wrapper class.

Date.
Page No.

Eg.

```
class Boxing {
    public static void main (String args [])  {
        int a = 50;
        Integer a2 = new Integer (a);
        Integer a3 = 5;
        System.out.println (a2+"  "+a3);
    }
}
```

Unboxing is the conversion of an object of a wrapper type (Integer) to its corresponding primitive (int) value.

The Java Compiler applies unboxing when an object of a wrapper class is:

• Passed as a parameter to a method that expects a value of the corresponding primitive type.

• Assigned to a variable of the corresponding primitive type.

Eg.

```
class Unboxing {
        public static void main (String args [])  {
            Integer i = new Integer (50);
            int a = i;
            System.out.println (a);
        }
}
```

Date.
Page No.

Q4
```
class Dimension
{
   public static void main(String args[])
   {
      int a[] = {10, 20, 30, 40, 50};
      System.out.println("1D Array elements are:");
      for(int i=0; i<a.length; i++)
            System.out.println(a[i]);
      int a[][] = {{10,20},{30,40},{50,60}};
      System.out.println("2D Array elements are:");
       for(int i=0; i<3; i++)
           for(int j=0; j<2; j++)
              System.out.println(a[i][j]);
   }
}
```

Q5: The String is immutable, so its value can't be changed. If the string doesn't remain immutable, any hacker can cause a security issue in the application by changing the reference value.
The String is safe for multithreading because of its immutableness. Different threads can access a single "String instance". It removes the synchronization for thread safety because we make strings thread-safe implicity

Date. ________
Page No. ________

E.g.

```
class TestImmutable
{
    public static void main (String args[])
    {
        String s = "Parth";
        s.concat ("Sarthi");        //appends at the end of string
        System.out.println (s);     //prints "Parth"
    }
}
```

Q6. A Jagged Array is an array of arrays such that member arrays can be of different sizes i.e. we can create a 2D array but with a variable no. of columns in each row
E.g.

```
class Main {
    public static void main (String args[]) {
        int arr[][] = new int [2][];
        arr[0] = new int[3];
        arr[1] = new int[2];
        int count = 0;
        for (int i=0; i< arr.length; i++)
            for (int j=0; j< arr[i].length; i++)
                arr[i][j] = count++;
        System.out.println("Contents of 2D Array :");
        for (int i=0; i< arr.length; i++) {
            for (int j=0; j< arr[i].length; j++)
                System.out.println(arr[i][j] + " ");
            System.out.println();
        }
    }
```

Date.
Page No.

Q7

```
~~import java.io.*;~~
~~class lexicographic~~
~~{~~
~~   public static void main(String args[]) throws IOException~~
~~   {~~
~~      BufferedReader br = new BufferedReader(new InputStreamReader~~
~~(System.in));~~
~~      string s = br.readLine();~~
~~      int k = Integer.parseInt(br.readLine());~~
```

Q7.

```
import java.io.*;
class lexicographically
{
   public string getSmallestAndLargest(string s, int k)
   {
      String s1 = null, s2 = null, s3 = null;
      s1 = s2 = s.substring(0, k);
      for(int i = 1; i < s.length() - 2; i++)
      {
         ~~String~~ s3 =
         ~~if (s.substring(i, i+3)).compareTo(s1)~~
         if (s3.compareTo(s1) < 0)
            s1 = s3;
         else if (s3.compareTo(s2) > 0)
               s2 = s3;
      }
      System.out.println(s1);
      System.out.println(s2);
   }
}
```