Name : Nikam Khushi Sahebrao.
PRN : 22620004
Branch : Information Technology
Batch : S1
Subject : Java Programming

ASSIGNMENT 01

- Answer the following.

1. Describe JDK, JRE, and JVM.

Ans :

I) JDK :

- JDK stands for Java Development Kit.
- It is a software development kit that includes the Java Runtime Environment (JRE), as well as tools and utilities for developing, testing, and debugging Java applications.
- The JDK includes the Java compiler, which is used to compile Java source code into bytecode, as well as other tools such as the Java Virtual Machine Debugger (JDB), Java Archive (JAR) tool, and Java documentation generator (Javadoc).

II) JRE :

- JRE stands for Java Runtime Environment.
- It is a software package that includes the JVM, class libraries, and other necessary components to run Java applications on a computer.
- When a Java program is written, it is compiled into bytecode, which is then executed by the JVM.
- The JRE provides the necessary runtime environment for the JVM to execute the bytecode. It also includes the class libraries that provide the core functionality of the Java language, such as I/O operations, network communication, and graphics.

III) JVM :

- JVM stands for Java Virtual Machine.
- It is a virtual machine that enables a computer to run Java bytecode, which is the compiled form of Java source code.
- When a Java program is compiled, it is translated into bytecode, which can be executed on any platform that has a JVM installed.
- The JVM is responsible for interpreting the bytecode and executing the program. It provides a layer of abstraction between the Java program and the underlying operating

system, allowing the program to be run on different platforms without modification.

2. Differentiate between C, C++, and JAVA.

C	C++	JAVA
C programming language was developed by Dennis Ritchie at Bell Labs in the early 1970s.	C++ programming language was developed by Bjarne Stroustrup at Bell Labs in the early 1980s.	Java programming language was developed by James Gosling & his team at Sun Microsystems in the mid-1990s.
C has a simple syntax with a small number of keywords and constructs.	C++ has a more complex syntax with more features for object-oriented programming.	Java has more complex syntax with more features for object-oriented programming.
Requires explicit memory management.	Require explicit memory management.	Has automatic memory management through garbage collection.
It is a procedural programming language	It is an Object-oriented programming language.	It is a Purely Object-oriented programming language.

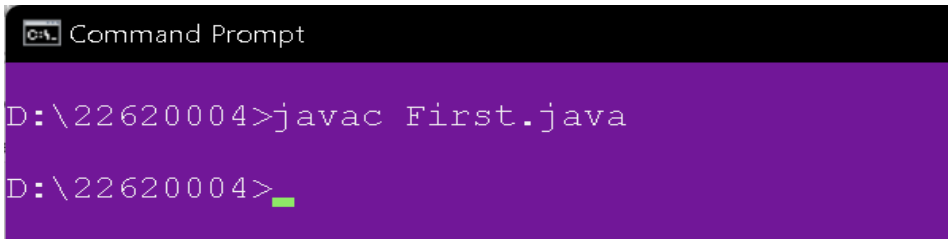
3. Explain a simple hello word program in Java. Explain the compilation and execution of the java program.

Ans :

1) Create a file with the First.java file(standard practice to save your file with your java class_name.
2) Write the following code in it.

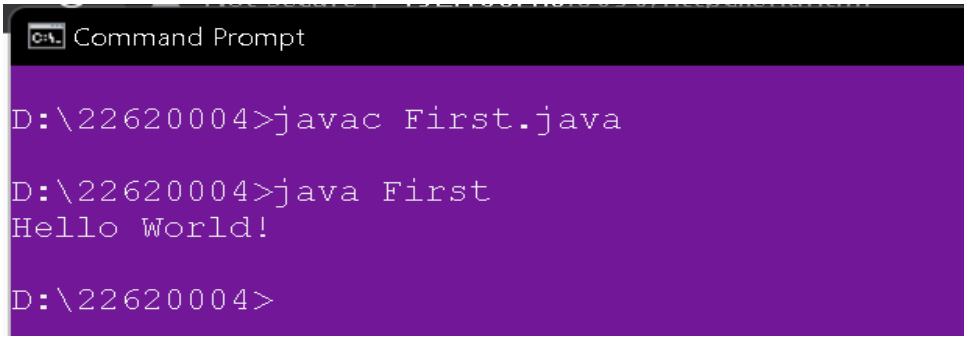
```
public class First{
    public static void main(String args[])
    {
        System.out.println("Hello World!");
    }
}
```

3) Now Save the File.
4) Open the Command Prompt and compile the java program.



The above command will create the .class file (that .class file contains the bytecode)

5) Now run the "java First" command :



4. Write a java program for reading input of various data types from users using the scanner class.

Program :

```
import java.util.*;
public class First{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int prn;
        String name;
        float marks;
        System.out.print("Enter PRN :");
        prn=sc.nextInt();
        System.out.print("Enter Name :");
        name=sc.next();
        System.out.print("Enter Marks :");
        marks=sc.nextFloat();
        System.out.println("");
        System.out.println("PRN :"+prn);
        System.out.println("Name:"+name);
        System.out.println("Marks :"+marks);
```

```
    }
}
```

Output :

```
D:\22620004>javac First.java

D:\22620004>java First
Enter PRN :220001
Enter Name :saee
Enter Marks :94.4

PRN :220001
Name:saee
Marks :94.4
```

5. Write a Java program to convert seconds to hour, minute and seconds.

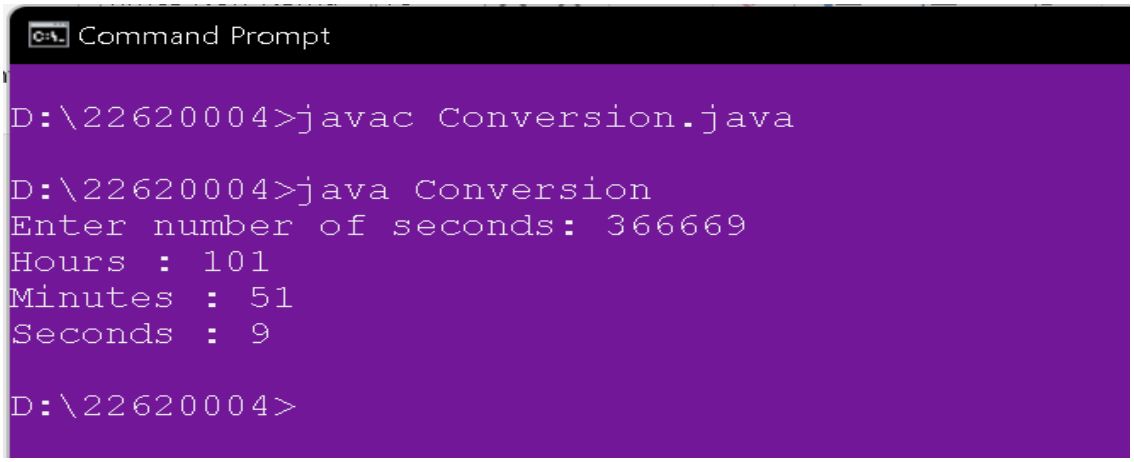
Program :

```
import java.util.Scanner;
public class Conversion {
  public int hours,minutes,remainingSeconds,seconds;
  public void convert()
  {
    hours = seconds / 3600;
    minutes = (seconds % 3600) / 60;
    remainingSeconds = seconds % 60;
    System.out.println("Hours : "+ hours+ "\n" + "Minutes : " +minutes + "\n" +
"Seconds : "+ remainingSeconds );
  }

    public static void main(String[] args) {
        Conversion time=new Conversion();
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter number of seconds: ");
        time.seconds = scanner.nextInt();
        time.convert();

    }
}
```

Output :



6. Write a Java program to check if there is a 10 in a given array of integers.

Program :

```
import java.util.*;
public class Search {
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the size of array : ");
        int n=sc.nextInt();
        int []array=new int[n];
        for(int i=0;i<n;i++)
        {
            System.out.print("Element["+(i+1)+"] : ");
            array[i]=sc.nextInt();
        }
        int cnt=0;
        for(int i=0;i<n;i++)
        {
            if(array[i]==10){
            System.out.print("10 is present in array at location "+(i+1)+" !");
            cnt++;
            }
        }
        if(cnt==0)
        System.out.print("10 is not present in arrray");
```

```
    }
}
```

Output :

```
D:\22620004>javac Search.java

D:\22620004>java Search
Enter the size of array : 6
Element[1] : 120
Element[2] : 302
Element[3] : 10
Element[4] : 40
Element[5] : 69
Element[6] : 10
10 is present in array at location 3 !
10 is present in array at location 6 !

D:\22620004>
```

7. Write a program to calculate the factorial of a number. (The number is passed as the command-line argument whose factorial we need to calculate).

Program :

```
public class Factorial{
    public static void main(String args[])
    {
        int fact=Integer.parseInt(args[0]);
        int result=1;
        for(int i=1;i<=fact;i++)
        result=result*i;
        System.out.println("Factorial of "+fact+" is "+result);
    }
}
```

Output :

```
D:\22620004>javac Factorial.java

D:\22620004>java Factorial 6
Factorial of 6 is 720

D:\22620004>
```

8. Write a Java Program to find transpose of Matrix.

Program :

```
import java.util.*;
public class Transpose{
    public static void main(String args[])
    {
        int i,j;
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter number of Rows : ");
        i=sc.nextInt();
        System.out.print("Enter number of Columns : ");
        j=sc.nextInt();
        int [][] matrix=new int[i][j];
        for(int m=0;m<i;m++)
        {
            for(int n=0;n<j;n++)
            {
             matrix[m][n]=m+1;
            }
        }
        System.out.println("Original Matrix : ");
         for(int m=0;m<i;m++)
        {
            for(int n=0;n<j;n++)
            {
             System.out.print(" "+matrix[m][n]);
            }
            System.out.println();
        }

         int [][] Tmatrix=new int[j][i];
        for(int m=0;m<j;m++)
        {
            for(int n=0;n<i;n++)
            {
             Tmatrix[m][n]=matrix[n][m];
            }
        }
        System.out.println("Transpose Matrix : ");
        for(int m=0;m<j;m++)
        {
            for(int n=0;n<i;n++)
            {
             System.out.print(" "+Tmatrix[m][n]);
            }
            System.out.println();
        }
```

```
    }
}
```

Output :

```
D:\22620004>javac Transpose.java

D:\22620004>java Transpose
Enter number of Rows : 4
Enter number of Columns : 5
Original Matrix :
 1 1 1 1 1
 2 2 2 2 2
 3 3 3 3 3
 4 4 4 4 4
Transpose Matrix :
 1 2 3 4
 1 2 3 4
 1 2 3 4
 1 2 3 4
 1 2 3 4
```

9. Write a program to implement different types of constructors.

Program :

```
public class Constructor {
    String name;

    // Default constructor
    public Constructor() {
        name = "WCE";
    }

    // Parameterized constructor
    public Constructor(String value) {
        name = value;
    }

    // Copy constructor
    public Constructor(Constructor other) {
        name = other.name;
    }

    public static void main(String[] args) {
        // Creating objects using different constructors
        Constructor o1 = new Constructor();
        Constructor o2 = new Constructor("Sangli");
```

```
        Constructor o3 = new Constructor(o2);
        System.out.println("obj1.name = " + o1.name);
        System.out.println("obj2.name = " + o2.name);
        System.out.println("obj3.name = " + o3.name);
    }
}
```

Output :

```
D:\22620004>javac Constructor.java

D:\22620004>java Constructor
obj1.name = WCE
obj2.name = Sangli
obj3.name = Sangli

D:\22620004>
```