**PRACTICAL # 02**

**OBJECT:**

**Introduction to Java, JVM, JSP.**

**A Java I/O program.**

**THEORY:**

Java is a high-level programming language originally developed by Sun Microsystems and released in 1995. Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX.

**Java Features**

1. Object-Oriented
2. Platform Independent
3. Secure
4. Robust
5. Architecture Independent
6. Portable
7. Dynamic
8. Interpreted
9. High Performance
10. Multithreaded
11. Distributed

**JVM**

JVM (Java Virtual Machine) is a specification that provides runtime environment in which Java **bytecode** can be executed.

JVMs are available for many hardware and software platforms. JVM, JRE and JDK are platform dependent because configuration of each OS differs. But, Java is platform independent.

The JVM Loads, verifies and executes the code and provides the runtime environment.

### **JRE**

JRE (Java Runtime Environment) provides the runtime environment. It is the implementation of JVM. It contains set of libraries and other files that JVM uses at runtime. The libraries in Java have jar extension.

### **JDK**

Java Development Kit contains JRE and the other tools required for Java program development. Two most important programs of JDK are javac(compiler) and java(run time environment).

**Program:**

The program below uses the classes **BufferedReader** and **InputStreamReader** to read user input from console. The value is then printed on the console. Save this program file as ***Io.java***

import java.io.BufferedReader;

import java.io.InputStreamReader;

class Io

{

public static void main(String[] args){

try{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

System.out.print("Enter String: ");

String s = br.readLine();

System.out.println(s);

int i = Integer.parseInt(br.readLine());

System.out.println(i);

}catch(Exception ex){

System.out.println(ex);

}

}

}

To compile and execute the code.

***javac Io.java***

***java Io***

By default, the BufferedReader object returns String object. If you need input of some other data-type, you need to parse the String to get that data-type value. The statement Integer.parseInt(br.readLine()) converts the String to integer and returns it.

Remember; to use the classes, import them in the code first. The BufferedReader and InputStreamReader classes are imported at the start of the program.

Another class used to get user input is Scanner available as java.util.Scanner. Here is a sample code to get user input.

Scanner scanner = new Scanner(System.in);

try {

int a = scanner.nextInt();

System.out.println(a);

}

catch (Exception ex) {

System.out.println(ex);

}

Note that the statement where you get user input must be within a try catch block. Since there may occur some IO exception, so Java requires the code to be encapsulated in try catch block.

**Activity:**

Perform the user input activity using the above two methods of the user input.

**REVIEW QUESTIONS**

1. What are the important features of Java?
2. What is the difference between JDK and JRE?
3. What is meant by buffer and stream?
4. What does the readLine() function of BufferedReader return?
5. Whys is it necessary to encapsulate the code that get user input in try catch block?