Experiment No.2
Accepting Input Through Keyboard
Date of Performance:
Date of Submission:

Aim: To apply basic programing for accepting input through keyboard.

Objective: To use the facility of java to read data from the keyboard for any program

Theory:

Java brings various Streams with its I/O package that helps the user perform all the Java input-output operations. These streams support all types of objects, data types, characters, files, etc. to fully execute the I/O operations. Input in Java can be with certain methods mentioned below in the article.

Methods to Take Input in Java

There are two ways by which we can take Java input from the user or from a file

1. BufferedReader Class
2. Scanner Class

Using BufferedReader Class for String Input In Java

It is a simple class that is used to read a sequence of characters. It has a simple function that reads a character another read which reads, an array of characters, and a readLine() function which reads a line.

InputStreamReader() is a function that converts the input stream of bytes into a stream of characters so that it can be read as BufferedReader expects a stream of characters. BufferedReader can throw checked Exceptions.

Using Scanner Class for Taking Input in Java

It is an advanced version of BufferedReader which was added in later versions of Java. The scanner can read formatted input. It has different functions for different types of data types.

The scanner is much easier to read as we don't have to write throws as there is no exception thrown by it.

It was added in later versions of Java

It contains predefined functions to read an Integer, Character, and other data types as well.

Syntax of Scanner class

Scanner scn = new Scanner(System.in);

Code:

```
import java.util.*;

public class ScannerClassExample1 {
   public s
tatic void main(String args[]) {
      String s = "Hello";
      Scanner scan = new Scanner(s);
      System.out.println("Boolean Result:" + scan.hasNext());
      System.out.println("String:" + scan.nextLine());
      System.out.println("-------Enter Your Details-----");
      Scanner in = new Scanner(System.in);
      System.out.print("Enter your name");
      String name = in.next();
      System.out.println("Name:" + name);
      System.out.print("Enter your age:");
      int i = in.nextInt();
      System.out.println("Age " + i);
      System.out.print("Enter your salary:");
      double d = in.nextDouble();
      System.out.println("Salary: " + d);

   }
}
```

OUTPUT:

```
Boolean Result:true
String:Hello this is java point
-------Enter Your Details-----
Enter your nameHarsh
Name:Harsh
Enter your age:19
Age 19
Enter your salary:200000
Salary: 200000.0
```

```
import java.io.FileReader;
import java.io.BufferedReader;

class Main {
  public static void main(String[] args) {
    char[] array = new char[100];

    try {

      FileReader file = new FileReader("input.txt");

      BufferedReader input = new BufferedReader(file);

      input.read(array);
      System.out.println("Data in the file");
      System.out.println(array);

      input.close();
    }

    catch (Exception e) {
      e.getStackTrace();
    }

  }

}
```

OUTPUT:

```
?) { javac Main.java } ; if ($?) { java Main }
Data in the file
Hello
```

Conclusion:

Comment on how you have used BufferedReader and Scanner Class for accepting user input

Both BufferedReader and Scanner have their own use cases. While BufferedReader is more efficient for reading large amounts of data, Scanner is more convenient for simple input parsing. It's important to handle exceptions properly and close these resources after usage to prevent resource leaks. Choose the appropriate class based on the requirements and complexity of the input reading tasks in your Java application.