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.

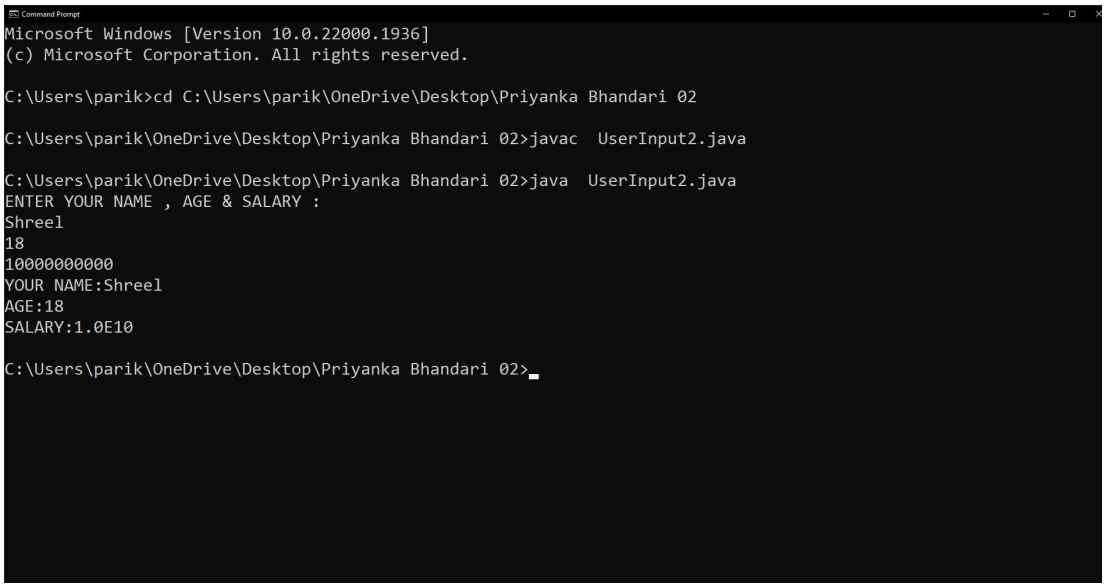
Syntax of Scanner class

Scanner scn = new Scanner(System.in);

Code:

1) Scanner class

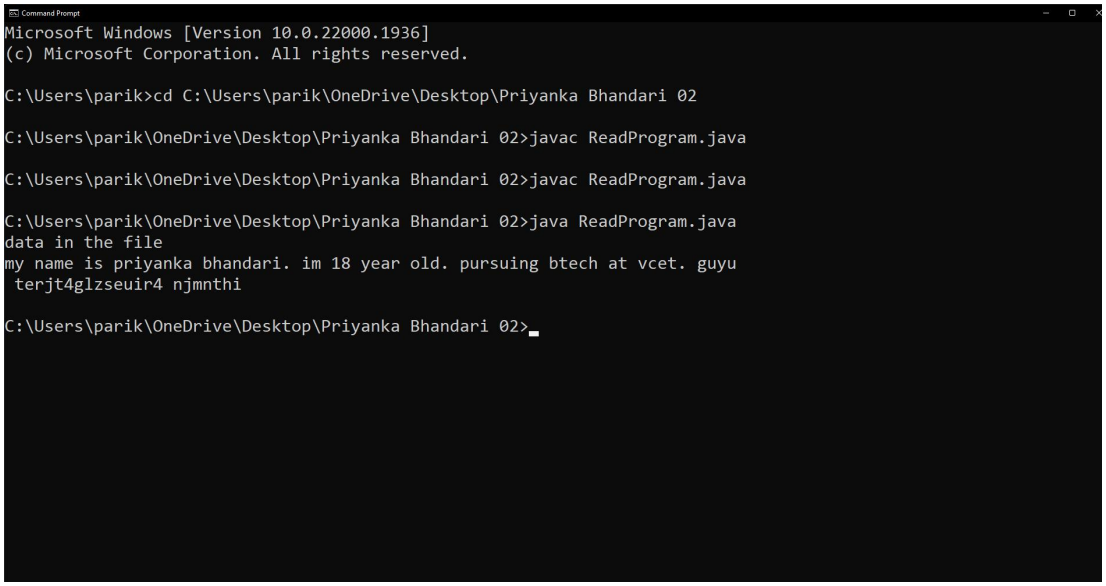
```
import  java.util.Scanner;

class UserInput2

{

        public static void main(String args[])

        {

        Scanner n=new Scanner(System.in);

        System.out.println("ENTER YOUR NAME , AGE & SALARY :");

        String name=n.nextLine();

        int age=n.nextInt();

        double salary=n.nextDouble();

        System.out.println("YOUR NAME:"+name);

        System.out.println("AGE:"+age);

        System.out.println("SALARY:"+salary);

        }

}
```



2) Buffer Reader class

```
import java.io.FileReader;

import java.io.BufferedReader;

class ReadProgram

{

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();

}

}

}
```



Conclusion:

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

In Java, both the BufferedReader and Scanner classes are commonly used for accepting user input from the command line or other input sources. Each of these classes has its own

advantages and use cases, and I'll provide some insights into how they can be used for this purpose.

BufferedReader:

BufferedReader is part of the java.io package and is primarily used for reading text from character input streams. It's efficient for reading large amounts of text efficiently.

Scanner:

The Scanner class is part of the java.util package and is a more high-level and user-friendly way to parse and tokenize input. It can be used for both reading from files and user input.