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.Code for Scanner Class:

```
import java.util.Scanner;
class ScannerInput
{
    public static void main(String args[])
        {
        Scanner s = new Scanner(System.in);
        System.out.println("What's your Name? : ");
        String str = s.nextLine();
        System.out.println("Name : " + str);
        System.out.println("What's your Age? : ");
        int age = s.nextInt();
        System.out.println("Your Age : " + age);
        System.out.println("What's your Salary (in USD)? : ");
        Double salary = s.nextDouble();
        System.out.println("Salary : " + "USD "+ salary);
        }
}
```

Output:

```
C:\Users\User.DESKTOP-VKOH6B7\Documents\Java Projects>javac ScannerInput.java

C:\Users\User.DESKTOP-VKOH6B7\Documents\Java Projects>java ScannerInput.java
What's your Name? :
Sunit
Name : Sunit
What's your Age? :
18
Your Age : 18
What's your Salary (in USD)? :
1232456
Salary : USD 1232456.0
```

Code for Buffer Reader Class:

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

```
import java.io.*;
class BufferedReaderInput
{
    public static void main(String args[])
        {
        char[] array = new char[25];
        try
        {
        FileReader file = new FileReader("file.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:

```
C:\Users\User.DESKTOP-VKOH6B7\Documents\Java Projects>javac BufferedReaderInput.java

C:\Users\User.DESKTOP-VKOH6B7\Documents\Java Projects>java BufferedReaderInput.java
Data in the file:
JAVA ROCKS!!!
```

Conclusion:

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

- I have used BufferedReader and Scanner class for accepting user input in different scenarios. BufferedReader is a class that reads text from a character-input stream, buffering characters for efficient reading. Scanner is a class that parses text from various sources and converts them into primitive types and strings. I have used BufferedReader to read a file line by line and store the contents in an array. I have created a BufferedReader object by passing a FileReader object as an

argument, and then used the readLine () method to read each line until the end of the file. I have used Scanner to read user input from the console and perform different operations based on the input. I have created a Scanner object by passing System.in as an argument, and then used the next (), nextInt (), nextDouble (), etc