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.

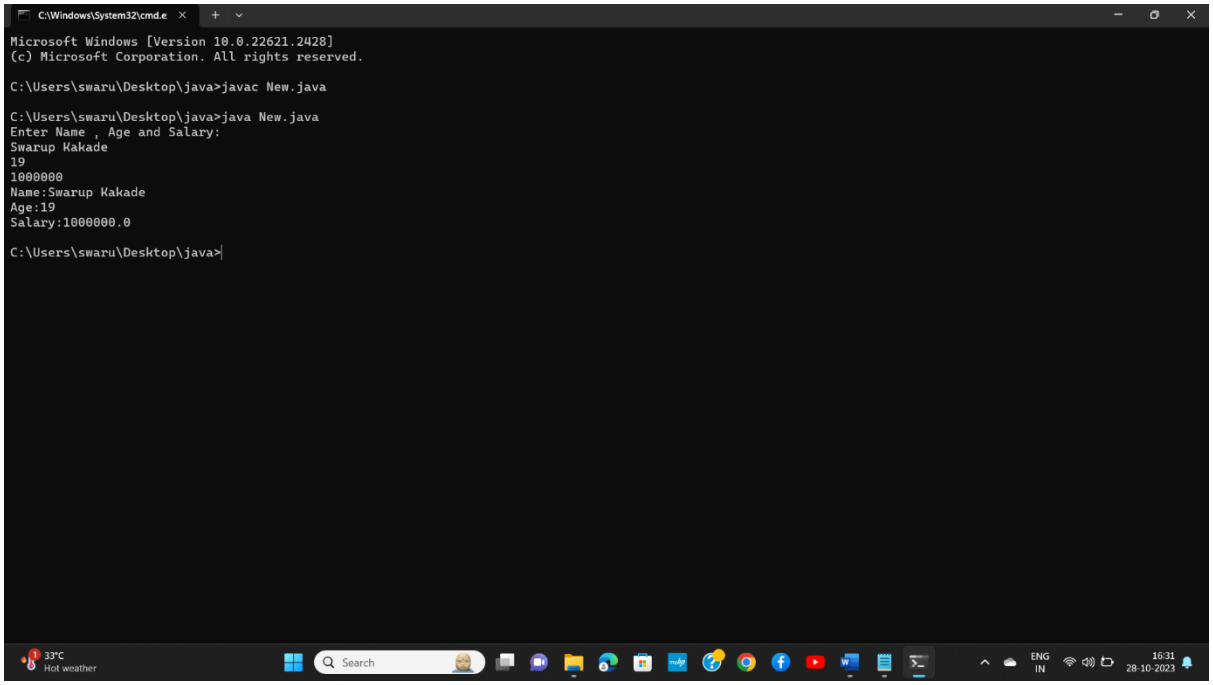
Syntax of Scanner class

Scanner scn = new Scanner(System.in);

Code:

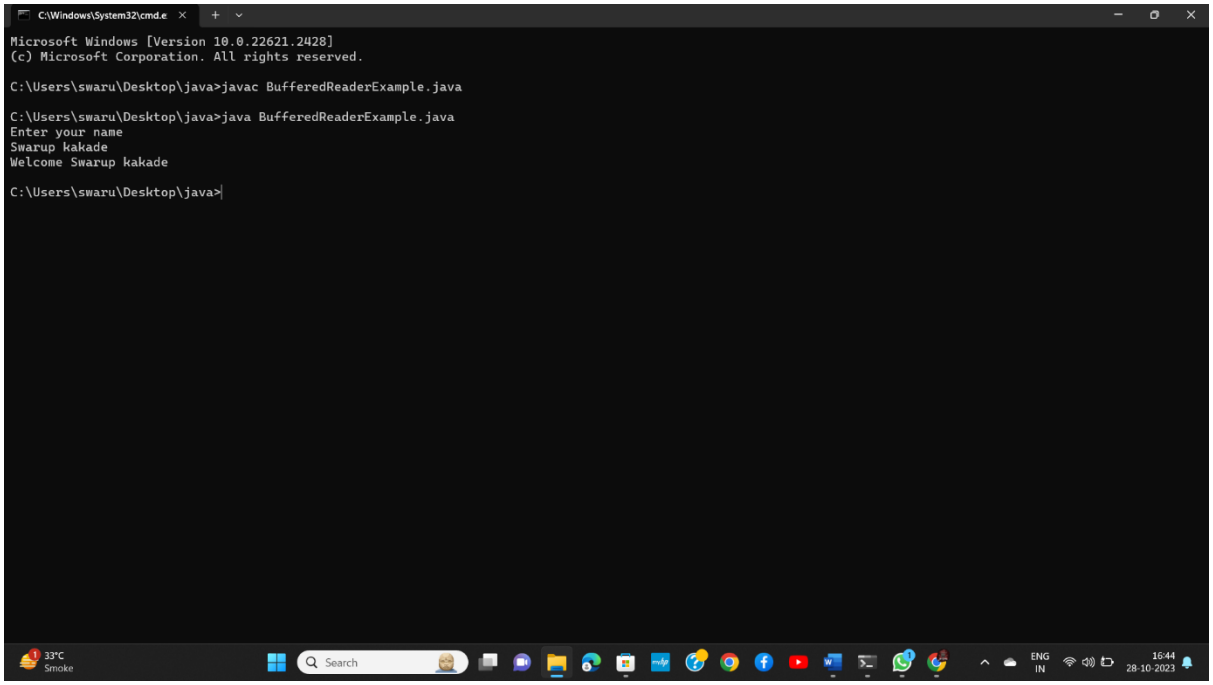
1} Scanner class

```
import java.util.Scanner;
class UserProgram
{
  public static void main(String args[])
   {
    Scanner a = new Scanner(System.in);
    System.out.println("Enter Name , Age and Salary:");
    String str = a.nextLine();
    int age = a.nextInt();
    Double salary = a.nextDouble();
    System.out.println("Name:" + str);
    System.out.println("Age:" + age);
    System.out.println("Salary:" + salary);
   }
}
```



2} Buffer reader class

```
package com.javatpoint;
import java.io.*;
public class BufferedReaderExample{
public static void main(String args[])throws Exception{
    InputStreamReader r=new InputStreamReader(System.in);
    BufferedReader br=new BufferedReader(r);
    System.out.println("Enter your name");
    String name=br.readLine();
    System.out.println("Welcome "+name);
}
}
```



Conclusion:

In the realm of Java programming, two commonly employed classes for obtaining user input are BufferedReader and Scanner. These classes each have their own unique advantages and are adept at catering to specific use cases when it comes to collecting data from users or other sources.

BufferedReader: Hailing from the java.io package, BufferedReader specializes in reading text from character input streams. It stands out for its efficiency when dealing with substantial amounts of text input, making it a preferred choice for such scenarios.

Scanner: On the other hand, the Scanner class, found in the java.util package, offers a more user-friendly and high-level approach to parsing and tokenizing input. This versatility allows it to be employed for reading from both files and user inputs, making it an excellent choice for a wide range of input processing tasks.