



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Aim: To apply basic programming 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);`



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

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);

    }
}
```

```
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>cd C:\Users\HP\OneDrive\Desktop\Charmi
C:\Users\HP\OneDrive\Desktop\Charmi>javac UserProgram.java
C:\Users\HP\OneDrive\Desktop\Charmi>java UserProgram.java
Enter Name , Age and Salary:
Charmi
18
80000
Name:Charmi
Age:18
Salary:80000.0
C:\Users\HP\OneDrive\Desktop\Charmi>
```



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

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);

    }

}
```

A screenshot of a Windows Command Prompt window. The window title is 'Command Prompt'. The text inside shows the following sequence of commands and output:
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.
C:\Users\HP>cd C:\Users\HP\OneDrive\Desktop\Charmi
C:\Users\HP\OneDrive\Desktop\Charmi>javac BufferedReaderExample.java
C:\Users\HP\OneDrive\Desktop\Charmi>java BufferedReaderExample.java
Enter your name
Charmi
Welcome Charmi
C:\Users\HP\OneDrive\Desktop\Charmi>
The background of the Command Prompt is dark grey. The window is open over a file explorer window showing the 'Charmi' folder on the Desktop.



Conclusion:

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.