



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

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



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



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

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. import java.util.Scanner;

class userinput{

    public static void main(String args[]){

        Scanner s=new Scanner(System.in);

        System.out.println("Enter your name");

        String name=s.nextLine();

        System.out.println("My name is"+name);

    }

}
```

OUTPUT:



```
C:\Windows\system32\cmd.e: X + v

C:\Users\ayush\Desktop\java>javac userinput.java

C:\Users\ayush\Desktop\java>java userinput.java
Enter your name
AYUSH
My name isAYUSH

C:\Users\ayush\Desktop\java>|
```

```
2. 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 file:");

System.out.println(array);

input.close();

}

catch(Exception e){
```



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

```
e.printStackTrace();
```

```
}
```

```
}
```

```
}
```

OUTPUT:

A screenshot of a Windows command prompt window. The title bar shows 'C:\Windows\system32\cmd.e' with a close button, a plus sign, and a dropdown arrow. The command prompt shows the following text:

```
C:\Users\ayush\Desktop\java>javac 2input.java  
C:\Users\ayush\Desktop\java>java 2input.java  
Data in file:  
hello,buddy!!  
C:\Users\ayush\Desktop\java>|
```

Conclusion:

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

I've used 'BufferedReader' and 'Scanner' classes in Java to handle user input in various programming assignments and projects. Here's how I've used them:

1. BufferedReader:

- I've used 'BufferedReader' when I needed to read user input character by character or line by line.
- It's particularly useful when working with text files or situations where input may span multiple lines.



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

- In assignments related to text processing, I've employed 'BufferedReader' to efficiently read and manipulate textual data.

2. Scanner:

- I've used 'Scanner' for its simplicity and versatility when handling user input.

- It's my go-to choice for reading various data types, such as integers, doubles, and strings, from the console.

- In assignments that require user interaction or calculations based on numeric inputs, I've relied on 'Scanner' to gather and process user data.

Both 'BufferedReader' and 'Scanner' have been valuable tools for acquiring user input, depending on the specific requirements of the programming task. They've made it easier to handle different input scenarios while developing Java applications as a student.