How to Take Input From User in Java?

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

- BufferedReader Class
- Scanner Class

1. 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.

Below is the implementation of the above approach:

- Java

```java
// Java Program for taking user
// input using BufferedReader Class
import java.io.*;

class GFG {

    // Main Method
    public static void main(String[] args)
        throws IOException
    {
        // Creating BufferedReader Object
        // InputStreamReader converts bytes to
        // stream of character
        BufferedReader bfn = new BufferedReader(
            new InputStreamReader(System.in));

        // String reading internally
        String str = bfn.readLine();
```

```
        // Integer reading internally
        int it = Integer.parseInt(bfn.readLine());

        // Printing String
        System.out.println("Entered String : " + str);

        // Printing Integer
        System.out.println("Entered Integer : " + it);
    }
}
```

Output

```
Mayank Solanki
888
Entered String : Mayank Solanki
Entered Integer : 888
```

Using Buffer Reader Class To Read the Input

Below is the implementation of the above approach:

- Java

```
/*package whatever //do not write package name here */

import java.io.*;
import java.io.BufferedReader;
import java.io.InputStreamReader;
class Easy {
    public static void main(String[] args)
    {
        // creating the instance of class BufferedReader
        BufferedReader reader = new BufferedReader(
            new InputStreamReader(System.in));
        String name;
        try {
            System.out.println("Enter your name");
```

```
            name = reader.readLine(); // taking string input
            System.out.println("Name=" + name);
        }
        catch (Exception e) {
        }
    }
}
```

Output:

```
Enter your name:
Geeks
Name=Geeks
```

2. 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);
```

Importing Scanner Class

'To use the Scanner we need to import the Scanner class from the util package as

```
import java.util.Scanner;
```

Inbuilt Scanner functions are as follows:

- **Integer:** nextInt()
- **Float:** nextFloat()
- **String :** next() and nextLine()

Hence, in the case of Integer and String in Scanner, we don't require parsing as we did require in BufferedReader.

- Java

```
// Java Program to show how to take
// input from user using Scanner Class

import java.util.*;
```

```java
class GFG {

    public static void main(String[] args)
    {
        // Scanner definition
        Scanner scn = new Scanner(System.in);

        // input is a string ( one word )
        // read by next() function
        String str1 = scn.next();

        // print String
        System.out.println("Entered String str1 : " + str1);

        // input is a String ( complete Sentence )
        // read by nextLine()function
        String str2 = scn.nextLine();

        // print string
        System.out.println("Entered String str2 : " + str2);

        // input is an Integer
        // read by nextInt() function
        int x = scn.nextInt();

        // print integer
        System.out.println("Entered Integer : " + x);

        // input is a floatingValue
        // read by nextFloat() function
        float f = scn.nextFloat();
```

```
        // print floating value
        System.out.println("Entered FloatValue : " + f);
    }
}
```

Output :

```
Entered String str1 : Geeks
Entered String str2 : Geeks For Geeks
Entered Integer : 123
Entered FloatValue : 123.090
```

Example 2:

- Java

```
// Java Program to implement
// Scanner Class to take input
import java.io.*;
import java.util.Scanner;

// Driver Class
class Easy {
    // main function
    public static void main(String[] args)
    {
        // creating the instance of class Scanner
        Scanner obj = new Scanner(System.in);
        String name;
        int rollno;
        float marks;

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

        // taking string input
        name = obj.nextLine();
        System.out.println("Enter your rollno");

        // taking integer input
```

```
        rollno = obj.nextInt();
        System.out.println("Enter your marks");

        // taking float input
        marks = obj.nextFloat();

        // printing the output
        System.out.println("Name=" + name);
        System.out.println("Rollno=" + rollno);
        System.out.println("Marks=" + marks);
    }
}
```

Output

```
Enter your name
Geeks
Enter your rollno
5
Enter your marks
84.60
Name=Geeks
Rollno=5
Marks=84.60
```

Differences Between BufferedReader and Scanner

- BufferedReader is a very basic way to read the input generally used to read the stream of characters. It gives an edge over Scanner as it is faster than Scanner because Scanner does lots of post-processing for parsing the input; as seen in nextInt(), nextFloat()
- BufferedReader is more flexible as we can specify the size of stream input to be read. (In general, it is there that BufferedReader reads larger input than Scanner)
- These two factors come into play when we are reading larger input. In general, the Scanner Class serves the input.
- BufferedReader is preferred as it is synchronized. While dealing with multiple threads it is preferred.
- For decent input, and easy readability. The Scanner is preferred over BufferedReader.