



User Input in Java

By Medha XL

Topics

- Using Function Argument
- Using Stream Class
- Using Scanner Class
- Using Command Line Argument
- Types of Errors

User Input

Input means to provide data to a computer for processing. While writing a program, we use input statement to accept data from the user.

Hence, input statement enables user to enter data at run time (during execution of a program).

In Java, we can use following ways for users to enter data for processing.

1. Using Function Argument
2. Using InputStreamReader Class
3. Using Scanner Class
4. Using Command Line Argument

1. Using Function Argument

This is one of the methods to accept the value from the user at the time of execution of the program. The variables whose values are to be input must be provided as arguments to the main() function.

Example: `public static void main(int a, int b)`

The above function will accept two integer numbers from the user. The output of the program is obtained on the screen after execution.

Sample program: Write a program to input `principal`, `rate` and `time` and display the difference between **Simple Interest (SI)** and **Compound Interest (CI)** by using function argument.

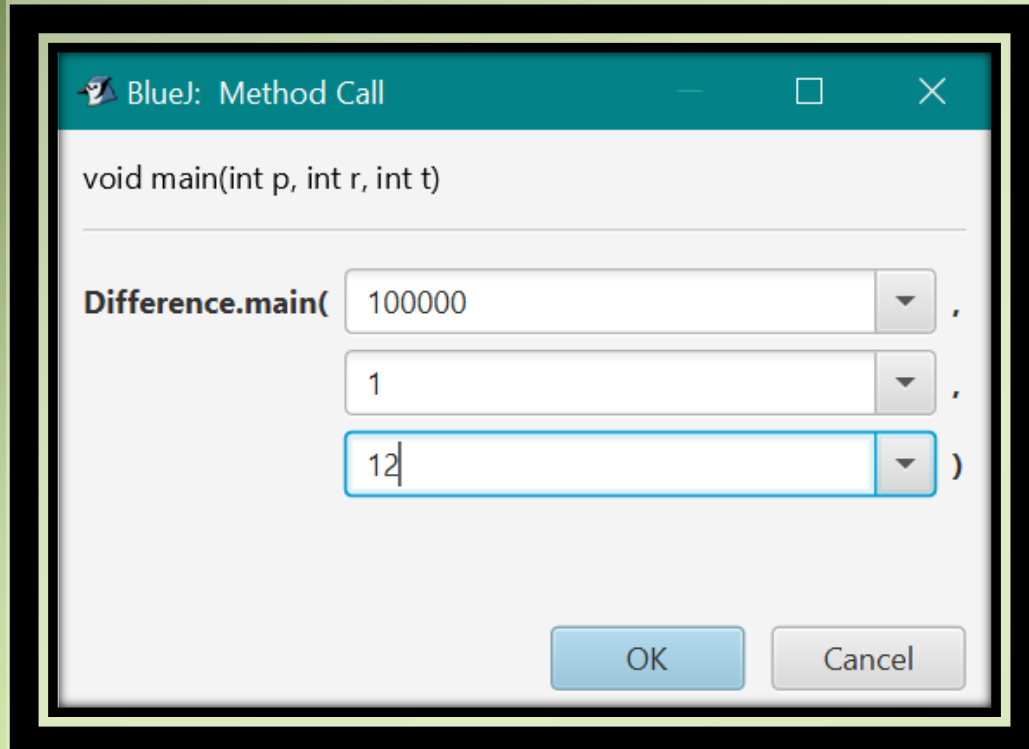
1. Using Function Argument (Sample Program)

// To find the difference between Compound Interest and Simple Interest

```
public class Difference {  
    public static void main(int p, int r, int t) {  
        double si, ci=0, amt, diff=0;  
        si = p*r*t/100.0;  
        amt = p*(Math.pow(1+r/100.0,t));  
        ci = amt - p;  
        diff = ci - si;  
        System.out.println("The Compound Interest = Rs. " + (float)ci);  
        System.out.println("The Simple Interest = Rs. " + si);  
        System.out.println("The difference between CI and SI = Rs. " + (float)diff);  
    }  
}
```

1. Using Function Argument (Sample Program)

Execution of the program



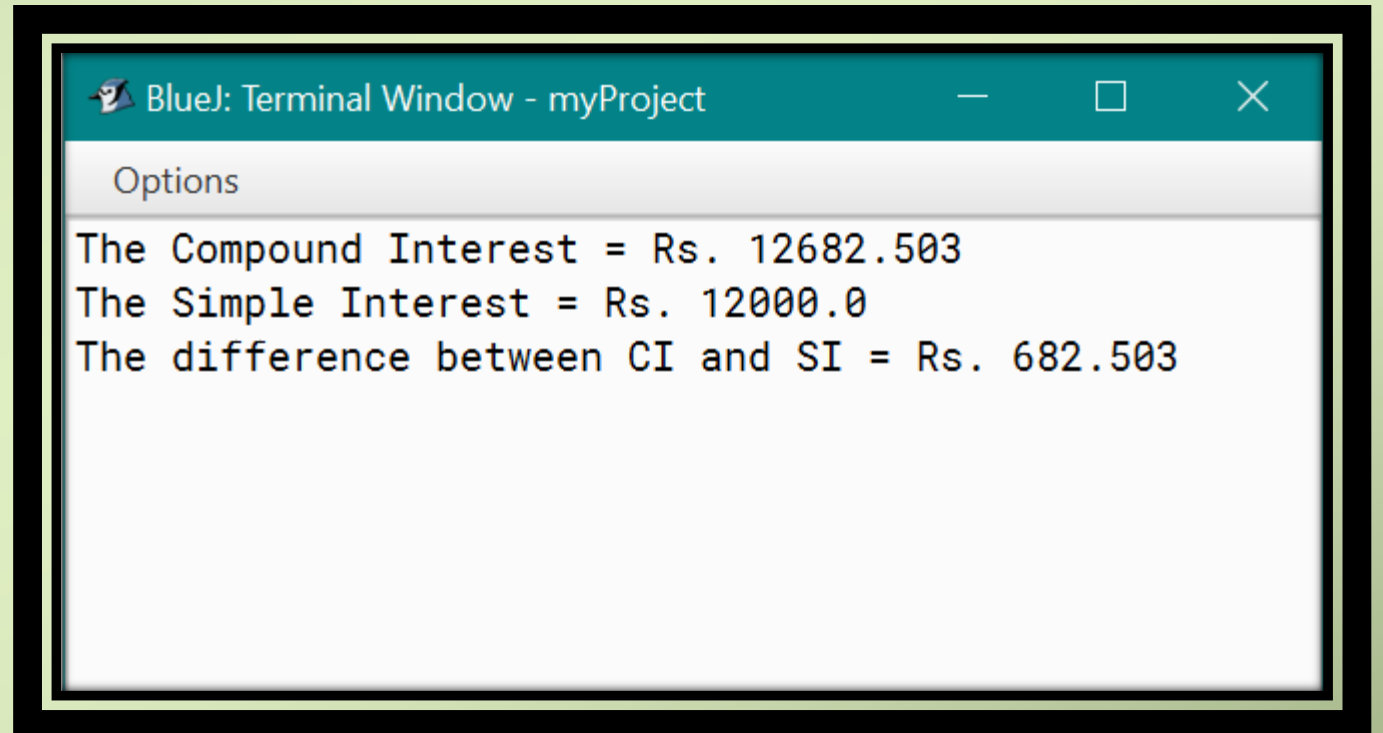
BlueJ: Method Call

void main(int p, int r, int t)

Difference.main(100000 ,
1 ,
12)

OK Cancel

The dialog box shows the method signature `void main(int p, int r, int t)` and the arguments being passed: `100000`, `1`, and `12`. The arguments are entered into three separate input fields, each with a dropdown arrow on the right. The first field contains '100000', the second contains '1', and the third contains '12'. The dialog has 'OK' and 'Cancel' buttons at the bottom.



BlueJ: Terminal Window - myProject

Options

```
The Compound Interest = Rs. 12682.503  
The Simple Interest = Rs. 12000.0  
The difference between CI and SI = Rs. 682.503
```

The terminal window displays the output of the program. It shows three lines of text: 'The Compound Interest = Rs. 12682.503', 'The Simple Interest = Rs. 12000.0', and 'The difference between CI and SI = Rs. 682.503'. The window has a title bar with 'BlueJ: Terminal Window - myProject' and standard window controls.

Assignment

Write a program in Java to accept the number of days and display result after converting it into number of years, number of months and the remaining number of days.

2. Using Stream Class

To input a value by using InputStreamReader class :

- 1) InputStreamReader class is available in java.io package. First of all we need to import this package by using this statement:

```
import java.io;
```

- 2) In the main function, create objects of InputStreamReader and BufferedReader classes as shown below:

```
InputStreamReader read = new InputStreamReader(System.in);
```

```
BufferedReader in = new BufferedReader(read);
```


2. Using Stream Class

To accept an integer:

```
int n;  
System.out.println("Enter a number : ");  
n = Integer.parseInt(in.readLine());
```

To accept a decimal number:

```
float n;  
System.out.println("Enter a decimal number : ");  
n = Float.parseFloat(in.readLine());
```

To accept an Character:

```
char ch;  
System.out.println("Enter a Character : ");  
ch = (char)(in.read());
```

To accept a String:

```
String str;  
System.out.println("Enter a String: ");  
str = in.readLine();
```

3. Using Scanner Class

To input a value by using Scanner class :

- 1) Scanner class is a member of java.util package. First of all, import java.util package in your program.

```
import java.util.*           (or)           import java.util.Scanner;
```

- 2) Create object of Scanner class in the main function as shown:

```
Scanner obj = new Scanner(System.in);
```

↓ ↓ ↓ ↓

Class Scanner New Class

 Object Operator

3. Using Scanner Class

The values of different data types can be input by using next() functions of Scanner class as per the table show below:

Types of data to be entered	Functions to enter data
integer	<code>int n = obj.nextInt();</code>
float	<code>float f = obj.nextFloat()</code>
double	<code>double d = obj.nextDouble();</code>
String	<code>String s = obj.next()</code> or <code>String s = obj.nextLine()</code>

Assignment : Write a Sample Program to use Scanner Class

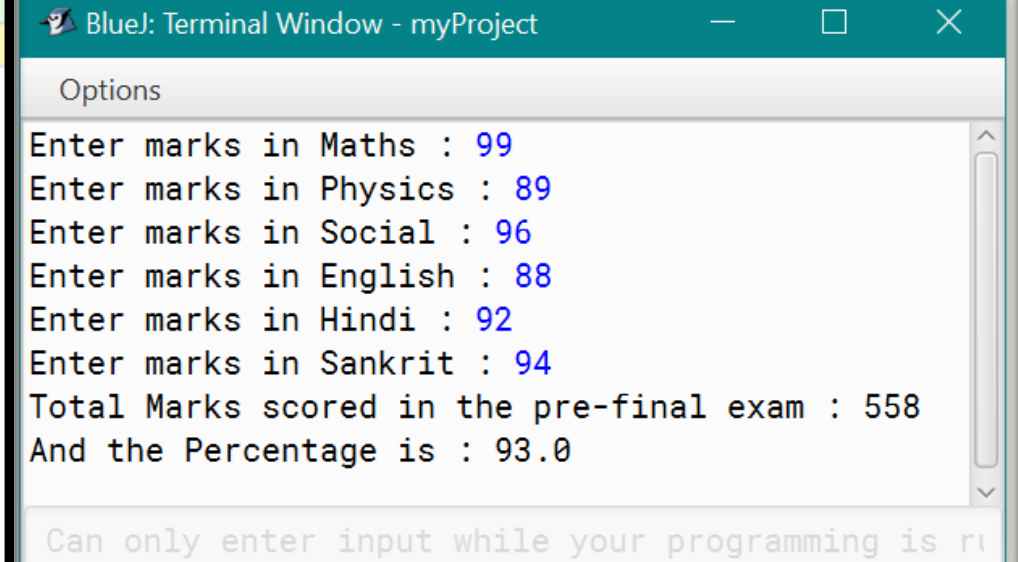
Sample Program: Write a program to read the marks and provide the **total marks** and **percentage** for the student.

```
import java.util.Scanner;

public class student
{
    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.print("Enter marks in Maths : ");
        int maths = input.nextInt();
        System.out.print("Enter marks in Physics : ");
        int physics = input.nextInt();
        System.out.print("Enter marks in Social : ");
        int social = input.nextInt();
        System.out.print("Enter marks in English : ");
        int english = input.nextInt();
        System.out.print("Enter marks in Hindi : ");
        int hindi = input.nextInt();
        System.out.print("Enter marks in Sankrit : ");
        int sankrit = input.nextInt();
        int totalMarks = maths + physics + social + english + hindi + sankrit;
        float percentage = totalMarks/6;
        System.out.println("Total Marks scored in the pre-final exam : " + totalMarks);
        System.out.println("And the Percentage is : " + percentage);
    }
}
```



```
BlueJ: Terminal Window - myProject
Options
Enter marks in Maths : 99
Enter marks in Physics : 89
Enter marks in Social : 96
Enter marks in English : 88
Enter marks in Hindi : 92
Enter marks in Sankrit : 94
Total Marks scored in the pre-final exam : 558
And the Percentage is : 93.0
Can only enter input while your programming is ru
```

4. Using Command Line Argument

This is one of the way to accept the data values from the user and pass the arguments(known as command line arguments) to the main function. While accepting the data values system stores the values in to an array of strings (args[0], args[1], args[2] and so on).

Syntax: `public static void main(String[] args)` or

`public static void main(String args[])`

4. Using Command Line Argument

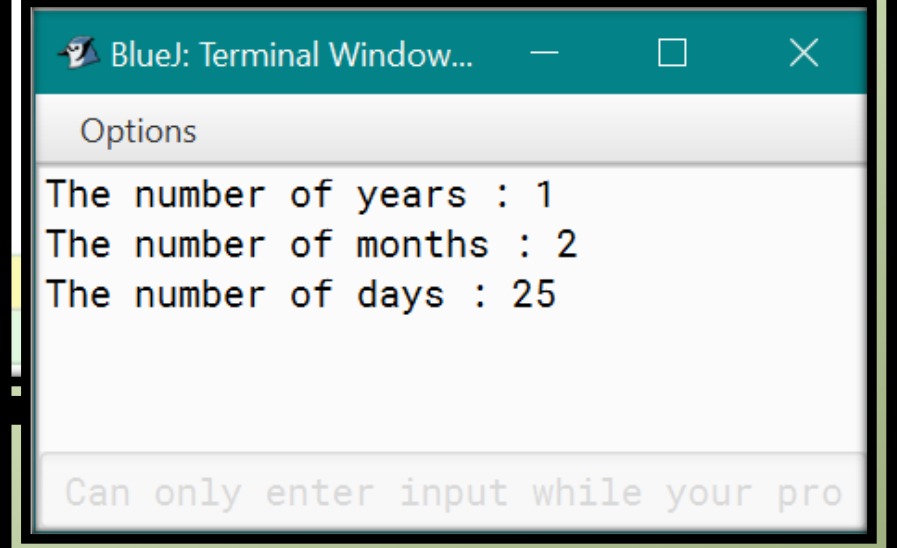
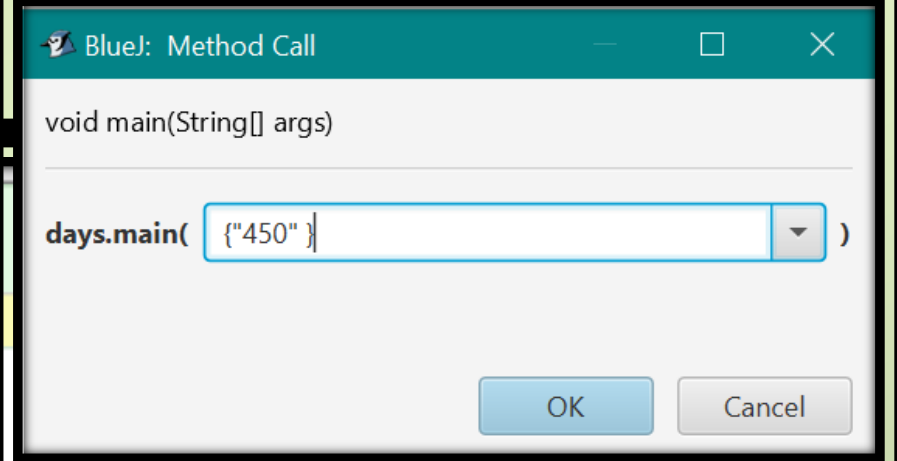
The values of different data types can be passed into command line arguments.

Types of data to be entered	Functions to input data
Integer	<code>Int n = Integer.parseInt(args[0])</code>
Float	<code>Float f = Float.parseFloat(args[0])</code>
Double	<code>Double d = Double.parseDouble(args[0])</code>
String	<code>String s = args[0]</code>

Assignment : Write a Sample Program by using command line Arguments

Sample Program: Write a program to read the number of days and print the **years** and **months** and **remaining days** on the console.

```
public class days
{
    public static void main(String[] args) {
        int days = Integer.parseInt(args[0]);
        int years = days/365;
        int temp = days%365;
        int months = temp/30;
        int rdays = temp%30;
        System.out.println("The number of years : " + years);
        System.out.println("The number of months : " + months);
        System.out.println("The number of days : " + rdays);
    }
}
```



Types of Errors

When you run a program some time you may not get desired results due to incorrect input data or some error in the program. There are three types of errors i.e.,

1. Syntax error
2. Logical error
3. Run time error

Syntax error

These errors due to the grammatical errors in the programming language. It may be missing semi colon, incorrect instructions or undefined variables, etc.

Example

```
Int a=10, b=90; c=20;
```

```
P = (a+b)/c;
```

```
System.out.println("The value of the expression = " + p);
```

In this program, '**p**' is **not defined**, thus the program shows a **syntax error**.

Logical error

The error which occurs in the programming logic is known as logical error. Below example $(a+b)/c$ to be calculated.

Example

```
int a=10, b=90; c=20;
```

```
float p=0;
```

```
P = a+b/c;
```

```
System.out.println("The value of the expression = " + p);
```

Here user wants to divide by **c** after **(a+b)**, but the statement $p=a+b/c$ will compute **b/c** first and then add to **a**. which will produce wrong results due to missing () for a+b.

Run Time error

It is type of error that occurs at runtime when the compiler does not respond properly while executing a statement.

Example : Dividing a number by Zero (0)

```
int a=10, b=90; c=0;
```

```
float p=0;
```

```
P = (a+b)/c;
```

```
System.out.println("The value of the expression = " + p);
```

Here, the value of $c = 0$, so the result will come to an infinite value, which is a run time error.