

# Utilizing User Input in Java

In Java, the `Scanner` class is used to get user input, and inject it into our code. The `Scanner` class is part of a package called `java.util`, and comes pre-installed with every JDK. Just because it's installed doesn't mean we can just use it, though. Because Java was developed with package size in mind, even built-in classes like those found in `java.util` aren't always automatically available. While it is technically installed on our computers already, Java's developers know that not every Java program will need the `Scanner` class. Rather than just include extra code that we may not need, what they've created for us is the ability to use it **if we need it**.

In order to bring the `Scanner` class into our package, all we have to do is **import** it. To do so, we write what's called an **import statement** at the top of our code:

CopyCC#C++ClojureCSSDartGoHaskellHTMLJavaJavaScriptJSONJSXKotlinMarkdownPascalPerlPHPPlain

TextPythonRRubyRustSchemeShellSQLSwiftTypescriptVB.NETVBScriptXMLYAML

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

1

Once we've imported it, we can use the `Scanner` class by creating an object of the class as shown below. We'll talk more about what classes and objects are later, but for now, just think of classes as prototypes of products that then become actual objects. With a class instantiated, we can use any of the available methods found in the `Scanner` [class documentation](#).

In our first example, we will use the `nextLine()` method, which is used to read `String` s:

Run

```
import java.util.Scanner; // import the Scanner;
```

```
class Main {  
    public static void main(String[] args) {  
        Scanner myObj = new Scanner(System.in);  
        String userName;
```

```

    System.out.println("Enter Username;"); //Ask the user to enter the user name
    userName = myObj.nextLine(); // Read the input given by the user

    // Print the username
    System.out.println("Username is: " + userName);

}
}

```

## CONSOLESHELL

### InfoWarningTip

Many complex Java programs will start with several imported classes, and we'll be seeing a lot of them as we go. If your IDE is throwing errors related to classes that come from outside packages, **always check the import statements** related to the packages/classes you imported. Remember that spelling, punctuation, and capitalization all matter.

## #Input Types

In the example above, we used the `nextLine()` method, which is used to read Strings. To read other types, take a look at the table below:

Input Type:	What it takes in:
<code>nextBoolean()</code>	Reads a boolean value from the user
<code>nextByte()</code>	Reads a byte value from the user
<code>nextDouble()</code>	Reads a double value from the user
<code>nextFloat()</code>	Reads a float value from the user
<code>nextInt()</code>	Reads a integer value from the user
<code>nextLine()</code>	Reads a string value from the user
<code>nextLong()</code>	Reads a long value from the user
<code>nextShort()</code>	Reads a short value from the user

### InfoWarningTip

Don't forget to close your scanner after using it by calling the **scanner.close()** method to avoid memory leak.

In computer science, a memory leak is a type of resource leak that occurs when a computer program incorrectly manages memory allocations in such a way that memory that is **no longer needed** is **not** released.

In the example below, we use different methods to read data of various types:

Run

```
import java.util.Scanner;
class UserInfo {
    private static String favCity;
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter your name:");
        // String input
        String name = scanner.nextLine();
        System.out.println("Enter your favorite city: ");
    }
}
```

CONSOLE SHELL

## #Creating an Algorithm To Detect Negative Numbers

Your next task is to instruct your computer to display whether a number is positive or negative.

### #Steps to be followed

1. Using Eclipse, create a new class for this exercise just like you did for your Hello World example, but give it a new name. You can name the class/file **NegNumbers.java** for example.
2. **Import** the scanner library.
3. In your **main** method, declare an **integer** variable.
4. Create a **Scanner** object to take input from a user.
5. Print a prompt for the user to enter a number.
6. Receive the number from the user using the appropriate **scanner** method.
7. Create the conditions to check if the number is positive or negative
8. Print the output.
9. Test your code.
10. If it works as expected, copy/paste your code into the file below.

## #Code

```
// Import the Scanner Libray

public class NegNumbers {

    public static void main(String[] args) {
        // Paste your code here.
    }
}
```

## Problem Statement

ASCII is American Standard Code for Information Interchange. ASCII values include special characters, numbers, and alphabets from various writing systems. At its core, ASCII is a numerical representation of characters: for example, the ASCII value for 'A' is 65, and the ASCII value for 'a' is 97.

If you'd like to take a look at it, here's the ASCII table:

<https://www.techonthenet.com/ascii/chart.php>

To print out the ASCII value of a character, you can assign it to an integer and print out the integer, as shown below:

CopyCC#C++ClojureCSSDartGoHaskellHTMLJavaJavaScriptJSONJSXKotlinMarkdown  
PascalPerlPHPPlain

TextPythonRRubyRustSchemeShellSQLSwiftTypescriptVB.NETVBScriptXMLYAML

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

char c = 'A';
int ascii = c;
System.out.println("The ASCII value of " + c + " is: " +
ascii);

// output: The ASCII value of A is: 65
```

```
}
```

## #Steps to be followed

---

- In Eclipse, create a new Java class called **ASCIIExample**. Note that the file name should have the `.java` extension.
1. In your main method, create a variable of type `char`.
  2. Assign the variable you created above to another variable of type `int`.
  3. Write a statement to print the value of the `int` variable you created.
  4. Run the code and check the output in the terminal.
  5. If your code is working as expected, copy and paste your code in the IDE below to submit it.

## #Code

Run

```
public class ASCIIExample {  
    public static void main(String[] args) {  
        // Fill in the code below  
        char myChar='A';  
        int myInt=myChar;  
        System.out.println("Value myInt: "+myInt);  
    }  
}
```

CONSOLE SHELL

You can get the ASCII value of a character in a java program by  
by printing a character

by assigning a character to an integer variable and then printing the variable

by typecasting a character to a string

by looking up the value in the ASCII table and hard coding it.