

Getting Started With Java – Input & Output of Variables

The `print` and `println` methods have already been used to output strings. They can also be used to output values of any other type of variable.

```
class PrintValue
{
    // Assigns a value to "number" and then prints that value
    public static void main (String [] args)
    {
        int number = 37;
        System.out.println(number);
    }
}
```

It is important to assign some value to a variable before we attempt to use it in any way, including output. This is one reason why initialization of variables is such a good practise.

The following code, attempting to output an uninitialized variable, will produce an error.

```
class NoValue
{
    // Produces an error because it tries to output
    // an uninitialized variable
    public static void main (String [] args)
    {
        int number;
        System.out.println(number);
    }
}
```

Although we can output data of any type using the `print` and `println` methods, all output in Java is actually performed using strings. Thus, if we give an argument to `print` or `println` that is not a string, it will be converted to a string automatically, and then output.

This fact allows us to use a mixture of different data types in the same output statement. The *concatenation operator* (+) can be used to combine strings. Since all output is actually made up of string, this operator can be used to combine any data type in an output statement.

Be aware, however, that (+) is also the arithmetic addition operator, and mathematical operations will take precedence over the automatic conversion to a string.

Consider the following two statements, where the variables `height` and `width` have already been set to the values of 4 and 6.

```
System.out.println("Dimensions are " + height + width);
System.out.println("Dimensions are " + height + " and " + width);
```

The first will incorrectly output "Dimensions are 10", since the `height` and `width` are added (math) first, and then the answer is converted to a string and concatenated with the other string.

The second will correctly output "Dimensions are 4 and 6".

Getting Started With Java – Input & Output of Variables

Input with Java

Java is a language designed for compatibility with all computer systems, and as a result, the input and output (particularly the input) can be quite difficult for those new to Java. In order to simplify this process, we will be using some custom classes, called `In` and `Out`, which are available from:

<http://www.ecf.utoronto.ca/~jcarter/>

A copy of these classes is already located in the handout folder. Some Integrated Development Environments (IDEs) will allow you to automatically reference and use custom classes such as these by making changes in the options (speak to your instructor about this).

Warning: The classes `In` and `Out` are not part of Java, so you probably won't see them outside of this course, unless you bring them yourself (as the files `In.java` and `Out.java`).

The `In` class provides the following methods for input: `getInt`, `getLong`, `getChar`, `getString`, `getFloat`, and `getDouble`. They are all used the same way.

Each method assumes that each input is followed by the `<enter>` key. If incorrect input is entered for a numeric value, a value of zero will be used. If the user hits `<enter>` during string input, the result will be an *empty string*, which has the value `""` (no characters in the string). This is also known as the *null string*.

For example,

```
class Interactive
{
    public static void main (String [] args)
    {
        // obtain name and age
        System.out.println("What is your first initial?");
        char firstInitial = In.getChar();
        System.out.println("What is your family name?");
        String familyName = In.getString();
        System.out.println("What is your age in years?");
        int age = In.getInt();

        // respond to input
        System.out.println("Well, " + firstInitial + "."
            + familyName + " - "
            + "I see that your age is " + age);
    }
}
```