

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 may take precedence over the automatic conversion to a string, depending upon the

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

The first will incorrectly output "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);
    }
}
```

Getting Started With Java – Input & Output of Variables

Exercises

1. What would be printed by each fragment?

(a)

```
int i = -47, j = 35;
System.out.println("The value of i is " + i
                   " while the value of j is " + j + ".");
```

(b)

```
boolean done = false;
System.out.println("We are not " + done + " yet.");
```

(c)

```
double x = 0.012, y = 2.7374e1;
System.out.println("x -> " + x + "\ny -> " + y);
```

2. Write a program that creates a **double** variable called `myMass`, asks the user to give his/her mass, and then prints a message giving the person's mass.

3. Write a program that creates four **integer** variables, asks the user to enter four course grades, and then calculates and prints the average grade.

4. Write a program for calculating a student average for a semester. Build your program in stages as outlined below:

(a) Create a single real (double) variable called `mark1`, and ask the user to input their first mark. Output the mark that was just given.

(b) Add a second variable, called `mark2`, and ask for the first and second mark. Output both marks.

(c) Create a real (float or double) variable called `markTotal`. Calculate the sum of the user marks and store the result in `markTotal`. Output the result.

```
markTotal = mark1 + mark2;
```

(d) Add third and fourth variables, `mark3` and `mark4`. Ask the user for these marks, output the results. Add these marks to the `markTotal` as well.

(e) Create a real variable called `average`. Calculate the student average and output the result.

$$\text{average} = \frac{\text{sum of marks}}{\text{number of marks}}$$

5. The area of a circle is $A = \pi r^2$ and the circumference is $C = 2\pi r$. Write a program that asks the user for a radius and calculates the area and circumference, with each stored in a variable. Your program should include a **constant** for pi, where $\pi = 3.1415$.

6. Write and run an interactive program that asks the user to provide a string, a character, an integer, and a floating point value. Your program should also have a constant value built into it.

After reading the values, the program should print each one (including the constant) on a separate line with a suitable message for each line of output.