Return to Module Overview Page (https://onlinelearning.berkeley.edu/courses/665040/pages/module-one)

1.4: Introducing the Java Programming Language

At the single-line level, Java code looks a lot like its closest relatives, C and C++:

```
int i, sum=0;
for (i=1; i<=10; i++)
{
  sum = sum + i;
}
```

computes the sum of the numbers from 1 to 10 in any of Java, C, or C++. If you're not a C programmer, you need to know that the curly brackets delimit blocks of code in the same way that sub ... end sub do in Visual Basic or begin... end do in Pascal. i and sum are variables that can hold integers (actually, C allows them to hold virtually anything; but by default, it's integers). The for(...) construct tells the computer to perform the block of code ten times, setting the value of the counter variable i successively to each of the values starting at 1 and up to and including 10. At the end, sum contains the sum of 1 through 10, or 55.

When we look at an entire program, some structural differences between Java and C are apparent on a slightly larger scale. Here is the code above embedded in a complete C program:

```
#include <stdio.h>

int main(int argc, char ** argv)
{
    int i, sum=0;
    for (i=1; i<=10; i++)
    {
        sum = sum + i;
    }
    printf("The sum of 1 through 10 is %d\n",
           sum);
    return 0;
}
```

Every C program contains a function called main, which is always the first function to be called in the program. The declaration above says that when main is called, it expects two parameters: an integer (which should contain the number of arguments the program was passed when it was started from a command line) and a list of character strings, each one of which should contain one of those command line arguments. When our loop completes, this program prints the result using the C printf function, then uses the return keyword to exit the function and hence the program.

Here is the same code fragment cast as a complete Java program (your first!):

```
public class SumOneToTen
{
   public static void main(String[] argv)
   {
     int i, sum=0;
     for (i=1; i<=10; i++)
       {
         sum = sum + i;
       }

       System.out.println("The sum of " +
                        "1 through 10 is " +
                        sum);
   }
}
```

There are a number of differences between these two programs. Some of them are pretty important, although you won't be in a position to understand why until we've gotten a bit further along.

Classes

The most obvious difference between the two programs is that outer set of brackets and the mysterious incantation public class SumOneToTen. In Java, every function and variable is inside such a declaration, called in general a *class*. Eventually we'll see that classes are really the cornerstone of maintainability in Java programming. For now, it's enough for you to know that you have to use one. Furthermore, for most Java development tools, the name of the class (SumOneToTen) must correspond to the name of the file in which you write this code (SumOneToTen.java), including the case. Neither SUMONETOTEN.java nor sumonetoten.java will do; even though some operating systems would consider all three of these names to refer to the same file, the JDK Java compiler and the JDK Java runtime consider case to be significant in file names. If you're not using the JDK, see the documentation for your own Java development environment for details about file names.