**To run a HelloWorld.java from the terminal**

First we need to compile it using

**javac HelloWorld.java**

this will compile the file and create a HelloWorld.class

type **java HelloWorld** and the program will run

**Static Typing**

One of the most important features of Java is that all variables and expressions have a so-called static type. Java variables can contain values of that type, and only that type. Furthermore, the type of a variable can never change.

**Extra Thought Exercise**

In Java, we can say **System.out.println(5 + " ");.** But in Python, we can't say **print(5 + "horse"),** like we saw above. Why is that so?

Consider these two Java statements:

String h = 5 + "horse";

and

int h = 5 + "horse";

The first one of these will succeed; the second will give a compiler error. Since Java is strongly typed, if you tell it h is a string, it can concatenate the elements and give you a string. But when h is an int, it can't concatenate a number and a string and give you a number.

Python doesn't constrain the type, and it can't make an assumption for what type you want. Is **x = 5 + "horse"** supposed to be a number? A string? Python doesn't know. So it errors.

In this case, **System.out.println(5 + "horse");,** Java interprets the arguments as a string concatentation, and prints out "5horse" as your result. Or, more usefully, **System.out.println(5 + " ");** will print a space after your "5".

What does **System.out.println(5 + "10");** print? 510, or 15? How about **System.out.println(5 + 10);?**

**Defining Functions in Java**

In languages like Python, functions can be declared anywhere, even outside of functions. For example, the code below declares a function that returns the larger of two arguments, and then uses this function to compute and print the larger of the numbers 8 and 10:

def larger(x, y):

if x > y:

return x

return y

print(larger(8, 10))

Since all Java code is part of a class, we must define functions so that they belong to some class. Functions that are part of a class are commonly called "methods". We will use the terms interchangably throughout the course. The equivalent Java program to the code above is as follows:

public class LargerDemo {

public static int larger(int x, int y) {

if (x > y) {

return x;

}

return y;

}

public static void main(String[] args) {

System.out.println(larger(8, 10));

}

}

The new piece of syntax here is that we declared our method using the keywords public static, which is a very rough analog of Python's def keyword. We will see alternate ways to declare methods in the next chapter.

The Java code given here certainly seems much more verbose! You might think that this sort of programming language will slow you down, and indeed it will, in the short term. Think of all of this stuff as safety equipment that we don't yet understand. When we're building small programs, it all seems superfluous. However, when we get to building large programs, we'll grow to appreciate all of the added complexity.