Intro

John Marchand
jhmarchand@gmail.com

I have a degree in Computer Science from the University of Kentucky. I work for Mercer as Software Developer, primarily responsible for the login and security code for a Workforce Management system licensed to large companies across the globe.

Class Objectives

The focus of this class is to learn Java. While we may refer to robot code from time to time, it will not be the taught much if at all. After 2 seasons of robotics, I have found there to be very little program logic and design needed for the robot code. Programming logic and design as well as the Java language will be the focus of this class. I expect though, that writing the robot code in an object oriented and well maintainable manner will be very easy for anyone who completes this class. You will also be well prepared to learn to write Android Apps, or other object oriented languages.

This class is 2 hours a week, but I hope you work on this outside of class. You are welcome to email me questions at the email address above.

I learned to program in the early 80s, and back then, you bought books. Now, languages change very fast and new ones are being created all the time. Online references are up to date, and cheaper, so I will be teaching out of the Oracle docs for Java.

https://docs.oracle.com/javase/tutorial/java/index.html

Java Software

Historically, you needed many different programs to edit, compile, link and prepare your code for use. While you can still use them separately, no one really does. Today, we use a Integrated Development Environment, or IDE. This gives us access to all the tools we need to develop our software in one program. It is usually customizeable so that you can choose which tools you use, but for this class, we will stick with the basics.

For this class, I will be using Eclipse Neon. You may use any Java IDE you want, or even different versions of Eclipse. Eclipse Oxygen is the latest release, but I have not downloaded it yet, and sometimes it IS best to wait a couple of months before getting the latest IDE. Developing software can be frustrating enough, you don't need to deal with all the bugs in the latest release.

To install Eclipse Neon (or any version of Eclipse), you will need to install a Java Development Kit (JDK). Here is the latest release for the Java SE JDK

http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

Here is the link to Eclipse Neon for Java developers.

http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/neon3

HS Robotics Team

Introductions. Can you guys help anyone who doesn't have a IDE installed?

Java Overview

Java was first released in 1995, so even though I will refer to it as a modern programming language, it is 22 years old. It is highly influenced by C++, but attempts to remove much of the C language holdovers so that it is more of a pure object oriented language.

You may have heard that C++ is much more efficient than Java. While this is true, most software performance is limited by hard disk speed and algorithm implementation. Java is a compiled language. Most compiled languages are compiled to run on a specific processor and operating system. Java is compiled to run on a Java Virtual Machine (JVM). This means that your compile code can run on Windows, Linux, MAC OS, or any other operating system for which there is a JVM.

According to Oracle - “As a platform-independent environment, the Java platform can be a bit slower than native code. However, advances in compiler and virtual machine technologies are bringing performance close to that of native code without threatening portability.”

Students Intro

Who has used Scratch or App Inventor?

Who has used Java?

Who has used C++, C#, Object C or Swift?

Who has used C, Pascal, Python, Java Script (no relation)?

Has anyone written code to control a robot?

Has anyone written a game?

Has anyone written a mobile app?

Hello World

It is convention to have your first program simply say “Hello, World!” Every time I learn a new programming language or target environment, I write one. I have probably done 20, 30 or even 50 of them in my lifetime. This is a good starting point and will let everyone know their development environment is set up.

```
// Create a public class called HelloWorld
public class HelloWorld
{
	// The default function for a console app function that is the starting point
	public static void main(String[] args)
	{
		// This will display anything in the quotes on the screen
		System.out.println("Hello, World!");

	}

}
```

First we declare a public class called HelloWorld. We will discuss more about a class later as we get into object oriented programming. For now, just realize Java requires all code to be in a class.

Next we declare a public static function. We will get into functions later in the class. For now, just realize, this is the start of your code.

We will also be covering what public (versus private) and static mean later on, again, let's just ignore these first 2 lines for now.

The third line means we do have to talk about functions and classes a bit, but for now we are just using predefined ones. System.out is a predefined object that Java provides to us. It allows us to write (or print) to the screen. We will use the predefined function println for now. It prints whatever you put in between the () on the console.

SIDE NOTES

1. System.out has several predefined functions, and if you type System.out. (including that last .) Eclipse will show you some of the functions available to you.)
2. Each statement in Java is terminated with a semicolon.

3. Java is case sensitive. So System.Out is not a predefined object and Java doesn't know what you mean.

Variables and Types

Variables are used to store information and they have a type. So a string variable would hold a name, while a int (integer) would hold a number. Consider the following code snippet.

```
String myName = "John";
int myAge = 47;

System.out.println(myName + " is " + myAge + " years old");
```

Here I have create a String variable called myName and assigned it a value of “John”. I also have a int variable called myAge and assigned it a value of 47. I can then use those variables in a print statement to show the information on the screen.

User Input

Now let's get some information from the user. This uses a predefined class called java.util.scanner. Again, we’ll just use it for now.

```
Scanner scanner = new Scanner(System.in);

System.out.println("Enter your name:");
String name = scanner.nextLine();

System.out.println("Enter your age:");
int age = scanner.nextInt();
```

Goals

1. Run Hello World on your IDE.
2. Change the Variable and Type code snippet to be your name and age. Add that to your Hello World Program and run it.
3. Ask the User how old they are using the User Input Snippet. Can you say Hi to them and tell them how old the are? (The variable age and myAge store 2 different values.)
4. Can you make a adding machine? Ask the user for 2 different numbers and add them together.