Learning How To Code

Monday, February 10, 2020 11:38 AM

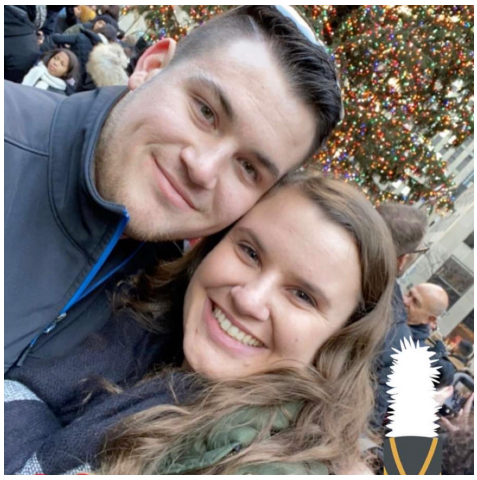
Congrats on wanting to learn more about Computer Science. This will be here to teach you the basics of what computer science is, and how to actually write computer programs. Throughout these lessons, I will help guide you to some basic results in how different things work. In this chapter we will cover:

- What a computer code is
- Why computers are the way they are
- How to write your first program and run it

- This guide is written by Nicholas M Edwards. My contact information is show below:

Email: Nicholas.Edwards@nedwads.us

Phone: (215)909-0448



What is a Computer Code?

A computer code is a list of commands written by a programmer, with the goal to get a computer to do something. This works in a way that many people see "confusing" but it's all logic based.

The best way to think about how a
computer works is as if you are teaching a monkey.

Here we have Tom. Tom is a Rhesus Macaque monkey, and good news, I think he likes you. There is an instant connection you have with Tom so I suggest you bring home your new pet and begin to train him.

Why are computers the way that they are?

Well a computer, when you think about it, is just a bunch of electrical currents, that all work together to do something. Similar to our new friend Tom, these all connect to the brain of the computer, the cpu.

Computers are something where out of the box, they arleady know something, sort of how Tom knows how to walk and move around, but he and the computer don't know any real tricks. Now you could take Tom to school and let him learn tricks, kind of how you could pay someone else to either use their program or to make you one, or you can make it yourself and get really good at it and take care of other people's computers (the good

news is no poop duty like the animal trainer.)

The first computers were about the size of a house and were made of a bunch of switches that either designate yes or no. This is where the first idea of logic comes into play. Remember how I mentioned that computers are just electric running around, well if it weren't for computers it would be hard to tell how much electric there is so the easiest way to do it is, if there is electric, then put yes, if not, then put no.

These yes and no options are still technically used today in the form of a compiler. The compiler serves to change your text code into the actual computer language that we call binary. Binary is the counting of base 2 numbers.

I'm getting a little ahead of myself for an intro level but base 2 counting is something that is very important later on in your knowledge of computer science since there are other things that use different bases such as password encryption that uses base 16 a lot.

The best way to show base 2 is with an example.

What we call counting numbers is base 10. so if we think of the number 15, that can be re-written as

$$1(10^1)+5(10^0)$$

The number in front is the amount in that point, which because it is base 10 can be any number 0-9. The 10 is the base number, and the exponent is the spot that it is in (think about when you are learning decimal places to fractions, where it's like 10ths 100ths, 1000ths, it's kind of like that so any number in this case can be written but $0[10^2]$ would still just be 0 so who cares)

With this in mind we think of the same number 15 but in base 2

$$1(2^3)+1(2^2)+1(2^1)+1(2^0)=15=1111_2$$

So in this case the original computer, if you were trying to input the number 15, you would switch the first 4 switches all to on and leave the other who knows how many off.

Pretty cool huh? Like I said, knowing the full how-to for this isnt really necessary for a beginner class but now you look smarter than the kid sitting next to you playing Fortnite on his phone, claiming how he wishes he made the game and had all the money they make off of it.

Now, I don't want to keep you too long so it's time to learn how to code your first project.

How to code your first project

If I have not scared you off yet then good job. Some of the stuff I talked about is saved at schools for at least 2 semesters, if not longer, but as a software engineer logic is my strong suit and I get very excited over it. Now on to the code.

1. Start by opening Jgrasp and Locating the folder you downloaded from github.
2. In the Do_It folder you should see a file labelled HelloWorld.java. Open it up and take a look at the beautiful nothingness inside it.
3. Time to Code

For the bulk of these courses, we will be doing what is called OOP or Object Oriented Programming. This is generally the start to how people learn how to code.

On the first line, we must define the class name. The class name must match the file name perfectly, or the computer will become upset that it cannot find the instruction book (file name) for the set of instructions (class name).

To do this, we will create a public class and name it HelloWorld (Instructions cannot have spaces in their names so what we generally do is call CamelCasing, where the first letter of a new word is upper case).

```
public class HelloWorld{
```

The bracket on the end is added to tell the computer anything within the bracket from here, until it ends will be included in this code.

Next, we need to create our main method. We will do this by creating the method called main, giving it the "ability" (String [] args) and by making it static. When it comes to making the method static, determining if it should be public, or any "abilities" don't worry about them, they are for a higher level when you move into a much more deeper type of coding, where it matters which classes use what methods.

```
public static void main (String [] args){
```

Now for the part we have all been waiting for, the body of the code. This is the part of the code that the computer will execute for you and place in the Run (I/O) screen. For this tutorial we will do one of the most basic but necessary things that any coder should be able to do. Print… a message.

To print a message, there are some very intricate details that goes into the code that for a beginner, are completely unnecessary, so for our case I'm just going to tell you the 2 different print options to print to the run console built in to Jgrasp.

```
System.out.println("Hello World!");
```
Or
```
System.out.print("Hello World!");
```

By adding the ln at the end, the computer will automatically return for you (or if you are looking at it like a word document, press enter). This is important if you have other information to be taking in, or printing out so that you don't run them together.

Now just close up your curly braces from the class and method and your code should be ready to go.

4. Run the Program

Now it's finally time to run the program! Click on the little running man at the top and watch as your code prints out the very exciting phrase Hello World! I know it's so much fun, who ever needs to use Microsoft Word again when I can just code all my essays now (Joke's on you Dr. Smith!!!).

If you are having problems running your code ever, don't be afraid to google the error that you get. The best coders spend more time writing tests and solving problems then actually writing code from scratch. The full working code will be posted in the answers folder as well as below here:

```
public class HelloWorld{
   public static void main(String [] args){
      System.out.println("Hello World!");
   }
}
```

Note that in my code, I have indented the method and then the body even further than that. Whenever you are containing something in a code it is best to indent it because it helps make the code look cleaner and if there are any errors they will be easier to find.