

Introduction to Java

Java is the programming language we use on the robot.

Creating a Java File

Every Java file is made with the extension .java. Your file would be [The Name of your File].java.

Examples of File Names Include:

```
bob.java  
Steve.java  
Hello7.java  
Hello_Steve.java
```

Illegal Name Examples:

```
7hello.java  
b-d.java  
$ja.java
```

Structure of a Java File

Every Java file needs to be structured in the same way.

Example of a Bob.java File:

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

Only change `public class Bob` to the name of your file. The syntax is `public class [The Name of your File]`.

The Main Method

All your code will be written in the main method. Only code in the main method will ever run. For now we write all our code in between the curly brackets `{ }` of the main method.

The Main Method:

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

A Statement

A statement is like a sentence in English. It is a single line of code. Instead of ending with a `.` like we do for sentences, we end with a `;`.

Example of a Statement:

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

`System.out.println("[Your Message]");` is the first statement we will learn. It takes a message and prints it out. The message must be in quotes `"`.

Lets Put Everything Together

Lets build a simple program that prints out "Hello World!". Tada! We just made our first program!

Example of a Bob.java File:

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

Adding More Statements

We can have as many statements as we want. They will run from top to bottom.

Example of a Bob.java File:

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

This will print out:

```
Hello World!  
Hello Steve!  
Hello Raccoon!
```

Comments

Our programs are starting to get longer so lets start adding comments. Comments are lines of code that are ignored by the computer and serve as notes for us. We can add comments by putting `//` in front of the line.

Example of a Bob.java File:

```
public class Bob {  
    public static void main(String[] args) {  
        // This is a comment  
        System.out.println("Hello World!");  
        System.out.println("Hello Steve!");  
        System.out.println("Hello Raccoon!");  
    }  
}
```

This is a comment will never run because it is ignored by the computer. Use this to take notes.

Variables

Variables are like boxes that we can store information in.

Types of Data

Before we can make a variable we need to know the types of data we want to use. The ones we usually use are: `int`, `double`, `boolean`, and `String`.

- `int`: Whole Numbers
- `double`: Decimal Numbers
- `boolean`: True or False
- `String`: Text

Making a Variable

The syntax for making a variable is `[Type of Data] [Name of Variable] = [Value];`. The rules for naming variables are the same as naming files.

Example of a Bob.java File:

```
public class Bob {  
    public static void main(String[] args) {  
        int age = 15;  
        double height = 5.5;  
    }  
}
```

```
        boolean isTall = false;
        String name = "Bob";
    }
}
```

Using a Variable

Remember variables are just placeholders for information so we can use them to print out messages.

Example of a Bob.java File:

```
public class Bob {
    public static void main(String[] args) {
        // Creating Variables
        int age = 15;
        double height = 5.5;
        boolean isTall = false;
        String name = "Bob";

        // Prints out Bob
        System.out.println(name);

        // Println can take in any type of data
        System.out.println(12);

        System.out.println(age);
        System.out.println(height);
        System.out.println(isTall);
    }
}
```

The output of this program is:

```
Bob
12
15
5.5
false
```

Changing a Variable

We can change the value of a variable by using the syntax `[Name of Variable] = [New Value];`.

Example of a Bob.java File:

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

```
// Creating Variables
int age = 15;
double height = 5.5;
boolean isTall = false;
String name = "Bob";

// Prints out Bob
System.out.println(name);

// Println can take in any type of data
System.out.println(12);

System.out.println(age);
System.out.println(height);
System.out.println(isTall);

// Changing Variables
age = 16;
height = 5.6;
isTall = true;
name = "Steve";

System.out.println(name);
System.out.println(age);
System.out.println(height);
System.out.println(isTall);
}
}
```

The output of this program is:

```
Bob
12
15
5.5
false
Steve
16
5.6
true
```

Operators

Lets do some math. We can use operators with the numbers to do math. We can do addition `+`, subtraction `-`, multiplication `*`, and division `/`.

Example of a Bob.java File:

```
public class Bob {  
    public static void main(String[] args) {  
        // Doing Math  
        System.out.println(5 + 5);  
        System.out.println(5 - 5);  
        System.out.println(5 * 5);  
        System.out.println(5 / 5.0);  
  
        // Doing Math with Variables  
        int age = 15;  
        int age2 = 16;  
        System.out.println(age + age2);  
  
        // The math is done first  
        int age3 = 12 + 17;  
  
        int age4 = 12 + 17 * 5;  
        int age5 = (12 + 17) * 5;  
    }  
}
```

Order of Operations

This is just like in math...

```
Parentheses come first.  
There are no exponents.  
Multiplication and division are next.  
Addition and subtraction are last.
```

Special Operator

There is one more operator we need to know about. The `%` operator is called the modulo operator. It gives us the remainder of a division.

So `5 % 2` is `1` because `5 / 2` is `2` with a remainder of `1`.

```
System.out.println(5 % 2); // Will print out 1
```

Styling Guide

This is a guide and less of a rule but these are generally what Java programmers do to make their code look nice and readable.

- Name variables with camelCase. So `ageLimit` where the first word is lowercase and the first letter of each word after that is capitalized.

- Tab every code inside curly braces ({ and }). This makes it easier to read.
- Keep each statement on its own line.
- Use variable names that make sense. So `ageLimit` is better than `x`.
- Keep your comments on their own line unless its at most 2 words.

Coding Challenges...

1. Print out a figure of a rocket ship. An Example is below. Customize it to your liking.

```
  *
 ***
*****
*****
*       *
*       *
*       *
*       *
*       *
*****
*****
*****
```

2. Print out a christmas tree. Customize it to your liking.
3. Use variables to calculate the area and perimeter of a rectangle with a `width` of 5.0 and a `height` of 10.0. Print out the area and perimeter.
4. Look up the circumference of a circle formula. Write code that use the variables `radius` and `pi` to calculate the circumference of a circle. Print out the circumference of a circle with a radius of 5.
5. Look up the area of a circle formula. Write code that use the variables `radius` and `pi` to calculate the area of a circle. Print out the area of a circle with a radius of 5.
6. Turn this into code, `my magic number is the remainder of 2 time x plus 7 more than y all divided by y plus 12` and print out the result.
7. Write $12x^3 + 3x^2 - 5x + 2$ in code.
8. Do challenge 7 but only using the variable 3 times in total in your code like `x * x * x` is used 3 times.