Java I

Java Basics

How Java Works

Java is a compiled language. (generally speaking)

- Some languages are compiled - C/C++, C#, Cobol, Visual Basic
- Some languages are interpreted - JavaScript, Python, Ruby

What you'll do:

How Java Works

What you'll do:

1. Write code in a source file; save as .java
2. Use the javac compiler to compile the code
3. If there are no errors, you get a file with Java bytecode; a .class file
4. Run the bytecode in a Java Virtual Machine (VM)

Write Once Run Anywhere

The JVM is specific to an operating system, different one for each OS.

(This is the JRE on your computer)

So the same compiled byte code runs on all systems.

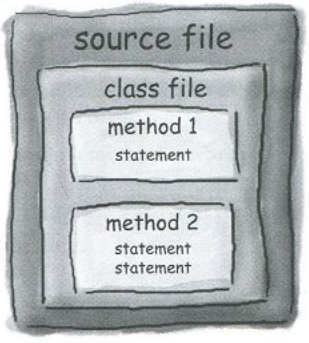
Review

Is Java a compiled language or interpreted?

What is a .java file? How about a .class file?

What runs your Java code?

Java Code Structure



Put a class in a source file.

Put methods in a class.

Put statements in a method.

The `main` method

When the JVM starts to run, it looks for a special method - the `main` method.

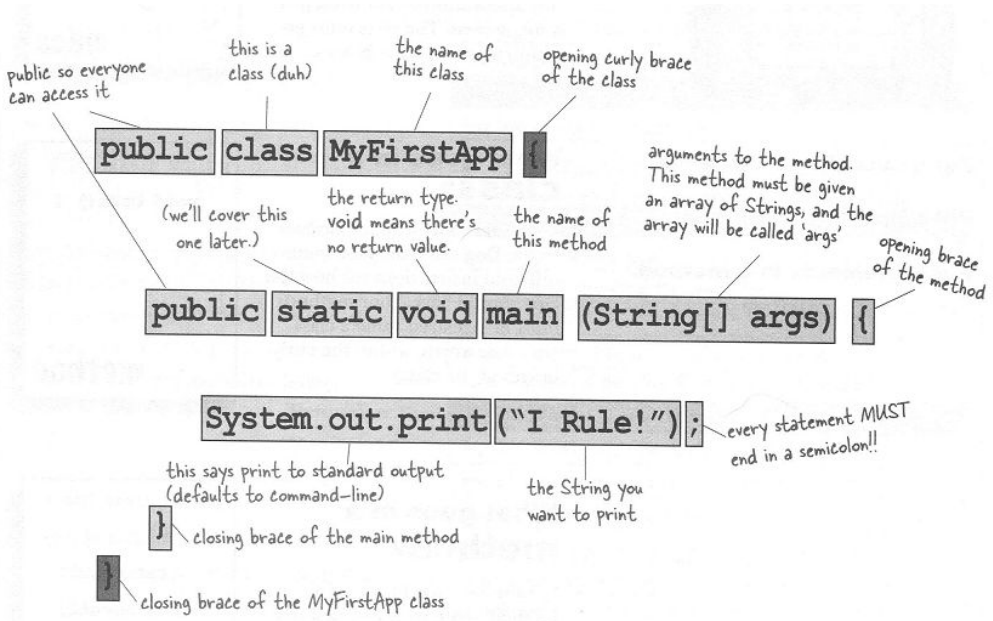
Every Java application must have at least one class AND at least one main method.

(There could be more than one main method, but the JVM will only run the one in the class you tell it to run.)

Template `main` method

The `main` method must look like this:

```
public static void main (String[]
args) {
  // your code goes in here
}
```



Simple Class

```
public class MyFirstApp {
   public static void main (String[] args) {
      System.out.println("I, Your Name, will be
            an awesome Java developer!");
   }
}
```

Save this file in C:\Java as MyFirstApp.java

Check for Java

Now we need Powershell… (or Mac Terminal)

At the prompt enter `java -version`

```
C:\Java>java -version
java version "1.7.0_25"
Java(TM) SE Runtime Environment (build 1.7.0_25-b17)
Java HotSpot(TM) 64-Bit Server VM (build 23.25-b01, mixed mode)
```

This is the Java Runtime version (JRE).

Try to compile

Make sure you are in the C:\Java directory.

- Use `cd C:\Java` to get there.

Then enter:

```
javac MyFirstApp.java
```

If you see no output, you're ready to run.

- If you get errors we'll need to fix them. Look at your code carefully.

Command not recognized

If you this, need to set the environment path

```
C:\Java>javac MyFirstApp.java
'javac' is not recognized as an internal or external command,
operable program or batch file.
```

If you see no output, you're ready for step 3!

- If you get errors look carefully at your code and the errors to fix them.

Look for .class file

OK, let's double check we got our class file…

At the prompt type `ls` and you should see both a .java and a .class file

Run the code

Now to run the program, enter:

```
java MyFirstApp
```

Java Syntax

Java has all sorts of control structures - loops, if/else, comparisons - all the things you'd expect from a programming language.

The format is C-based, and is similar to C/C++, C#, JavaScript, Perl and PHP

Syntax rules...

Every statement must end with a semicolon ;

Comments begin with double forward-slashes

```
//This is a comment.
```

Block based - Match your curly braces!!! { }

- Curly braces make blocks, and generally speaking your code goes in between them.

Variables have types

Variables are given a type when declared

```
String name;
int x = 3;
```

Java is picky about types.
We say it is a strongly typed language.

Variables

Java has two types of variables:

- primitive types - like `int` (integer)
- reference types - refers to an instance of a class (like `String` or `BeerSong`)

Variables

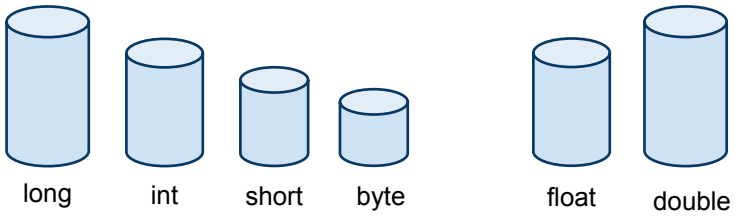
The differences between the variable types and where they are used (their scope) is very important in understanding how Java code behaves.

Primitive variables

Have different sizes… like cups.

Integer Types:	
`byte`	8 bits
`short`	16 bits
`int`	32 bits
`long`	64 bits

Floating Point Types:	
`float`	32 bits
`double`	64 bits



Primitive values

Think of a primitive variable like a cup.

The size of the cup matters!
Don't overflow the cup!

```
int x = 24;
byte b = x;   //error!
```

Type checking.

The compiler will *try* to ensure that your data doesn't "overflow" by giving you a compile error.

More primitives

There are two primitive types that are not numeric:

- `char`
- `boolean`

Character Type

A `char` is 16 bits, and corresponds to a single character - like `'A'` or `'z'`.

It is important to note that a char is not the same as a single character string.

```
'a' != "a"
```

Boolean Type

A `boolean` is either `true` or `false`.

- `true` and `false` are Java keywords

boolean values in Java are strict.
They do not equate to numeric 1 or 0

Naming variables

Use descriptive names & avoid abbreviations.

Good Variable names should:

- Start with a letter (generally a lower case letter)
- Can contain letters or numbers (underscore and dash are OK, but avoided)
- Be written in camelCase (capitalize the first letter of each word)

Naming variables

Use descriptive names & avoid abbreviations.

Good Variable names should:

- Start with a letter
- Can contain letters or numbers
- Be written in camelCase

Naming variables

Variable names cannot be a Java keyword.

Some keywords we've seen are:

```
public  static  void  class
boolean  char  byte  short  int
long  float  double  if  else
while  new
```