Overview Of Programming

- You tell the computer what to do in your chosen programming language (Java, C, …)
- The computer **compiles** that human readable code in to processor instructions (ASM, bytecode) that the computer understands
- A user can then **execute** the **compiled** code to run the program

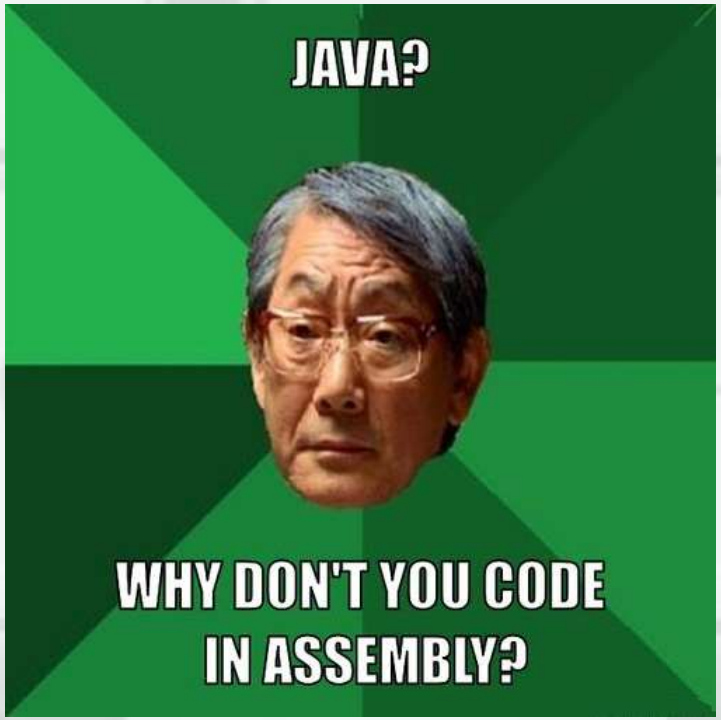
Java Overview

- Java runs in a Java Virtual Machine (JVM) so that Java code can be run on many different operating systems (Windows, Lunix, ...)

Human readable	int i = 1 + 5;
JVM bytecode	iconst_1 istore_1 iinc 1, 5

Machine code	MOV AX, 1
	MOV BX, 5
	ADD AX, BX

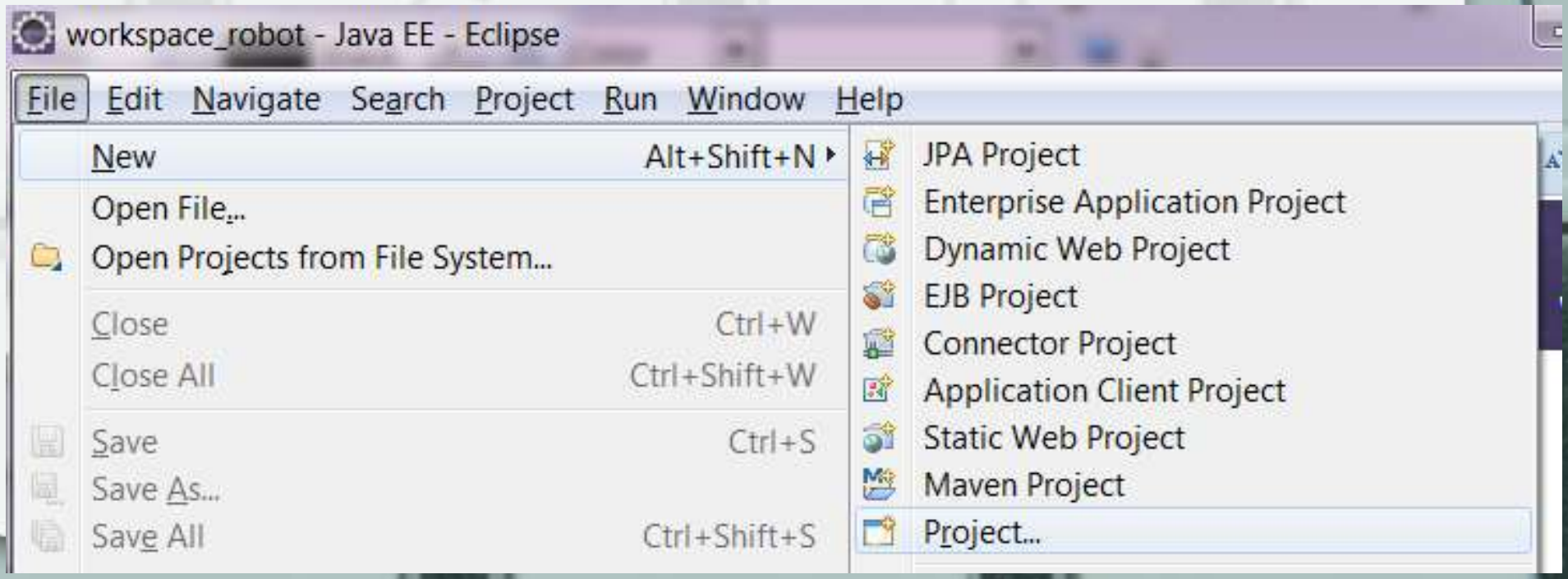
Why Not?



- I made an OS once. 1.6K lines to get text, mouse, and keyboard working
- No need to reinvent the wheel

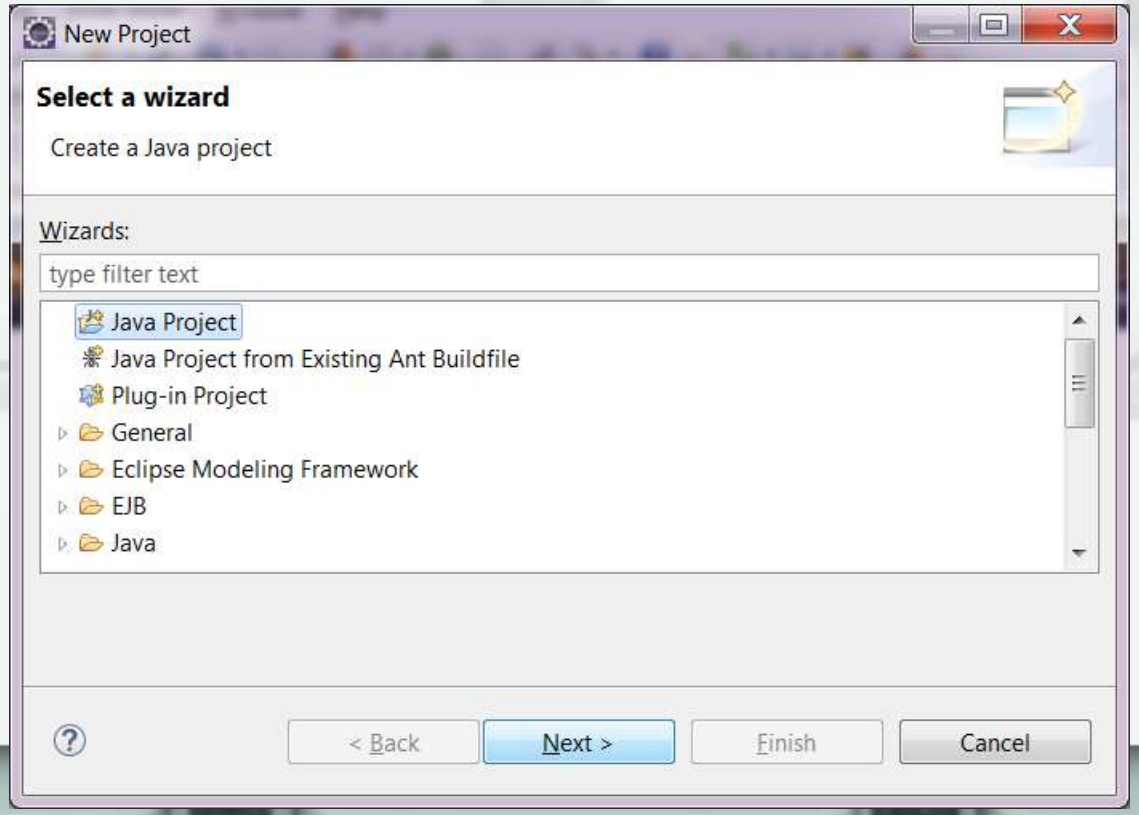
Hello World

- Traditional first program
- File > New > Project



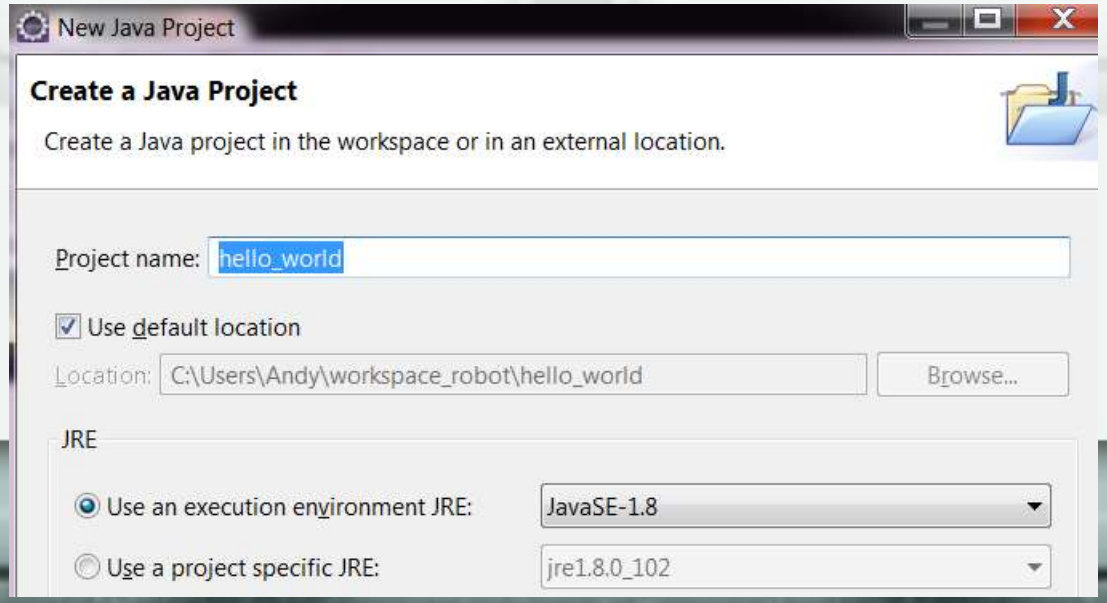
Hello World – Create Project

- Java Project, Next



Hello World – Create Project

- Project name “hello_world”, Finish
- TIP: good idea to use “_” rather than a space when programming. A space let the computer know you have finished something, so don't use it unless that's what you want

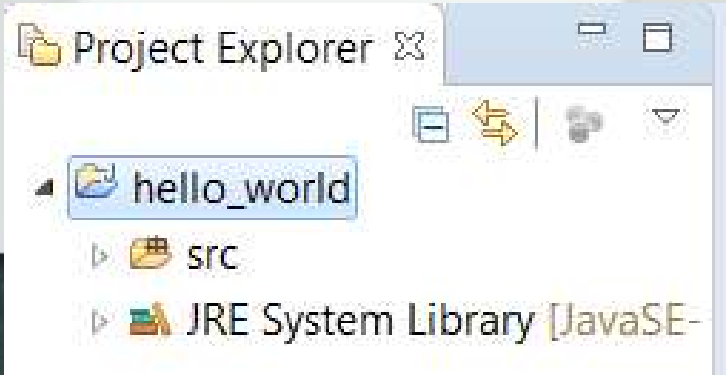


Hello World – Create Project

- Yes

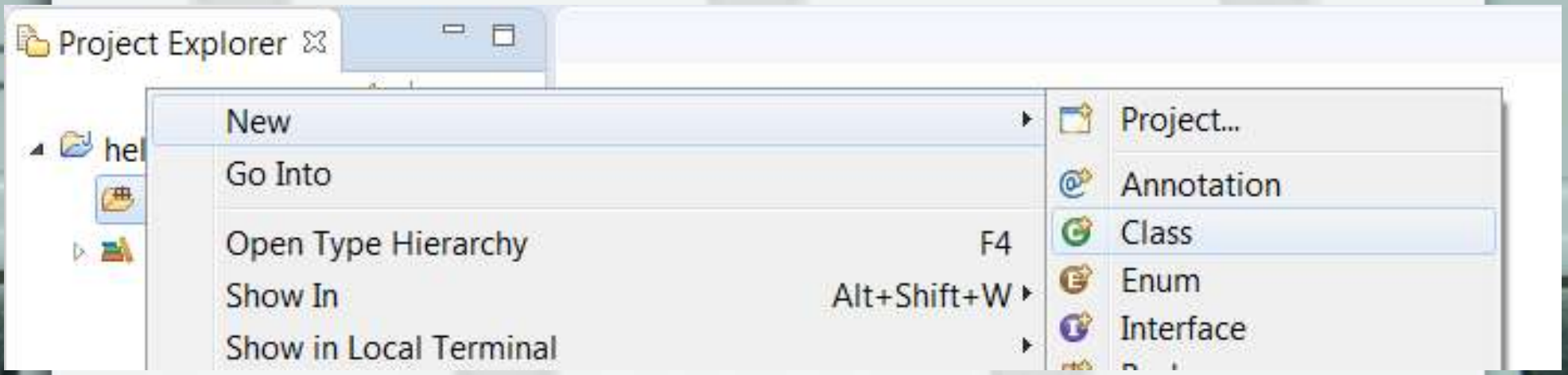
Hello World – Create Project

- We just created a Java “Project”
- A project is a group of Java classes that work together to achieve a program's goal (or part of the goal)
  - A calculator
  - A controller for a robot



Hello World – Create a class

- Right click “src” and then new > class

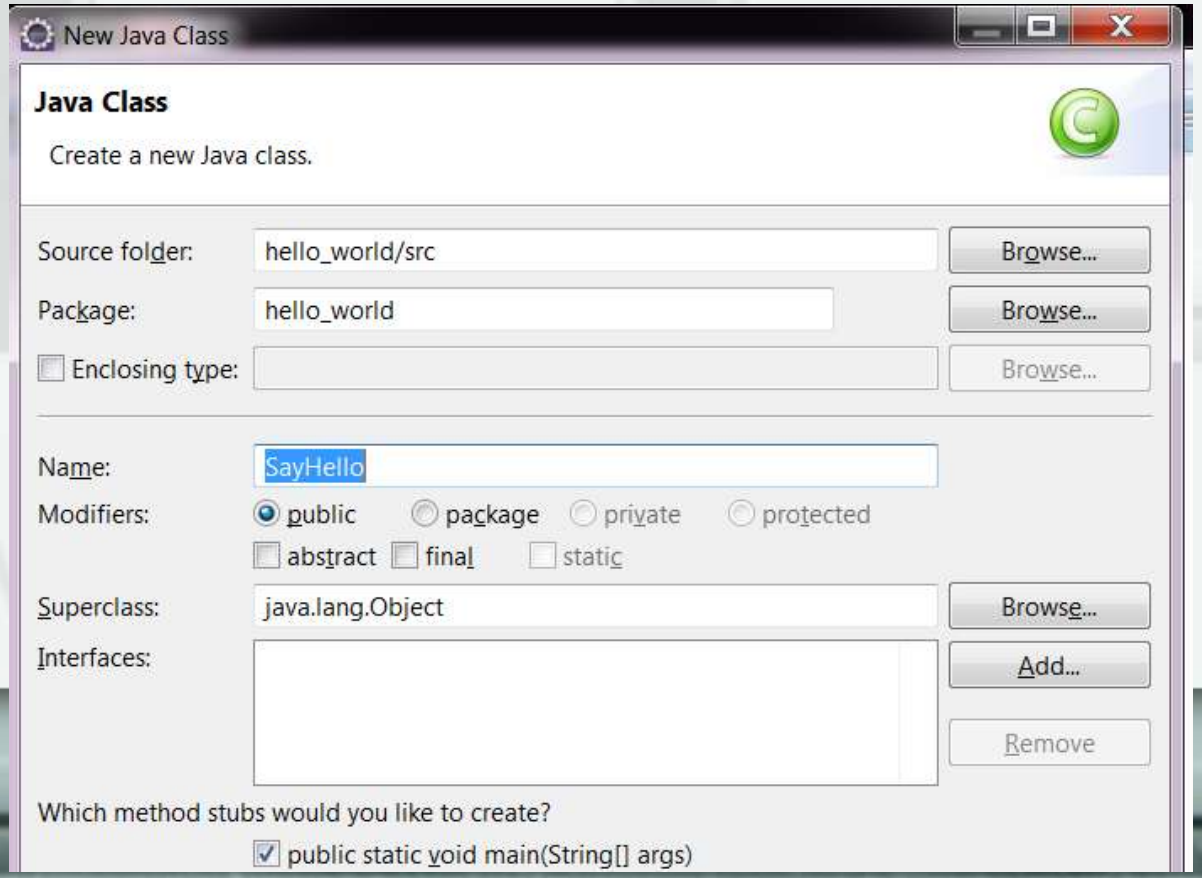


Hello World – Create a class

- A class is a Java file that contains program code
- A Java project is made up of 1 or more classes that work together
- Generally each “object” in the program will be in its own class
- A programmer can **instantiate** a class to create a instance of it to use in their program

Hello World – Create a class

- Name: SayHello
- Check: public static void main(...)



Hello World – SayHello.java

SayHello.java

```java
package hello_world;

public class SayHello {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

    }

}
```

Hello World - Package

- **Package**: A group of related classes. Like a “cars” package that may contain the classes:
  - Ford, Audi, BMW, Honda, ...
- When someone is looking to create a Toyota the “cars” package would be a good place to look

SayHello.java

```java
package hello_world;

public class SayHello {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

    }

}
```

Hello World - Class

- **Class**: The object that the code inside the file relates to. Almost anything could be a class if it makes sense.
  - We could have a “calculator” class that performed math operations
  - If our calculator was complex we may split down further and have “button”, “display”, “power”, etc. classes

Hello World - Method

- **Method**: Some functionality that we want grouped together within our class so that we can "**call**" it. In a calculator we may want to call "add", "subtract", "multiple", etc.
- **Arguments**: Data that we can pass to a method, such as add(1, 5)

SayHello.java

```
1  package hello_world;
2
3  public class SayHello {
4
5      public static void main(String[] args) {
6          // TODO Auto-generated method stub
7
8      }
9
10 }
11
```

Hello World – Return Type

- **Return Type**: What does our method return? Currently nothing. A calculator "add" method would probably return a number, like:
  - public int add (int x, int y)
  - It takes the arguments x and y and **returns** another integer which is x and y added together

SayHello.java

```
1  package hello_world;
2
3  public class SayHello {
4
5      public static void main(String[] args) {
6          // TODO Auto-generated method stub
7
8      }
9
```

Hello World - Visibility

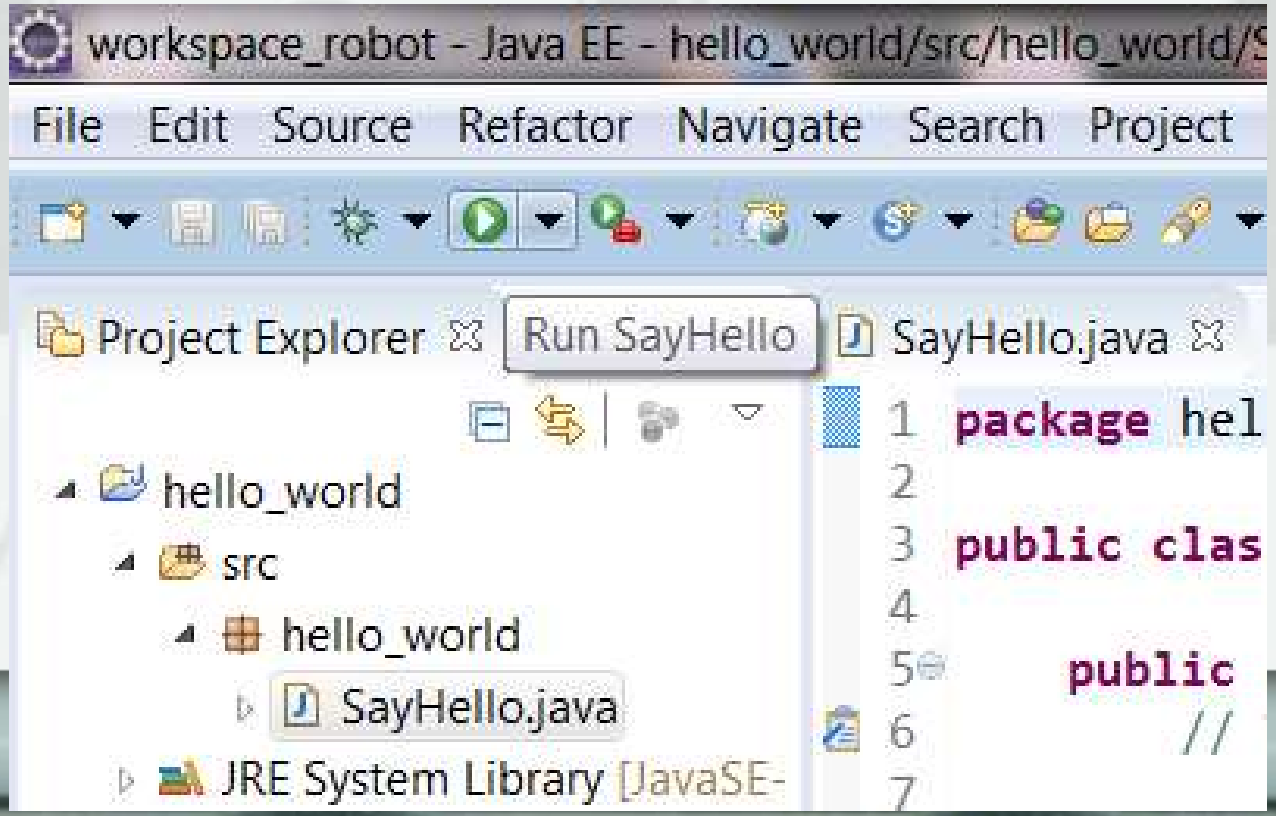
- **Visibility**: Who can call this method? If we have a method “delete_all_files()” then perhaps we don't want to allow everyone access to it.
- Our method is “public” so everyone can current call it

SayHello.java

```
1  package hello_world;
2
3  public class SayHello {
4
5      public static void main(String[] args) {
6          // TODO Auto-generated method stub
7
8      }
9
10 }
11
```

Hello World – Run it

- Click on HelloWorld.java
- Click the “Run” button



Hello World – What happened

- Our program ran
- It did exactly what we've told it to do – nothing
- Eclipse called javac to compile our HelloWorld.java to HelloWorld.class (bytecode) and then ran it

Hello World – Objects

- Java is an OBJECT ORIENTED language
- We organize our code based on unit that make sense to place together as an Object
- For example, we may have a Car object
  - It may have a Tire Object
  - A door Object
  - A seat Object
- The objects will have properties such as color, pressure, etc.

Hello World – Say Hello

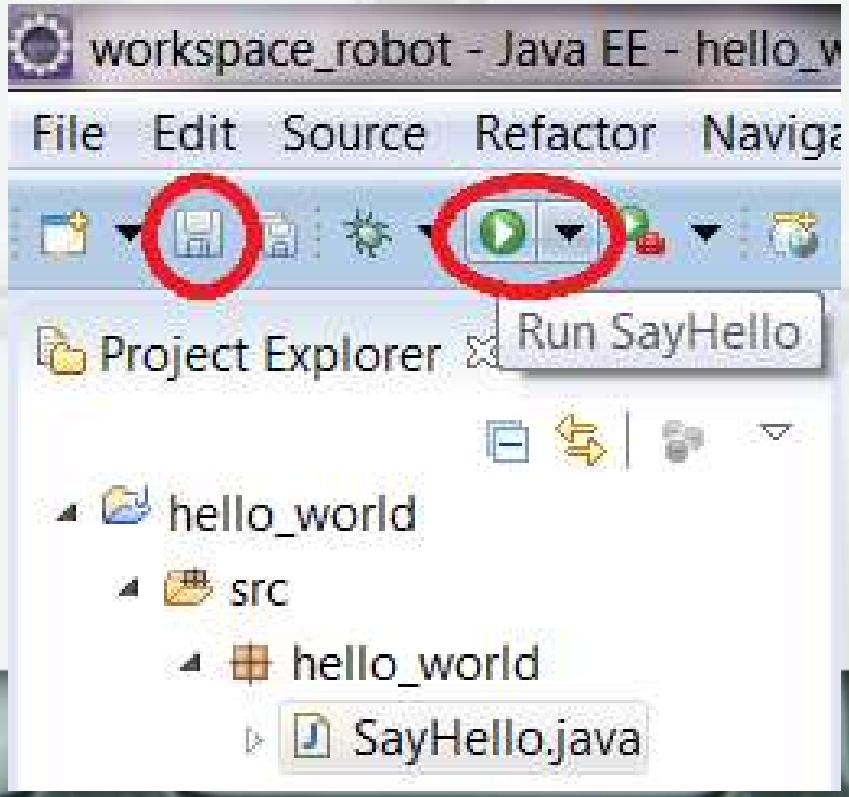
- Use the System object to output text to the computer system
- System.out.println("Hello world!")

SayHello.java

```
1 package hello_world;
2
3 public class SayHello {
4
5     public static void main(String[] args) {
6         System.out.println("Hello world!")
7     }
8
9 }
```

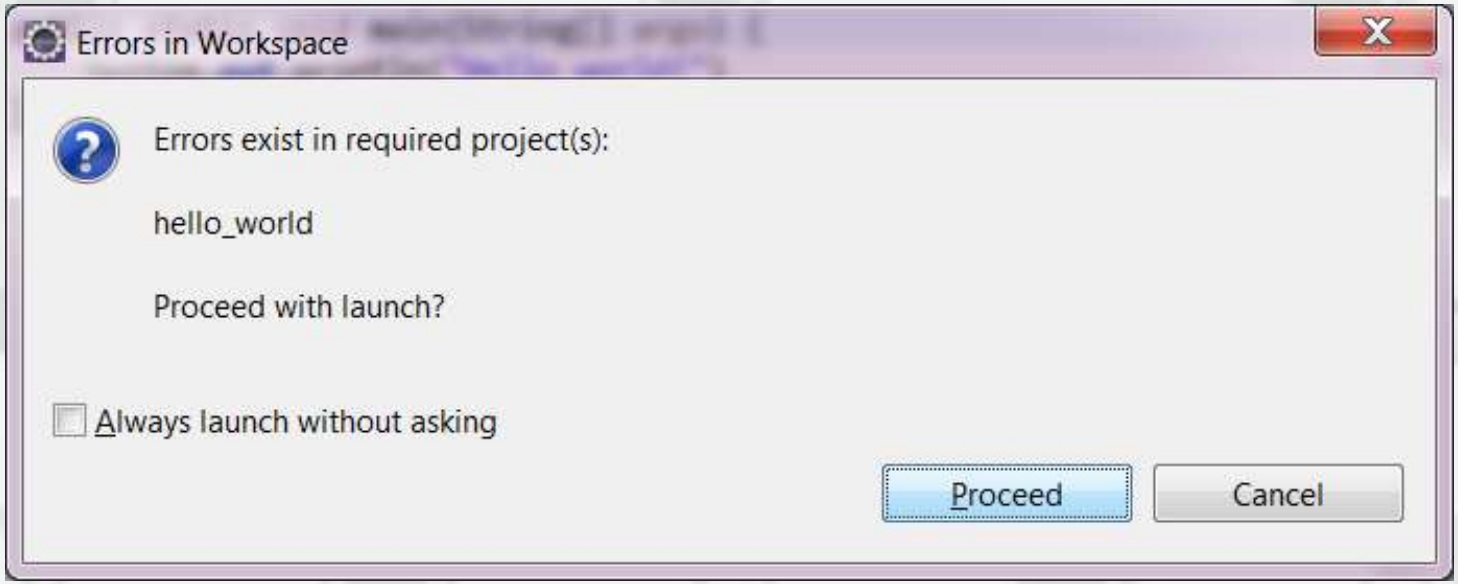
Hello World – Save and Run

- Click “Save”
- Click “Run”

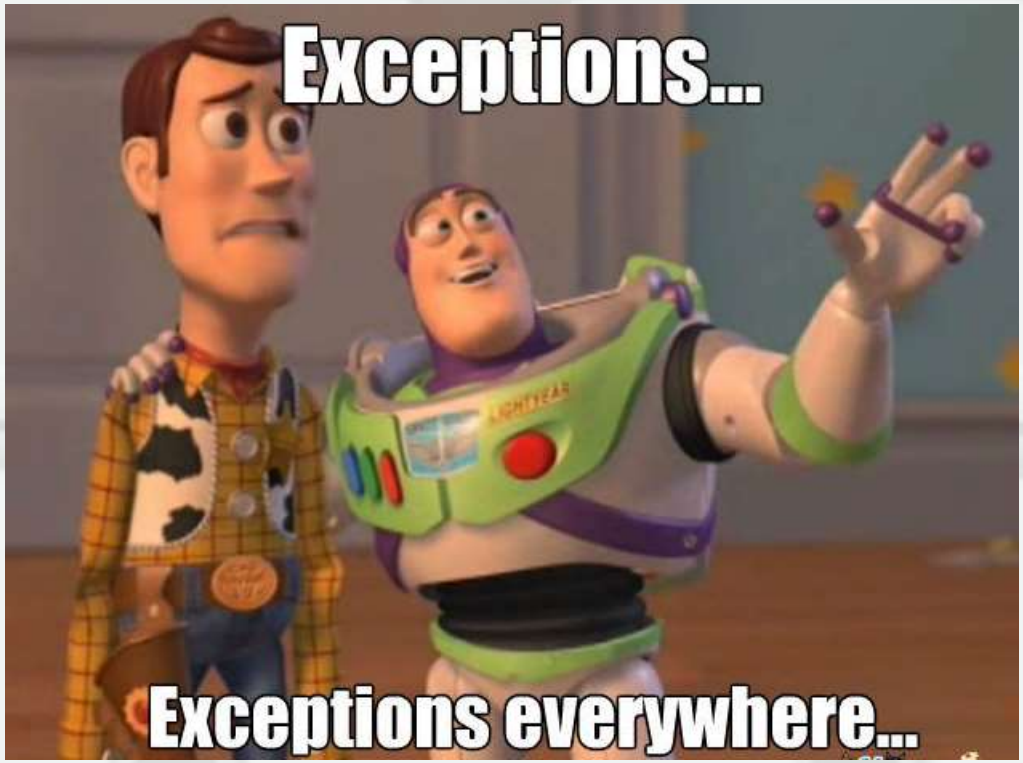


Hello World – Save and Run

- Click “Proceed”



Hello World – Oh no!



Hello World – IDE help

- The console will show us the problem

Markers | Properties | Servers | Data Source Explorer | Snippets | Console

```
<terminated> SayHello [Java Application] C:\Program Files\Java\jre1.8.0_102\bin\javaw.exe (Aug 7, 201
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
	Syntax error, insert ";" to complete BlockStatements

	at hello_world.SayHello.main(SayHello.java:6)
```

- Also, click “Markers”

Markers | Properties | Servers | Data Source Explorer | Snippets | Console

1 error, 0 warnings, 2 others

Description	Resource	Path	Location
Java Exception Breakpoints (2 items)			
Java Problems (1 item)			
Syntax error, insert ";" to complete BlockStatements	SayHello.java	/hello_world/src/hell...	line 6

Hello World - ;-)

- ; tells Java that you have finished a command. Other space is just “white space”

```
5⊖    public static void main(String[] args) {
6         System.out.
7         println
8         ("Hello world!");
```

```
5⊖    public static void main(String[] args) {
6         System.out.println("Hello world!");
7     }
```

Hello World – Fixing it

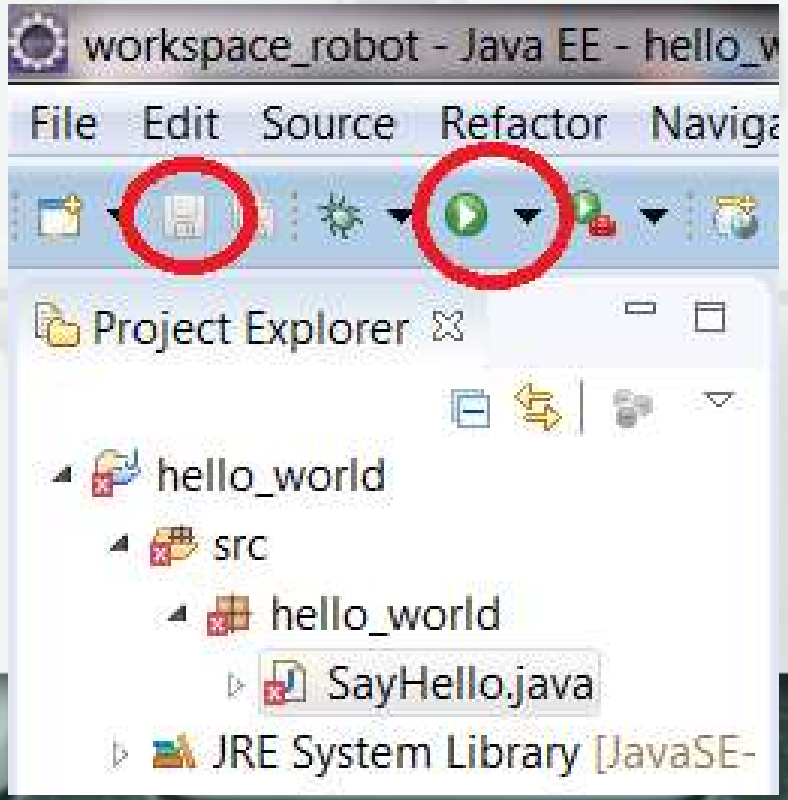
- System.out.println("Hello world!");

SayHello.java

```
1 package hello_world;
2
3 public class SayHello {
4
5     public static void main(String[] args) {
6         System.out.println("Hello world!");
7     }
8
9 }
```

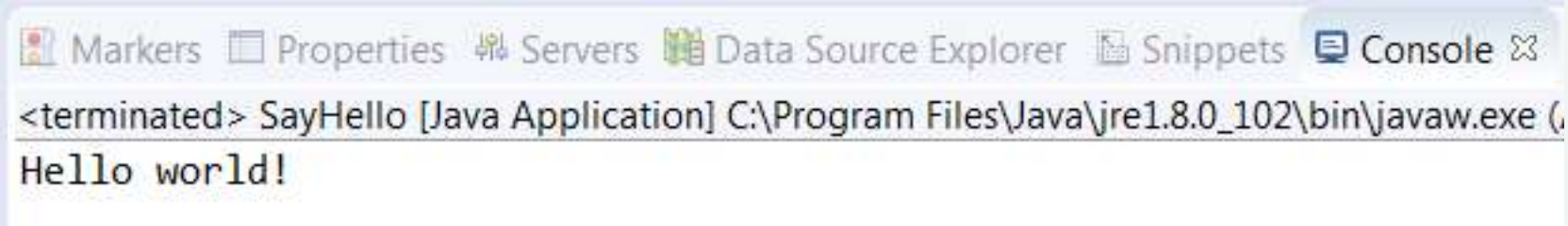
Hello World – Save and Run

- Click “Save”
- Click “Run”



Hello World – Hello!

- Look at the console at the bottom of the screen



Hello World – What happened?

- We used the “System” object to interact with out compute system
- We used the “out” object as we wanted to output to the system
- By default the System.out directs output towards our programming console
- We used “out”'s println() method to print a line of text

Debugging

- If we run the program we get the result

```
5⊖    public static void main(String[] args) {
6         String say = "Hello world!";
7         System.out.println(say);
8     }
```

Problems Javadoc Declaration Search Console

```
<terminated> test [Java Application] C:\Program Files\Java\jre1.8.0_74\bin\
Hello world!
```

Debugging

- We can double click at the left edge of a line to pause the execution of the program

```
5⊖    public static void main(String[] args) {
6         String say = "Hello world!";
7         System.out.println(say);
8     }
9
```

Debugging

Debug - test/src/test/test.java - Eclipse

File Edit Source Refactor Navigate Search Project WPILib Run Window Help

Quick Access | Java | Debug

Debug

- test [Java Application]
  - test.test at localhost:49543
    - Thread [main] (Suspended (breakpoint at line 7 in test))
      - test.main(String[]) line: 7
  - C:\Program Files\Java\jre1.8.0_74\bin\javaw.exe (22 Aug 2016, 13:18:16)

Variables | Breakpoints

Name	Value
args	String[0] (id=16)
say	"Hello world!" (id=18)
hash	0
value	(id=23)

FOXACID.java | EditedWebcam... | ShootDistanc... | PIDTurnCont... | test.java

```
1  package test;
2
3  public class test {
4
5      public static void main(String[] args) {
6          String say = "Hello world!";
7          System.out.println(say);
8      }
9
10     /*  Integer x = new Integer(10);
11         Integer y = new Integer(10);
```

Outline

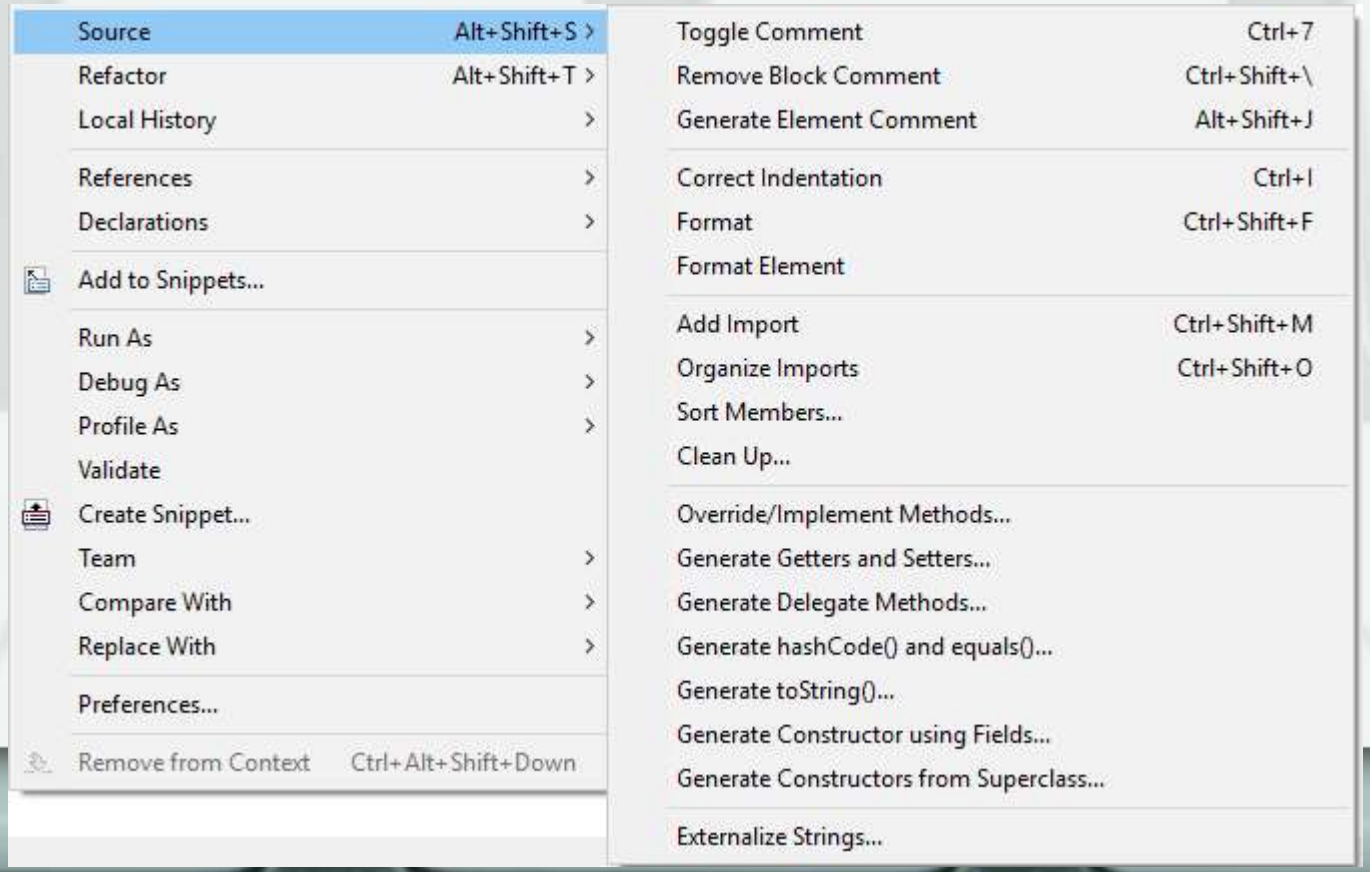
- test
- test
  - main(String[]) : void

Console | Tasks | Problems | Executables

test [Java Application] C:\Program Files\Java\jre1.8.0_74\bin\javaw.exe (22 Aug 2016, 13:18:16)

Generate Code

- We can get Eclipse to generate code for us



Hello World - Overview

- We make our first Java class
- We interacted with the System object
- We fixed an Exception
- We ran our program
- You're now officially a Java programmer

How to become a good programmer

- Just do it ©®™
  - I started by cheating at computer games
- THINK BEFORE you type
  - Less is more in programming
- Solve the problem first, don't just use the language
  - Learn pseudocode - the best programming language
- Use Stack Overflow, it's the programmer's version of a dictionary
- Divide and conquer
  - Always the best approach (...most of the time)
  - If you can't solve a problem, make it two smaller ones, then four...
  - The best go-to answer in interviews
- Zero is the first number: 0, 1, 2, 3, …
- **Never listen to Evan or Rob**