Introduction to Programming & Java

Brandon Krakowsky

Penn Engineering

What is Programming?

- A *program* is a collection of instructions that performs a specific task (or set of tasks) when executed

What is Programming?

- A *program* is a collection of instructions that performs a specific task (or set of tasks) when executed
- *Programming* is a way of specifying (or writing) the instructions

What is Programming?

- A *program* is a collection of instructions that performs a specific task (or set of tasks) when executed
- *Programming* is a way of specifying (or writing) the instructions
- *Programming languages* vary in many ways:
  - Syntax: Structure or grammar of the language
  - Semantics: Meaning of the code. What will it do when I run it?
  - Speed
  - Memory management
  - Etc.

Client-Side vs. Server-Side Programming

- *Client-side* programs run on a *client*
  - Client-side programming has mostly to do with a user's interaction with a user interface
    - For example, a web page is a client-side program that runs in a web browser, the client
  - Common client-side programming languages are:
    - HTML, CSS, and JavaScript

Client-Side vs. Server-Side Programming

- *Client-side* programs run on a *client*
  - Client-side programming has mostly to do with a user's interaction with a user interface
    - For example, a web page is a client-side program that runs in a web browser, the client
  - Common client-side programming languages are:
    - HTML, CSS, and JavaScript
- *Server-side* programs run on a *server* (or computer)
  - Server-side programming has mostly to do with the interaction between a user interface and a program on a server
    - For example, a web page sends messages (or requests) to a program on a server and it processes user input and interacts with a database
  - Common server-side programming languages are:
    - Java, Python, PHP, and ASP.NET

What is Java?

- Java is a *high-level programming* language
  - Provides abstraction from the details of the computer
    - Does most of the work in communicating with the computer

What is Java?

- Java is a *high-level programming* language
  - Provides abstraction from the details of the computer
    - Does most of the work in communicating with the computer
- Java is an *object-oriented programming (OOP)* language
  - Organized around objects rather than “actions”

What is Java?

- Java is a *high-level programming* language
  - Provides abstraction from the details of the computer
    - Does most of the work in communicating with the computer
- Java is an *object-oriented programming (OOP)* language
  - Organized around objects rather than “actions”
- Java can be used for:
  - Desktop GUI applications
  - Mobile app development
  - Web-based applications
  - Gaming applications
  - Big data technologies
  - Distributed systems
  - Cloud-based applications
  - *Many other purposes!*

Java is Compiled

- Java is a *compiled* language
  - It's converted to binary machine code (or Java *bytecode)*

Java is Compiled

- Java is a *compiled* language
  - It's converted to binary machine code (or Java *bytecode)*
- On the other hand, for comparison, Python is an *interpreted* programming language
  - Does not need to be *compiled*
  - Is *interpreted* by a Python *interpreter*

Java is Compiled

- Java is a *compiled* language
  - It's converted to binary machine code (or Java *bytecode)*
- On the other hand, for comparison, Python is an *interpreted* programming language
  - Does not need to be *compiled*
  - Is *interpreted* by a Python *interpreter*
- *Compiled* languages have many advantages over *interpreted* languages
  - When code is compiled, it's optimized under the hood
  - Since your program will be inspected for errors, many kinds of potential bugs will be caught early (e.g. using the same variable name twice)

Java is Compiled

- Java is a *compiled* language
  - It's converted to binary machine code (or Java *bytecode)*
- On the other hand, for comparison, Python is an *interpreted* programming language
  - Does not need to be *compiled*
  - Is *interpreted* by a Python *interpreter*
- *Compiled* languages have many advantages over *interpreted* languages
  - When code is compiled, it's optimized under the hood
  - Since your program will be inspected for errors, many kinds of potential bugs will be caught early (e.g. using the same variable name twice)
- Your program will not run if it is not compiled!

Java is Compiled

- Java is a *compiled* language
  - It's converted to binary machine code (or Java *bytecode)*
- On the other hand, for comparison, Python is an *interpreted* programming language
  - Does not need to be *compiled*
  - Is *interpreted* by a Python *interpreter*
- *Compiled* languages have many advantages over *interpreted* languages
  - When code is compiled, it's optimized under the hood
  - Since your program will be inspected for errors, many kinds of potential bugs will be caught early (e.g. using the same variable name twice)
- Your program will not run if it is not compiled!
- The IDE we'll be using for Java development, Eclipse, will compile your code for you (on the fly) as you save your work
  - It will also help you fix MANY problems in your code

Popularity of Java vs. Other Languages Using TIOBE

- The TIOBE Programming Community index is an indicator of the popularity of programming languages

Ref: https://www.tiobe.com/tiobe-index/

Popularity of Java vs. Other Languages Using TIOBE

- The TIOBE Programming Community index is an indicator of the popularity of programming languages
- It can be used to:
  - Check whether your programming skills are up to date
  - Make a decision about what programming language(s) to use when starting new projects

Ref: https://www.tiobe.com/tiobe-index/

Popularity of Java vs. Other Languages Using TIOBE

- The TIOBE Programming Community index is an indicator of the popularity of programming languages
- It can be used to:
  - Check whether your programming skills are up to date
  - Make a decision about what programming language(s) to use when starting new projects
- The ratings are:
  - Based on the number of skilled engineers world-wide, courses and third party vendors
  - Calculated based on popular search engines

Ref: https://www.tiobe.com/tiobe-index/

Popularity of Java vs. Other Languages Using TIOBE

- The TIOBE Programming Community index is an indicator of the popularity of programming languages
- It can be used to:
  - Check whether your programming skills are up to date
  - Make a decision about what programming language(s) to use when starting new projects
- The ratings are:
  - Based on the number of skilled engineers world-wide, courses and third party vendors
  - Calculated based on popular search engines
- The index is updated once a month

Ref: https://www.tiobe.com/tiobe-index/

Popularity of Java vs. Other Languages Using TIOBE

- Top 10 of the TIOBE index for March 2022

Mar 2022	Mar 2021	Change	Programming Language	Ratings	Change
1	3	↑	Python	14.26%	+3.95%
2	1	↓	C	13.06%	-2.27%
3	2	↓	Java	11.19%	+0.74%
4	4		C++	8.66%	+2.14%
5	5		C#	5.92%	+0.95%
6	6		Visual Basic	5.77%	+0.91%
7	7		JavaScript	2.09%	-0.03%
8	8		PHP	1.92%	-0.15%
9	9		Assembly language	1.90%	-0.07%
10	10		SQL	1.85%	-0.02%

- General highlights:
  - Java (and Python) are in the top 3 most popular programming languages
  - Currently, both languages have *almost the same rating*

Ref: https://www.tiobe.com/tiobe-index/

A Bit More About Java

- Java has been around for over 25 years
- It was initially called “Oak”, but renamed based on the large amounts of Java coffee consumed by the creators (true story)
- There are some recommended (but entirely optional) textbooks available
  - Head First Java (by Kathy Sierra)
  - Java in Easy Steps (by Mike McGrath)
  - Java Concepts, Early Objects (by Cay Hortsmann)

Configuring Java & Tools

Installing & Running Java

- In order to use Java, you need to first install the Java Development Kit (JDK)
  - This is the package of tools for *developing* Java-based software

Installing & Running Java

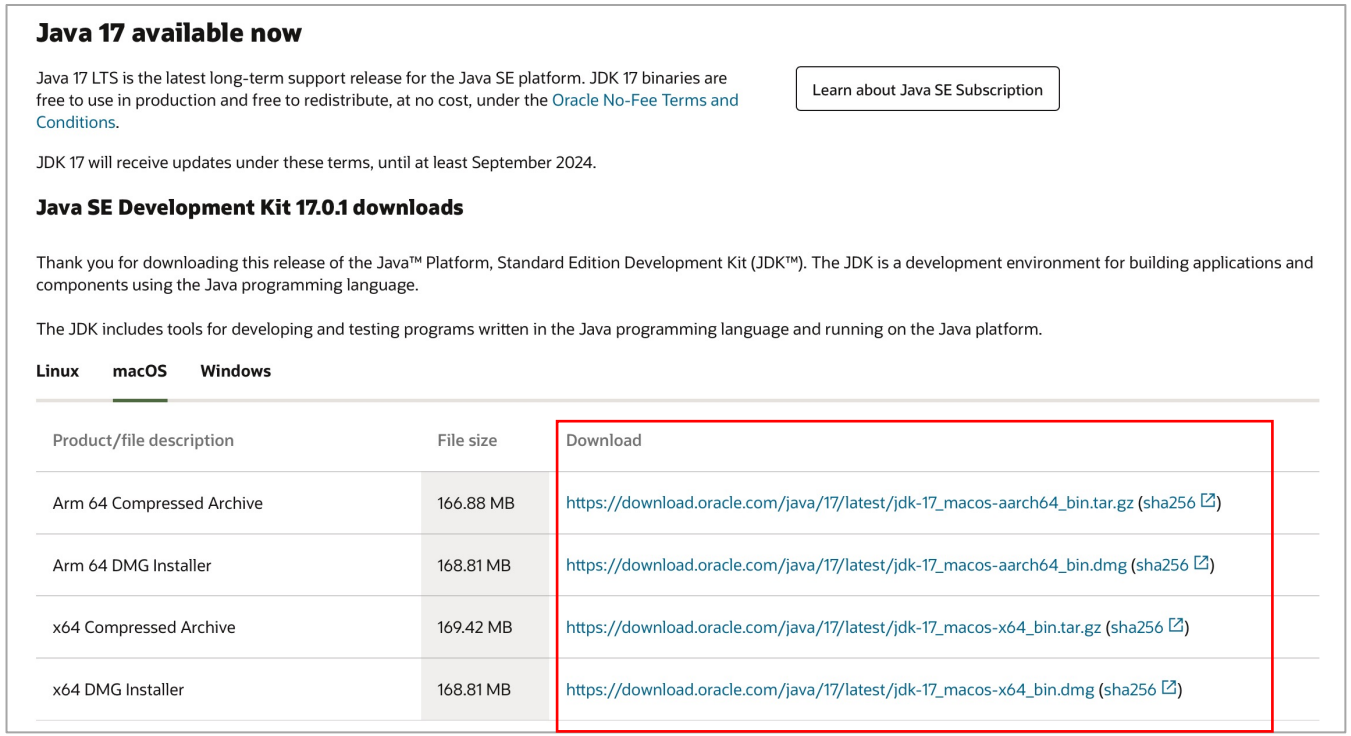
- In order to use Java, you need to first install the Java Development Kit (JDK)
  - This is the package of tools for *developing* Java-based software
- You'll also need the Java Runtime Environment (JRE) which includes the Java Virtual Machine (JVM)
  - This is the environment for *running* Java applications
    - The JVM is what actually runs compiled Java bytecode

Installing & Running Java

- In order to use Java, you need to first install the Java Development Kit (JDK)
  - This is the package of tools for *developing* Java-based software
- You'll also need the Java Runtime Environment (JRE) which includes the Java Virtual Machine (JVM)
  - This is the environment for *running* Java applications
    - The JVM is what actually runs compiled Java bytecode
- Download and install the JDK, which includes the JRE (and JVM): https://www.oracle.com/java/technologies/downloads/

Downloading and Installing the JDK

- Download and install the JDK, which includes the JRE (and JVM): https://www.oracle.com/java/technologies/downloads/
  - Download the latest version of the JDK for your OS



Java 17 available now

Java 17 LTS is the latest long-term support release for the Java SE platform. JDK 17 binaries are free to use in production and free to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions.

Learn about Java SE Subscription

JDK 17 will receive updates under these terms, until at least September 2024.

Java SE Development Kit 17.0.1 downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications and components using the Java programming language.

The JDK includes tools for developing and testing programs written in the Java programming language and running on the Java platform.

Linux **macOS** **Windows**

Product/file description	File size	Download
Arm 64 Compressed Archive	166.88 MB	https://download.oracle.com/java/17/latest/jdk-17_macos-aarch64_bin.tar.gz (sha256)
Arm 64 DMG Installer	168.81 MB	https://download.oracle.com/java/17/latest/jdk-17_macos-aarch64_bin.dmg (sha256)
x64 Compressed Archive	169.42 MB	https://download.oracle.com/java/17/latest/jdk-17_macos-x64_bin.tar.gz (sha256)
x64 DMG Installer	168.81 MB	https://download.oracle.com/java/17/latest/jdk-17_macos-x64_bin.dmg (sha256)

Eclipse

- Eclipse is one of the main IDEs for Java development

Eclipse

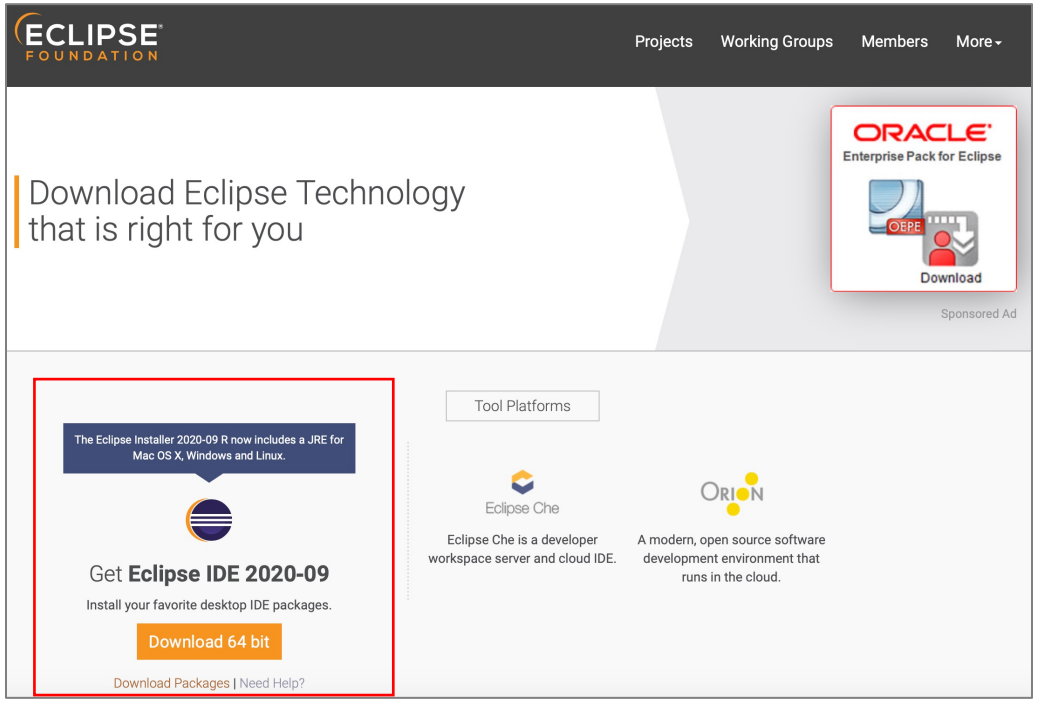
- Eclipse is one of the main IDEs for Java development
- Eclipse makes it very easy to write well-formatted Java, with good style
  - It's similar to Python's PyCharm, in that it has a TON of features
  - It compiles code on the fly, provides autocomplete suggestions, and fixes simple bugs
  - Overall, Eclipse greatly speeds up Java programming

Eclipse

- Eclipse is one of the main IDEs for Java development
- Eclipse makes it very easy to write well-formatted Java, with good style
  - It's similar to Python's PyCharm, in that it has a TON of features
  - It compiles code on the fly, provides autocomplete suggestions, and fixes simple bugs
  - Overall, Eclipse greatly speeds up Java programming
- Getting Eclipse:
  - Go to https://www.eclipse.org/downloads/ and download the latest version

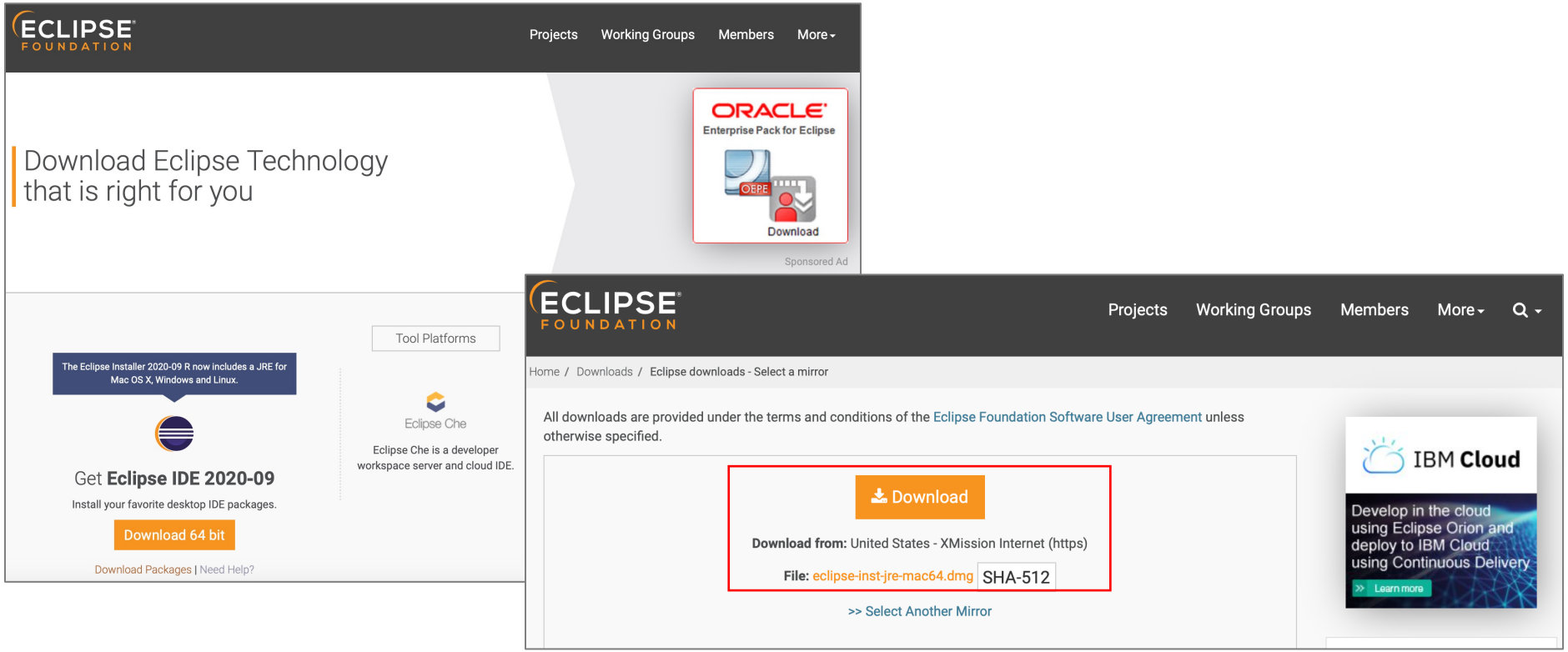
Installing & Configuring Eclipse

- Install Eclipse via https://www.eclipse.org/downloads/
  - Scroll down to get the latest version of Eclipse



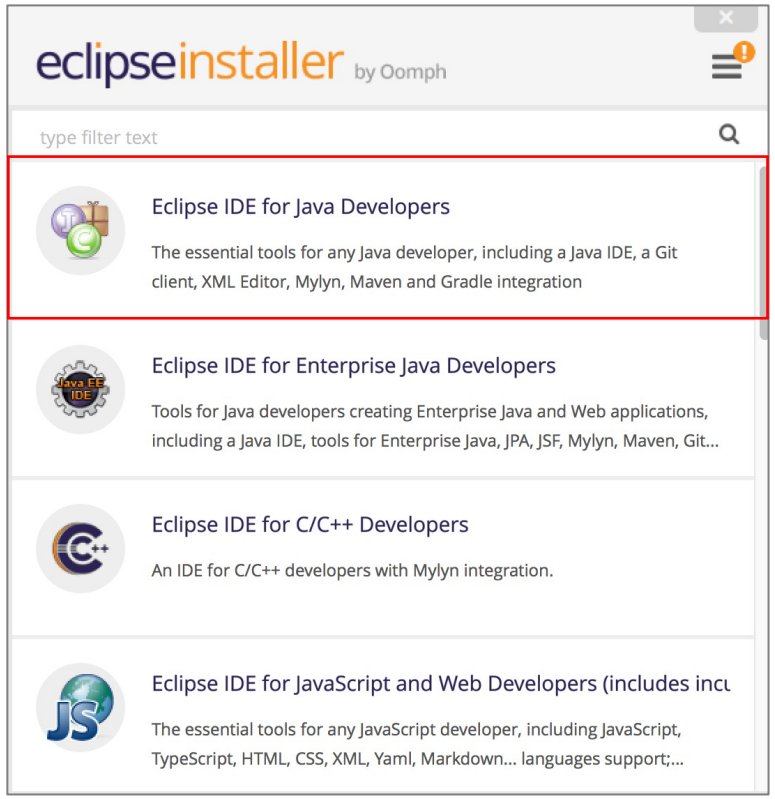
Installing & Configuring Eclipse

- Install Eclipse via https://www.eclipse.org/downloads/
  - Click to download the latest version of the IDE for your OS



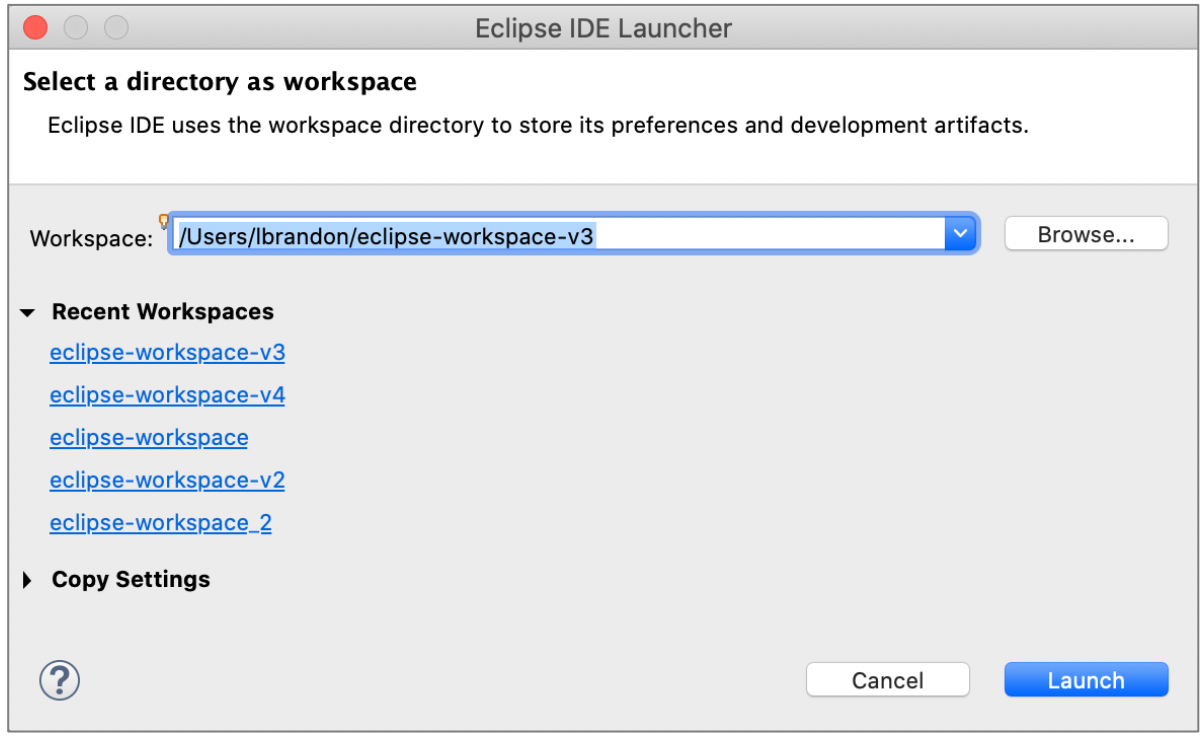
Installing & Configuring Eclipse

- When you extract and run the Eclipse Installer
  - Choose Eclipse IDE for Java Developers



Installing & Configuring Eclipse

- When you launch Eclipse, you need to specify a workspace location
  - You can use the default option (unless you have a really strong need to change it)
  - Click “Launch”



Java & Eclipse

- Eclipse stores projects in a workspace

Java & Eclipse

- Eclipse stores projects in a workspace
- When you use Eclipse to create a project (a single “program”), it creates a directory with that name in your workspace

Java & Eclipse

- Eclipse stores projects in a workspace
- When you use Eclipse to create a project (a single “program”), it creates a directory with that name in your workspace
- Within the project, you create an *optional* package (a sub-directory)

Java & Eclipse

- Eclipse stores projects in a workspace
- When you use Eclipse to create a project (a single “program”), it creates a directory with that name in your workspace
- Within the project, you create an *optional* package (a sub-directory)
- Finally, within the package, you create a class (a file)

Java & Eclipse

- Eclipse stores projects in a workspace
- When you use Eclipse to create a project (a single “program”), it creates a directory with that name in your workspace
- Within the project, you create an *optional* package (a sub-directory)
- Finally, within the package, you create a class (a file)
- For the simplest program, you’ll only need a single package (or the default “no” package), and only one (or very few) classes
  - Java is object-oriented and class-based, which means you have to create *at least one class* to write a Java program

My First Java Project

Simple Introductory Java Program

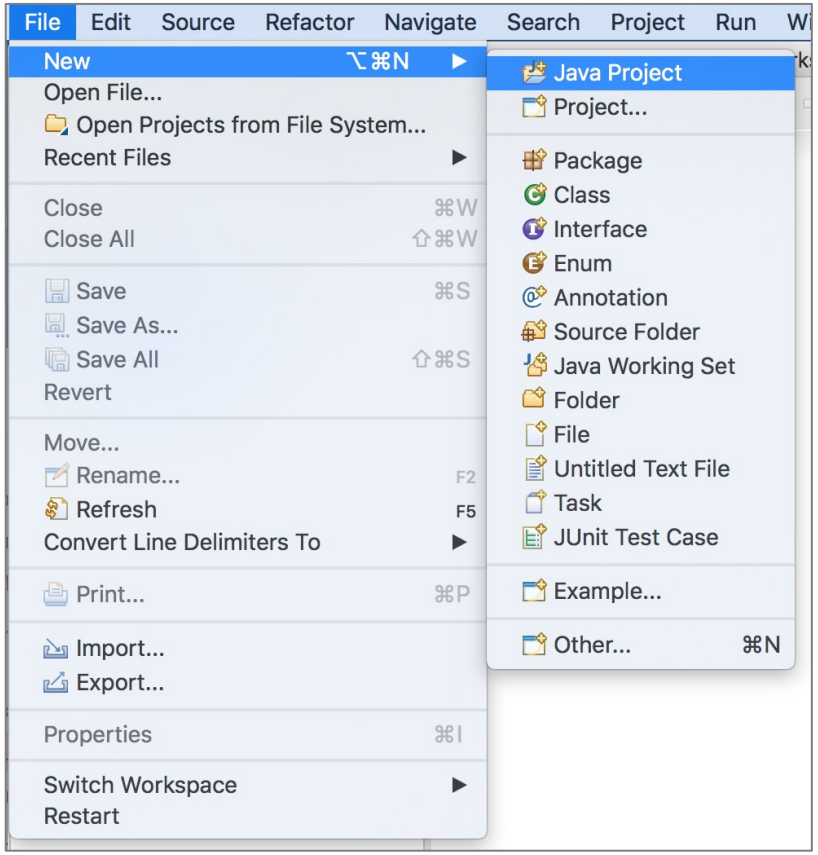
```
//Optional package declaration
package myPackage; //Should begin with a lowercase letter

//Class declaration
public class MyClass { //Should begin with a capital letter
//The Java file will be named (and saved in) ‘myPackage/MyClass.java’

    //Main method -- the starting point of any Java program
    //In Java, the name “main” is special and reserved for the main
method
    public static void main(String[] args) {
       System.out.println(“Hello World”); //Prints ‘Hello World’
    }
}
```

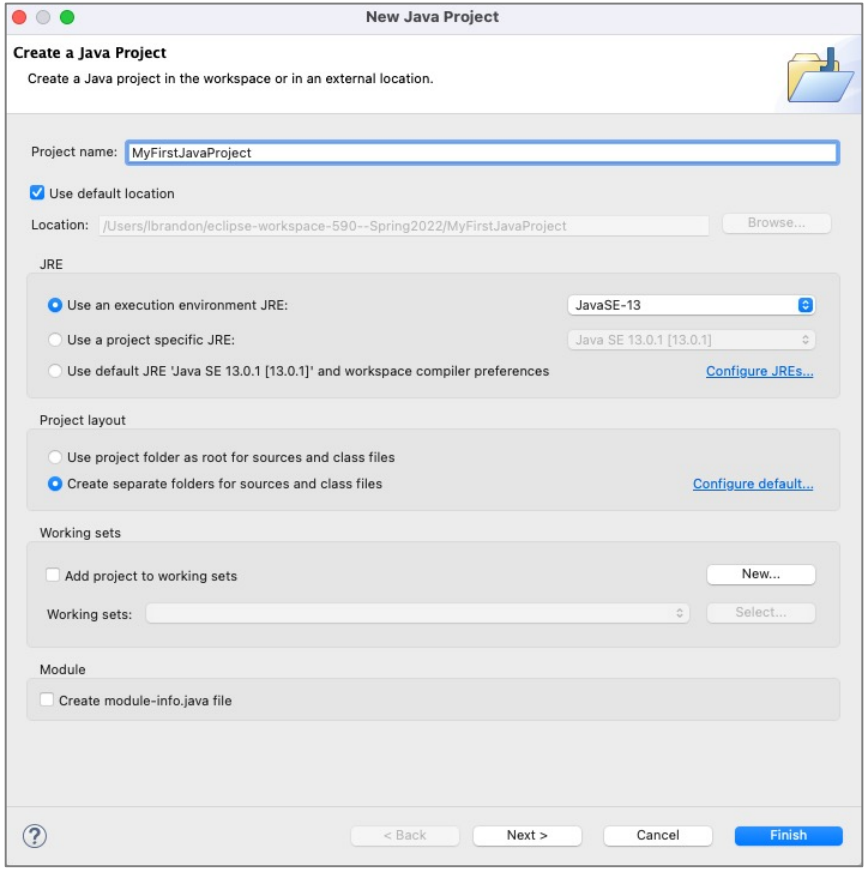
My First Java Project

- In Eclipse, go to “File” → “New” → “Java Project”



My First Java Project

- Create a Java Project in your workspace



Provide a Project name
- Project names should be capitalized

Use the default location

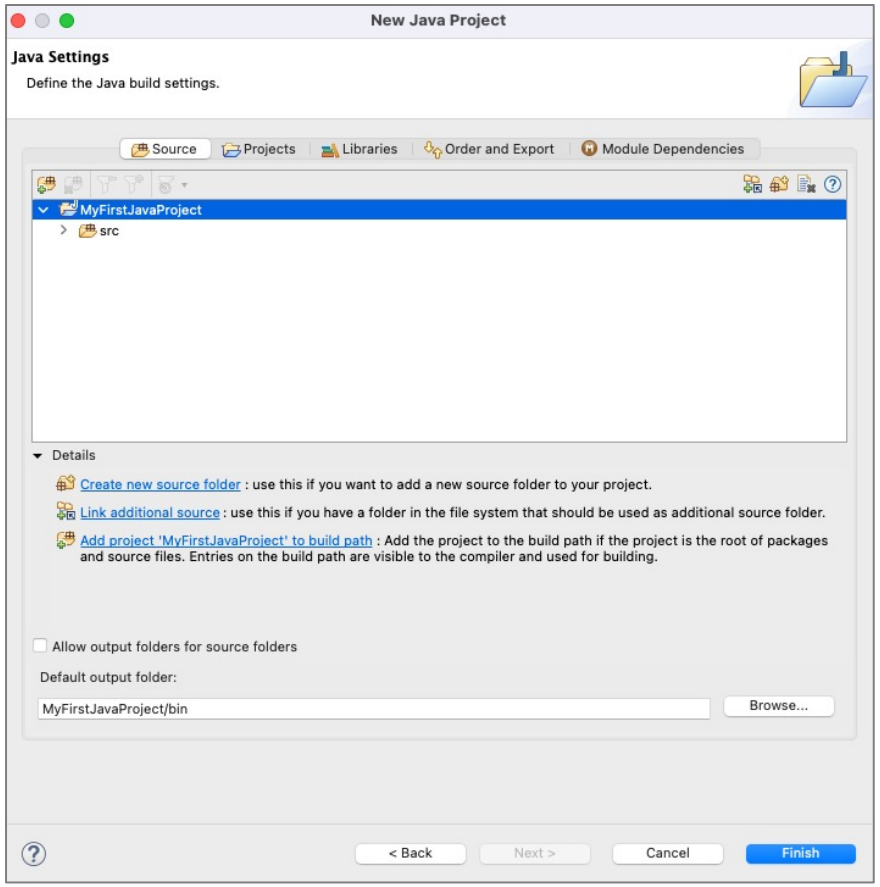
Use the default JRE and project layout

Make sure Create module-info.java file IS NOT checked

Click “Next”

My First Java Project

- Define the compilation/build settings

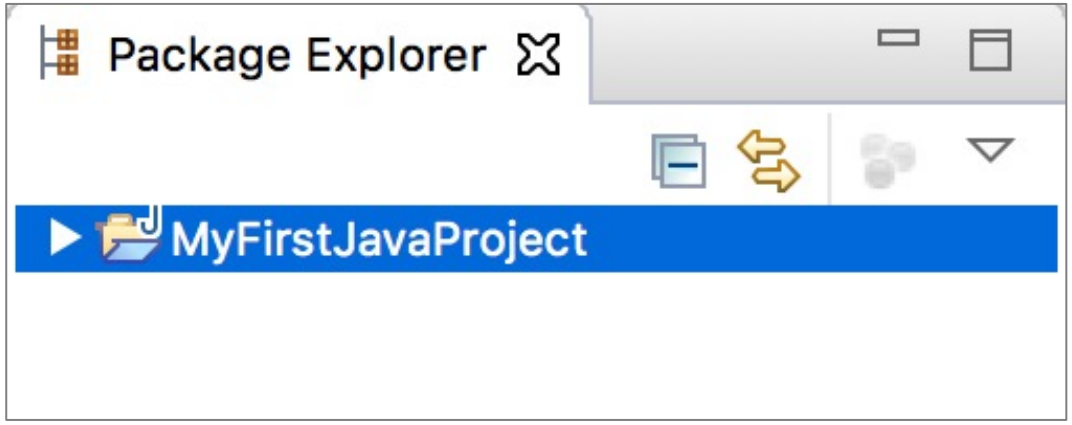


Use the default output folder

Click “Finish”

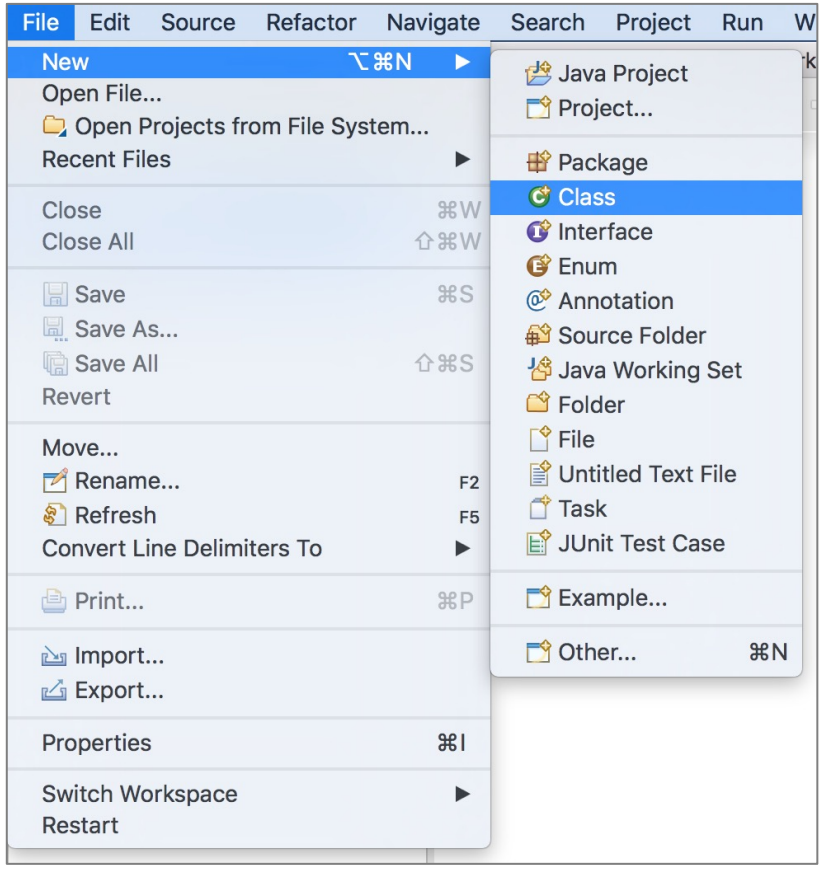
My First Java Project

- The project will appear in the Package Explorer on the left hand side in the IDE



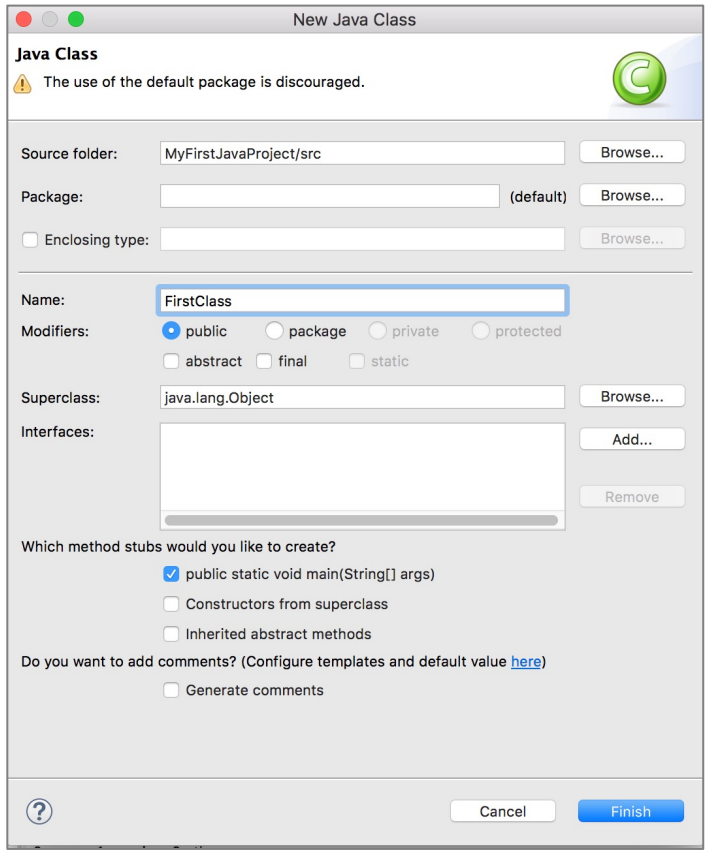
My First Java Project

- In Eclipse, go to “File” → “New” → “Class”



My First Java Project

- Create a Java Class in your Java Project



Provide a Name
- Class names should be capitalized

Make sure public static void main(String[ ] args) IS checked

Make sure Inherited abstract methods IS NOT checked

Click “Finish”

My First Class

- The entry point of any java program is the *main* method

FirstClass.java

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

Some General Rules for Java

- Individual statements end in a semicolon
  - New lines do not mean anything in Java

Some General Rules for Java

- Individual statements end in a semicolon
  - New lines do not mean anything in Java
- For example, here's a statement

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

Some General Rules for Java

- Individual statements end in a semicolon
  - New lines do not mean anything in Java
- For example, here's a statement

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

- Here's another statement

```
String myString = "My String";
```

Some General Rules for Java

- Indentation doesn't matter
  - Unlike a language like Python, where it's required, indentation in Java is a matter of style
  - While it won't make your program fail (the way it does in Python), you should indent your programs!

Some General Rules for Java

- Indentation doesn't matter
  - Unlike a language like Python, where it's required, indentation in Java is a matter of style
  - While it won't make your program fail (the way it does in Python), you should indent your programs!
- You can use these shortcuts in Eclipse
  - Fixes format of your code

  CTRL/Cmd + SHIFT + F

  - Selects all code in Java file and fixes indentation

  CTRL/Cmd + A, CTRL/Cmd + I

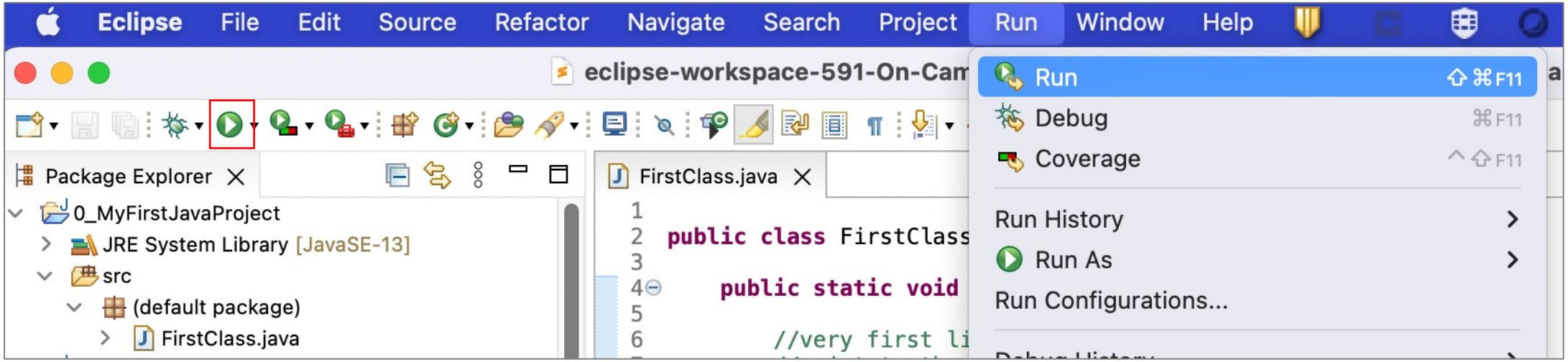
My First Class

FirstClass.java

```
1
2  public class FirstClass {
3
4      public static void main(String[] args) {
5
6          //use println to print to the console and end the line
7          System.out.println("Hello World!");
8
9      }
10 }
11
```

My First Class

- To run your Java program in Eclipse, go to Run → Run
  - Or click the “Run” button
- Keyboard shortcuts will vary based on your install of Eclipse and operating system
  - On a Mac, you should use CMD + (Fn) F11



My First Class

FirstClass.java

```
1
2  public class FirstClass {
3
4      public static void main(String[] args) {
5
6          //use println to print to the console and end the line
7          System.out.println("Hello World!");
8
9          //use print to print to the console and do not end the line
10         //(so the next thing you print will go on the same line)
11         System.out.print("Welcome, ");
12
13         //print to the console and end the line
14         System.out.println("Brandon.");
15
16         System.out.println(); //print a blank line
17
```

Java Comments

- You can, and SHOULD, add comments to your code
  - Comments can be used to document your code, and to make it more readable

Java Comments

- You can, and SHOULD, add comments to your code
  - Comments can be used to document your code, and to make it more readable
- Comments are ignored by the Java compiler

Java Comments

- You can, and SHOULD, add comments to your code
  - Comments can be used to document your code, and to make it more readable
- Comments are ignored by the Java compiler
- Here is a single line comment, using double slashes //

```
//Here is an int, initially set to 0
int myInt = 0;
```

Java Comments

- Here is a block comment, using /* */

```
/*
 * Here is an int
 * It's initially set to 0
 */
int myInt = 0;
```

Java Comments

- Here is a block comment, using /* */

```
/*
 * Here is an int
 * It's initially set to 0
 */
int myInt = 0;
```

- As a shortcut in Eclipse, you can type the following

```
/*
```

and then hit Enter

- It will add a block comment and you can fill in the rest

Java Comments

- Here is a block comment, using /* */

```
/*
 * Here is an int
 * It's initially set to 0
 */
int myInt = 0;
```

- As a shortcut in Eclipse, you can type the following

```
/*
```

and then hit Enter

- It will add a block comment and you can fill in the rest
- Adding comments to a Java file keeps your code organized & readable!

Java Comments

- Here is a block comment, using /* */

```
/*
 * Here is an int
 * It's initially set to 0
 */
int myInt = 0;
```

- As a shortcut in Eclipse, you can type the following

```
/*
```

and then hit Enter

- It will add a block comment and you can fill in the rest
- Adding comments to a Java file keeps your code organized & readable!
- At the very least, you should add comments to all non-trivial lines of code

My First Class

```
18
19          /*
20           * Data types.
21           * Every value has a type associated with it.
22           */
23
24          //int (integer): Positive or negative whole number with no decimal points
25          System.out.println(8);
26
27          //You can do math
28          System.out.println(2 + 3);
29          System.out.println(5 - 6);
30          System.out.println(2 * 3);
31
32          //Remember the order of operations.  You can use parentheses ()
33          System.out.println(3 + 5 - 2 * 6);
34          System.out.println((3 + 5 - 2) * 6);
35
36          //double (decimal): A positive or negative number that contains a decimal point
37          System.out.println(1.3);
38          System.out.println(23.0);
39          System.out.println(-5.1);
40          System.out.println(2 * 3.5);
41          System.out.println(7 / 2.0);
42
43          System.out.println(); //print a blank line
44
```

Basic Arithmetic Operators

- Arithmetic operators
  + addition
  - subtraction
  * multiplication
  / division
    - with two ints: divides and returns the largest whole number, discarding the fractional result (ex. 3 / 2 = 1)
    - with at least one double: divides and returns fractional result (ex. 3 / 2.0 = 1.5)

  Math.pow exponent (ex. Math.pow(2, 3) = 8)
  % modulus, divides and returns the remainder (ex. 7 % 5 = 2)

My First Class

```
45
46          /*
47           * Division
48           */
49
50          //Division with ints uses integer division (drops the remainder)
51          System.out.println(3 / 2);
52
53          //Division with decimals doesn't drop the remainder
54          System.out.println(3.0 / 2);
55
56          //Modulus divides and returns the remainder
57          System.out.println(3 % 2);
58          System.out.println(4 % 2);
59          System.out.println(3.1 % 2);
60
61          //The power operation raises one number to another number
62          System.out.println(Math.pow(3, 4));
63
64          System.out.println(); //print a blank line
65
```

My First Class

```
66
67          //boolean: true or false
68          System.out.println(1 == 2);
69          System.out.println(1 < 2);
70          System.out.println(1.2 >= 1.2);
71          System.out.println(8.9 != 8);
72
73          //How do we know that 500002 is an even number?
74          System.out.println(500002 % 2 == 0);
75
76          //Is 500003 odd?
77          System.out.println(500003 % 2 >= 1);
78
79          System.out.println(); //print a blank line
80
```

Basic Comparison Operators

- Comparison operators compare values and determine their relationship
  == equal, used for comparing primitive values (e.g. 2 == 2)
  != not equal, used for comparing primitive values (e.g. 2 != 3)
  < less than
  > greater than
  <= less than or equal to
  >= greater than or equal to

My First Class

```
81
82          /*
83           * Strings
84           */
85
86          //String: Characters enclosed within double quotes
87          System.out.println("Nice!");
88
89          //Concatenate (link together) strings using a +
90          System.out.println("Wow!" + " Java is cool!");
91
```

My First Class

```
91
92          //Printing strings with special characters
93          //In Java strings, the backslash (\) is a special character, also called
94          //the "escape" character
95
96          //Prefixing a special character (e.g. double quotes) with a backslash (\)
97          //turns it into an ordinary character
98          System.out.println("Brandon says \"Hello!\"");
99
```

My First Class

```
99
100         //Concatenate strings with numbers
101         System.out.println("4 % 2 = " + 4 % 2);
102
103         //Concatenate strings with booleans
104         System.out.println("Is 4 even? " + (4 % 2 == 0));
105
106         System.out.println(); //print a blank line
107
```

My First Class

```
108
109         /*
110          * Casting.
111          * Converting from one data type to another.
112          */
113
114         //Cast an int to a String and get the type (Class)
115         System.out.println(Integer.toString(1).getClass());
116
117         //Cast a String to an int
118         System.out.println(Integer.parseInt("1"));
119
120         System.out.println(); //print a blank line
121
122
123         System.out.println("Goodbye!");
124
125     }
126 }
127
```

Homework 1

Homework 1

Will be assigned by tonight, Wednesday, 08/31/22 at midnight and due Wednesday, 09/07/22 at midnight

- It's designed to give you practice writing Java and using Eclipse
- It's deliberately easy and shouldn't take you more than an hour
- The topics are: Math, Data Types, Strings, and Printing

To complete the assignment:

- Submit your completed Java code file to Canvas

Announcement: Analytics Accelerator

Analytics Accelerator

Wharton Customer Analytics (WCA) will be hosting the 10th Analytics Accelerator during the fall semester. This is a unique experiential learning opportunity for Wharton and Penn students.

- The Analytics Accelerator will match selected students, working in teams, with companies to provide implementable solutions to their most pressing marketing and analytics problems
- Great opportunity to broaden your analytics skillset through a monthlong experiential learning project and work with actual companies using real-world data
- The student application is now open to work with WCA partners: Align Technology, Lowe's, Master Kong, and Nielsen
- Applications from interested students will be accepted until 11:59 p.m. ET on September 18

Go here: https://wca.wharton.upenn.edu/students/student-analytics-accelerators/ for more information and to apply!