



CS 351: Design of Large Programs Hello World with the IntelliJ IDE

Instructor: **Joel Castellanos**
e-mail: joel.unm.edu

Web: <http://cs.unm.edu/~joel/>
Office: Electrical and Computer Engineering building (ECE), Room 233



8/31/2016

Install Java Development Kit (JDK) 1.8

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Most computers already have the **Java Runtime Environment** installed. However, to create Java programs, the **Java Development Kit** is required. Note: the JDK includes a copy of the matching version of the JRE.



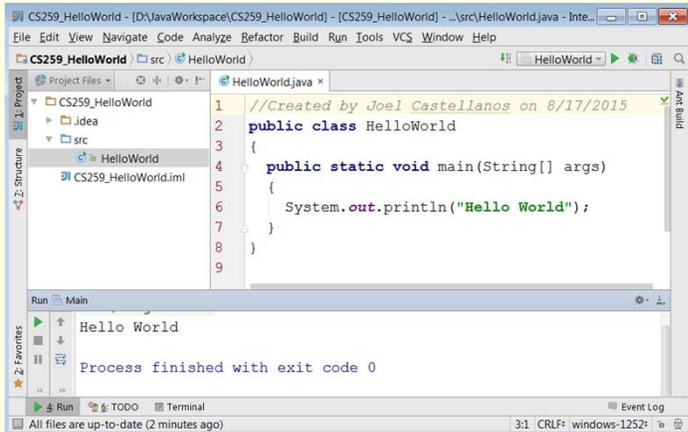
Download and Install JDK 1.8

IntelliJ IDE (Integrated Development Environment)

- <https://www.jetbrains.com/idea/download/>
- IntelliJ Community Edition (free) version 14.1.

**Java is a
Programming
Language.**

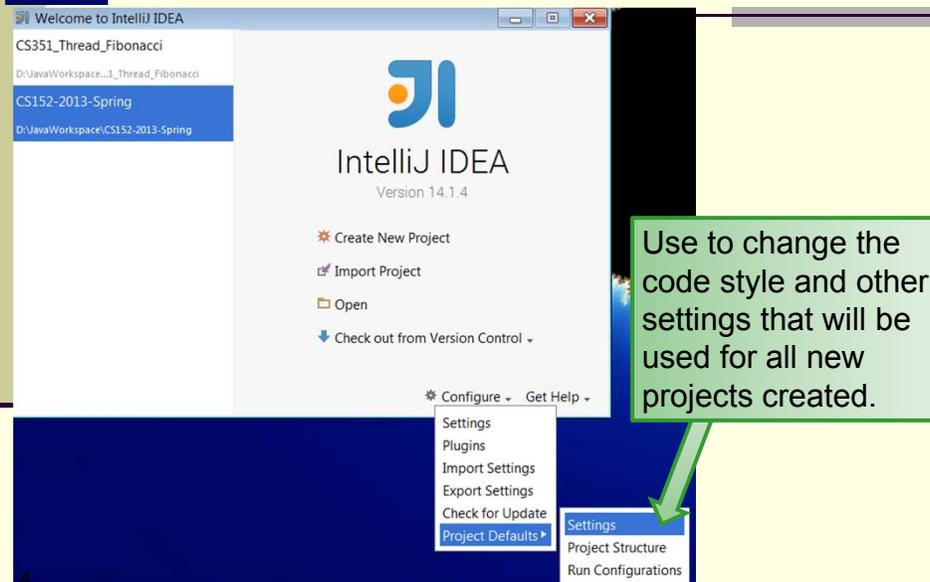
**IntelliJ is an
Integrated
Development
Environment.**



```
CS259_HelloWorld - [D:\JavaWorkspace\CS259_HelloWorld] - [CS259_HelloWorld] - ...src\HelloWorld.java - Inte...
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
CS259_HelloWorld [src] HelloWorld
Project Files
CS259_HelloWorld
  idea
  src
    HelloWorld
    CS259_HelloWorld.iml
HelloWorld.java
1 //Created by Joel Castellanos on 8/17/2015
2 public class HelloWorld
3 {
4     public static void main(String[] args)
5     {
6         System.out.println("Hello World");
7     }
8 }
9
Run Main
Hello World
Process finished with exit code 0
Run Run TODO Terminal Event Log
All files are up-to-date (2 minutes ago) 3:1 CRLF windows-1252
```

3

Project Default Settings (Code Style)



Welcome to IntelliJ IDEA

CS351_Thread_Fibonacci
D:\JavaWorkspace..._Thread_Fibonacci
CS152-2013-Spring
D:\JavaWorkspace\CS152-2013-Spring

IntelliJ IDEA
Version 14.1.4

- * Create New Project
- Import Project
- Open
- Check out from Version Control -

Configure - Get Help -

- Settings
- Plugins
- Import Settings
- Export Settings
- Check for Update
- Project Defaults

Settings
Project Structure
Run Configurations

Use to change the code style and other settings that will be used for all new projects created.

4

Code Style: IntelliJ → File → Settings...

“Manage...” to create custom scheme.

5

Code Style: Braces

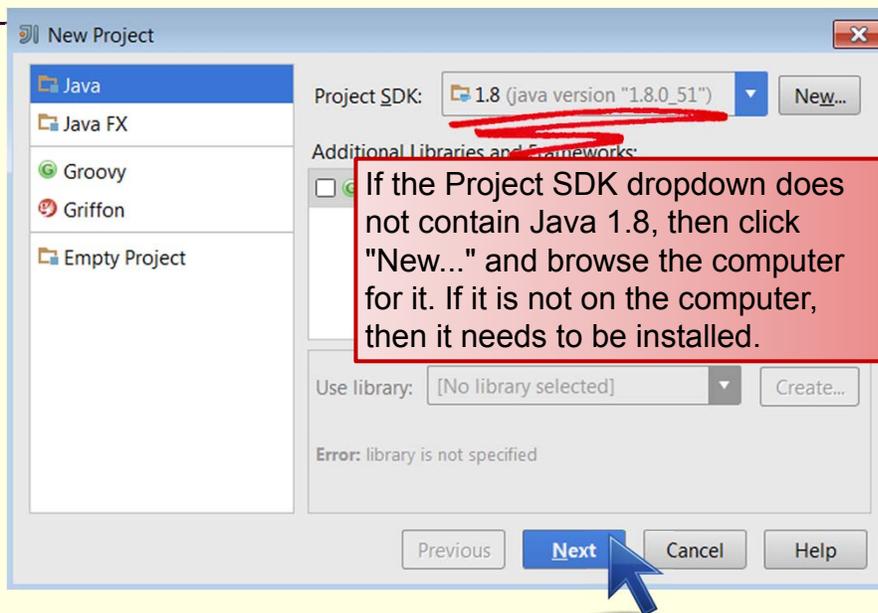
6

IntelliJ: Create New Project (1 of 4)



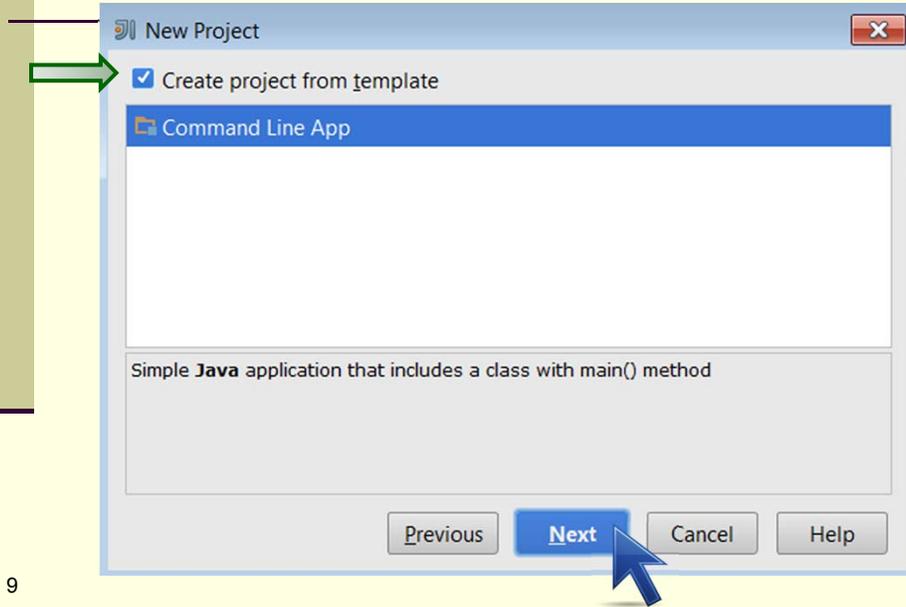
7

IntelliJ: Create New Project (2 of 4)



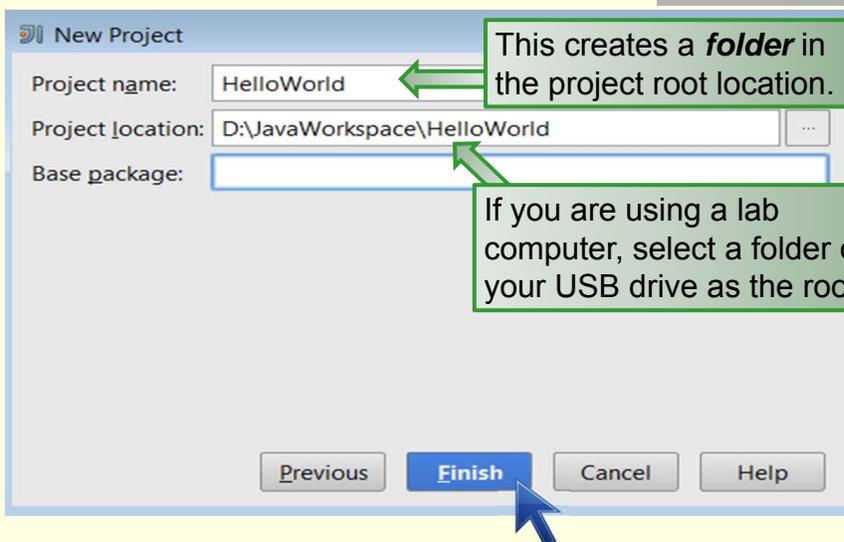
8

IntelliJ: Create New Project (3 of 4)



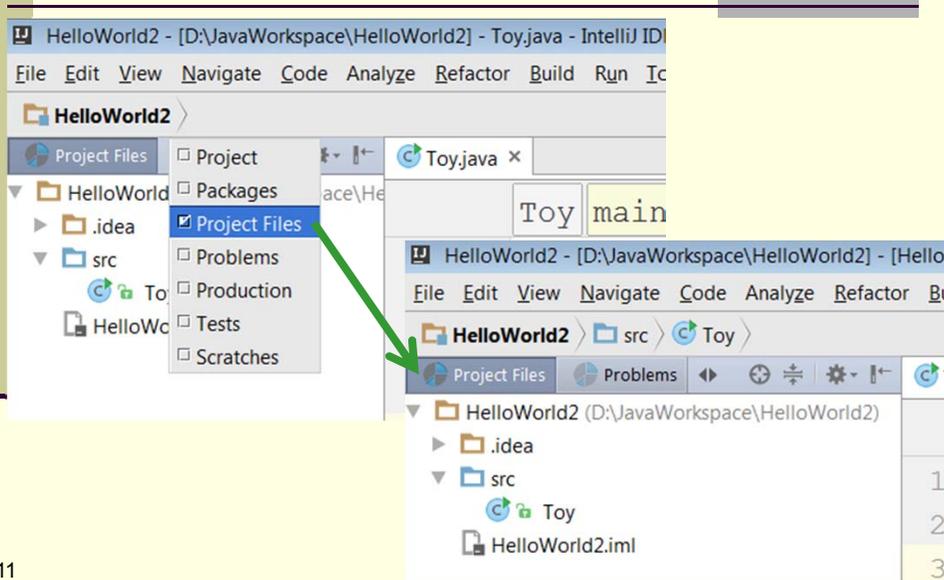
9

IntelliJ: Create New Project (4 of 4)



10

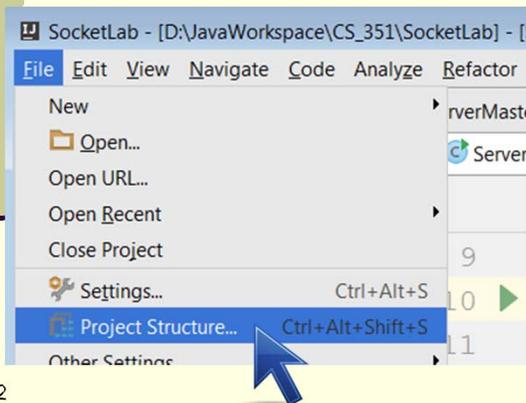
Setting IntelliJ to Project Files View



11

Creating an Executable JAR (1 of 6)

- An Executable JAR is one that can be run by double-clicking on it. That is, the JAR includes a Manifest file that tells Java which class contains `main`, and all the required resources.



12

Creating an Executable JAR (2 of 6)

2 Click Add Icon

1

3

Select:
JAR → From modules
with dependencies...

13

Creating an Executable JAR (3 of 6)

1

2

3

4

In this first lab, there are no tests, so checking does nothing.

14

Creating an Executable JAR (4 of 6)

1 Add your name to the .jar file name.

2 Create .jar in project root.

"compile output" include all class files in correct package structure.

3

4 The MANIFEST.MF file tells Java where to find `main` and the classpath.

15

Creating an Executable JAR (5 of 6)

1 Click the Add icon

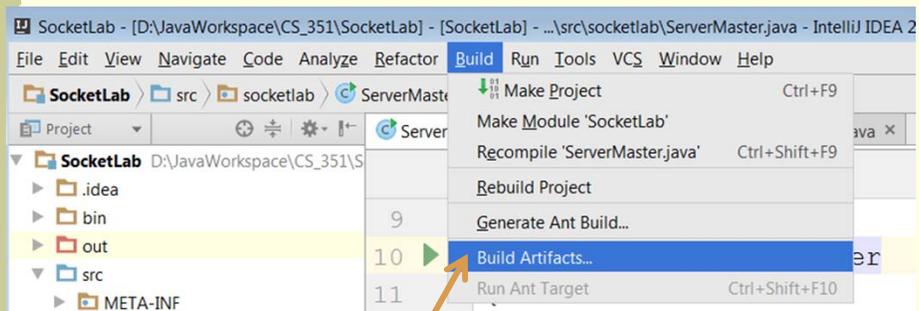
2 Select "Directory Content" and browse for the `src` directory.

This will add all the .java files along with their correct package structure.

3

16

Creating an Executable JAR (6 of 6)



- 1 The previous steps configure the JAR. After being configured, the JAR can be built. The JAR must be rebuild after code changes and compiles in order to create a JAR that includes those changes .