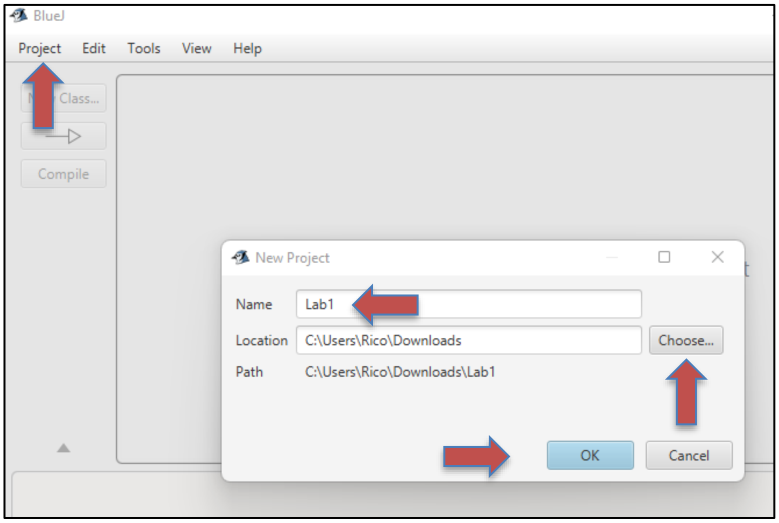
Lab 1 (5 points)
The purpose of this lab is to introduce students to the BlueJ Integrated Development Environment (IDE) or online Replit IDE environment. There are several learning objectives to this assignment. Students using Windows PCs should use BlueJ. Those using Chromebooks or struggling to get Macs set up with BlueJ can use Replit. However, it is advised to use BlueJ when possible.

- Set up BlueJ IDE or create a Replit account to become familiar with either or both Replit and BlueJ environments.
- Students are HIGHLY SUGGESTED to create GitHub accounts to store their projects. See https://github.com/ This will allow you to get practice on an industry tool that all the pros use. After creating a GitHub account, you can link your Replit and GitHub accounts from Replit by going to Account>Connected Services, and then connecting Replit with GitHub.

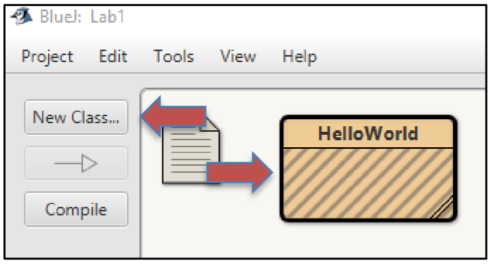
BlueJ Instructions (see below/click here for Replit instructions)

1) Download Java Development Kit (JDK), Standard Edition at the following link per your Operating System. NOTE - Please use Java17 Long Term Support (LTS). Each 6 months, Oracle comes out with new features (ie 18, 19,...). Java 17 will be supported through Sept 2024 at which time a new LTS (ie Java 23) will be supported through Sept 2027.
2) Download BlueJ at the following link depending on your OS
3) Select New Project, enter a name, select Choose to select a storage Location, and select OK.

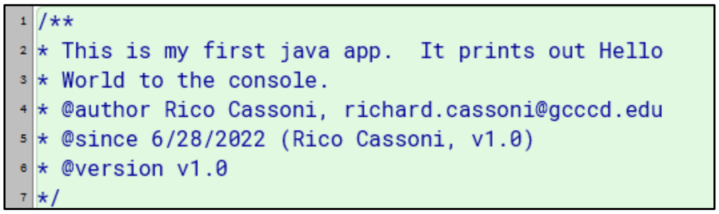


Name Lab1
Location C:\Users\Rico\Downloads
Path C:\Users\Rico\Downloads\Lab1

4) Select the New Class button and type in **HelloWorld** for the Class Name. Then select and open the HelloWord class.

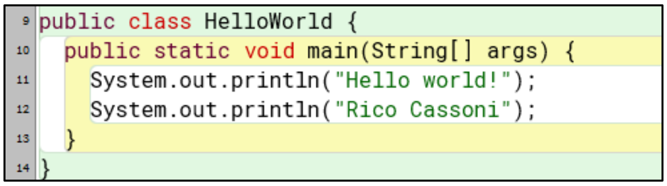


5) Update the javadocs at the top, lines 2-7 (select Options>Preferences>Display Line Numbers), with your information per the below example.



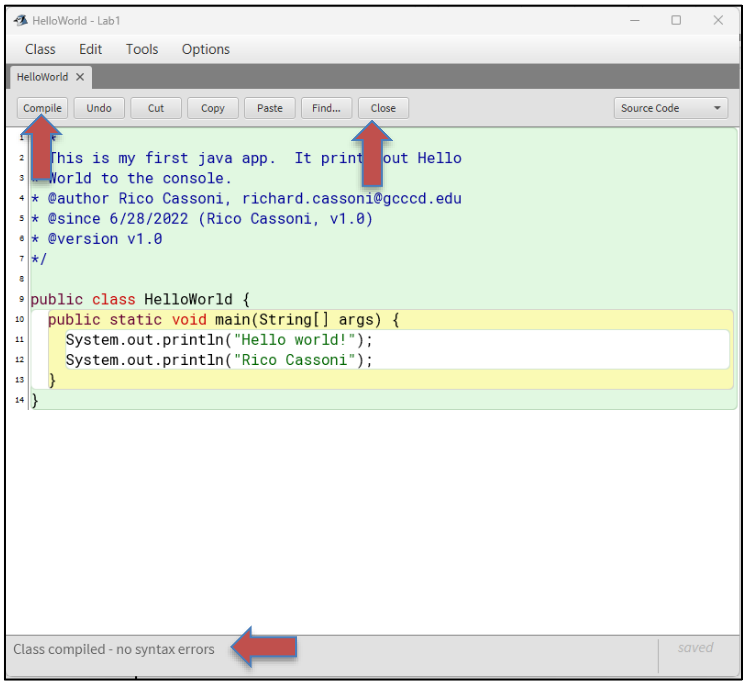
```
1 /**
2 * This is my first java app.  It prints out Hello
3 * World to the console.
4 * @author Rico Cassoni, richard.cassoni@gcccd.edu
5 * @since 6/28/2022 (Rico Cassoni, v1.0)
6 * @version v1.0
7 */
```

6) Delete lines 12 to 34 and type in the following. You can use Edit>Auto-layout to line up your code so it looks nice and uniform. This can also catch mismatched blocks { } or methods ( ).



```
 9 public class HelloWorld {
10   public static void main(String[] args) {
11     System.out.println("Hello world!");
12     System.out.println("Rico Cassoni");
13   }
14 }
```

7) Select the Compile button. If all goes well, you will see `Class Compiled-no syntax errors` at the bottom of the screen. Next, select Close.

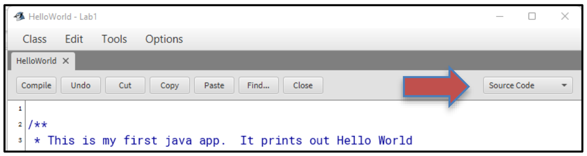


```
 1 /**
 2 * This is my first java app.  It prints out Hello
 3 * World to the console.
 4 * @author Rico Cassoni, richard.cassoni@gcccd.edu
 5 * @since 6/28/2022 (Rico Cassoni, v1.0)
 6 * @version v1.0
 7 */
 8
 9 public class HelloWorld {
10   public static void main(String[] args) {
11     System.out.println("Hello world!");
12     System.out.println("Rico Cassoni");
13   }
14 }
```

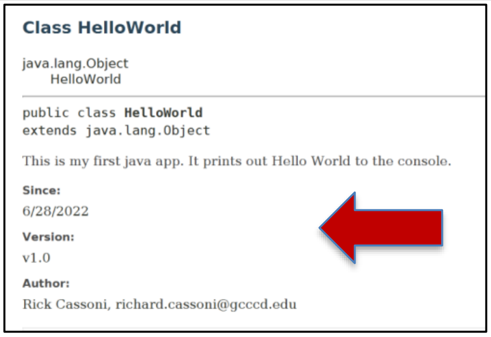
Class compiled - no syntax errors

8) On the BlueJ main screen, right click HelloWorld and select `void main (String[] args)` and then select OK. You should see two lines, Hello World and Your Name (ie Rico Cassoni). Select Options>Save to File with the name of `Lab1Output.txt`.

9) On the main BlueJ screen, double click HelloWorld rectangle to open up the source code. In the upper right, toggle Source Code to Documentation. This generates and displays javadocs in BlueJ. Take a screen shot of the Javadoc page and save. This will be one of three items needed to turn in.



```
/**
 * This is my first java app.  It prints out Hello World
```



Class HelloWorld

java.lang.Object
    HelloWorld

```
public class HelloWorld
extends java.lang.Object
```

This is my first java app. It prints out Hello World to the console.

Since:
6/28/2022

Version:
v1.0

Author:
Rick Cassoni, richard.cassoni@gcccd.edu

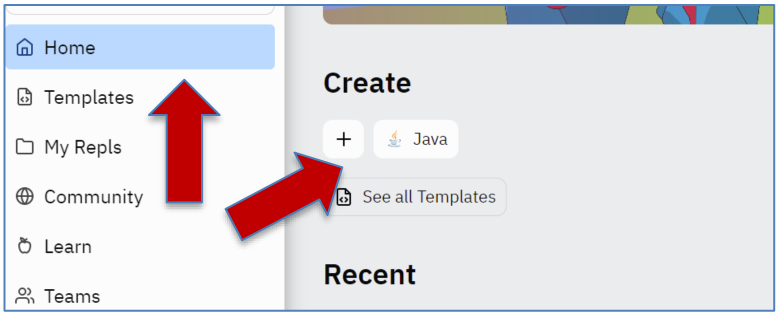
Lab1 BlueJ Submission - You will need to submit the following for Lab1

1) HelloWorld.java file – For BlueJ you will need to go the location where you saved the project (3pts)
2) Lab1Output.txt text file (or a screen cap of the console) per Step 8 above
3) Screen cap of your Javadocs from BlueJ (1pt)
   Here are the best ways to capture screen caps depending on your platform (1pt):
   a. Window-Snipping Tool
   b. Mac - https://support.apple.com/en-us/HT201361

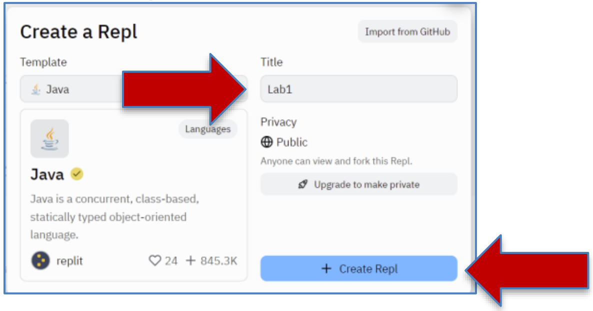
Replit Instructions

Replit provides students with a free of charge, online development environment for creating Java and other programming language source code. This is convenient in the fact that students can consistently create java applications on Windows, Mac, Linux, and Chrome OS environments that includes Chromebooks. In the case of Java, Replit will compile the source code into byte code that can be interpreted in Replit or downloaded to any device that has a java virtual machine (jvm) called a Java Runtime Environment (jre).

1) Go to replit.org. For any students that already have an account you can log in. For most, you will need to select the top right Sign Up button to create an account. This will require email verification.
2) To create a program, select the Home button from the left hand navigation section. Then in the create section, select the + icon.



3) Type java in the templates or look in Favorites to select Java. In the Title section, enter something like Lab1, Lab2, etc. For those wishing to maintain a free of charge account, you will only be able to save as Public. Then select the + Create Repl button



Template: Java

Title: Lab1

Privacy: Public

Anyone can view and fork this Repl.

Java

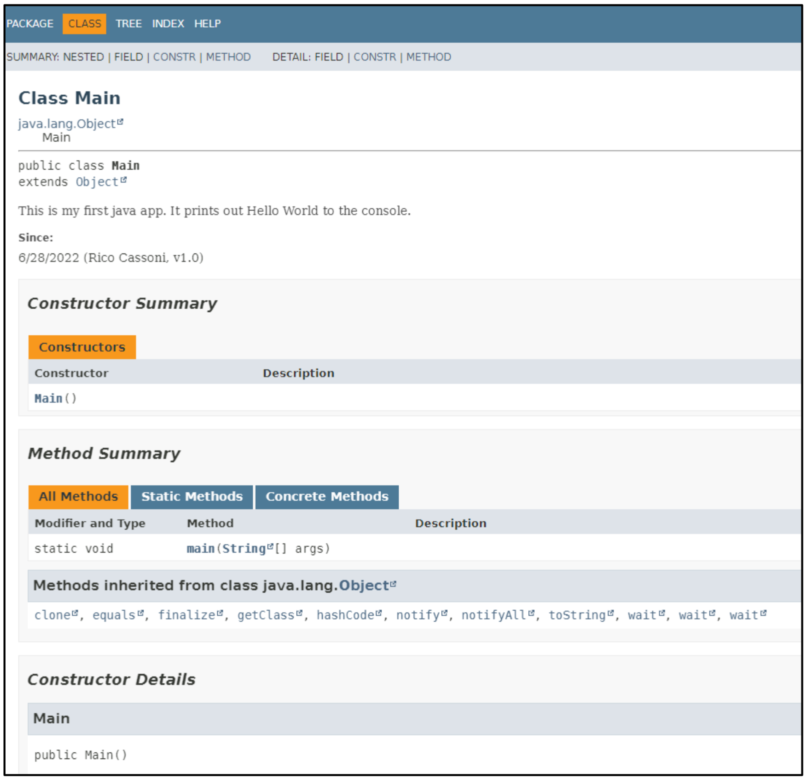
Java is a concurrent, class-based, statically typed object-oriented language.

4) While the book covers javadocs a little bit later, I always like to get students started with javadocs right away. Javadocs are created by starting with /**, you should then type in a short description of what your program will do. At the end of each line, hit return and then start the next line with an *. You should also use the @author tag to add your name and optionally your email address, the @since tag to note when you created or updated, and then the @version tag to list the versions. The convention for versions is to have v1 for the first **major** version, v1.1 for the first **minor** version of a major version and v1.2 for the second minor version and so on. This same convention is used for mobile OS versioning as well. For example, at the time of writing this lab, the version of iOS is 15.5. There can also be further minor versions such as 15.5.1 for example. *Unfortunately, replit only recognizes the @since tag, but you should still add the @author and @version tags with applicable information and can add (Name, vX.x) on the @since line.* For example in Lab 1, you would have something like below before the class declaration, but you will need to update the name, optionally add/update the email address, @since with your information, and verify/update the version info:

```
/**
* This is my first java app.  It prints out Hello World
* to the console.
* @author Rico Cassoni, richard.cassoni@gcccd.edu
* @version v1.0
* @since 6/28/2022 (Rico Cassoni, v1.0)
*/
```

NOTE - Once we get to methods in Ch5, we will add additional Javadoc descriptions and tags as applicable for each method. More in Chapter 5.

5) For Replit, the class that is the application, the one that has `public static void main(String[] args)` will need to be `public class Main`, since Replit must have Main.java be the file name. **NOTE - The file name and public class name must match. However, in java the class name should never be Main. However, this is something required for Replit but is atypical.**
6) Once all your code is written, you will need to go to the upper right-hand side (Console). Then type `javadoc Main.java` to create javadocs. Once you do this, you will see the number of files significantly increase. On the upper left-hand side, where it says Files, select the 3 vertical dots (ellipse) and then select Download as zip. Once you download, right click the zip folder (ie Lab1.zip) and extract. In the extracted folder, you can select the index.html file to launch the web page that java auto generated. It should look something like below:


PACKAGE
CLASS
TREE
INDEX
HELP
SUMMARY: NESTED | FIELD | CONSTR | METHOD
DETAIL: FIELD | CONSTR | METHOD

Class Main

java.lang.Object
    Main

```
public class Main
extends Object
```

This is my first java app. It prints out Hello World to the console.

Since:
6/28/2022 (Rico Cassoni, v1.0)

Constructor Summary

Constructor	Description
`Main()`	

Method Summary

Modifier and Type	Method	Description
`static void`	`main(String[] args)`	

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Details

Main

`public Main()`

7) After creating the Lab1 project per steps 2 and 3 above, students will need to:
    a. copy and paste the javadoc comments per step 4 above the code that Replit auto creates and then update the info in the @author, @since, and @version tags.
    b. update `class Main {` to `public class Main {`

c. Copy line `System.out.println("Hello world!");` and paste below that line and update `“Hello world!”` to your name (ie `“Rico Cassoni”`). What do you think will happen in the Console when run?

d. Select Console upper right and then select the Run button.

8) Copy and paste the output below java -classpath .:target/dependency/* Main

```
Hello world!
Rico Cassoni
```

into a text editor and save as Lab1Output.txt

<u>Lab1 Replit Submission - You will need to submit the following for Lab1</u>

1) Main.java file – For Replit this will be in the zip folder, remove from the zip folder (3pts)
2) Lab1Output.txt text file (or a screen cap of the console) per Step 8 above
3) Screen cap of your Javadocs from the web browser per (index.html) (1pt)
   Here are the best ways to capture screen caps depending on your platform (1pt):
   a. Chromebook - Shift+Ctrl+Show keys
   b. Window-Snipping Tool
   c. Mac - https://support.apple.com/en-us/HT201361