ITSC 1213 Introduction to Computer Science 1 - Module 0 Lab

Lab Instructions

Each student must complete their own lab assignment. However, you are free to discuss with other students.

IMPORTANT! I cannot overemphasize the importance of following these instructions EXACTLY! 99% of the time, if you have a problem with the lab, it will be because you skipped a sentence, didn't copy something precisely, or skimmed the instructions. Read every sentence completely and follow the instructions to the letter. If you find an error, please let me know.

OVERVIEW

1) Preliminaries: Setup Environment (1point)
2) Part A: First NetBeans Project (1 point)
3) Part B: Adding Classes (1 point)
4) Part C: Exporting a project using NetBeans (1 point)

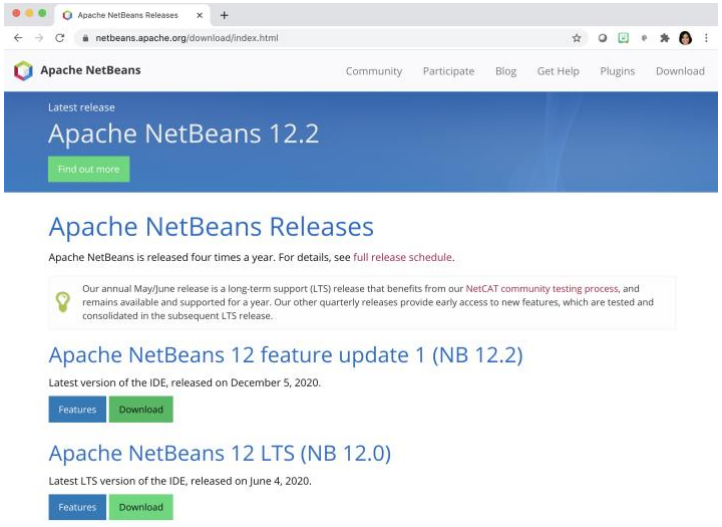
Preliminaries: Installing NetBeans IDE

Objective: Setup development environment with NetBeans

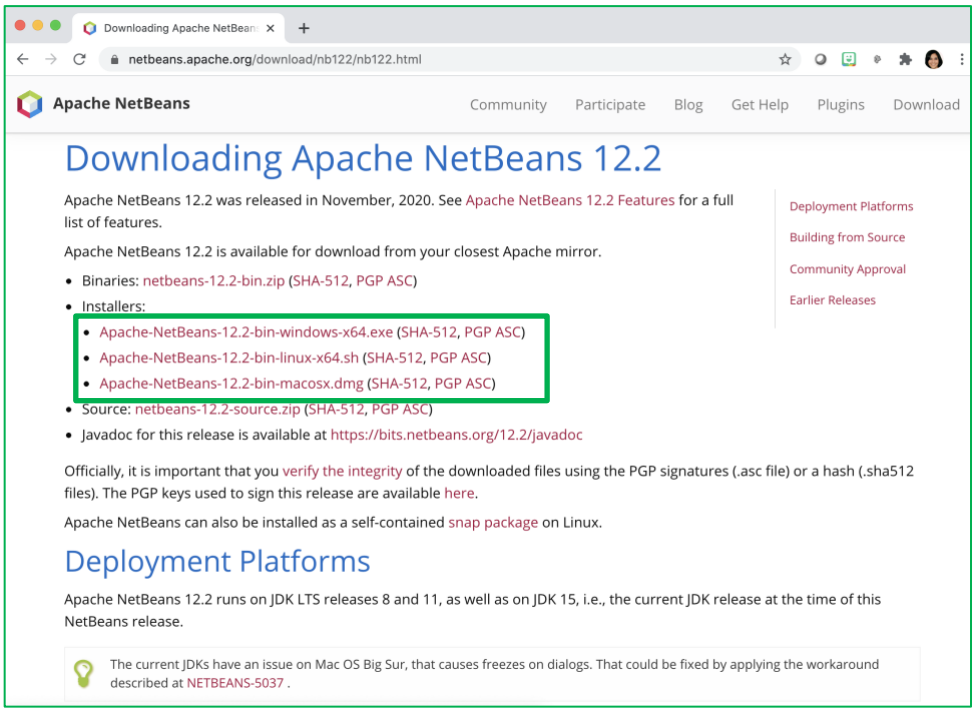
We are assuming that you already have downloaded and used a Java JDK on the device that you plan to use for this course. If you do not have a JDK installed on your device please visit https://www.oracle.com/technetwork/java/javase/downloads/, and download the desired Java version. For this course you can use any version that is 8 or higher. Please consult the instructional team if you have any doubts about this.

In this course we will be using Apache NetBeans to develop Java applications. NetBeans is a tool that aids in the development process. There are many other tools that provide this type of support and can be used. We choose NetBeans as it is a stable and well supported tool.

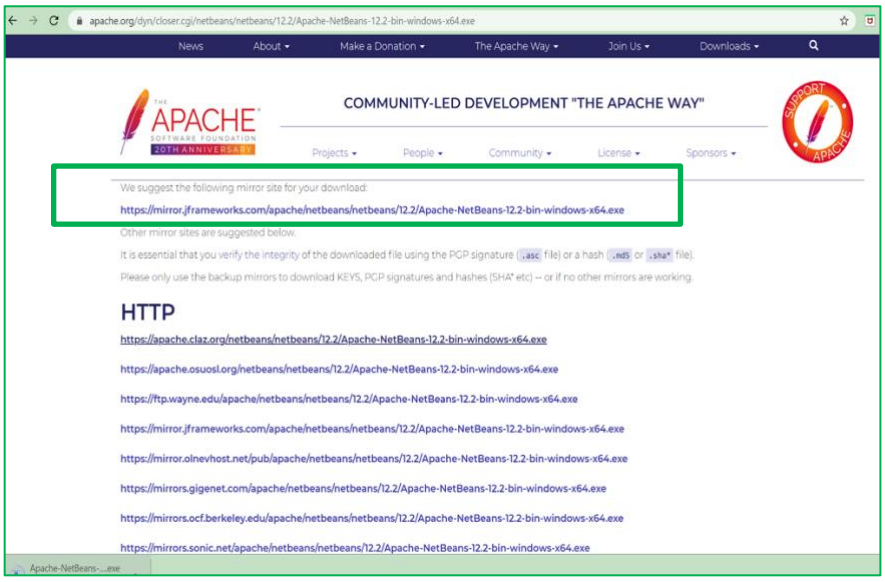
- To download NetBeans visit https://netbeans.apache.org/. At this time the latest version available for download is 12.2. Click the download button for the latest version:



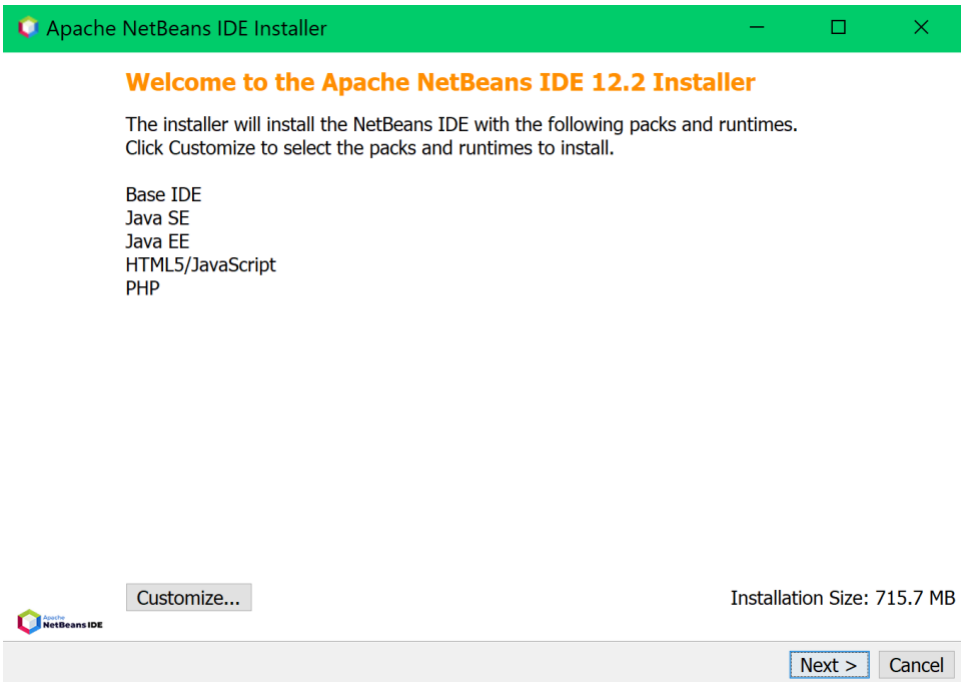
- Find the link for the installer for your device type:



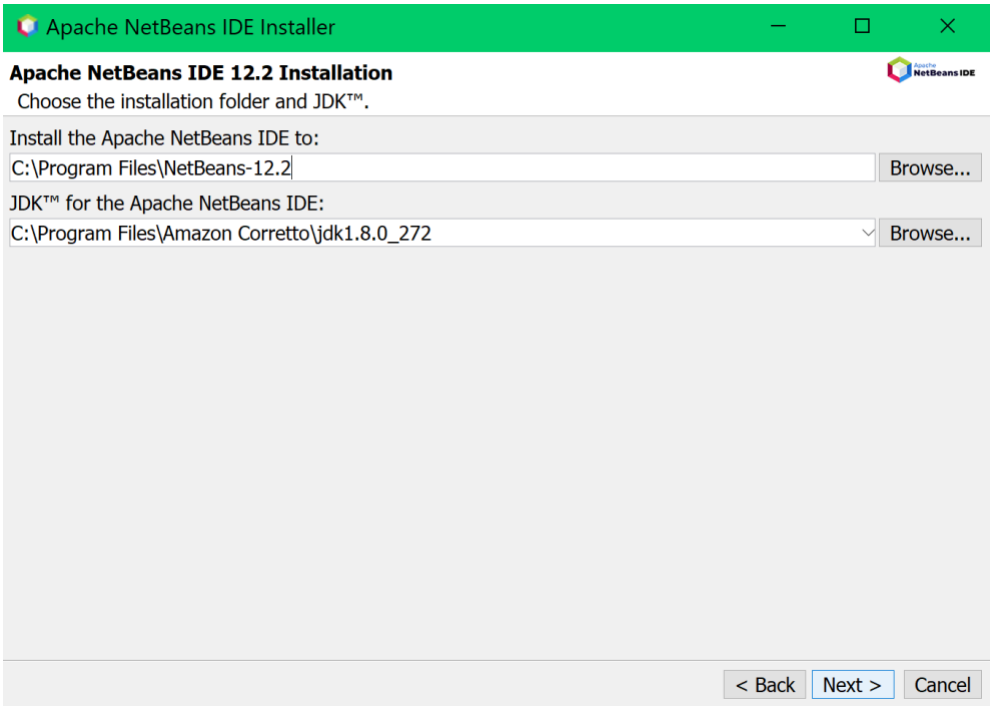
- Click the recommended mirror site to start the download

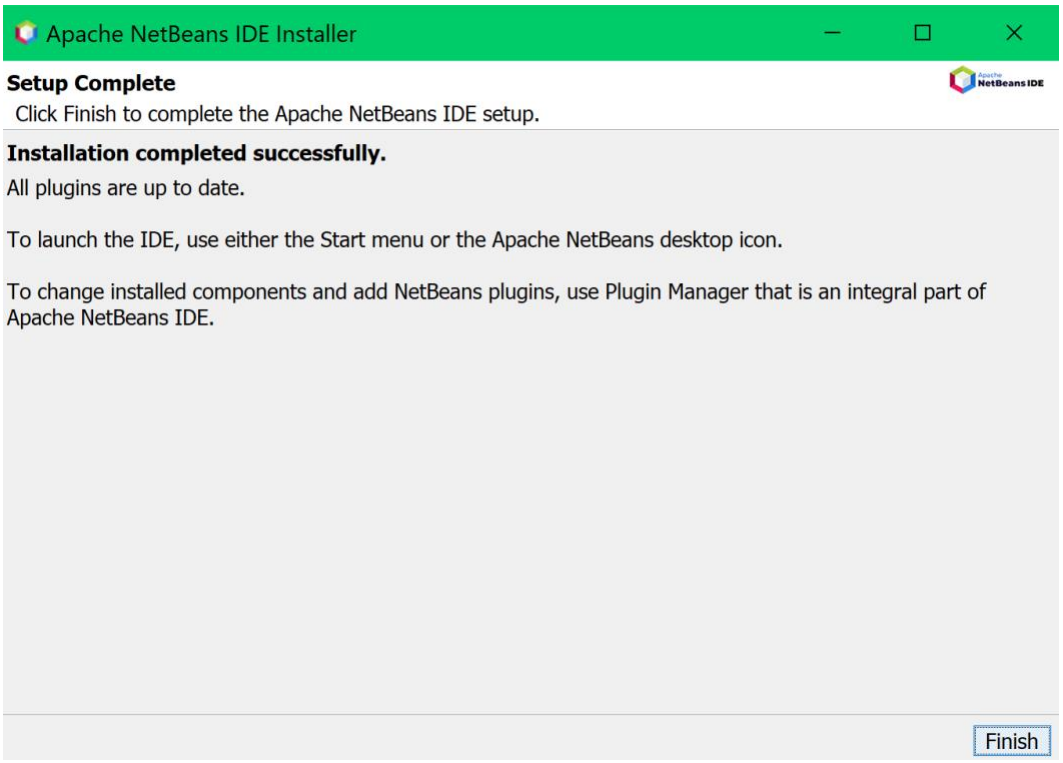


- Locate the downloaded file on your device and double click to start the installation process. Follow the installer steps to install.



- Choose where you want to store the IDE and the JDK





- NetBeans should now be installed and can be launched from your desktop:

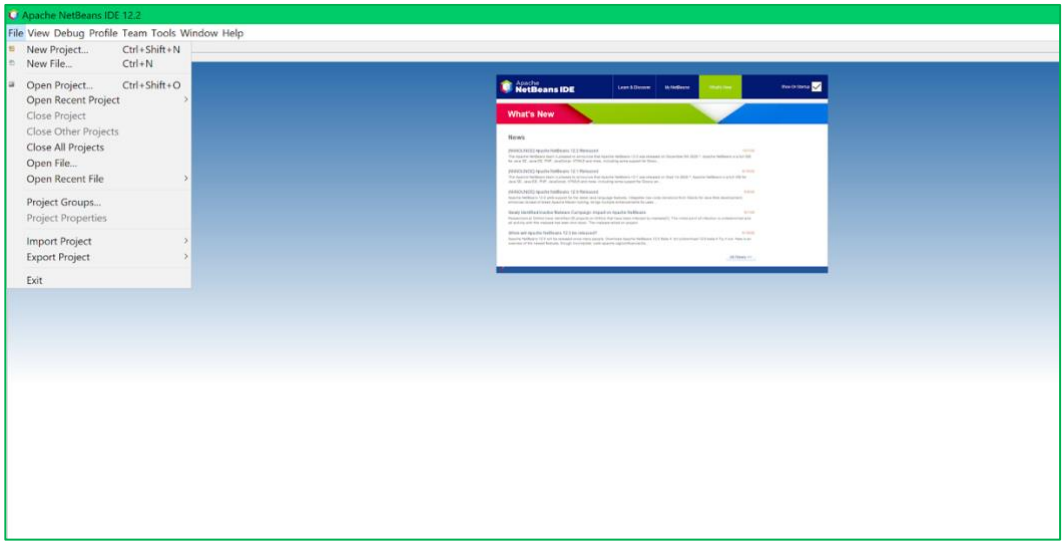
- Double click to launch the application.

Congratulations, you have successfully installed NetBeans and are ready to start using it to develop Java programs.

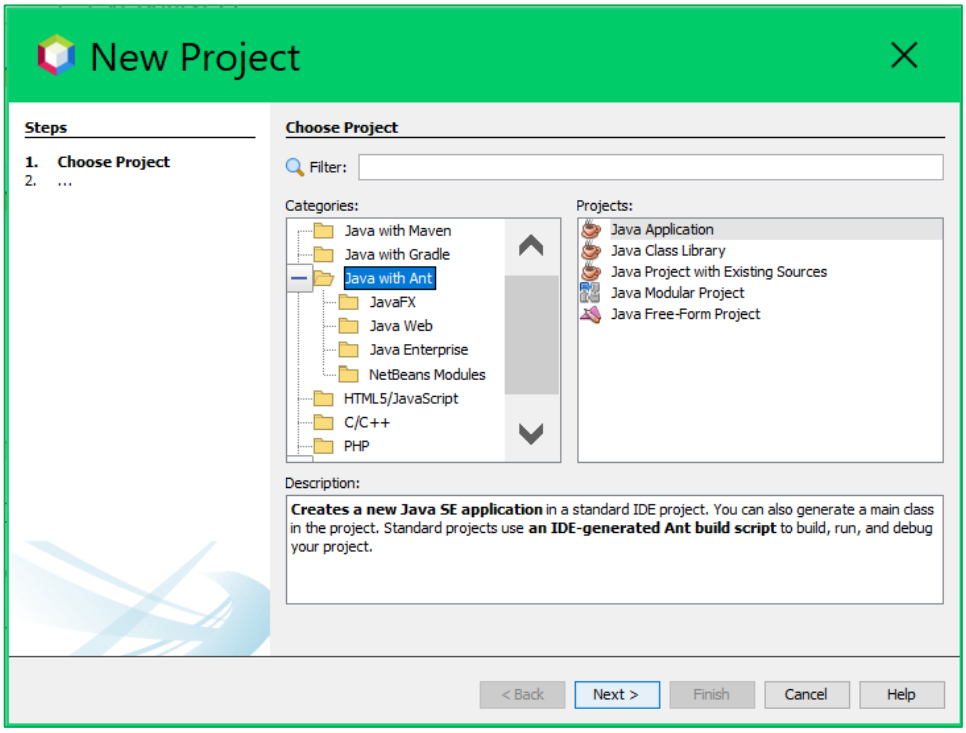
ITSC 1213 Introduction to Computer Science 1 - Module 0 Lab

Part A: First NetBeans Project
Objective: Create a new project with NetBeans

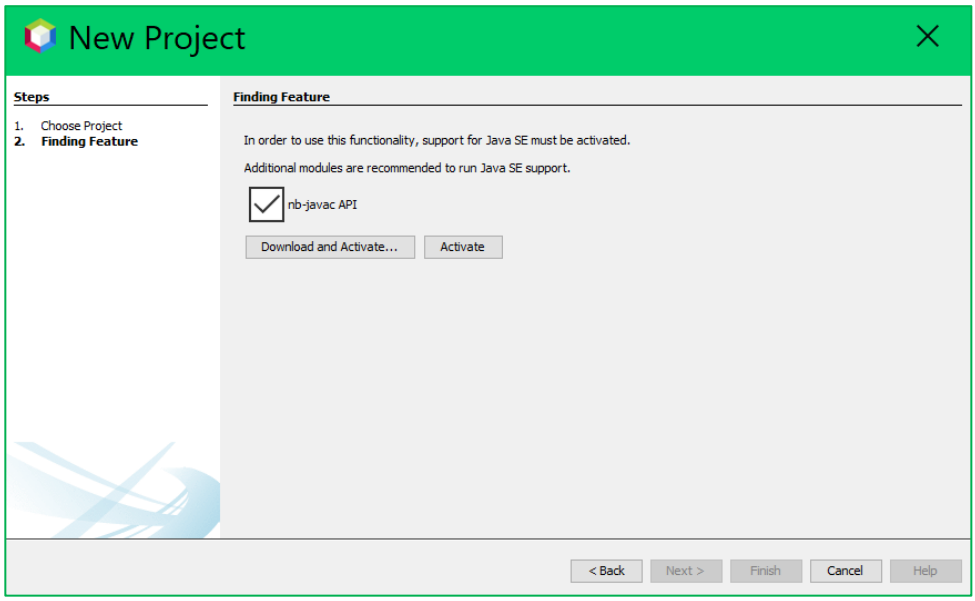
1) Open ***NetBeans*** and choose ***File --> New Project***



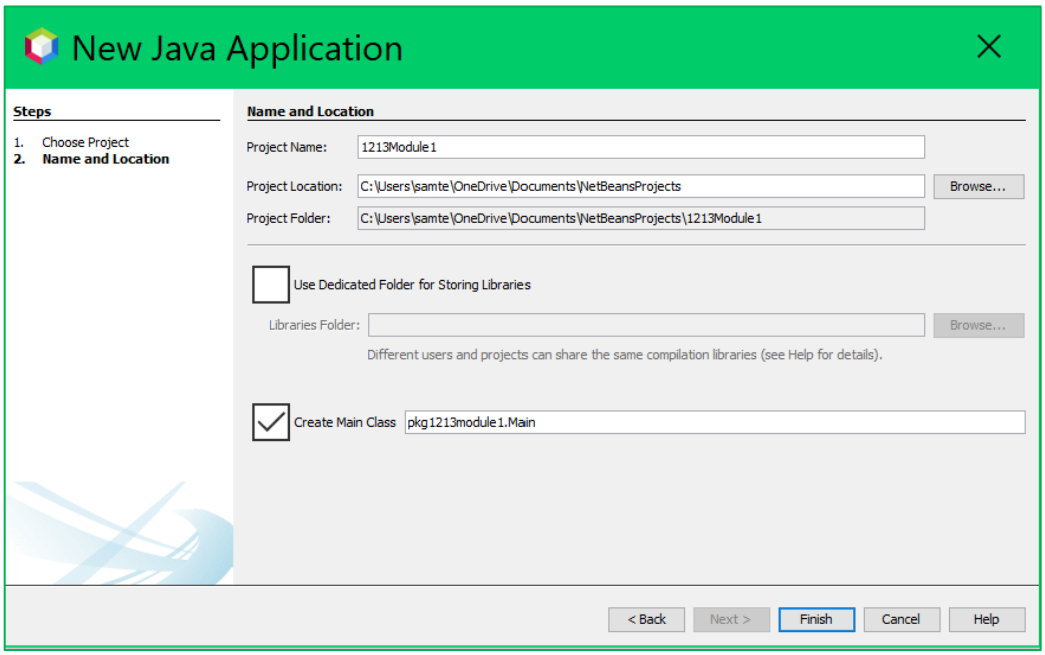
2) In this class we will always be creating a Java Application, Choose 'Java with Ant', then choose 'Java Application'. You should see a dialog box like this. Click '***Next >***' to proceed



- If you encounter this screen select “Download and Activate” and follow the steps.

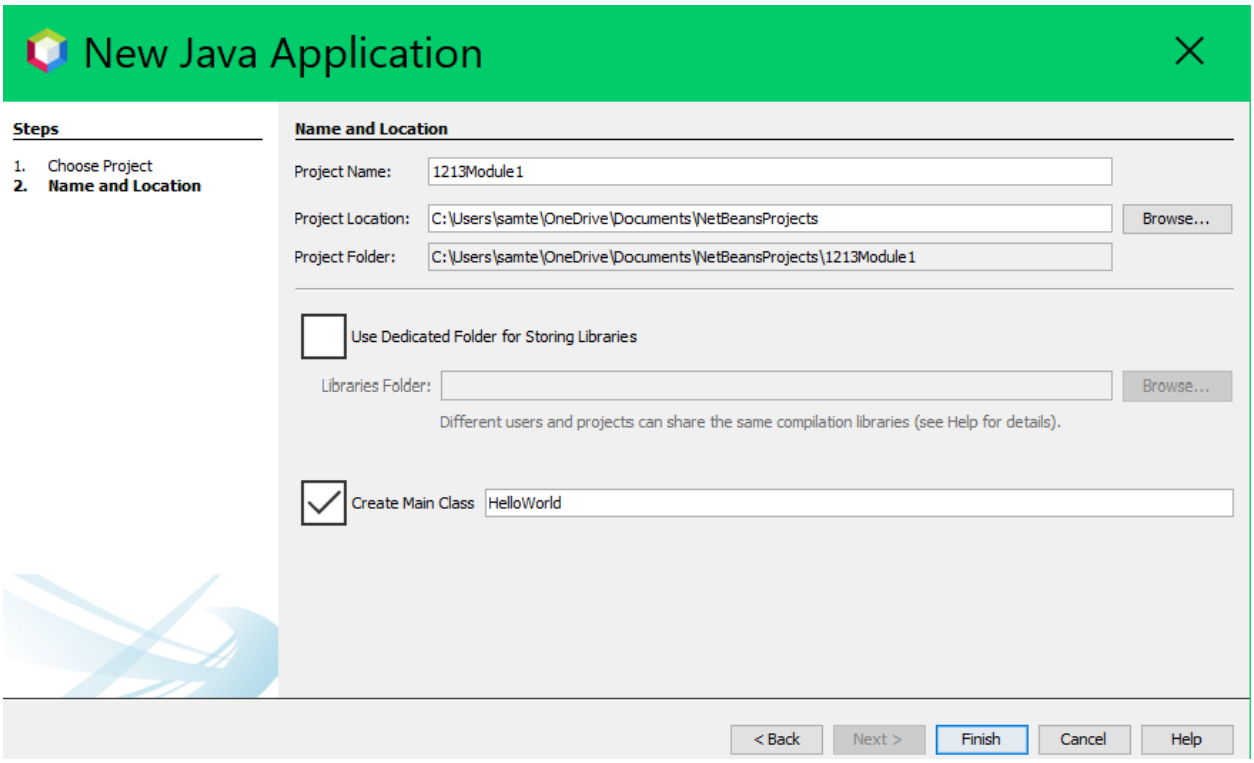


3) Once this step is complete you will see the dialog box to specify the name and location of the project (note that what appears as default values will be based on your machine):



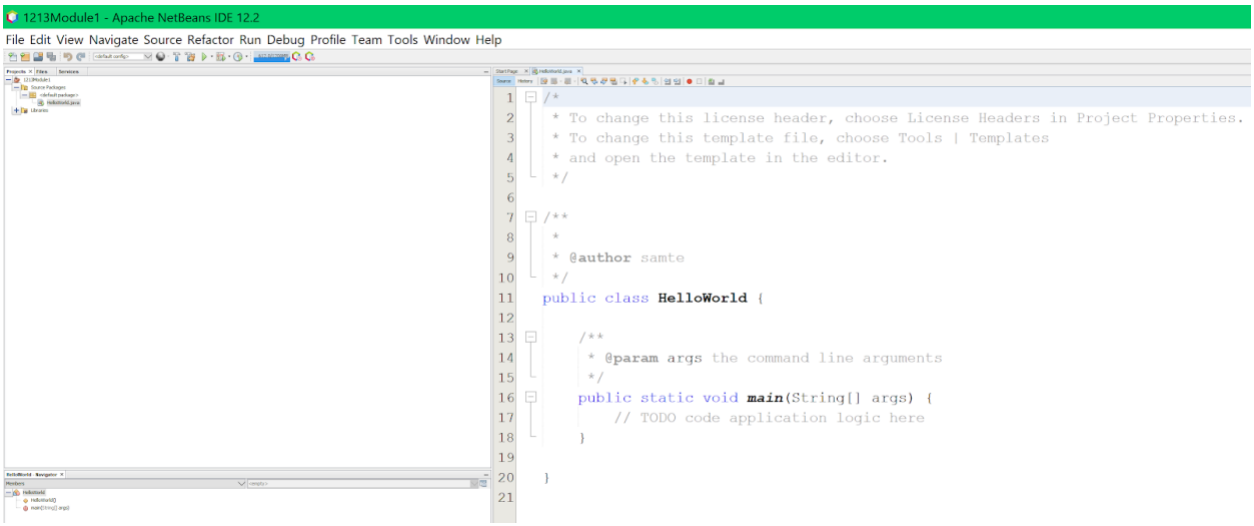
Let's call the first NetBeans Java application you create ***1213Module0***, so type that in the first box at the top, replacing the default Project Name. **NO SPACES IN THE PROJECT NAME: *1213Module0* not *1213* <space>*Module*<space>*0*.** In fact, NEVER put spaces in filenames or folder names. Use underscores or dashes but never spaces. **Never, ever, ever.** The Project Location should be a folder named ***NetBeansProjects***. You can keep this default location or change it to a different folder. The main thing here is to remember where you are saving your files and projects and keep things organized. You can browse to whatever location you want to use but for simplicity's sake I'd make sure the folder is named **NetBeansProjects** like in the example above.

NetBeans will automatically create a new folder in the location you specify and it will reflect the project name. You want to leave the '***Create Main Class***' checkbox checked, because we want to be able to run ***1213Module0*** as an executable file and we want to use a class with the name ***HelloWorld*** as the class with the main method that will get executed when we run this project. Your dialog box should now look something like this:



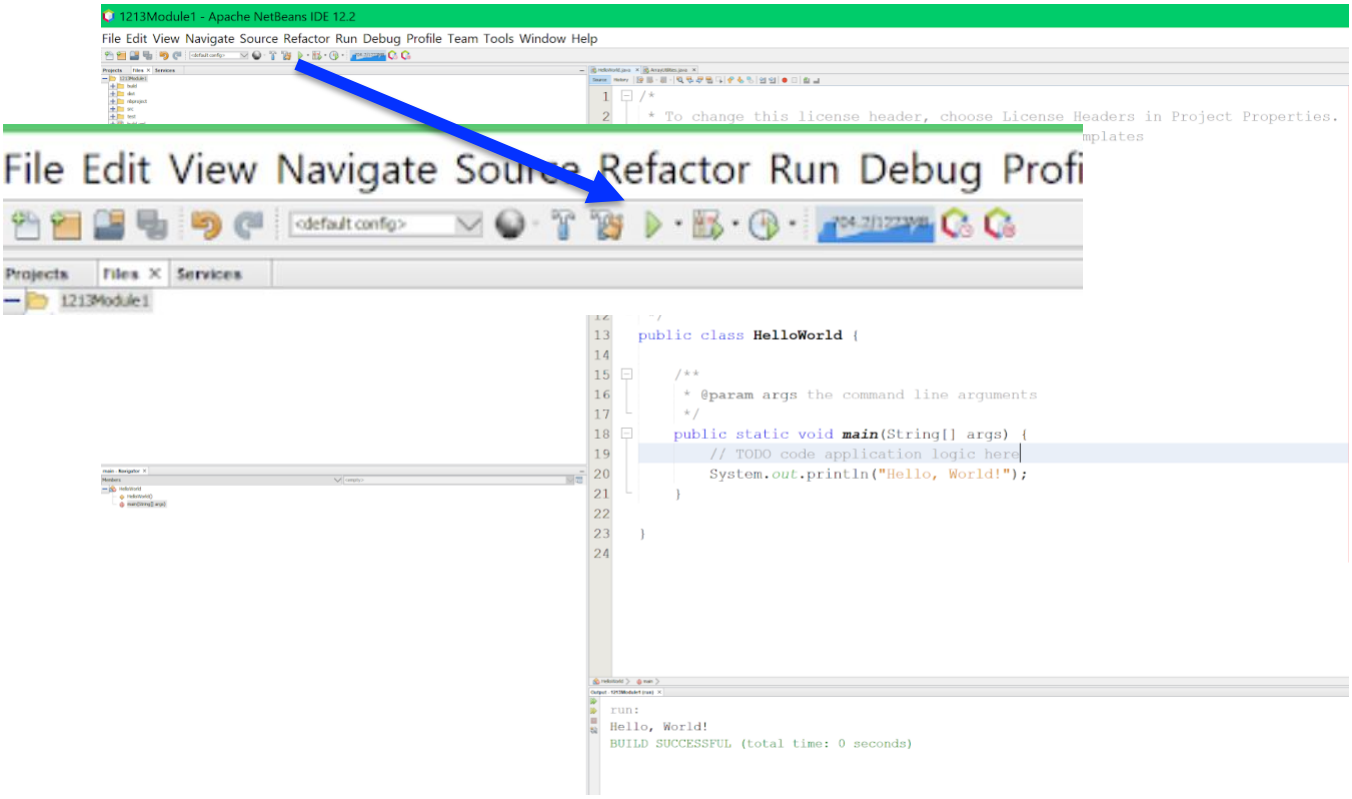
Once you are done making these updates click '***Finish***'.

4) What you should see now is the main NetBeans screen (be patient, this may take while), with the automatically generated **HelloWorld.java** file in the code editing pane, like this:



5) Add a **System.out.println("Hello World.");** statement inside the main method of the file to print out the words: "***Hello, World!***". Place your cursor at the end of the line that says **TODO code application logic here**, press ***Enter***, type **sout** and press ***Tab*** to see what happens. NetBeans automatically creates a statement for you. Look at the *NetBeans Shortcut Commands* document in the *Resources* section on your Canvas page for more examples of these types of shortcuts.

6) To test that your program runs, click the green '***Run***' button (the one shaped like a triangle or "play" button) at the top of the NetBeans window. You should see the words appear in the Output window at the bottom, like this:



It's important to know that messages from System.out.println(…) are system messages that go to the output console. They are typically useful for you the programmer, and for your programming team members, but these aren't usually seen by the users of your program. To see what they see, let's do a 'Clean and Build', which is the tool you use in NetBeans to create an executable file.

7) Click on the '***Clean and Build Project***' button at the top of the screen (the hammer and broom icon located to the left of the green run button). This will compile your code and create an executable JAR file that you can send to your friends to run. (If the Clean and Build button is grey, click anywhere in the pane with the HelloWorld source code. If you get an error about not being able to delete a directory click the ***Build Project*** (hammer) button.) When you click '***Clean and Build***' (or '***Build***'), watch the Output pane at the lower right. (If you don't see one press ***Ctrl-4***.) You should see messages about the code compiling and some information about where the executable JAR file is, like this:

```
ant -f C:\\Users\\samte\\OneDrive\\Documents\\NetBeansProjects\\1213Module1 -Dnb.internal.action.name=rebuild clean jar
init:
deps-clean:
Updating property file: C:\Users\samte\OneDrive\Documents\NetBeansProjects\1213Module1\build\built-clean.properties
Deleting directory C:\Users\samte\OneDrive\Documents\NetBeansProjects\1213Module1\build
clean:
init:
deps-jar:
Created dir: C:\Users\samte\OneDrive\Documents\NetBeansProjects\1213Module1\build
Updating property file: C:\Users\samte\OneDrive\Documents\NetBeansProjects\1213Module1\build\built-jar.properties
Created dir: C:\Users\samte\OneDrive\Documents\NetBeansProjects\1213Module1\build\classes
Created dir: C:\Users\samte\OneDrive\Documents\NetBeansProjects\1213Module1\build\empty
Created dir: C:\Users\samte\OneDrive\Documents\NetBeansProjects\1213Module1\build\generated-sources\ap-source-output
Compiling 1 source file to C:\Users\samte\OneDrive\Documents\NetBeansProjects\1213Module1\build\classes
compile:
Created dir: C:\Users\samte\OneDrive\Documents\NetBeansProjects\1213Module1\dist
Copying 1 file to C:\Users\samte\OneDrive\Documents\NetBeansProjects\1213Module1\build
Nothing to copy.
Building jar: C:\Users\samte\OneDrive\Documents\NetBeansProjects\1213Module1\dist\1213Module1.jar
To run this application from the command line without Ant, try:
java -jar "C:\Users\samte\OneDrive\Documents\NetBeansProjects\1213Module1\dist\1213Module1.jar"
deploy:
jar:
BUILD SUCCESSFUL (total time: 1 second)
```

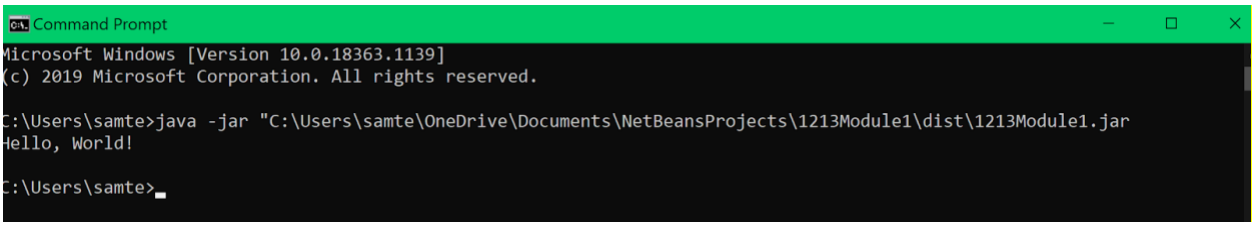
Note the green message that says: "***BUILD SUCCESSFUL***". Also note a few lines above that, it tells you where the executable ***1213Module0.jar*** file is located. The default is that it is in your project folder, in a sub-folder called '***dist***', which is short for "distribution". The contents of the ***dist*** folder are what you would send to someone so they could run your application. Regardless of the kind of computer you used to write your program, anyone who has Java installed should be able to run your program if you send them the contents of your '***dist***' folder. That's part of the power of Java.

8) Navigate to the ***dist*** folder in your file system and double-click on the ***1213Module0.jar*** file. You should see the system seeming to start to do something, but then nothing happens. That's because this HelloWorld program doesn't generate anything that the user can see.

9) Now, open a terminal window (command prompt in Windows) and run the program from the command line. You can get a terminal on a Mac by typing "terminal" in Spotlight. On windows 10 click the windows icon in the lower left, select All Apps, scroll down to and click on Windows System, then click Command Prompt.

10) Copy the line from the bottom of the NetBeans window (the line that begins "java –jar...") and paste it into the terminal window. Hit 'Enter'. MAC users should see something like this:

```
Created dir: C:\Users\samte\OneDrive\Documents\NetBeansProjects\1213Module1\dist
Copying 1 file to C:\Users\samte\OneDrive\Documents\NetBeansProjects\1213Module1\build
Nothing to copy.
Building jar: C:\Users\samte\OneDrive\Documents\NetBeansProjects\1213Module1\dist\1213Module1.jar
To run this application from the command line without Ant, try:
java -jar "C:\Users\samte\OneDrive\Documents\NetBeansProjects\1213Module1\dist\1213Module1.jar"
deploy:
jar:
BUILD SUCCESSFUL (total time: 1 second)
```



Here, you can see the output of the program, just like in the NetBeans window. But this isn't how your users are typically going to run your programs. Usually, they are going to run your programs by double-clicking on the JAR file. So, how can we give them some output that they can see? The answer is to use a dialog box to display our message instead.

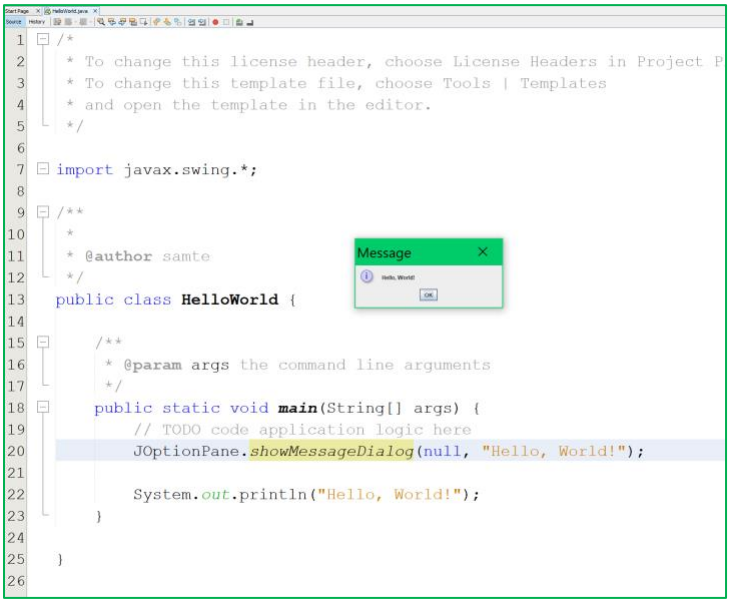
11) Go back to the NetBeans window, and add the following import statement below the package statement in the ***HelloWorld.java*** file:

    import javax.swing.*;

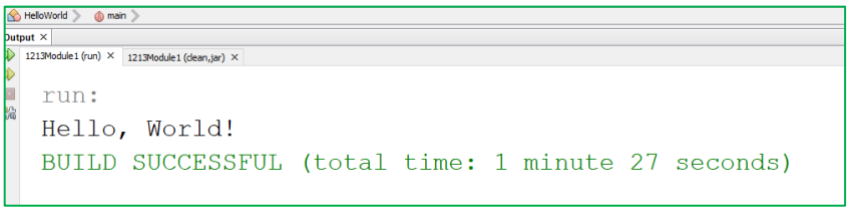
12) Now, add the following line of code above the **System.out.println(…)** statement in the main method:

    JOptionPane.showMessageDialog(null, "Hello, World!");

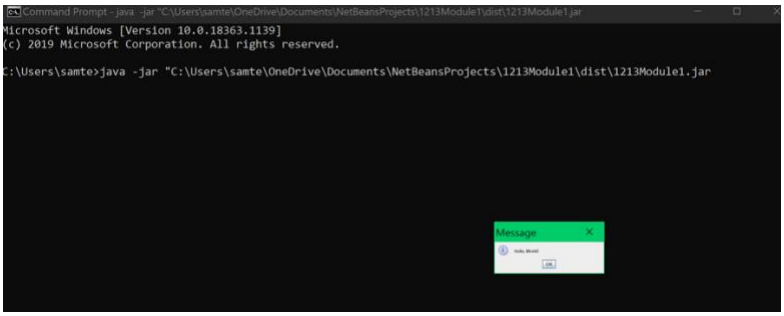
13) If you click the '**Run**' button now, you should see a dialog box with your "***Hello, World!***" message and an '***OK***' button:



14) After you click the '***OK***' button, you'll also see the "***Hello, World!*** " message appears in the NetBeans Output window, because the **System.out.println(…)** statement is still in your code.



15) Finally, let's verify that this dialog box is visible to a user of your program. Click the '***Clean and Build***' button to build a new JAR file. Find the JAR file in your file system and double-click it (or repeat the first part of step 10 above). You should see something like this (and after you click on the 'OK' button, the HelloWorld program ends and the dialog box will disappear):



So, now you should understand the difference between programmer's output (created with System.out… statements) and user interface output that the user can see when they execute a JAR file, which is generated through some kind of Java user interface code (in this case the JOptionPane dialog box). You should also now know how to build an executable JAR file and find it in your file system.

16) When your program works, take a screenshot of your terminal window and the Java dialog box and save the file as lab1_partA to include with your submission.

I know a lot of this is new and confusing. What's a **JOptionPane**, for example? That's OK, we'll get to things like that (if they're important) as the course progresses.

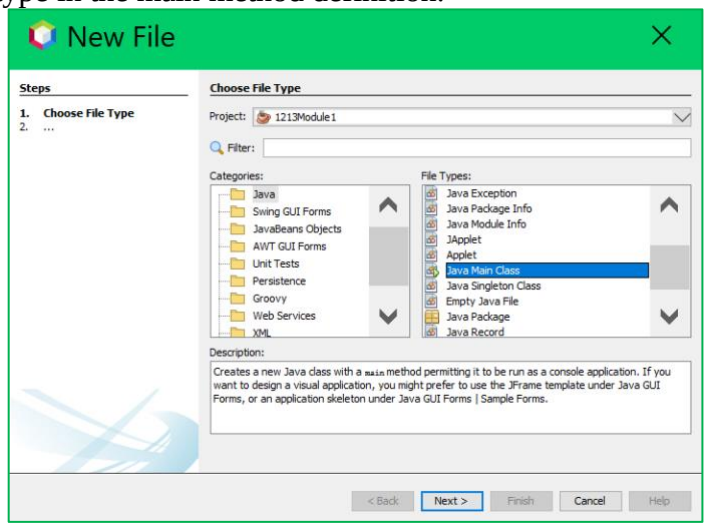
NOTE: Sometimes in Windows systems the build fails because the .jar file is still open even though the program has stopped. If this happens use Ctrl-Alt-Delete to open the Task Manager, click the 'Processes' tab, locate and select the javaw.exe process, then click 'End

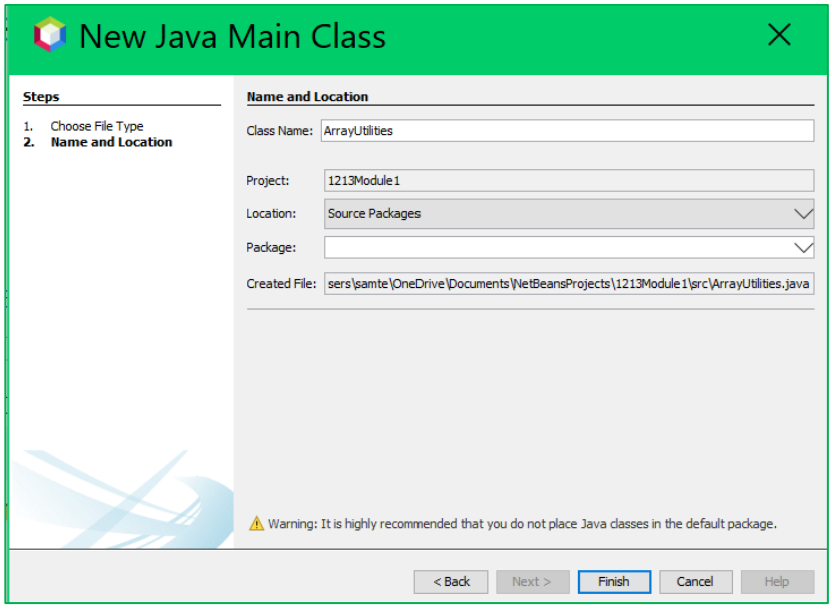
Process'. If the Task Manager asks if you want to end javaw.exe click the 'End Process' button in the dialog box. Close the Task Manager. Attempt the clean and build again.

Part B: Adding Classes and Methods to a NetBeans project

Objective: Adding a second Java source file to a NetBeans project and implementing methods to process arrays

1) Add a new Java class file called ***ArrayUtilities***. Note that NetBeans gives you the option of creating a Java class or a Java Main Class. The difference between the two is that the Java Main Class template will include a main method definition where the Java Class template will not include any methods. Either one will work. If you use the Java Class you will need to type in the main method definition.



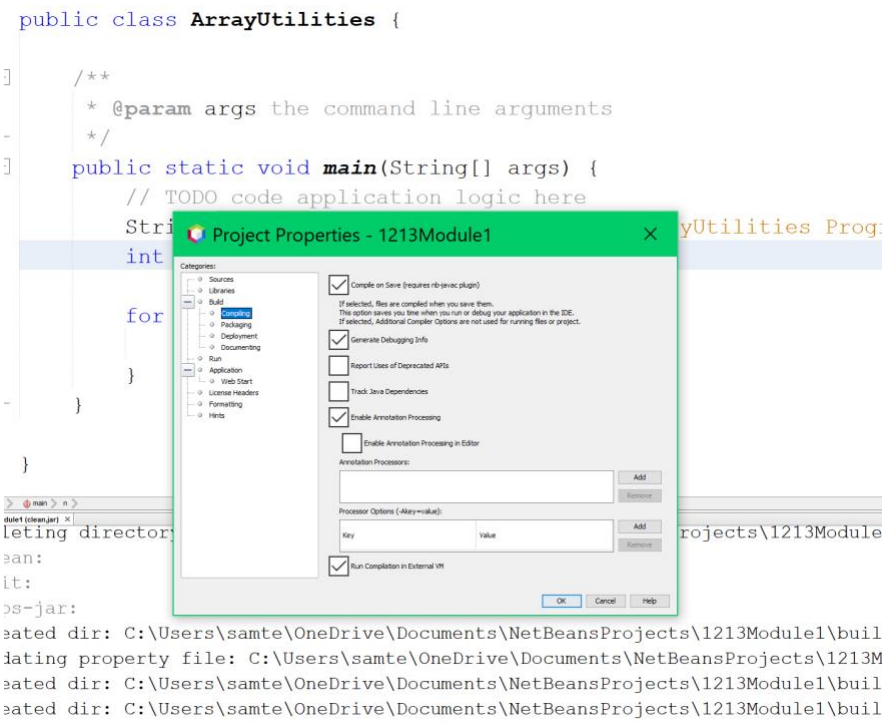


2) In your main method, write the code that will display "***Welcome to ITSC1213 ArrayUtilities Program!***" five times using a for loop.

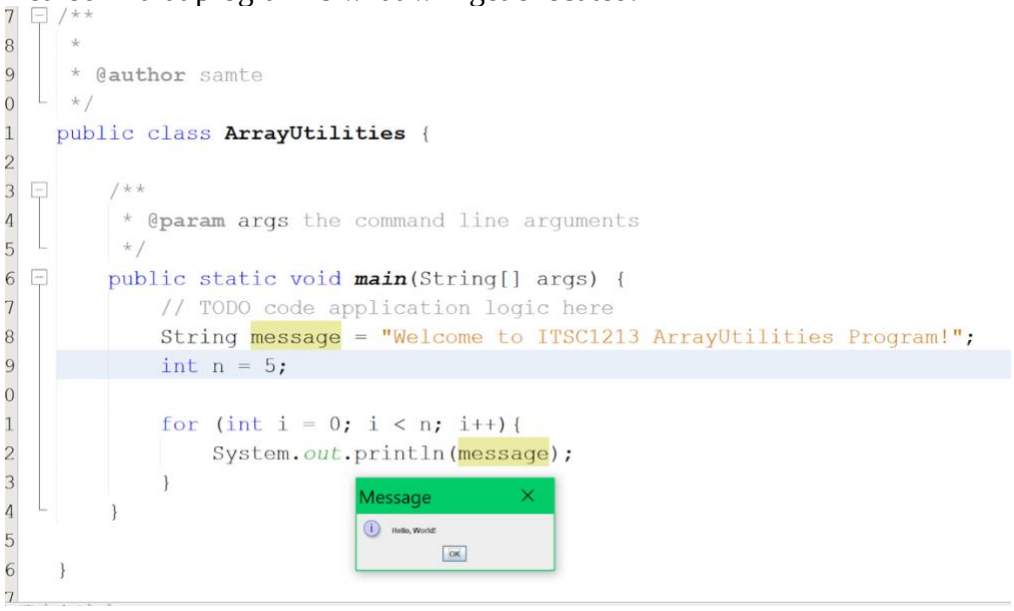
```java
/**
 *
 * @author nanajjar
 */
public class ArrayUtilities {

    public static void main(String[] args) {

        String message = "Welcome to ITSC1213 ArrayUtilities Program!";
        int n = 5;

        for (int i = 0; i < n; i++) {
            System.out.println(message);
        }
    }

}
```

There are many ways to accomplish this; you are free to implement it as you see fit. For example, you might not want to use variables for the message or the number of times we want the message to be outputted.

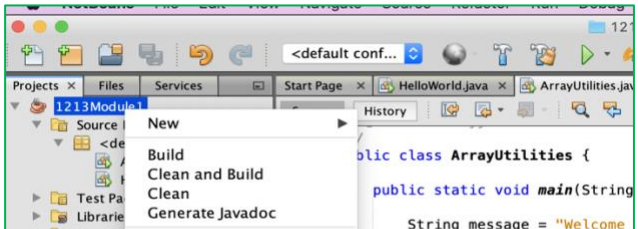
Once you are done updating the main method you should test and run your program. The first step is to make sure that the newly created Java class is compiled. By default, projects in NetBeans are created with the Compile on Save enabled. So, you do not need to compile your code first in order to run the application. Clicking the "Clean and Build Project" button will compile all valid classes in your project and create a new JAR file.

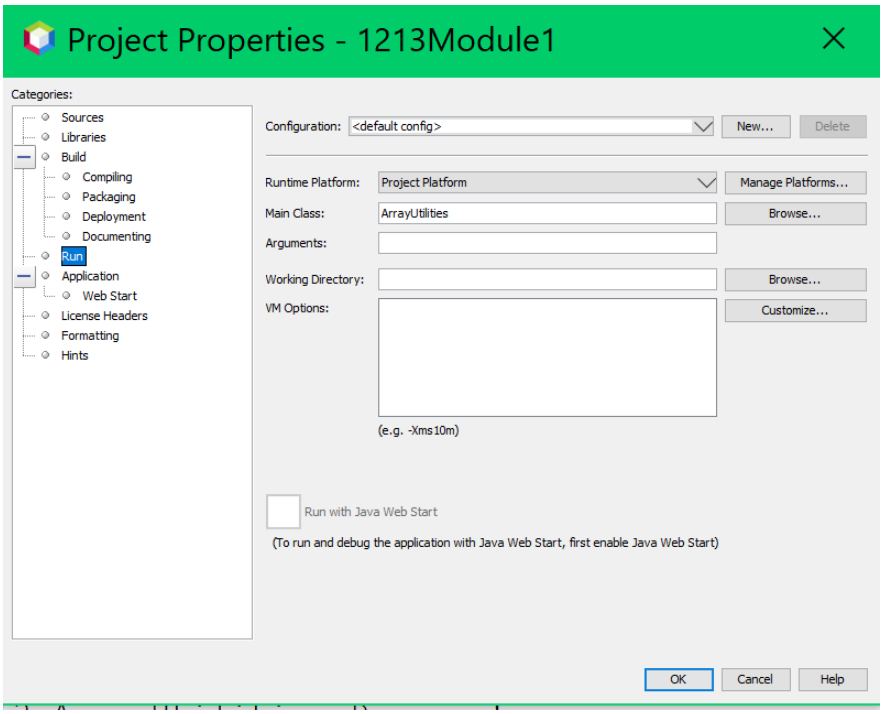


If you click the Run button you will notice that the dialog box from the HelloWorld program is what was actually executed not the ***ArrayUtilities*** file. If you remember in Part B when we created the project, we set the ***HelloWorld*** class as the main class for this project which means the main method in that program is what will get executed.

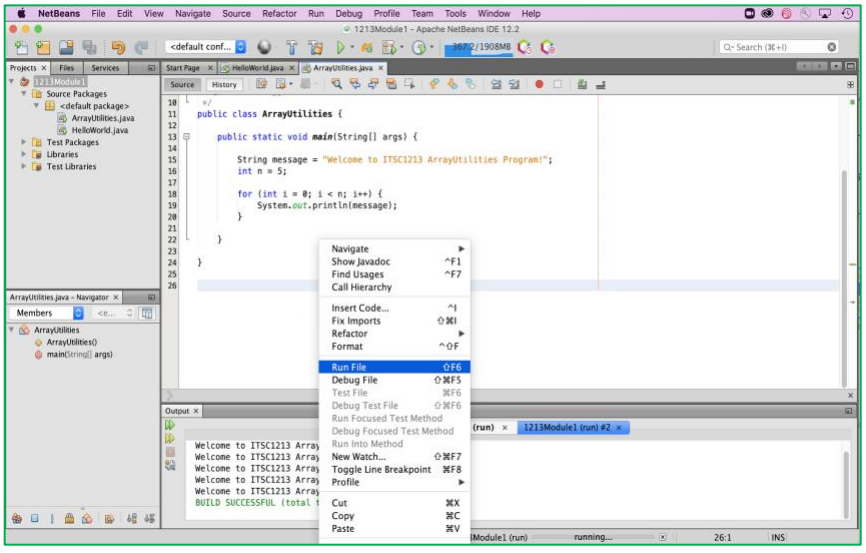


We can change this in the project properties if we want to make this file the main file for this project





Or you can simply right click in the edit pane for the file you want to run and click 'Run File'



Run your file and verify the printed messages in the output window. If you do not see the desired output click the Clean and Build Project and then try running the file again.

3) Now we are going to add more methods to our ***ArrayUtilities*** class. Create a new method: ***public static int sum(int[] numbers)***

   This method should return the sum of all the elements in the parameter array. Inside this method add the logic to perform this task.

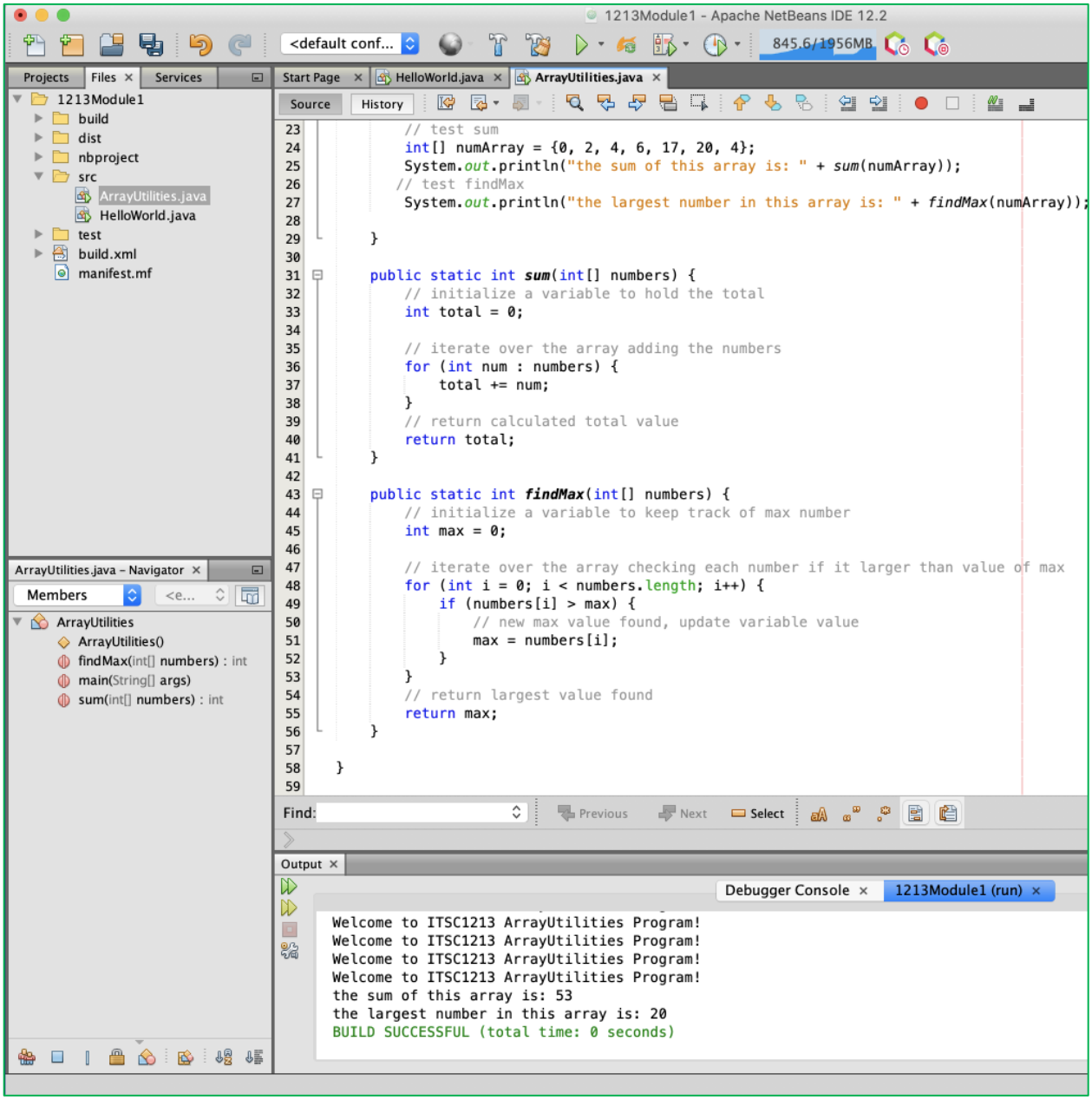
```java
/**
 *
 * @author samte
 */
public class ArrayUtilities {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        String message = "Welcome to ITSC1213 ArrayUtilities Program!";
        int n = 5;

        for (int i = 0; i < n; i++){
            System.out.println(message);
        }
    }

}
```

```
run:
Welcome to ITSC1213 ArrayUtilities Program!
Welcome to ITSC1213 ArrayUtilities Program!
Welcome to ITSC1213 ArrayUtilities Program!
Welcome to ITSC1213 ArrayUtilities Program!
Welcome to ITSC1213 ArrayUtilities Program!
BUILD SUCCESSFUL (total time: 0 seconds)
```

4) Now test this method in your main method by creating an array of integers, calling **sum** with the array you created, and print out the result of **sum**.

5) Now we want to create another method called ***findMax***. This method should return the largest number in the array of **ints**. Inside this method add the logic to perform this task.

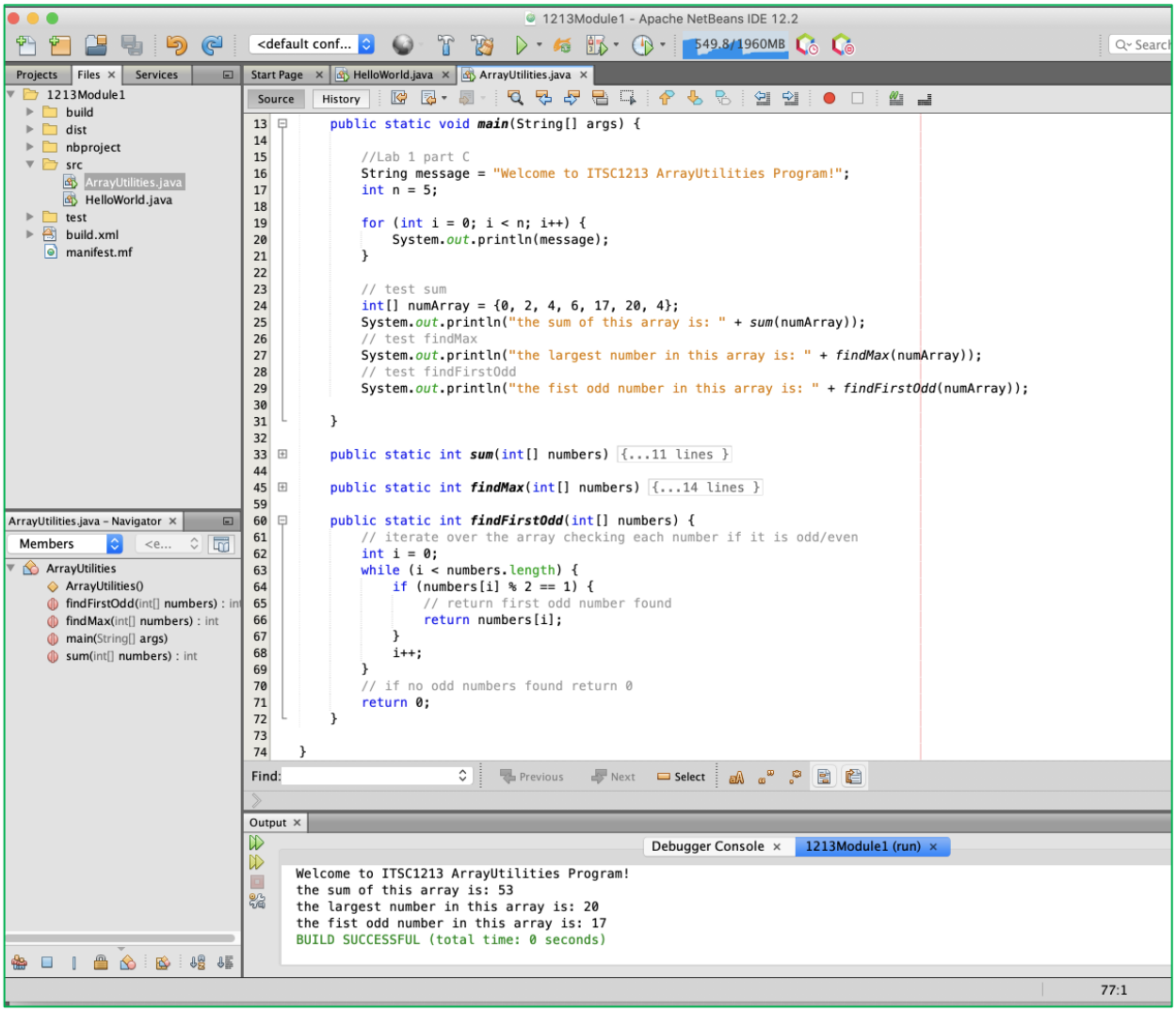
This method signature would look like this: ***public static int findMax(int[] numbers)***



6) Test this new method in your main method by creating an array of integers, calling ***findMax***, and printing the result. Make sure this method will work with **any** array of integers, even ones with all negative numbers.

7) Now we want to add our last method ***findFirstOdd***. Create a new method: ***public static int findFirstOdd(int[] numbers).***
This method should return the first odd number in an array, or return 0 if no odd numbers were found. Use a loop to find the first odd number in an array. When you find one, stop

looping and return the odd number. Test this by passing the method with different integer arrays. Remember to test with an array that contains no odd numbers.



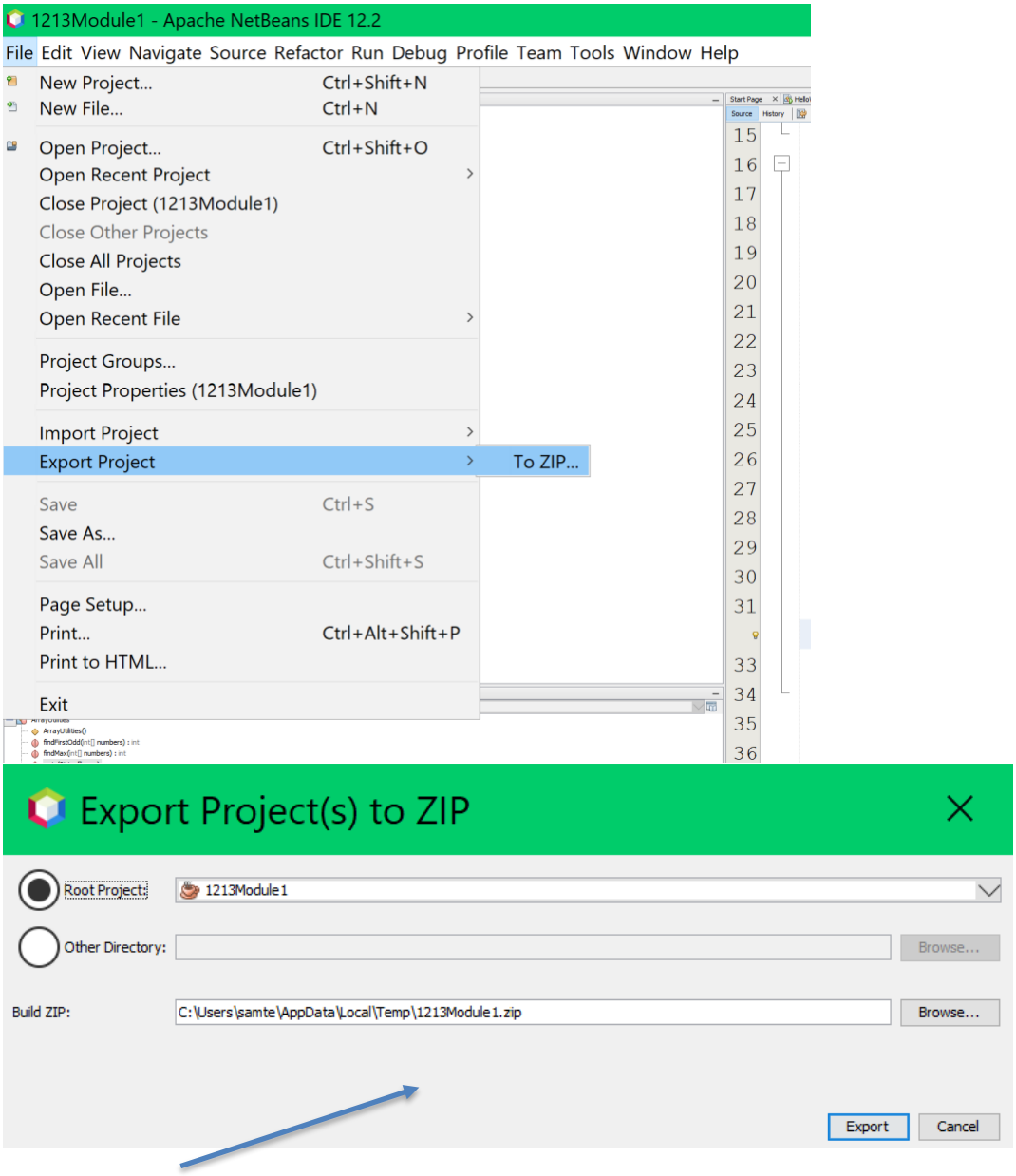
8) When your program works take a screenshot of your NetBeans window and output and save the file as lab1_partB to include with your submission.

Part C: Exporting a project using NetBeans
Objective: Be able to export projects using NetBeans.

1) Sometimes, you may need to move your project from one computer to another or to share it with someone else. An easy way to do this is to export your project as a zip file.

2) To export your 1213Module0 project, select this project in the Projects Pane, then go to File → Export Project To ZIP.... In the window that pops up don't change the Root Project line. On the 'Build ZIP:' line, you click 'Browse' to select the directory where you want your zip file to be stored, but you'll have to enter the filename yourself. Make sure to include the zip

file extension when you enter the filename. Then click 'Export', a new file window will pop up, and you can find the 1213Module0.zip there.



Click the 'browse' button to specify the folder you want this to be saved in. You are encouraged to keep your files for this course in one folder for easier reference when it is time to submit or review your work.

Make sure you note where you saved your newly created project export file.

ITSC 1213 Introduction to Computer Science 1 - Module 0 Lab

You're done. In summary, in this lab, we discussed:

- **How to create a new NetBeans Project**
- **How to close an open NetBeans project**
- **How to specify where the code for this project will be stored**
- **How to run a program**
- **How to "Clean and Build" a program to create an executable .JAR file**
- **How the folders and files associated with NetBeans projects are organized and where they are stored**
- **How to run a Java JAR file from a command line prompt**
- **How to add files to NetBeans project**
- **How export a NetBeans project folder**

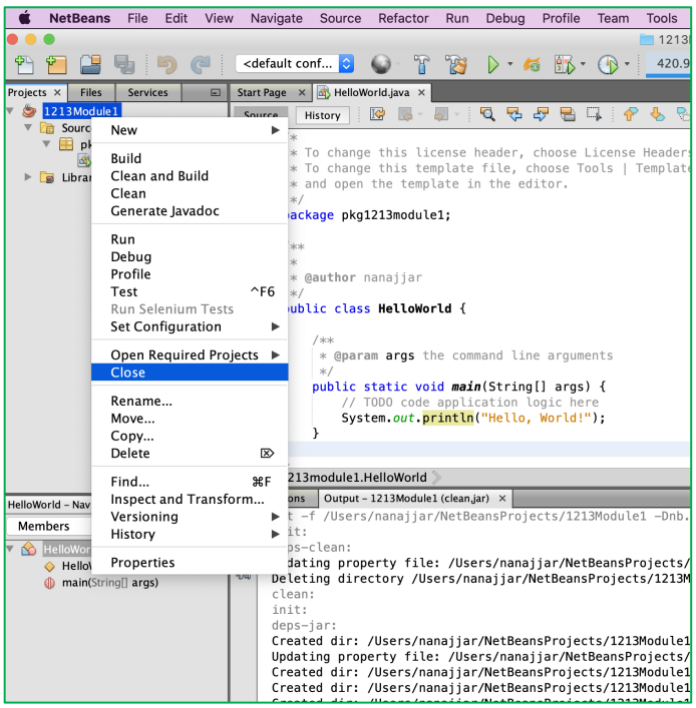
Lab 0 Canvas submission

- In this week's module, you will find the link to the lab completion submission. Submit the following files
    a. screenshots for parts A and B
    b. your project export for part C - 1212Module0.zip

Explore More in NetBeans

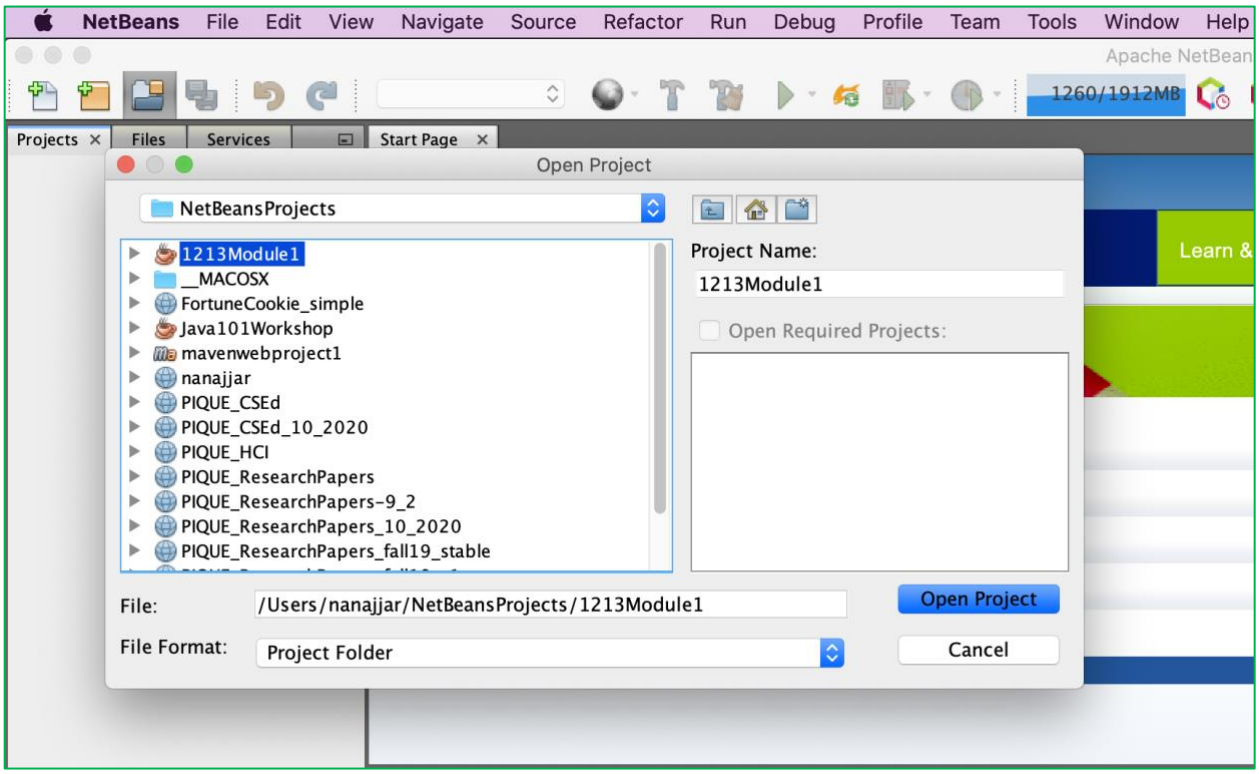
How to close a project?

You can right-click on the project and select 'Close'. This will close the project, including any source files that are open in the edit pane on the right.



ITSC 1213 Introduction to Computer Science 1 - Module 0 Lab

How to open an existing project?



Compare Project structure to File Structure in NetBeans: