Project 1: Introduction to Java and jGrasp

How to get project code, make changes, test, and upload to Gradescope.

Overview

In this project, you will gain experience downloading, extracting and opening a project in jGRASP. Then you will make some changes to a java program, compile and run it to make sure it is correct. Finally, you will create a zip file and upload it to Gradescope to verify that your project is correct in the autograder.

Learning Goals:

- How to download, extract, and open a project in jGrasp.
- Work with files and folders (directories) on your machine.
- Practice editing a Java file in jGRASP.
- Experience re-assigning variables and writing a print statement.
- Use recommended style guide.
- Practice compiling and executing code to verify correctness.
- Create the correct zip file for upload to Gradescope.

Project Specification:

You will download, extract, and open the project code in jGrasp. Then, you will make some modifications to the code, compile and run the program. You can see if your program works by examining the output that is printed to the console. There are five modifications that you need to make in the code. Each modification is specified by the comments in the source code file.

After you have made all five modifications as specified by the comments, you will create a zip file and then upload it to Gradescope. Gradescope will run an autograder on your code and report your score. Note that if your code does not compile and run or if your zip file is not correctly made you will receive zero points. If your code runs and produces a correct output, you will receive 100 points.

The project files you see when you extract the project code are:
`Project1.java`
`Project1.gpj`

The first file is a Java source code file where you will make your modifications. The second file is a jGrasp project file. You do not modify this file.

The rest of this document describes the steps you need to take for a successful project outcome.

Download the Project & Tasks to Complete.

1: Log in to Moodle, download the `Project1.zip` file and save it in your COMPSCI121 folder which you should have created. Do not work with project files in your Desktop, Downloads or your OneNote folder. Note that a folder is also called a directory.

2: Unzip the downloaded zip file by either clicking on it or right click and extract the files. (You should then delete the zip file so that you don't use it). If you use a mac and double click on the zip file, a `Project1` folder will be created. Otherwise, create a `Project1` folder and extract the zip files to that folder. See Figure 1.

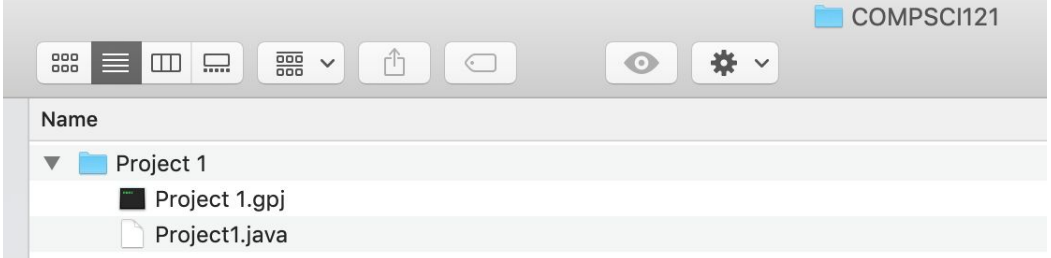
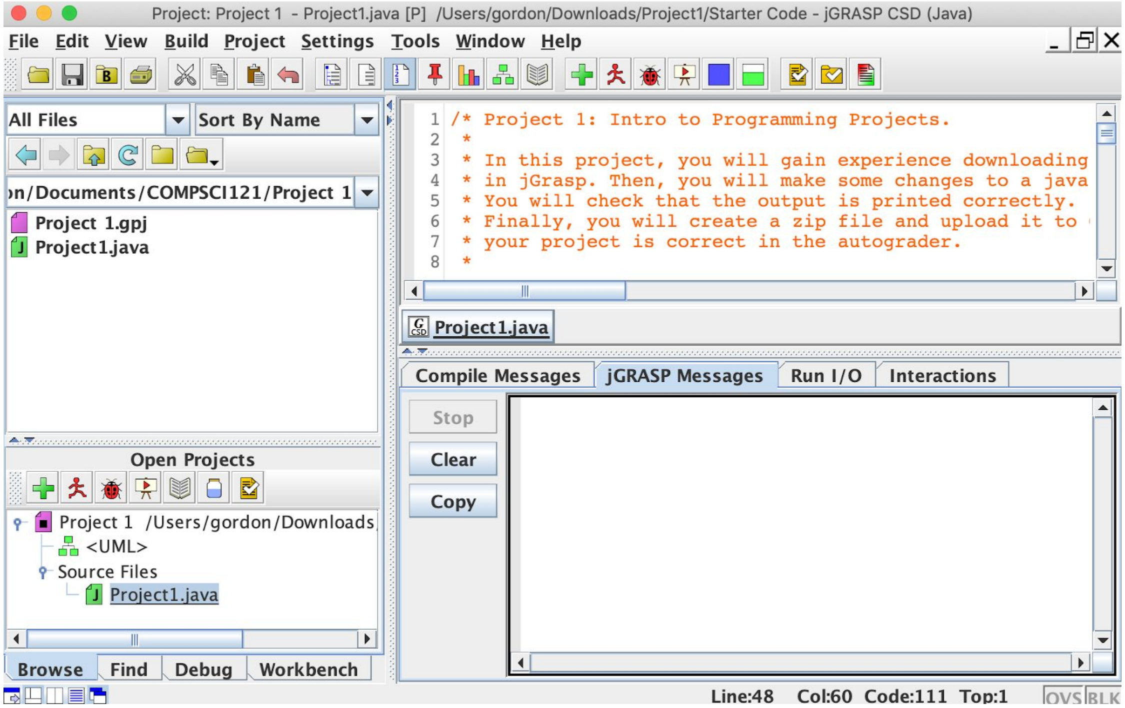


Figure 1: Extracted files in the COMPSCI121 folder.

3: Open jGRASP by clicking on the icon on your laptop. Using the navigation pane navigate to the `Project1` folder. Click on `Project1.gpj` to open the jGRASP project file. Check that you can see the **"Open Projects**" window as in Figure 2.

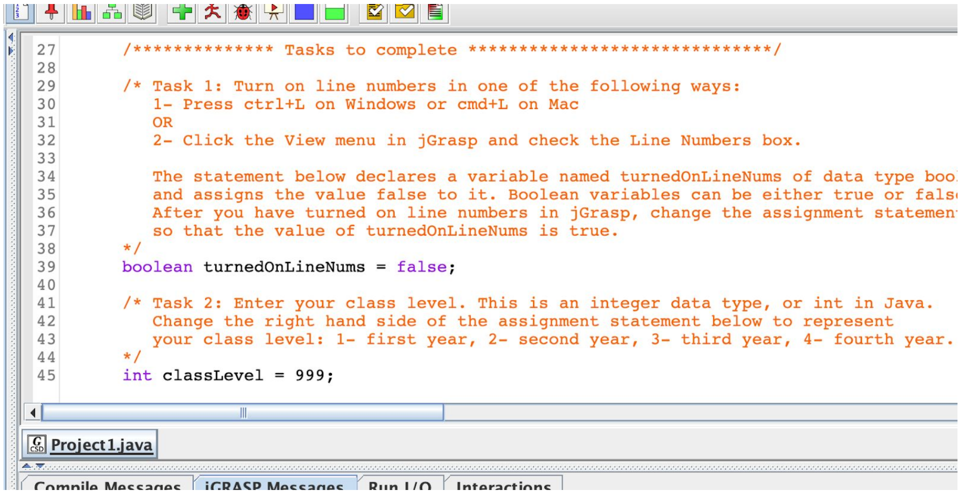

Project: Project 1 - Project1.java [P] /Users/gordon/Downloads/Project1/Starter Code - jGRASP CSD (Java)
File Edit View Build Project Settings Tools Window Help
All Files
Sort By Name
›n/Documents/COMPSCI121/Project 1
Project 1.gpj
Project1.java
Open Projects
Project 1 /Users/gordon/Downloads
<UML>
Source Files
Project1.java
Browse Find Debug Workbench
Project1.java
Compile Messages jGRASP Messages Run I/O Interactions
Stop
Clear
Copy
Line:48 Col:60 Code:111 Top:1
OVS BLK

```
/* Project 1: Intro to Programming Projects.
 *
 * In this project, you will gain experience downloading
 * in jGrasp. Then, you will make some changes to a java
 * You will check that the output is printed correctly.
 * Finally, you will create a zip file and upload it to
 * your project is correct in the autograder.
 *
```

Figure 2: View of Project1 files opened as a project in jGrasp.

Remember that you must ***open the code as a project***, not by opening individual java file/s.

4: Read the comments in the source code file, `Project1.java,` and start your work. The image below shows that the comments appear in orange in jGrasp (Figure 3).



```
/************** Tasks to complete ******************************/

/* Task 1: Turn on line numbers in one of the following ways:
   1- Press ctrl+L on Windows or cmd+L on Mac
   OR
   2- Click the View menu in jGrasp and check the Line Numbers box.

   The statement below declares a variable named turnedOnLineNums of data type boo
   and assigns the value false to it. Boolean variables can be either true or fals
   After you have turned on line numbers in jGrasp, change the assignment statemen
   so that the value of turnedOnLineNums is true.
*/
boolean turnedOnLineNums = false;

/* Task 2: Enter your class level. This is an integer data type, or int in Java.
   Change the right hand side of the assignment statement below to represent
   your class level: 1- first year, 2- second year, 3- third year, 4- fourth year.
*/
int classLevel = 999;
```

Figure 3: View of comments and a Java statement in the code editing pane.

The java source code file, `Project1.java`, has five tasks that you must complete. Each task is specified by comments that appear in orange color just above the Java statement that you will modify. The last task requires you to write a complete print statement. Please read all of the comments in this file.

When you have completed a task, compile and run the program. You can verify that the changes you made appear in the output.

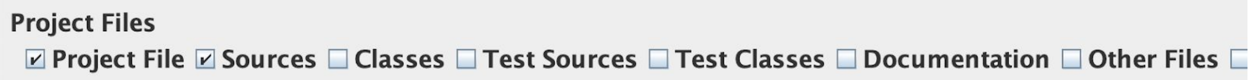
If you have any questions or encounter any problems getting your code to work, please seek support offered by the course staff or your peers.

Export and Submit

When you have completed this project, you need to create a zip file that contains the entire Java project. This is done in jGrasp in the following manner.

1: Create the zip file.

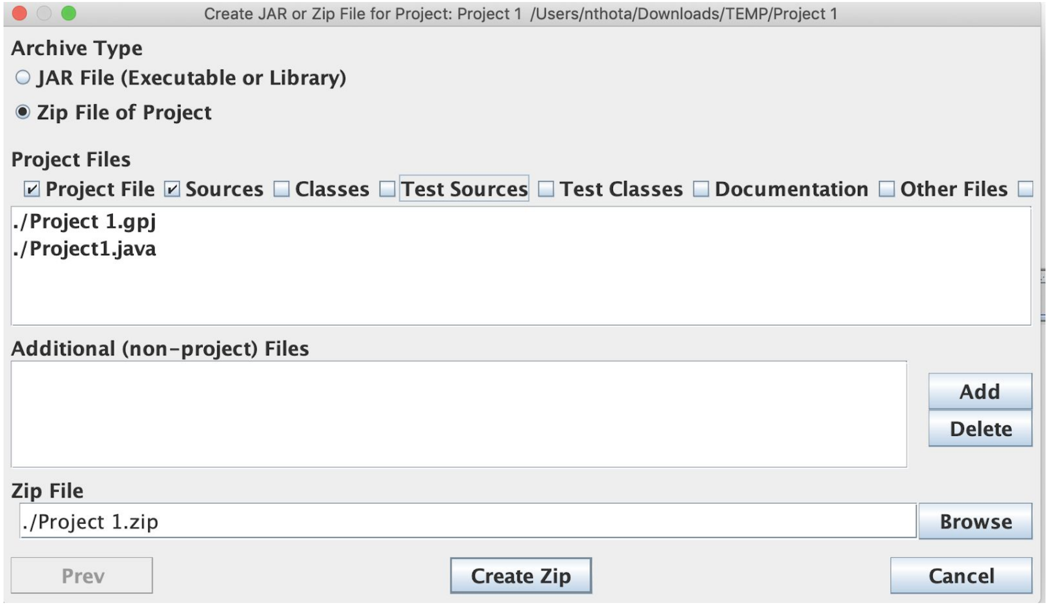
Click on the "**`jar`**" icon. Select **Project File** and **Sources.**



Project Files

☑ Project File ☑ Sources ☐ Classes ☐ Test Sources ☐ Test Classes ☐ Documentation ☐ Other Files ☐

Click **Create Zip** and the **`.zip`** file is created. See figure 4 below.


Create JAR or Zip File for Project: Project 1 /Users/nthota/Downloads/TEMP/Project 1

Archive Type

○ JAR File (Executable or Library)

● Zip File of Project

Project Files

☑ Project File ☑ Sources □ Classes □ Test Sources □ Test Classes □ Documentation □ Other Files □

./Project 1.gpj
./Project1.java

Additional (non-project) Files

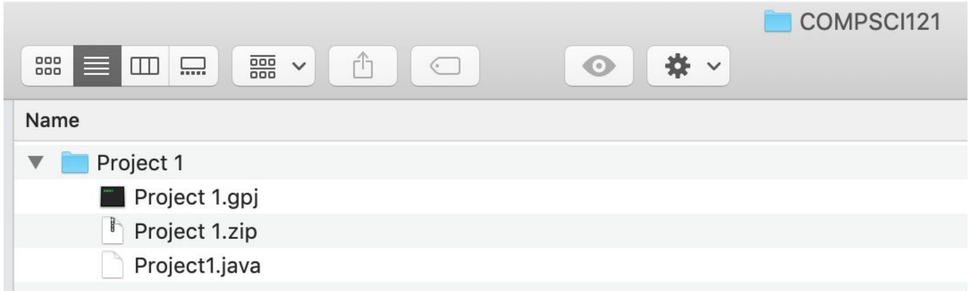
Zip File

./Project 1.zip

Figure 4: Create a zip file window that appears in jGrasp.

Note: A `zip` file is a compressed file. The data in every file and folder contained in the zip file is compressed. The data is uncompressed when you extract a zip file. A `jar` file is a type of compressed file that is used to deliver a java program. Jar files are often used to deliver java library code. However, we require that you create a zip file to upload to Gradescope.

After clicking on the "*Create Zip*" button, you should see the file **`Project1.zip`** in the Project1 directory (see figure 5). Note that the name of the zip file will be the same as the project name.


COMPSCI121

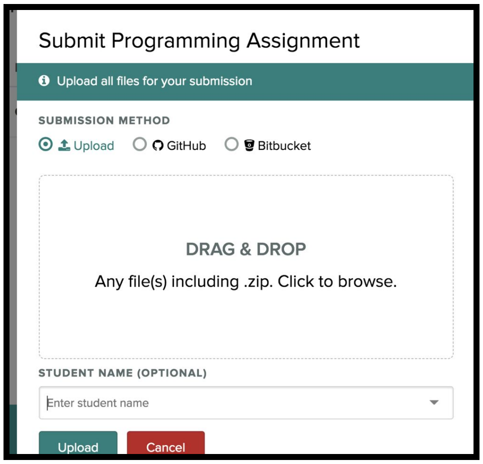
Name

Project 1
Project 1.gpj
Project 1.zip
Project1.java

Figure 5: View of the Project 1 directory after the Project1.zip file has been created.

Important Note: If you are not using jGrasp to develop your code, you must ensure that your zip file has the correct file structure. The correct structure is that all project files are at the top level of the zip file. You can verify this by opening the zip file. Many IDEs will create a zip file with one command, but the file/directory structure may not be correct for the Gradescope autograder. If you upload a zip file with incorrect structure the autograder will not evaluate your code and you receive zero points even if your code works correctly. Contact the course staff if you have any questions about the correct structure of the project zip files.

2: Submit the file to Gradescope.
Log into Gradescope, select the assignment, click to browse, navigate to your `Project1.zip` file and select it, then click on the Upload button.


Submit Programming Assignment

Upload all files for your submission

SUBMISSION METHOD

● Upload ○ GitHub ○ Bitbucket

DRAG & DROP

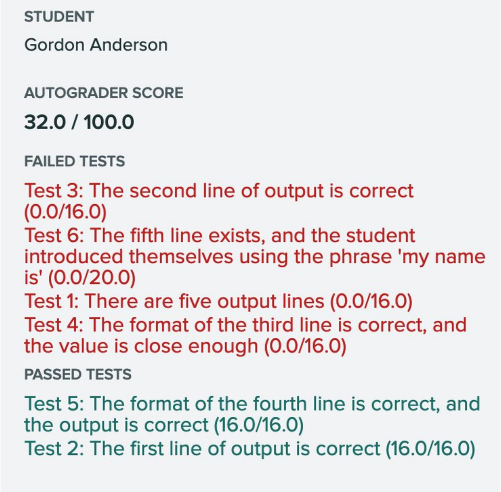
Any file(s) including .zip. Click to browse.

STUDENT NAME (OPTIONAL)

Enter student name

Figure 6: Gradescope file submission window.

Gradescope will extract your zip file and run tests on your code. It may take a few minutes. If your zip file is not structured correctly or if your code does not compile, you will not receive any credit as the autograder could not be run. If your code does run, you will get a report of how many tests passed and failed with the points received (figure 7). You can fix hour code and resubmit again.


STUDENT
Gordon Anderson

AUTOGRADER SCORE
32.0 / 100.0

FAILED TESTS
Test 3: The second line of output is correct (0.0/16.0)
Test 6: The fifth line exists, and the student introduced themselves using the phrase 'my name is' (0.0/20.0)
Test 1: There are five output lines (0.0/16.0)
Test 4: The format of the third line is correct, and the value is close enough (0.0/16.0)

PASSED TESTS
Test 5: The format of the fourth line is correct, and the output is correct (16.0/16.0)
Test 2: The first line of output is correct (16.0/16.0)

Figure 7: An example of the autograder report where two tests passed, four tests failed.

Gradescope consent

Make sure that you have given consent to use Gradescope. You will be informed when your Grtadescope account is ready for you to upload your project.

NOTE: Remember, you can re-submit the assignment as many times as you want, until the deadline. If it turns out you missed something and your code doesn't pass 100% of the tests, you can keep working until it does. Start on the project early and seek help if necessary.

Tips.

- Do one task in the Java file and test it. Then move on to the next task. Do the tasks one at a time, testing each time.
- Use jGrasp to troubleshoot any problems with your code as it is designed for that purpose. Although you can submit many time to Gradescope, it is efficient to work on code in jGrasp and only submit when you think you have solved any problems.
- Start early! If you get stuck you need to make sure you have enough time to seek help.
- Seek help when you get stuck. We have office hours and forums specifically for you to ask questions when you need assistance. Use public posts as much as possible so we don't have to answer the same question multiple times, only use a private post if you need us to see your code or have questions specific to you.
- Look at examples in both your textbook and previous labs. While you shouldn't be copying code, looking at examples can help clear up semantic or simple questions should they come up.
- Submit to gradescope at least once, even if you aren't completely done. Grtadescope serves as a backup for your project.
- If you need help, you do not need to submit your code or pictures of your code to the forums on Moodle or Piazza. Submit your code to Gradescope and then the 121 staff can easily access it. Mention line numbers in your question when applicable.