CS 210, Fundamentals of Computer Science I
Lab 1: Your CS account, jGRASP and HelloWorld
Submit to Canvas

Computer Science

This lab is worth 100 points and has five parts. The goals for this lab are:

I. **Becoming familiar with jGRASP – your first Hello World Java program**
II. **Compile and run your HelloWorld.java from the command line (not graded, but useful experience for you)**
III. **Upload your work to Canvas**

If you do not complete the lab in the time allotted, then please return to the lab in your spare time, and complete it by the **due date, which is specified on Canvas**. All the labs PDFs for this course will be posted in Canvas: Files-> labs->lab<#>.

Using your BC login information log into the Canvas. Navigate to Files-> labs->lab1.

I. Becoming familiar with jGRASP – your first Hello World Java program

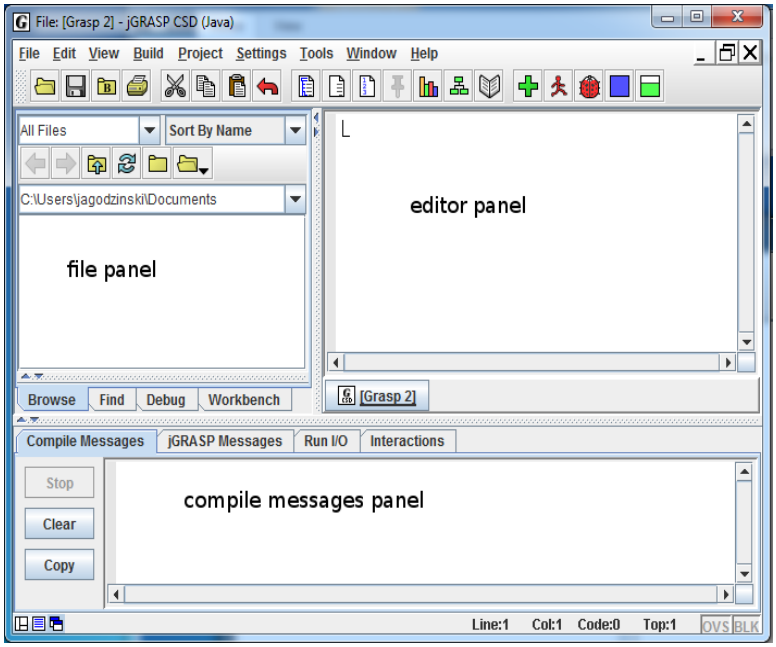
Now it's time to write some code. jGRASP is a lightweight Integrated Development Environment (IDE). You will be using it throughout this course to write, compile, debug, and execute your Java programs, as was discussed in lecture.

1. If you don't have Java or JGrasp downloaded on your computer, download them at Oracle website and http://www.jgrasp.org respectfully.

2. To start jGRASP, mouse-click on the windows icon in the lower-left hand corner of the desktop. If the jGRASP icon does not appear in the list of programs, type "jgrasp" into the search box. When found, click on the jGRASP icon, which looks like the following: jGRASP has several panels (Figure 9) and icon buttons. You'll explore some of these today. In this (and subsequent) labs, if you are asked to "write" java code, mouse-click inside the editor panel, and type java code into that panel.

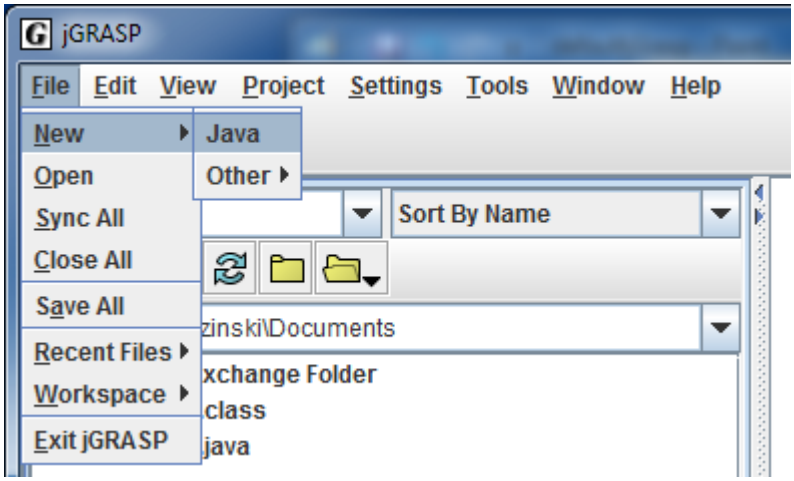
Figure 9: jGRASP's *editor panel* is where you input java code.

The *file panel* displays .java (and .class) files that you have created. The *compile messages panel* is where jGRASP displays compilation messages when you compile your code. The *run I/O* panel (not shown) shows the output that is generated by your program.



3. To create a new java file, mouse-click on the *file* menu, then select *new*, and finally click on *Java* (Figure 10). Note there may be several other options other than *Java* on your computer.

Figure 10: To create a new java file, select the New option from the File menu, followed by the Java option.



4. Using the keyboard, type the text that is shown in Figure 11, into the editor panel of the new java file that you just opened. **Type the code EXACTLY as shown**. Be careful to include semicolons and brackets, and to use proper capitalization. Change the comments accordingly, by adding **your name to the right of "Author" and the date**. Remember that adding blank lines makes your code easier to read.

```
// Author:
// Date:
// File: HelloWorld.java

// Simple java program that prints "Hello World!" to the screen
public class HelloWorld {

   // the main method
   public static void main(String[] args){

      // Invoke the println method from System.out, which
      // requires one argument, the item to be printed
      // to the screen
      System.out.println("Hello World!");
   }
}
```

Figure 11: Text for the HelloWorld java program.

5. Once you've entered the text in Figure 11 into the editor panel of jGRASP, save your code into a file that is called *HelloWorld.java*. Recall from lecture that the name of the .java file must end in .java. To save a file, click on the save button (Figure 12a). A menu/dialogue box will appear, and will ask you where you want to save your file. **Using the drop-down menu, save your *HelloWorld.java* file into the lab1 folder of your cs210Submissions file in your drive of your CS210 account.** When you have successful saved your file in the right location, it should appear in the *file panel* as *HelloWorld.java.*

Figure 12a
The save button

Figure 12b
The compile button

Figure 12c
The run button

Figure 12d
The line-numbering button

6. Recall from lecture and from the textbook readings, that the java compiler converts your human-readable text in a .java file into a .class file (the computer byte code) that is invoked using the Java Virtual Machine (JVM). To compile your java code, click on the compile button (Figure 12b).

7. If you have typed into the editor panel the code from Figure 11 exactly, with no errors, then the *compile messages panel* will display no errors (Figure 13). If you have typed java code that has a syntax error, an error message will be displayed in the *compile messages panel.* If that is the case, you'll need to find the syntax error in your .java code, and compile again. Ask Tatiana if you have questions.

Figure 13: jGRASP's compile message panel displays status and error messages that arise when you compile your program. When the code is error-free, the compile message is empty.

```
----jGRASP exec: javac -g HelloWorld.java


----jGRASP: operation complete.
```

8. If your *HelloWorld.java* code compiles correctly, you'll see a *HelloWorld.class* file in the *file panel*. Once you've done that, it is time to invoke the Java Virtual Machine. In other words, you want to run the program. To do that, click on the run icon (Figure 12c). The output of your program will appear in the *Run I/O* tab (Figure 14).

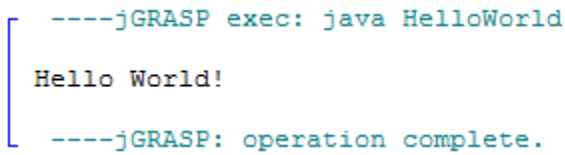


Figure 14: The output of your first java program, which prints "Hello World!" to the screen.

9. That's it. You now know how to write, compile, and execute a java program using jGRASP. One additional feature that you can enable is line numbering. Click on the line-numbering button (Figure 12d), and jGRASP will automatically line-number you code. Line numbering is useful, especially if you are talking to somebody about your code; it allows you to refer to different line numbers so that the person you are talking to knows which part of the code you are discussing. Be sure to save your *HelloWorld.java* file if you make any changes to it, so that it can be graded.

IV. Compile and run your HelloWorld.java from the command line.

1. In Windows click **Start**, go to **All Programs**, and then type "cmd" into the search box. A command window should open
2. At the command, make sure you are in the same directory where the Java program is. Use the following commands:

   cd --- tells you where you currently are

   dir --- lists files in your current directory

   cd *dirname* --- change directory. You always start out in your 'home directory', and you can get back there by typing 'cd' without arguments. 'cd ..' will get you one level up from your current position. You can make big leaps or avoid walking around by specifying pathnames.

3. Then, use the following command to compile your program:
   javac <FileName> or javac HelloWorld.java

4. To run the Java program, use the following command:
   java <ClassFilename> or java HelloWorld

5. Show the running program to your grader.

V. Upload your work to Canvas.

For this first lab, your code should be saved to your CS210 account, and uploaded to Canvas. Here's what we are looking for, when grading your submission. In later homework and lab submissions, there will be additional items that will be graded, which will be described in future lectures.

1. Each .java file must be thoroughly commented. If your code does not compile because you've been unable to fix a syntax error, then the comments will allow you to receive partial credit.
2. Your java code must be indented, so that the code is easy to read (more on this in future lectures).

For this lab, make sure that the following file in your lab1 folder and uploaded to Canvas:

HelloWorld.java

There will be additional files (the .class files, for example) in your lab1 folder, but don't worry about them. Those files are not graded; they are just the byte code files that are used by the Java Virtual Machine. **Do not upload them to Canvas**.

Rubric

	File / task	Points
I.	**Becoming familiar with jGRASP – your first Hello World Java program**	40
II.	**Compile and run your HelloWorld.java from the command line**	10
III.	**HelloWorld.java is properly formatted and uploaded to Canvas**	50
	Total	100