Lab #0 – Eclipse and GitHub

Total Points: 25

Upon completion of this lab, you should be able to do the following:

* Create a Java project in Eclipse
* Clone a repository from GitHub and import it into the Package Explorer in Eclipse.
* Understand the method for getting the starter code as well as submitting labs in this course
* Share source code files with a lab partner

**Instructions**

**1. Obtain starter code and create Git repository**

* Go to the "How to…" webpage (linked from the main course homepage) and follow the steps listed in the section "Obtaining starter code and creating Git repository". This will clone a Git repository containing the starter code for this lab and import that code into a Java project that appears in the Package Explorer tab in the left part of the Eclipse window.
* In the Package Explorer, you can edit the contents of the package by writing code, adding new classes, etc.
  + Locate the voting package in lab-00 project. (You will probably need to double-click the project then double-click the src directory. Note that "src" is the traditional abbreviation of "source", for directories that contain source code).
  + Expanding that package will reveal the one java source file you will be working on in this part of the lab (Candidate.java)
  + Before the next step, go back to the main labs webpage and download the file ExampleForLab00.java, available from a link next to the instructions for this lab. Open that file in Eclipse or any other suitable program. By browsing this file, you can see examples of how to do all the tasks listed next. Don’t worry about fully understanding the code. Because it is so early in the semester, it is fine to just copy and paste from this example code and make a few alterations as necessary.
  + Create a *main* *method* within the Candidate class. Inside the main method, create three Candidate objects. Add varied number of votes to each Candidate object. Then display each candidate’s name and the number of votes. Run Candidate.java by choosing Run from the Run menu. Candidate names and their vote counts should be printed to the console.
* Go to the "How to…" webpage and follow the steps listed in the section "Committing and pushing your changes to GitHub for backup and grading". This saves all the changes on your computer and also pushes them to the GitHub servers. This is a good way to back up your progress, and it is also the way you will submit code for grading. The instructor will grade the most recently committed version.

**2. Create a new project, a new class, and run a new Java program**

* Go to the "How to…" webpage and follow the steps listed in the section "Creating a new project in Eclipse" to create a new project called "MyFirstProject".
* Now follow the "How to…" webpage instructions for "Creating a new package in Eclipse" to create a new package within your project, called "MyFirstPackage"
* Now follow the "How to…" webpage instructions for "Creating a new class in Eclipse" to create a new class within your package, called "Person"
* In the editor window, paste in the Person class code from Person.java — this file is available on the Labs webpage, adjacent to the instructions for this lab.
* Run your Person program by following the "How to…" webpage instructions for "Running a Java program". The statement “This person is Nayer Montgomery and her email is nayer@yahoo.com” should be printed to the console. Take a screenshot that includes this output within Eclipse. Paste your screenshot into the lab report that you will eventually submit to Moodle for grading. (To see more details about this lab report, go to the "How to…" webpage and read the section "Submitting lab report, including self-assessment report, to Moodle".)

**3. Find your source files (e.g. to share with a lab partner)**

Go to the "How to…" webpage and read the section "Finding your source files to share with a lab partner". Use this technique to locate the file Candidate.java within your local file system. At this point, you could attach the file Candidate.java to an email and send it to a partner if desired. However, for the purposes of this lab you can demonstrate your mastery of this skill by simply pasting the full path name of the Candidate.java file into your lab report that will be submitted to Moodle.

**4. Finish your write up and submit to Moodle**

As with every lab, complete your lab report with a short self-assessment report, mentioning any aspects that are incomplete or not working correctly. For details about the self-assessment report, go to the "How to…" webpage and read the section "Submitting lab report, including self-assessment report, to Moodle".