Learning Activity: Class Tools

Back

The first learning activity is designed to give you some familiarity with the tools we will use in this course. This assignment consists of four parts.

- Part 1: Forum Posting
- Part 2: Java Application
- Part 3: Student Survey
- Part 4: Canvas Quizzes

Preliminary Grading

The learning activity is "officially" due on Sunday evening, 1 July at 6:00 pm (plus a 59-minute grace period, as outlined in the syllabus). There is a preliminary grading of Part 2 of this learning activity on Wednesday 27 June at noon. This is when the Comprehension Exercise would normally be due. Please think of this preliminary grading as if it were the CE for the week. This early deadline will give us all the opportunity to clear up any difficulties or questions that arise with using the support tools we'll be using this quarter. This is particularly important since we won't have class in the second week of the quarter, since that Wednesday is the Fourth of July. You still have until Sunday evening to complete this LA for full points, we just won't have any face-to-face time to clear up problems.

Part 1: Forum Posting

This is a little tricky. For this part of the assignment, you need to make two forum posts.

Claiming a Topic

For Part 2 of the learning activity, you will create a simple Java program that will print out three lines of text; the first one will include your name, and the second one will present some fact that is verifiable on the World Wide Web. More about that in the Part 2 write-up. However, each of you needs to have a unique fact. You can insure that your fact is unique by posting to the **My Topic** thread in the forum. The posts are date/time stamped, so if there is a "collision" the first poster "wins".

Introduction

Use this topic to "introduce" yourself to your classmates in the **Introductions** thread. This is the second forum posting required for Learning Activity 1, Part 1.

For this posting, you need to reply within this conversation in the forum. The posting should contain the following four pieces of information:

- Your name, minimally first and last
- What you want to be called
- Your favorite color
- The "something interesting" fact you shared on the first day of classes

The "something interesting" fact is to help other students (and me) to put a face with the name. If you weren't at the first class meeting, you need to write "something interesting" about yourself. (It doesn't have to be earthshatteringly interesting. It could be where you grew up, what your favorite pastime/food/movie/sport is, what kind of pet you have or if you're allergic to animal dander, etc.)

These four pieces of information should appear in four separate "paragraphs", that is, there should be blank space vertically separating the pieces of information. See the example posting in the forum.

The favorite color information does not have be accurate. You could answer "red" just because it's fast and easy to type, even if you hate red. However, the other three pieces of information should be accurate.

Grading Rubric

1 point.

Part 2: Java Application

For this part of the learning activity, you will create a Java class that you will submit as homework.

The Class File

The Java class shall be an application that prints three lines of text. The first line of text printed by the application shall include your name. This should be the name (first and last) that you're using in Moodle. The second line of text printed by the script shall be some World Wide Web-verifiable fact. Wikipedia is a good resource to use for finding some verifiable fact. These facts shall be unique within the class. Use the forum posting in Part 1 of this assignment to insure uniqueness of your fact. The third line will respond to input from the command-line. When a program is run, additional information can be given on the command-line, following the name of the application class, that is, the class with the application method. This information appears in the String array parameter of the application method.

There are exactly four possible third-lines:

1. `No command-line arguments given.`
   This text appears when there are no additional command-line arguments.
2. `The command-line arguments say "OOP".`
   This text appears when there is one command-line argument, the text "OOP".
3. `The command-line arguments say "CSC 143".`
   This text appears when there are two command-line arguments, the text "CSC 143", specifically, two three-character Strings, "CSC" followed by "143".
4. `The command-line arguments are not recognized.`
   This text appears when there are command-line arguments which do not match any of the three cases listed above.

For full credit, the text for the third line shall match one of the four options shown above, including the placement of the punctuation. (I realize that the placement of the period shown above is in conflict with the norms of American usage. The placement reflects the absence of the period on the command-line.)

For example, assuming my application class was named **Part2** and the command-line was:

```
java Part2 OOP
```

my script could print out:

```
My name is Dan Jinguji.
Java is the most populous island in the world.
The command-line arguments say "OOP".
```

Another example, same assumptions, if the command-line was:

```
java Part2 CSC
```

my script could print out:

```
My name is Dan Jinguji.
Java is the most populous island in the world.
The command-line arguments are not recognized.
```

Additional requirements for the source code file:

- The source code file shall compile.
- The compiled file shall run as a Java application.
- The application code shall appear in a public class.
- There shall be a JavaDoc comment (documentation comment) for the class definition.
- The class JavaDoc comment shall include your name (first and last) in the **@author** tag.
- The class JavaDoc comment shall include the number and name of this assignment (Assignment 1: Class Tools) in the **@version** tag.
- There shall be a JavaDoc comment for all non-private resources within the class. At this point, that will only include methods.
- There shall be a comment immediately before (above) the code that prints your name. A short descriptive phrase is sufficient, e.g., "prints my name".
- There shall be a comment immediately before (above) the code that prints the "interesting" fact. A short descriptive phase is sufficient, e.g., "my interesting fact".
- There shall be a comment immediately before (above) the code that prints the line responding to the command-line arguments. A short descriptive phase is sufficient, e.g., "handle args".
- All comments shall precede the associate Java code.
- All comments shall align vertically with the associated Java code.

Now, there is a built-in "trap" in this assignment. And, here's the hint for avoiding it:
When working with reference types, the == (two equal signs) operator does not check for equivalent values. It checks for identical instances.

If you find this to be a tricky assignment, you should talk with the instructor at your earliest convenience.

The Citation File

The submission will include a short textual file giving the citation information for the verifiable fact you include in the script.

The citation file shall be an ASCII (plain text) file with three sections.

- The first section shall include your name (first and last) and this activity name and number (LA1: Class Tools).
- The second section shall repeat the verifiable fact you used in your script. Use the heading: "Fact:".
- The third section shall give the citation (verification) for your fact. Since the information is taken from an Internet source, the citation is the URL for the webpage and the date retrieved. Use the headings "Source:" and "Date retrieved".

Here is an example of what the report file would contain:

```
Dan Jinguji
LA1: Class Tools

Fact: Java is the most populous island in the world.
```

```
    Source: http://en.wikipedia.org/wiki/Java
    Date retrieved: 15 August 2011
```

Submission

Two files:

- The Java source code file with comments
- The ASCII plain text file with the citation for your fact.

Grading rubric

2 points.

Source code: 1.5
 * The submission compiles and runs without error
 * Submitted code produces the required output
 * JavaDoc comments are present, produce no errors or warnings
Citation: 0.5
 * The citation is an ASCII (plain text) file
 * The citation has the correct format and content

Part 3: Student Survey

A way for me to learn more about you and your background. This is structured as a Canvas quiz. Now, there are no answers for this quiz that are wrong. So, completing the quiz on time will earn you normal successful completion.

The only way to sidestep Canvas grading this quiz is to make each question an essay question. After this quiz closes, I will go through and manually award points for the questions answered. Unanswered questions shall not receive points.

Grading Rubric

1 point for completing the student survey

Part 4: Canvas Quizzes

This is a sample Canvas quiz. That is, Canvas will collect your answers and score them. The purpose of this quiz is for you to get a sense of how rigidly Canvas grades things.

None of these questions should be difficult for you to answer. They are all on material covered in CSC 142.

This quiz will let you take and submit it multiple times. The highest score will be the one recorded.
Note: This is the only quiz this quarter which permits multiple attempts.

Grading Rubric

1 point.

The quiz will be automatically graded by Canvas.

Back