Queens College, CUNY, Department of Computer Science

Software Engineering

CSCI 370

Fall 2018

Instructor: Dr. Sateesh Mane

revised due date = Friday September 21, 2018

2 Project 2

2.1 Submission details

- The project 2 specification is to implement a GUI for a four function calculator.
- *The due date of Project 1 will be postponed, by one week, to 10/7/2018.*
- **Project 2 has emergency priority.**
- **All students will be required to demo their Project 2 GUI in class on 9/17/2018.**
- **Bring a computer to class, and demo your GUI on your computer.**
- **All students must submit individual written reports for Project 2.**
  1. This is because your work will be a "final submission" for the project.
  2. **See Sec. 2.2 for details of the report.**
- **Tutorial:**
  1. For Java code, a student has kindly written a tutorial how to make a jar in a zip file, which is safe to email.
  2. The instructions use the Eclipse IDE.
  3. The tutorial is posted online at the following link:
     `http://venus.cs.qc.cuny.edu/~smane/cs370/projects/java_Tutorial.pdf`
- **Teams:**
  1. Students may work in teams (either existing teams or form new teams).
  2. *For new teams, the names of all team members must be submitted to me before 9/17/2018.*

2.2 Report

- **The project 2 report is due on Friday 9/21/2018.**
- **There is no need to submit a report on Tuesday 9/18/2018.**
- The report should be submitted as a zip archive (do NOT use rar).
- Please employ the following naming convention for your report (zip archive).

  `StudentId_first_last_CS370_Project2.zip`

- The zip should contain an executable jar file.
- The zip should contain a "cover" document (txt/pdf/docx).
- *Every student must submit a report.*
  1. For students in teams, the jar file can be the same for all team members.
  2. However, the cover letter must be written individually (although it can be copy and paste mostly the same for team members).
- The cover letter should contain the following.
  1. Names of all team members, if in a team.
  2. A "mission statement" (= implement a GUI for a four function calculator).
  3. A description/summary of what happened in the demo in class on 9/17/2018.
  4. A list of bugs that were discovered in the demo.
  5. A description of debugging procedures and tests to check the functionality.
  6. *A description of additional software checks and tests to validate the functionality.*
- Students who did not demo a GUI in class on 9/17/2018 will receive a grade of C or D (to be decided). This does not apply to students in teams, where another team member gave the demo in class on 9/17/2018.
- **Red star**
  1. Students who received a red star for their demo in class on 9/17/2018 will probably receive a grade of A for Project 2.
  2. *However, students who do not submit an individual project report will not score A, regardless of a red star in class on 9/17/2018.*
  3. All "red star students" must sumbit (in their cover document) a description of additional software checks and tests to validate the functionality.
- ***Rigorous software testing is expected from all students.***

2.3 GUI details

- The project specification is to implement a GUI for a four function calculator.
- **The basic GUI layout was displayed in class.**
- **The basic GUI functionality was displayed in class.**
- The four functions are the basic arithmetic operations $+, -, *, \div$.
- **Buttons:**
  1. Digits $0 - 9$ and button "." for decimal point.
  2. Button "+/−" to change sign of operand.
  3. Button "=" to perform calculation and display result.
  4. Buttons "C" to clear the current operand and "AC" to clear everything from memory.
- **You are NOT required to write a programmable calculator.**
  1. You are required to support only two operands.
  2. A calculation such as $1 - 2 * 3$ will be performed as follows:
  3. Enter "2" then enter "$*$" then enter "3" then enter = (to display "6") then enter "+/−" to change sign (to display "−6") then enter "+" then enter "1" then enter = (to display "−5").
  4. You are NOT required to implement a "stack" or "binary tree" to support multiple nested operations and operator precedence, etc.
- Results (for large/small magnitudes) can be displayed in fixed format or scientific notation.
- Leading zeroes should not be displayed (enter "0" and "1" the display should show "1" not "01").
- Trailing zeroes should not be displayed (enter "1" and "+" and "2.0" and "=" the display should show "3" not "3.0").
- Input a fraction as "." and "4" the display should show "0.4" (not ".4").
- Division by zero should display "err" or a suitable error message.
  1. Bad input must be trapped.
  2. **The GUI must not crash.**
  3. *Do not worry about overflow or underflow.*
  4. The GUI will not be tested with extreme numbers.
- **There is no memory.** If the GUI is restarted all previous calculations are lost.