**Class Project CS342 Fall Semester 2013 due December 03, 2013, 11:00am.**
**Turn in the printed copy with Screen shots of sample run as discussed in the class.**
**Turn in the one jar file of the whole project executable as applet on Bb for grader to grade.**

**Project:** Write a program using new threads (Do not use old threads to get credit for the project) as discussed in the class for Algorithm Visualization. Your program should be menu driven as seen in the demos and previous assignments. Your program uses JInternalFrames (full functionality) and JPanels for proper display of the program components to do all the work. The JInternalFrames and JPanels may be placed in JApplet and/or JFrame. The program should be executable as an applet or an application. Use package command for package the appropriate files. Jar the whole package. Document the code using java style code. Every class, function and variable has javadoc descriptive comments. Make your program as fancy as you can.

The program implements sorting algorithms at least Bubble, Insertion, Selection, Quicksort, Heapsort, Shellsort. Extra credit for additional sorting algorithms. Program should be able to display at most three of the sorting algorithms simultaneously in executing in parallel for visual comparison. Scale the displays depending on the number of algorithms displayed. Data consists at most 800 numerical values in the range [1, 100].

Draw vertical-rectangle bar chart showing the numbers in the original data. For running multiple algorithms in parallel, draw bar charts side-by-side showing the numbers at each step of the selected sorting algorithms. Color code the bars corressponding to the exchanged data items. A titled border may be used to identify which algorithms are executing.

Use menu items to select algorithms and for various options for data, reset algorithms, reset data, speed control for each algorithm.

Use a slider with border to interactively input a count n of numbers to be sorted.  The program generates these n numbers randomly, upper bound on the numbers to be sorted is 800.  Let the min and max value of slider be 1 and 800. Border of the slider shows the count of numbers used to sort in the experiment.

Use Slider to control the speed of display. Slider shows the sleep time in the titled border

Use Button to start the simulation.

Use menuItem to select data version: Random, Best, Worst case data.

Use JCheckBoxMenuItems to select the algorithms at the same time.

You may use GridLayout to place drawPanels with vertical bar barchart of original numbers and sorting threads for the algorithms.

Use a Reset menu item to restart, reselect algorithms.

**Menus**

About

       Author,

       Problem,

       References:  Here you will specify what is referenced References should give complete citation to anything which is not your original. In this internal frame, write a statement that this is your original code if it is so.  If you borrowed any segment of code development, give reference (citation with complete info to the reader) to avoid plagiarism.

       Help: How to use the program. Any description of algorithms.

Demos

       Algorithms

          AlgorithmSelectionItems

          Unselect

       Data

          DataTypeItemes

       Main display

          SlidersToControl speed/input number,

          ButtonToStart

          ButtonToReset the data/speed

**Note.** The grader will grade it only if  the • Program is uploaded by due date and time. • File names are correct. • Program is your original code • Program is complete with package command in source files and correct link  • Fully document it with javadoc style comments • Program is uploaded with jar file containing all program files, no other files. Extra points will be taken off for not following the guidelines. The program should not contain any commented code. Any code not needed should not be part of submission.  You can do it in groups of at most three. I will need to know the grouping tomorrow.

**Grading Criteria**
**In office Presentation** and Demonstration of the project.
In the author frame, state code originality, your individual contribution, what did you learnt in this project.
Each member in the group should be able to present whole project.
Grading will be based on Implementation and Documentation, Presentation and Demonstration.

Implementation and Documentation          70%
Presentation and Demonstration          30%

Project  Implementation and Documentation
     70% of  the following distribution

I IMPLEMENTATION                                        (50%)
          Completeness                                        [15]
          Correct Working                                        [20]
          **UserFriendliness**:  Include author, problem, help, references in menus, use the skills learnt in this course,
                    e.g. Menus, ToolBars, Internal Frames, MainFrame, Images, Audios, MultiTasking, good use of
                    display area.                                        [10]
          Objected Oriented Design, the driver program must be named ***Demo.java***
                                                       [5]

II USER INTERFACE                                        (15%)
          **Delivery**:  Demo.jar.  Program can be used as applet and application.
          The Demo.jar must includes (whatever is applicable)
                    directories:
                              audios
                              code
                              docs
                              images
                              source
                    files:
                              m.txt
                              Demo.html
                              PPT presentation

III Users Manual  -- handout                                        (35%)
          Printed copy of the source code          [5]
          Printed copy of the demo powerpoint
          Sample run with screen shots                    [5]

          **Author** - phone number, email address, contribution, URL for project files. Originality, contribution,
                    learning from the project.
                                                       [5]
          **Problem Description**– descibe the problem you are solving,  your method of solution, any problems
                    encountered, document the format of images and format of audios you used in the project
                                                       [5]
          **References** Anything used in your program code but Not Discussed in the class must be clearly described.
                    Give complete references in your code to anything beyond the class lectures.  References are used
                    to help develop your code and help the reader to get to the resources.  It does not mean that you
                    have permission to copy the code from an external source.
                                                       [5]
          **Sample run and Documentation**:  Screen shots of program execution and Java doc documentation.
                                                       [10]

8. Have the ability to use current techniques, skills and tools necessary for computing practice and be
     able to apply or evaluate a computer-based system to solve a problem. (ABET Criterion 3: c, i)

|  | The program is | The program | The program | The program |
| --- | --- | --- | --- | --- |

| Project | producing incorrect results. Does not meet specifications, incorrect GUI. Not turned in on time. Or No submission. Does not compile. | has correct GUI, does not meet specifications produces incorrect results. | works, correct GUI, produces correct results. It does not meet all the requirements | works correctly and meets all of the specifications. The GUI is correct, meets all of the specifications. |
|---|---|---|---|---|
| Programming | Inadequate | Need help | Adequate | Excellent |
| Score Received | | | | |

**Criteria:**

**Summary assessment**:
**Assessment done by:** _____

**Grading Script**

```sh
#!/bin/sh
# Demo Grading Script
# For grading, I use Demo.jar file to check its contents
# expand the jar file
jar -xf Demo.jar

#move original .jar file
mv Demo.jar DemoOld.jar

#remove old code
rm -r code
rm -r docs

# create new code
#Compile
javac -d .  source/*.java

# Create javadoc documentation
javadoc -d docs -version -author -private -quiet source/*.java

# Next, to create jar file, do not forget *,
jar -cmf m.txt Demo.jar *

# execute the application from commandline
#java -jar Demo.jar

# execute the applet from appletviewer, you are on commandline anyway
# I need only Demo.jar, I will have my own Demoj.html
# <applet code = code.Demo archive=Demo.jar width=500 height=500></applet>
appletviewer Demoj.html
```