

CISC 3120 Fall 2012 Extra Credit Homework

Instructions

- This assignment **is due by midnight on December 23rd** and you will submit this online — instructions below:
 1. Create a ZIP archive from the files that make up your homework. (Instructions on what exactly you should put in this archive are given below).
 2. Go to:
`http://agents.sci.brooklyn.cuny.edu/parsons/cisc3120/`
 3. Login and submit your ZIP archive.
 4. You can submit as many times as you like before the deadline—only the final version of the homework will be graded.
 5. Failure to follow these instructions will result in points being taken away from your grade. The number of points will be in proportion to the extent to which you did not follow instructions... (since it can make it a lot harder for me to grade your work — grrrr!)
 6. Note that this deadline is a hard deadline. If you submit after the deadline, your work will not be marked.

1 Overview

This homework is an **extra credit** exercise — you do not need to complete it in order to complete the course.

This homework extends two of the previous homeworks:

1. The photo editor from Homework 7.
2. The simple chat server from Homework 9.

If you did not complete those earlier homeworks, this homework also gives you the opportunity to gain some credit that you missed out on.

2 Photo Editor

1. The starting point for this part of the exercise is the Photo Editor from Homework 7.
2. As indicated above, if you did not submit versions of these programs for Homework 7, or you submitted versions of those programs without all the required functionality, this is a chance to make up the shortfall. Add the functionality to the version of the program you submit for this homework and create a text file (not an RTF file or a .docx) `README-photo` which explains the differences between the new versions and what you submitted for Homework 7. `README-photo` should be included in the ZIP file of your submission.
3. If you already submitted a full version of these programs for Homework 7, a statement to this effect should also be included in `README-photo`.
4. The Photo Editor for Homework 7 was a mock-up of a real editor — it was supposed to have a placeholder for a gallery of images, and placeholders for 3 operations that were performed on the images.
5. This homework is to write code that carried out those operations for real.

6. In particular, you can use the `BufferedImage` object to manipulate images.
7. Make sure that you document the functionality that you implement in the Help facility of your program.

3 Secure chat

1. The starting point for this part of the exercise is the GUI Chat program from Homework 9. In particular, two Java programs
 - `ChatServer.java`
 - `GUIChat.java`
2. Details of the required functionality of those programs can be found in the description of Homework 9.
3. As indicated above, if you did not submit versions of these programs for Homework 9, or you submitted versions of those programs without all the required functionality, this is a chance to make up the shortfall. Add the functionality to the version of the program you submit for this homework and create a text file (not an RTF file or a .docx) `README-chat` which explains the differences between the new versions and what you submitted for Homework 9. `README-chat` should be included in the ZIP file of your submission.
4. If you already submitted a full version of these programs for Homework 9, a statement to this effect should also be included in `README-chat`.
5. You need to modify `ChatServer.java` and `GUIChat.java` to encrypt the text that they transmit, and to decrypt it so that it can be read after transmission.
6. The program `SecretWriting` provides the means for doing encryption, so your job is to integrate this functionality into the two parts of the chat program — both the client part (the bit with the GUI) and the server part will need to be able to handle both encryption and decryption.
7. Both parts of the program should display:
 - The plain text that is typed in.
 - The encrypted text that is generated from this plain text.
 - The decrypted text that is received from the other part.
 - The decrypted version of the text that is received.
8. As before, you would be well advised to test your program running the server part on the department machine so that you can be sure it works over the network.
9. You should list all the .java files used by your program in `README-chat`, and identify which of these files include `main()`.
10. This description has assumed you started from the basic version of the chat program from Homework 9. If you completed the extra credit version of that homework (which had two `GUIClient`s communicating through a `ChatServer`), you can add encryption/decryption to that version of the program. If you do this, document it in `README-chat`.

4 What to submit

The ZIP file you submit should include:

1. A folder Photo which contains all the .java files from the Photo Editor program along with the file README-photo.
2. A folder Chat which contains all the .java files from the Secure Chat program along with the file README-chat.

5 How I will mark your work

1. You will get credit for each of the things listed above that you have implemented and which work as I have described them.
2. You will also get credit for the **way** in which you do this. That is programs that do not adhere to what I consider to be good programming style will not gain as much credit.
You can get an idea of what I consider to be good style by looking at the code I have given you.
3. To get full credit you will also have to document your code. Again looking at my code will give you an idea of what I am looking for.
4. Also note that your program **must compile and run** for you to get any credit at all.
(At this point in your computer science career it is unacceptable to submit programs that don't compile and run).
Bear in mind that I will be using Java version 1.6 to assess whether your code compiles and runs.
5. Thus it is better for you to submit a program that works and contains less functionality than it is to submit a program with more functionality that does not work.