* Turn in pseudo-code for each of the methods specified in CryptoManager.java. Your pseudo-code should be part-way between English and java. There is no need to spell out all the details of variable declaration, etc., but by the same token, the pseudo-code needs to have enough detail that a competent Java programmer could implement it.
* Turn in a test table with
* at least two tests for the Caesar Cipher
* at least two tests for the Bellaso Cipher
* at least one string that will fail because it has characters outside the acceptable ones.
* Learning Experience: highlight your lessons learned and learning experience from working on this project.
* What have you learned? I learned new things about deciphering messages, and it was very interesting.
* What did you struggle with? I had difficulty with the GUI. GUI was not compiling so I had to change the method headers and fix the formats.
* What will you do differently on your next project? I have to start working on the projects earlier so that I don’t get stressed.
* Include what parts of the project you were successful at, and what parts (if any) you were not successful at. I was successful with almost everything. I had problem with GUI and I got some errors in the Junit test but I fixed the problems.
* GitHub: In your repository (see Lab 1), upload the files initially provided in Blackboard for the project. When you are finished with the design and programming, upload your Word file and java file. You will want to upload these files as contents of a directory so that future uploads can be kept separate. Take and submit a screen shot of the GitHub repository.

**Notes:**

* Proper naming conventions: All constants, except 0 and 1, should be named. Constant names should be all upper-case, variable names should begin in lower case, but subsequent words should be in title case. Variable and method names should be descriptive of the role of the variable or method. Single letter names should be avoided.
* Documentation: The documentation requirement for all programming projects is one block comment at the top of the program containing the course name, the project number, your name, the date and platform/compiler that you used to develop the project. If you use any code or specific algorithms that you did not create, a reference to its source should be made in the appropriate comment block. Additional comments should be provided as necessary to clarify the program.

Indentation: It must be consistent throughout the program and must reflect the control structure

**Deliverables / Submissions and Deliverable format:**

The Java application must compile and run correctly, otherwise Project grade will be 0.

The deliverables will be packaged as follows. Two compressed files in the following formats:

* FirstInitialLastName\_Assignment1\_Complete.zip, a compressed file in the zip format, with the following:
* Source Code: Java Files
* Word document with a name FirstInitialLastName\_Assignment3.docx should include:
* Pseudocode for each of the methods specified in CryptoManager.java.
* Test Plan
* Screen snapshots of outputs from Eclipse based on your Test Plan
* Screen snapshots of Junit Test for each method
* Screen snapshot of GitHub submission
* Lessons Learned
* Check List
* A zip file will only contain the .java files and will be named: FirstInitialLastName\_Assignment3\_Moss.zip. **This .zip will not have any folders in it – only .java files.**

**Pseudocode:**

Declare the LowerBound and UpperBound.

If plainText string length is not within allowable bounds show, return false.

Return true if all character are within allowable bounds.

For loop to add key for encryption and return result.

For loop to subtract key for decryption and return result.

If the key is greater than higher bound subtract UpperBound to LowerBound and add 1.

If characters in bellasoStr less than plaintext character, repeat characters until its equal to plaintext character length.

For encryption create for loop to add first character from bellasoStr and plaintext and return result.

If its more than the upperlimit subtract (UpperBound to LowerBound and add 1) from result.

For decryption create for loop to subtract first character from bellasoStr and plaintext and return result.

**Test Plan**

Test your program with at least five test cases. **Make sure your tests cover all the possible scenarios.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Input text | Input Key | Encrypted (method1) | Encrypted (method2) | Decrypt (method1) | Decrypt (method2) |
| CDF | 1 | DEG |  | CDF |  |
| EFG | 2 | GHI |  | EFG |  |
| #$&% | CD | CDCD | &()) | CDCD | #$&% |
| ,-./ | AB | ABAB | -//1 | ABAB | ,-./ |
| JLY | 198 | PR\_ |  | JLY |  |

**Make sure your tests cover all the possible scenarios.**

Assignment 3 Check List

|  |  |  |  |
| --- | --- | --- | --- |
| **#** |  | **Y/N** | **Comments** |
|  | **Assignment files:** |  |  |
|  | * FirstInitialLastName\_ Assignment#\_Moss.zip | **<Yes or No>** | **Yes** |
|  | * FirstInitialLastName\_Assignment#.docx/.pdf | **<Yes or No>** | **Yes** |
|  | * Source java files | **<Yes or No>** | **Yes** |
|  | **Program compiles** | **<Yes or No>** | **Yes** |
|  | **Program runs with desired outputs related to a Test Plan** | **<Yes or No>** | **Yes** |
|  | **Documentation file:** |  |  |
|  | * Comprehensive Test Plan | **<Yes or No>** | **Yes** |
|  | * Screenshots for each Test case listed in the Test Plan | **<Yes or No>** | **Yes** |
|  | * Screenshots of your GitHub account with submitted Assignment# (if required) | **<Yes or No or N/A>** | **Yes** |
|  | * UML Diagram (if required) | **<Yes or No or N/A>** |  |
|  | * Algorithms/Pseudocode (if required) | **<Yes or No or N/A>** |  |
|  | * Flowchart (if required) | **<Yes or No or N/A>** |  |
|  | * Lessons Learned | **<Yes or No>** | **Yes** |
|  | * Checklist is completed and included in the Documentation | **<Yes or No>** | **Yes** |

**Grading Rubric**

|  |  |  |
| --- | --- | --- |
| **CMSC203 Grading Rubric - Assignment 3** | **Possible total grade:** | **100** |
|  |  |  |
| **Name** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
|  |  |  |
| **TESTING** |  |  |
|  | Project must compile. If it doesn't compile | 0 |
|  | Project must run. If it's run time error | 0 |
|  | Passes Public JUnit tests and running project tests | 25 |
|  | Passes private instructor JUnit tests and running project tests | 75 |
|  | **Possible Sub-total** | **100** |
|  |  |  |
| **REQUIREMENTS (Subtracts from TESTING total)** |  |  |
| **Documentation:** |  |  |
| Documentation within source code is missing or incorrect |  | -10 |
|  | Description of what class does is missing |  |
|  | Author’s Name is missing |  |
|  | Methods not commented |  |
|  | Additional Comments to clarify a code inside a program are missing |  |
|  | Header Comments are missing |  |
| MOSS files were missing |  | -5 |
| Screenshots of at least two tests for the Caesar Cipher are missing |  | -5 |
| Screenshots of at least two tests for the Bellaso Cipher are missing |  | -5 |
| Screenshots of test cases based on Test Plan are missing |  | -5 |
| Screenshot of GitHub with uploaded Assignment 3 Files is missing |  | -5 |
| Pseudocode for each method specified in CryptoManager.java is missing |  | -15 |
| Test Plan is missing |  | -10 |
| Assignment 3 Checklist is missing |  | -5 |
| Learning Experience |  | -5 |
|  | Highlight your lessons learned and learning experience |  |
|  | from working on this project. What have you learned? What did you struggle with? |  |
|  | What would you do differently on your next project? What parts of the project were |  |
|  | you successful with, and what parts (if any) were you not successful with? |  |
| **Programming Style:** |  |  |
| Incorrect use of indentation, naming convention, etc. (see coding/style standards) |  | -15 |
| **Design:** |  |  |
|  | Data Manager – CryptoManager | -20 |
|  | Methods do not follow provided requirements |  |
|  |  |  |
| **Deliverables** |  |  |
|  | Files are submitted as compressed files using the format explained in assignment | -5 |
|  | **Possible decrements:** | **-100** |
|  |  |  |
|  | **Possible total grade:** | **100** |