CA2 – AES

Encrypt/Decrypt Java Application   
Value: 30%

Completion & Upload Date: Monday 2nd December 9am

Objectives

Build an application that will encrypt and decrypt a file **using AES encryption**.

Task 1: Create a Menu System

Create an initial menu system that will allow the user to:

1. Encrypt a File (Task 2)
2. Decrypt a File (Task 3)
3. Quit the application

After each task is finished the menu should be presented to the user until the user specifies the quit the application menu option.

Task 2: Encrypt a File

When the user selects encrypt a file, they should be prompted to enter the filename. The application should generate a random key to encrypt the file. The application should then encrypt the file using the random key. The encrypted data should be written to a file ciphertext.txt, the key should be written to the screen along with a notice regarding the encrypted text file.

Task 3: Decrypt a File

When the user selects decrypt a file, they should be prompted to enter the filename. Then they should be prompted to enter a valid key. The application should then decrypt the file using the key. The decrypted data should be written to a file plaintext.txt and a notice regarding the location of the decrypted data should be written to the screen

Additional Restrictions and Guidance

The solution must include appropriate exception handling and validation of external input. You must gracefully handle a scenario where a user enters an invalid menu option, invalid filename, or invalid key. Your application should not terminate abruptly.

Assessment Criteria

1. Use of exception handling and validation of external input (20%)
2. Implementation of AES encryption process (25%)
3. Implementation of AES decryption process (25%)
4. Code quality, abstraction, and reuse (20%)
5. Presentation and professionalism of terminal interface (10%)

## Submission Checklist

Make sure to submit all the following to Moodle or your submission will not be marked:

|  |  |
| --- | --- |
| **GitHub Link**  Regular commits, relevant commit messages, minimum 2 branches  Must be either public, or have invited your lecturer as a collaborator |  |
| **Screencast**  5 minutes **max** – talk through a running demo of your project including any extra functionality, point out relevant code |  |
| **Coversheet**  Signed |  |
| **Zipped project**  Include everything needed to run the project, including any instructions, test files, passwords, etc, |  |

## Academic Integrity

The assignment must be entirely the work of each student – in your own words. Students are not permitted to share any pseudocode or source code from their solution with any other student in the class. Students may not distribute the source code of their solution to any other student in any format (i.e., electronic, verbal, or hardcopy transmission). Any suspected plagiarism will be investigated, pursued, and reported to the Plagiarism Committee.

Generative artificial intelligence (AI) tools cannot be used in this assessment task. In this assessment, **you must not use** generative artificial intelligence (AI) (ChatGPT, ChatSonic, Bing Chat, Lex, DALL-E 2, or other tools) to generate any materials or content in relation to the assessment task.

The DkIT Academic Integrity Policy and Procedures, <https://www.dkit.ie/about-dkit/policies-and-guidelines/academic-policies.html>) states the following:

“Using generative artificial intelligence tools (e.g. ChatGPT) in an assessment unless explicitly permitted to do so and with proper acknowledgement, is a form of plagiarism”.

## Late Submissions

The institute’s standard policy on marking of late submissions will be applied:

* Submissions received up to 1 week late will have a 20% grade reduction applied.
* Submissions received up to 2 weeks late will have the above penalty applied and will be capped at 40%.
* Submissions received more than 2 weeks late cannot be graded.