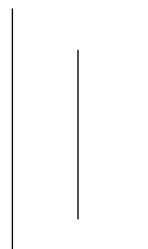




Introductory Programming And Problem Solving

**Encrypts and Decrypts text
using the Caesar Cipher**

Individual Coursework



Student ID: 2417335

Student Name: Nima Norbu Sherpa

Word Count: 622

Contents

| | |
|--|----------|
| 1. Challenge's Faced in the coursework..... | 1 |
| 2. Approach to Completing the Task | 1 |
| 3. Lesson Learned..... | 1 |
| 4. Conclusion | 2 |

Abstract

This report covers the difficulties encountered during the Caesar Cipher effort, as well as the approach and lessons learned. The key difficulty was keeping the code structure clear while allowing for smooth console and file inputs. The method used to investigate requirements, utilizing modular design, putting in place an intuitive console interface, and having reliable error management. The lessons covered included file input complexities, efficient error management, code organization, and the value of program adaptability.

The course improved students' ability to solve problems and deepened their understanding of error management and user-centric design. Gaining a comprehensive understanding of programming principles and improving coding proficiency were achieved by putting theoretical knowledge of Python to use in real-world circumstances.

Reflective Report

Student Name: Nima Norbu Sherpa

Student ID: 2417335

1. Challenge's Faced in the coursework

The primary challenge in our class assignment was creating a Caesar Cipher application that is user-friendly and capable of efficiently handling files and user inputs. The challenging aspect was structuring the code logically for both scenarios while maintaining readability. To ensure that the application functions consistently, we had to exercise caution while addressing mistakes such as incorrect inputs or filenames.

Keeping the code organized while maintaining program flexibility was a key objective. This required us to design with consideration. It taught us a lot about proper coding techniques and program design how to deal with mistakes and work with various input kinds.

2. Approach to Completing the Task

I started the project by understanding what the assignment needed and focused on creating a Caesar Cipher program that can both encrypt and decrypt messages. To make the code easy to read and maintain, I organized it into different parts, like dealing with files, getting input from users, and handling encryption and decryption. Making it simple for people to use was important, so I created a clear console interface. It helps users understand how to use the program, whether they want to type in messages or use a file.

I also made sure the program can handle potential problems by adding checks for things like wrong choices, invalid inputs, and shift values. This way, the program is protected from causing issues when people use it, especially when they choose to input messages from a file.

3. Lesson Learned

I learned a lot from this class assignment in a few key areas:

Code Organization: I understood how important it is to organize and split the code into different parts. This makes the code easier to read and maintain in the long run.

Error Handling: I now realize how crucial it is to find and fix errors. By thinking about what users might do wrong and providing helpful error messages, we can make the program more reliable and user-friendly.

File Handling: I explored how to work with files, which added another layer to the project. I got better at using Python for reading from files and including this functionality in the program.

Flexibility: Creating a program that can adjust and adapt to different situations was a key goal. This means the application can handle various inputs and still work well.

4. Conclusion

This class allowed me to apply what I learned in my Python programming studies to real-life situations. Facing challenges during the project was tough but it really helped me get better at solving problems. The things I learned will be super useful as I continue with programming.

Besides just coding, doing this work taught me a lot about making programs that are easy for people to use and handling errors well. I now understand more about the core ideas in programming because I got to actually use Python in a practical way, making the theoretical stuff more important to me.