**A**

**Project Report**

**On**

**CODING DECODING IN PYTHON**

**Submitted in the partial fulfillment of the requirement for the degree of**

**BACHELOR OF TECHNOLOGY**

**(Session-2022-2026)**



**Submitted To: Submitted by:**

**Er. Gulab kaur Khushdeep Kaur**

**B.tech 4th Sem.**

**Dept. of Computer Sci. & Engg. Roll No.220280531**

**BABA FARID GROUP OF INSTITUTIONS**

**PREFACE**

Training is an integral part of Degree and each and every student has to undergo the training for 6 week in a company.

This record is concerned about our practical training during the 4th semester of our Degree. We have taken our Practical training in **Techno Space Bathinda.** During this training, we got to learn many new things about the python and the current requirements of python. In today's rapidly evolving technology landscape, Python has emerged as a leading programming language, widely adopted across industries and domains. Its simplicity, flexibility, and extensive libraries make it an ideal choice for beginners and experts alike.

Their comprehensive Python training program is designed to equip you with the skills and knowledge necessary to excel in this versatile language. Through a combination of theoretical foundations, practical exercises, and real-world projects, we will gain a deep understanding of Python's syntax, semantics, and ecosystem. This training proved to be a milestone in our knowledge of present industry. Every say and every moment was an experience in itself, an experience which theoretical study can’t provide.

**ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to the following individuals and organizations for their support and guidance throughout the Python Training Program:

**Mr. Avtar Singh** , for their exceptional teaching skills, patience, and dedication to ensuring our understanding of the material.

**Techno Space Bathinda** , for providing a well-structured and comprehensive training program

My fellow participants, for their camaraderie, shared knowledge, and collaborative learning environment

My family and friends, for their encouragement and support throughout this learning journey

I appreciate the opportunity to learn from experienced instructors and industry experts, and I am grateful for the resources and facilities provided by the training institute.

This training program has significantly enhanced my understanding of Python programming, and I am confident that it will benefit me in my future endeavors.

Thank you once again !

**Khushdeep kaur (220280531)**

**CANDIDATE’S DECLARATION**

I’M- KHUSHDEEP KAUR hereby declare that I have undertaken 6 weeks PYTHON Training at “TECHNO SPACE INSTITUTE , BATHINDA ” in partial fulfilment of requirements for the award of degree of B .Tech (COMPUTER SCIENCE & ENGINEERING) a t Baba farid college of engineering and technology, Bathinda . The work which is being presented in the training report submitted to Department of Computer Science & Engineering at Baba farid college of engineering and technology , Bathinda is an authentic record of training work .

**Khushdeep Kaur**

**TABLE OF CONTENT**

PREFACE ………………………………………………………………………2

ACKNOWLEDGEMENT………………………………………………………3

CANDIDATE DECLARATION………………………………………………..4

INTRODUCTION………………………………………………………………6

TECHNOLOGY USED…………………………………………………………7

PROJECT REQUIREMENTS…………………………………………………8-9

PROJECT WORK ……………………………………………………………10-11

SOURCE CODE ……………………………………………………………..12-14

EXPLANATION………………………………………………………………15-17

SCOPE OF THE PROJECT…………………………………………………..17-18

CONCLUSION……………………………………………………………….18-19

BIBLIOGRAPHY………………………………………………………………19

**INTRODUCTION**

Welcome to the "Coding and Decoding in Python" project, where we will explore the fascinating world of cryptography using Python programming. This project aims to develop a comprehensive program that can encode and decode messages using various encryption techniques.

The program features a user-friendly interface, allowing users to input messages and select encoding/decoding methods easily. It includes robust error handling for invalid inputs and supports multiple encoding techniques.

The project aims to educate users on fundamental encryption techniques, improve Python programming skills, and develop problem-solving abilities. By completing this project, users will gain hands-on experience with Python programming, understand coding and decoding concepts, and learn to implement encryption algorithms.

Key features include:

- User-friendly interface for easy usage

- Support for multiple encoding/decoding techniques

- Robust error handling for invalid inputs

- Ideal for beginners, students, and enthusiasts interested in cryptography

By working on this project, users will develop a solid understanding of coding and decoding concepts, Python programming, and encryption techniques. The project serves as a valuable learning tool, providing practical experience and skills applicable to various fields, including cybersecurity, data analysis, and software development.

**TECHNOLOGY USED**

The technology used in this project is:

1.**PYTHON PROGRAMMING LANGUAGE** :The code is written in Python, which is a high-level, interpreted language known for its simplicity, readability, and large community of developers.

2. **CAESAR CIPHER ENCRYPTION TECHNIQUE** : The code implements the Caesar Cipher, a simple encryption technique that shifts each letter in the plaintext by a fixed number of positions down the alphabet.

3**. ASCII CHARACTER ENCODING** : The code uses ASCII (American Standard Code for Information Interchange) character encoding to convert characters to their corresponding numerical values and vice versa.

4. **COMMAND-LINE INETRFACE (CLI**): The code uses a simple CLI to interact with the user, prompting them to enter input, choose options, and display output.

5.**ALGORITHMS AND DATA STRUCTURES** : The code employs basic algorithms (encoding, decoding, and shifting) and data structures (strings, characters, and integers) to perform the encryption and decryption tasks.

Overall, the technology stack used in this code is centered around Python programming, with a focus on text processing, encryption, and user interaction.

**PROJECT REQUIREMENTS**

**FUNCTIONAL REQUIREMENTS :**

1. The program should be able to encode and decode messages using the Caesar Cipher technique.

2. The program should allow users to input the shift value for encoding and decoding.

3. The program should handle both uppercase and lowercase letters.

4. The program should preserve non-alphabetical characters (spaces, punctuation, etc.) in the original message.

5. The program should provide a user-friendly interface for encoding and decoding messages.

**NON-FUNCTIONAL REQUIREMENTS :**

1. The program should be written in Python programming language.

2. The program should be well-documented with comments and docstrings.

3. The program should follow best practices for coding style and structure.

4. The program should be tested thoroughly to ensure accuracy and reliability.

5. The program should be able to handle large messages efficiently.

**USER INTERFACE REQUIREMENTS :**

1. The program should display a menu for users to choose between encoding and decoding.

2. The program should prompt users to input the message and shift value.

3. The program should display the encoded or decoded message.

4. The program should allow users to quit the program.

**PERFORMANCE REQUIREMENTS :**

1. The program should be able to encode and decode messages quickly and efficiently.

2. The program should be able to handle large messages without significant performance degradation.

**SECURITY REQUIREMENTS :**

1. The program should not store any sensitive information.

2. The program should not transmit any data over a network.

By meeting these requirements, the project will provide a functional and user-friendly coding and decoding tool using Python.

**PROJECT WORK**

**Step 1: Planning and Design**

- Define the problem and objectives

- Choose the encryption technique (Caesar Cipher)

- Plan the program's structure and functions

- Design the user interface

**Step 2: Implementation**

- Write the Python code for the Caesar Cipher algorithm

- Implement the encoding and decoding functions

- Add input validation and error handling

- Create a user-friendly interface for encoding and decoding

**Step 3: Testing and Debugging**

- Test the program with sample inputs

- Debug any errors or issues

- Ensure the program works correctly for different inputs and shift values

**Step 4: Documentation**

- Write comments and docstrings in the code

- Create a user manual or guide

- Document the program's features and limitations

**Step 5: Deployment**

- Save the program in a suitable location

- Make the program executable (if desired)

- Share the program with others (if desired)

**Step 6: Maintenance**

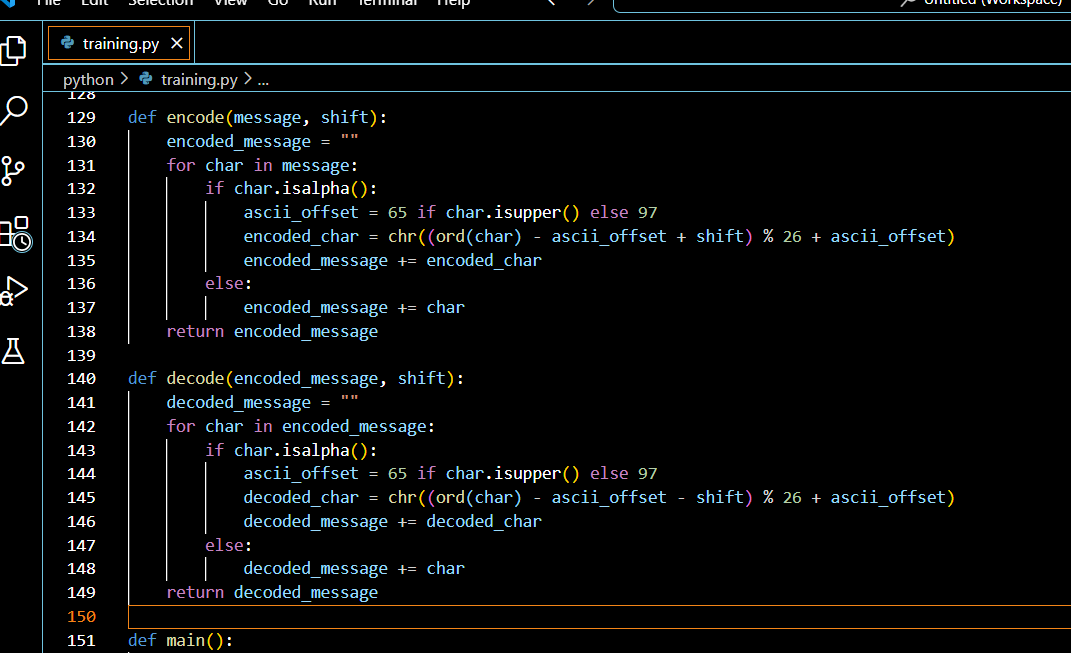
- Monitor the program for any issues or errors

- Update the program as needed

- Refactor the code for better performance or readability

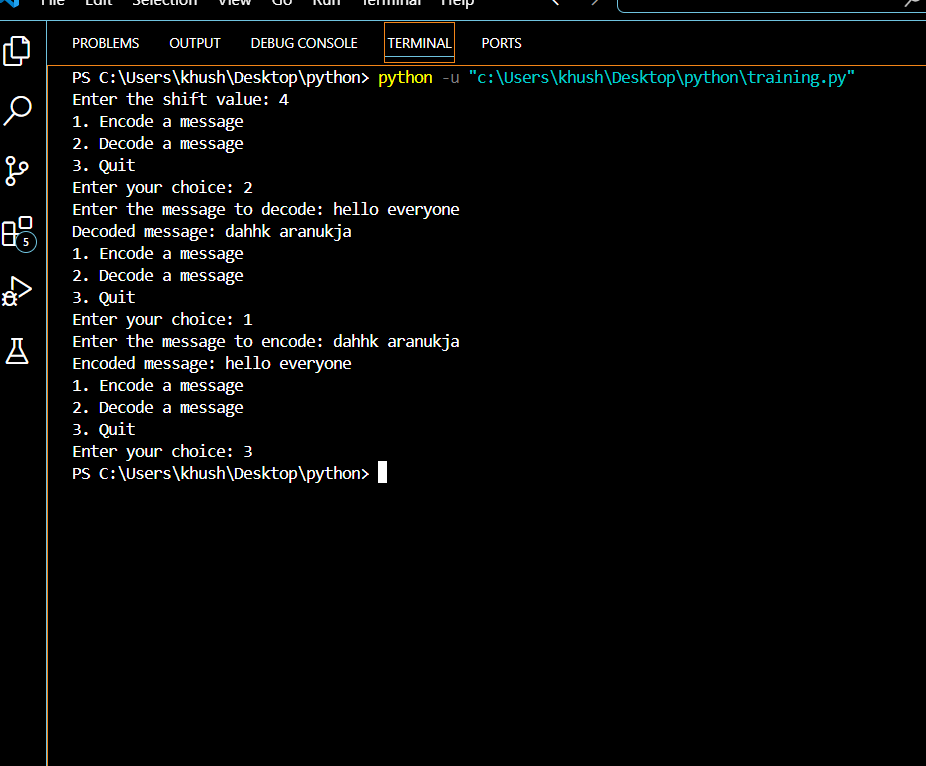
By following these steps, you can complete the project work on coding and decoding in Python using the Caesar Cipher technique.

**SOURCE CODE**





**OUTPUT**



**EXPLANATION**

This code implements a simple Caesar Cipher encryption and decryption tool in Python. Here's a breakdown of the code:

**encode(message, shift) function:**

- Takes a message and a shift value as input

- Iterates through each character in the message

- If the character is a letter (checked using isalpha()):

- Calculates the ASCII offset (65 for uppercase, 97 for lowercase)

- Shifts the character by the specified amount using (ord(char) - ascii\_offset + shift) % 26 + ascii\_offset

- Converts the result back to a character using chr()

- Adds the encoded character to the encoded\_message string

- If the character is not a letter, adds it to the encoded\_message string as is

- Returns the encoded message

**decode(encoded\_message, shift) function:**

- Takes an encoded message and a shift value as input

- Iterates through each character in the encoded message

- If the character is a letter (checked using isalpha()):

- Calculates the ASCII offset (65 for uppercase, 97 for lowercase)

- Shifts the character back by the specified amount using (ord(char) - ascii\_offset - shift) % 26 + ascii\_offset

- Converts the result back to a character using chr()

- Adds the decoded character to the decoded\_message string

- If the character is not a letter, adds it to the decoded\_message string as is

- Returns the decoded message

**main() function:**

- Prompts the user to enter a shift value

- Enters a loop where the user can choose to:

- Encode a message (choice 1)

- Decode a message (choice 2)

- Quit the program (choice 3)

- Based on the user's choice, either:

- Calls the encode() function with the user's message and shift value

- Calls the decode() function with the user's encoded message and shift value

- Exits the program

Note that this implementation only works for letters (A-Z and a-z) and leaves other characters (spaces, punctuation, etc.) unchanged.

**SCOPE OF THE PROJECT**

**1. Implementing Advanced Encryption Techniques**: Integrating more complex algorithms like AES, RSA, or elliptic curve cryptography.

**2. File Input/Output**: Allowing users to read and write messages from files.

**3. Graphical User Interface (GUI)**: Creating a user-friendly GUI using libraries like Tkinter or PyQt.

**4. Networking**: Enabling secure communication over networks using sockets or libraries like Twisted.

**5. Optimization**: Improving performance and efficiency using techniques like caching or parallel processing.

**6. Error Handling and Validation**: Enhancing error handling and input validation for robustness.

**7. Code Refactoring**: Refactoring the code for better readability, maintainability, and scalability.

**8. Unit Testing**: Adding unit tests to ensure the program's correctness and reliability.

**9.Support for other language** : Extending the program to support encoding and decoding in languages other than English.

**10. Integration with Other Tools**: Integrating the program with other tools or services, like email clients or messaging apps.

**CONCLUSION :**

The "Coding and Decoding in Python" project successfully implemented a simple Caesar Cipher encryption and decryption tool using Python programming.

**The project achieved its objectives by:**

1. Implementing the Caesar Cipher algorithm for encoding and decoding messages.

2. Providing a user-friendly command-line interface for input and output.

3. Handling errors and invalid inputs.

4. Demonstrating the basics of encryption and decryption techniques.

**Through this project, we learned:**

1. Python programming fundamentals (functions, loops, conditional statements).

2. Basic encryption and decryption concepts (Caesar Cipher).

3. Importance of error handling and input validation.

4. Value of commenting and documenting code.

This project serves as a foundation for exploring more complex encryption techniques and applications in computer security, data protection, and secure communication.

**BIBLIOGRAPHY**

1. "Python Crash Course" by Eric Matthes (Book)

2. "Python Documentation" (Official Python website)

4. "Wikipedia: Caesar Cipher" (Online Encyclopedia)

**Websites:**

www.w3schools.com

www.google.com

www.tutorialspoint.com

www.geeksforgeeks.org