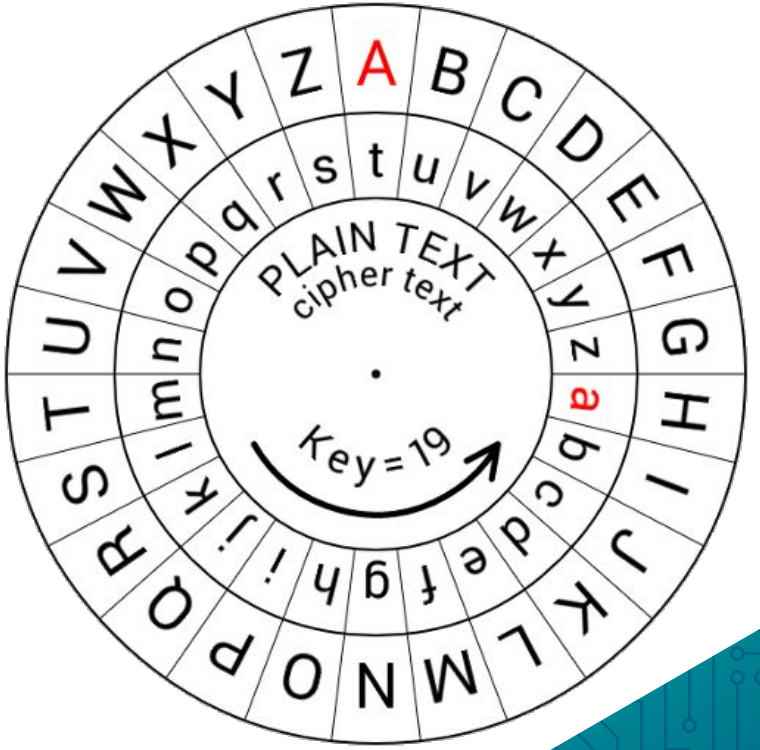
Caesar Cyphering

By: Joseph S. , Raul G

INTRODUCTION

- Developed in 100 B.C.
- Caesar cyphering was created by Julius Caesar
- Served the purpose of sending secret messages to his generals in the field



1

Object & Goals

Objects & Goals

- Using ‘key’ to shift letters and numbers to be able to decipher the phrase
- For special characters have them remain the same
- Have the cyphered and decyphered phrase shown on the GUI

2

Implementation

Implementation

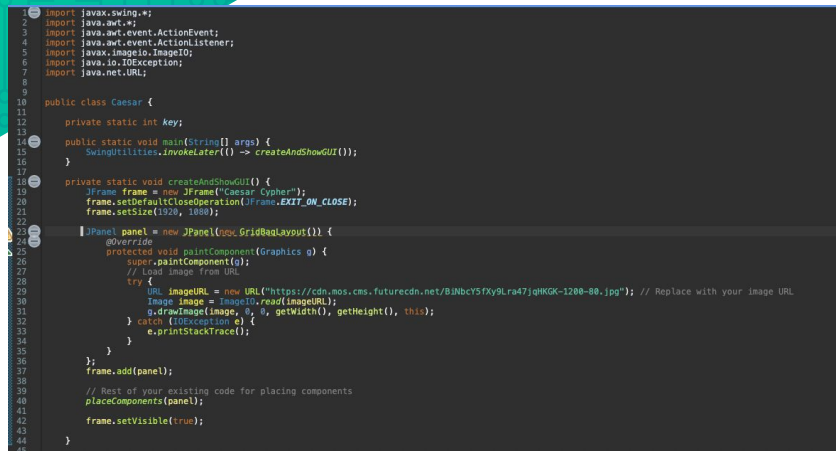
Original C code from CS50x

```c
#include <stdlib.h>
#include <cs50.h>
#include <stdio.h>
#include <ctype.h>
#include <string.h>
char rotate(char c, int n);

int main(int argc, string argv[])
{
    int i, stringtoint, k, cipherlower, cipherupper; //Declaration of variables
    char space; // space variable
    char specialchar; //variable for any other character besides alphabetical or space

    if (argc != 2)
    {
        return 1;
    }

    if (argc == 2) // makes sure the key is only at index 1
    {
        for (i = 0 ; i < strlen(argv[1]) ; i++) // For loop to go through each character in string
        {
            if (!isdigit(argv[1][i])) // Checks if key is a digit
            {
                printf("Usage: ./caesar key\n"); // tells user the correct way of using the program

                break; // if the key is not a digit then it stops the loop
            }

        }

    }

    else
    {
    {
        printf("Usage: ./caesar key\n"); // tells user the correct way of using the program
    }

    stringtoint = atoi(argv[1]); // Converts string to integer

    string plaintext = get_string("plaintext: "); // prompts user for plaintext to encypher

    printf("ciphertext: ");

    for (k = 0; k < strlen(plaintext); k++) //Loop for encyphering
    {
        if (isalpha(plaintext[k]))  //Checks if the characters are in the alphabet
        {
        if (isupper(plaintext[k]))
            {
                cipherupper = (((plaintext[k] - 65 + stringtoint) % 26) + 65) ; // rotating algorithm for upperc
                printf("%c", cipherupper);
            }

            else if (islower(plaintext[k]))
            {
                cipherlower = (((plaintext[k] - 97 + stringtoint) % 26) + 97) ; // rotating algorithm for lowerc
                printf("%c", cipherlower);
            }
        }

        else if (isspace(plaintext[k])) //Checks if the character is a space
        {
            space = plaintext[k];
            printf("%c", space);
        }

        else // if it's not in the alphabet or is a space then it prints out whichever character was typed
        {
            specialchar = plaintext[k];
            printf("%c", specialchar);
        }

    }
    printf("\n"); // Moves 1 line down for formatting purposes
```

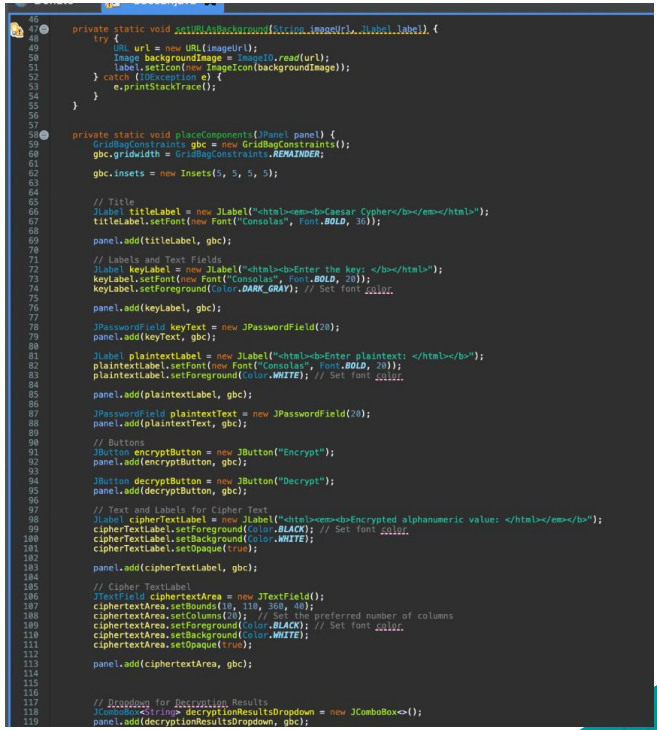
Java Code (Backend)

```java
        }
    });

    decryptButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            String ciphertext = ciphertextArea.getText();
            decryptAndDisplay(decryptionResultsDropdown, ciphertext);
        }
    });
}

private static void decryptAndDisplay(JComboBox<String> dropdown, String ciphertext) { //This is to display the decrypted dropdown box
    String decryptText = "Decrypted Text: " + decryptText(ciphertext, key);
    String decryptedKey = "Decrypted with Key " + key;

    // Add both results to the dropdown
    dropdown.addItem(decryptedKey); // Displays the decrypted key
    dropdown.addItem(decryptText); // Displays the decrypted message
}

private static String encryptText(String plaintext, int key) { //This is to encode the alphanumerical value from A-Z & 1-9
    StringBuilder result = new StringBuilder();

    for (int i = 0; i < plaintext.length(); i++) {
        char currentChar = plaintext.charAt(i);

        if (Character.isLetterOrDigit(currentChar)) { //Letter or Digit values
            char encryptedChar = encryptChar(currentChar, key);
            result.append(encryptedChar);
        } else if (Character.isWhitespace(currentChar)) {//If its a white space then add it to the resulted string, so keep it the same (output)
            result.append(currentChar);
        } else {
            result.append(currentChar);
        }
    }

    return result.toString();
}

private static String decryptText(String ciphertext, int key) { //This method is to decrypt the alphanumerical values from A-Z & 1-9
    StringBuilder result = new StringBuilder();

    for (int i = 0; i < ciphertext.length(); i++) {
        char currentChar = ciphertext.charAt(i);

        if (Character.isLetterOrDigit(currentChar)) {
            char decryptedChar = decryptChar(currentChar, key);
            result.append(decryptedChar);
        } else if (Character.isWhitespace(currentChar)) {
            result.append(currentChar);
        } else {
            result.append(currentChar);
        }
    }

    return result.toString();
}

private static char encryptChar(char c, int key) { // This method is to give instructions on what to do for certain characters
    int alphabetSize = 26; //Alphabet size

    if (Character.isUpperCase(c)) {
        return (char) (((c - 'A' + key) % alphabetSize + alphabetSize) % alphabetSize + 'A'); //For Uppercase characters
    } else if (Character.isLowerCase(c)) {
        return (char) (((c - 'a' + key) % alphabetSize + alphabetSize) % alphabetSize + 'a'); //For lowercase characters
    } else if (Character.isDigit(c)) {
        return (char) (((c - '0' + key) % 10 + 10) % 10 + '0'); // If it's a digit value then it'll iterate from 0-9 depending on the key value and number
    }

    // Non-alphanumeric characters remain unchanged
    return c;
}

private static char decryptChar(char c, int key) {  // This method is to decrypt the characters by reversing the key
    return encryptChar(c, -key);


}
}
```

Implementation (Front end GUI)



```java
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.imageio.ImageIO;
import java.io.IOException;
import java.net.URL;


public class Caesar {

    private static int key;

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> createAndShowGUI());
    }

    private static void createAndShowGUI() {
        JFrame frame = new JFrame("Caesar Cypher");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(1920, 1080);

        JPanel panel = new JPanel(new GridBagLayout()) {
            @Override
            protected void paintComponent(Graphics g) {
                super.paintComponent(g);
                // Load image from URL
                try {
                    URL imageURL = new URL("https://cdn.mos.cms.futurecdn.net/BiNbcY5fXy9Lra47jqHKGK-1200-80.jpg"); // Replace with your image URL
                    Image image = ImageIO.read(imageURL);
                    g.drawImage(image, 0, 0, getWidth(), getHeight(), this);
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        };
        frame.add(panel);

        // Rest of your existing code for placing components
        placeComponents(panel);

        frame.setVisible(true);

    }
```

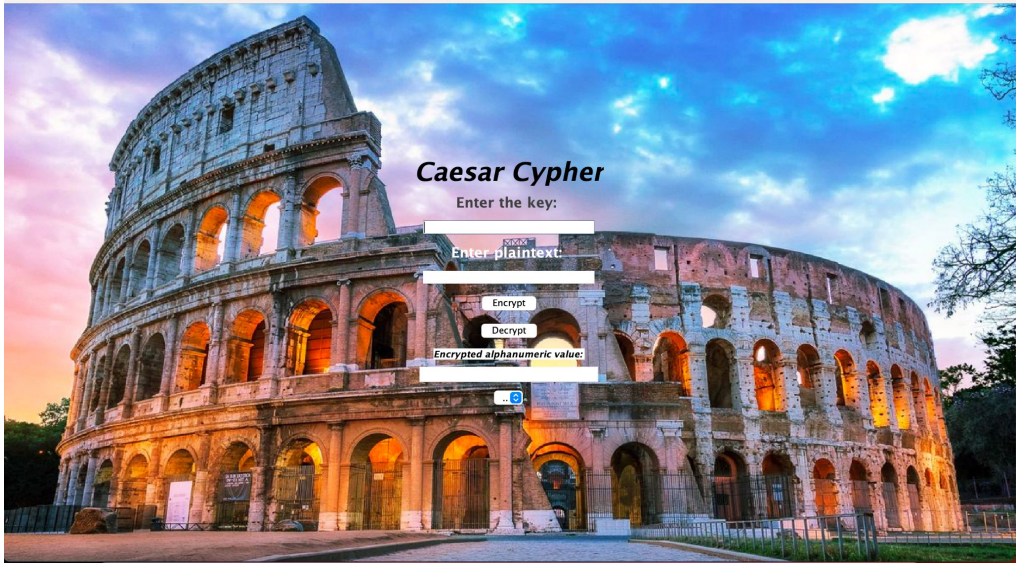


```java
    private static void setURLasBackground(String imageUrl, JLabel label) {
        try {
            URL url = new URL(imageUrl);
            Image backgroundImage = ImageIO.read(url);
            label.setIcon(new ImageIcon(backgroundImage));
        } catch (IOException e) {
            e.printStackTrace();
        }
    }


    private static void placeComponents(JPanel panel) {
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.gridwidth = GridBagConstraints.REMAINDER;

        gbc.insets = new Insets(5, 5, 5, 5);


        // Title
        JLabel titleLabel = new JLabel("<html><em><b>Caesar Cypher</b></em></html>");
        titleLabel.setFont(new Font("Consolas", Font.BOLD, 36));

        panel.add(titleLabel, gbc);

        // Labels and Text Fields
        JLabel keyLabel = new JLabel("<html><b>Enter the key: </b></html>");
        keyLabel.setFont(new Font("Consolas", Font.BOLD, 20));
        keyLabel.setForeground(Color.DARK_GRAY); // Set font color

        panel.add(keyLabel, gbc);

        JPasswordField keyText = new JPasswordField(20);
        panel.add(keyText, gbc);

        JLabel plaintextLabel = new JLabel("<html><b>Enter plaintext: </html></b>");
        plaintextLabel.setFont(new Font("Consolas", Font.BOLD, 20));
        plaintextLabel.setForeground(Color.WHITE); // Set font color

        panel.add(plaintextLabel, gbc);

        JPasswordField plaintextText = new JPasswordField(20);
        panel.add(plaintextText, gbc);

        // Buttons
        JButton encryptButton = new JButton("Encrypt");
        panel.add(encryptButton, gbc);

        JButton decryptButton = new JButton("Decrypt");
        panel.add(decryptButton, gbc);

        // Text and Labels for Cipher Text
        JLabel cipherTextLabel = new JLabel("<html><em><b>Encrypted alphanumeric value: </html></em></b>");
        cipherTextLabel.setForeground(Color.BLACK); // Set font color
        cipherTextLabel.setBackground(Color.WHITE);
        cipherTextLabel.setOpaque(true);

        panel.add(cipherTextLabel, gbc);

        // Cipher TextLabel
        JTextField ciphertextArea = new JTextField();
        ciphertextArea.setBounds(10, 110, 360, 40);
        ciphertextArea.setColumns(20); // Set the preferred number of columns
        ciphertextArea.setForeground(Color.BLACK); // Set font color
        ciphertextArea.setBackground(Color.WHITE);
        ciphertextArea.setOpaque(true);

        panel.add(ciphertextArea, gbc);



        // Dropdown for Decryption Results
        JComboBox<String> decryptionResultsDropdown = new JComboBox<>();
        panel.add(decryptionResultsDropdown, gbc);
```

3

Results

RESULTS


Caesar Cypher
Enter the key:
Enter plaintext:
Encrypt
Decrypt
Encrypted alphanumeric value:

Conclusion

Bring the attention of your audience over a key concept using icons or illustrations

Conclusion

- We've learned that you can use HTML tags for the Title labels
- We also learned how to place components and style them with different colors and font size
- Added a background image to a GUI
- Learned how to use action listeners to our action events for both our encrypt and decrypt methods
- Handled each error to make sure the alphanumeric values iterated from A-Z & 0-9
- Kept the special characters the same (e.g: ~ , / , . , * , [])

THANKS!

Any questions?