

1. Write a java program to display each word from a file in reverse order

```
import java.io.*;
import java.util.Scanner;

public class ReverseWordsFromFile {
    public static void main(String[] args) {
        String filePath = "input.txt";

        try {
            File file = new File(filePath);
            Scanner scanner = new Scanner(file);

            while (scanner.hasNextLine()) {
                String line = scanner.nextLine();
                String[] words = line.split("\\s+");
                for (String word : words) {
                    System.out.print(reverseString(word) + "
");
                }
                System.out.println();
            }
            scanner.close();
        } catch (FileNotFoundException e) {
            System.out.println("File not found: " + filePath);
        }
    }

    public static String reverseString(String word) {
        StringBuilder reversed = new StringBuilder(word);
        return reversed.reverse().toString();
    }
}
```


2. Create a hashtable containing city name & STD code. Display the details of the hashtable. Also search for a specific city and display STD code of that city.

```
import java.io.*;
import java.util.Scanner;

public class ReverseWordsFromFile {
    public static void main(String[] args) {
        String filePath = "input.txt";

        try {
            File file = new File(filePath);
            Scanner scanner = new Scanner(file);

            while (scanner.hasNextLine()) {
                String line = scanner.nextLine();
                String[] words = line.split("\\s+");
                for (String word : words) {
                    System.out.print(reverseString(word) + " ");
                }
                System.out.println();
            }
            scanner.close();
        } catch (FileNotFoundException e) {
            System.out.println("File not found: " + filePath);
        }
    }
}
```

```

        public static String reverseString(String word) {
            StringBuilder reversed = new StringBuilder(word);
            return reversed.reverse().toString();
        }
    }
}
-----
-----
-----

```

3. Define a class named Rectangle which can be constructed by a length and width.

The Rectangle class has a method which can compute the area and Perimeter.

```

class Rectangle {
    private double length;
    private double width;

    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }

    public double calculateArea() {
        return length * width;
    }

    public double calculatePerimeter() {
        return 2 * (length + width);
    }

    public void displayDetails() {
        System.out.println("Rectangle Details:");
        System.out.println("Length: " + length);
        System.out.println("Width: " + width);
        System.out.println("Area: " + calculateArea());
        System.out.println("Perimeter: " + calculatePerimeter());
    }
}

public class RectangleDemo {
    public static void main(String[] args) {
        Rectangle rect = new Rectangle(5.0, 3.0);

        rect.displayDetails();
    }
}

```

4. Write a Python program to convert a tuple of string values to a tuple of integer

values. Original tuple values: (('333', '33'), ('1416', '55'))

New tuple values: ((333, 33), (1416, 55))

```
original_tuple = (('333', '33'), ('1416', '55'))
```

```
new_tuple = tuple(tuple(int(val) for val in sub_tuple) for sub_tuple in
original_tuple)
```

```
print("Original tuple values:", original_tuple)
```

```
print("New tuple values:", new_tuple)
```

5. Write a Java program to calculate factorial of a number using recursion.

```
import java.util.Scanner;

public class Factorial {
    public static int factorial(int n) {
        if (n == 0 || n == 1) {
            return 1;
        } else {
            return n * factorial(n - 1);
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int num = scanner.nextInt();

        int result = factorial(num);

        System.out.println("Factorial of " + num + " is: " + result);
    }
}
```

Write a python class to accept a string and number n from user and display n repetition of strings by overloading * operator

```
class StringRepeater:
    def __init__(self, text):
        self.text = text

    def __mul__(self, n):
        if isinstance(n, int): # Ensuring n is an integer
            return self.text * n
        else:
            raise ValueError("The multiplier must be an integer.")

user_string = input("Enter a string: ")
n = int(input("Enter the number of repetitions: "))

string_repeater = StringRepeater(user_string)

result = string_repeater * n

print(result)
```

Write a python script to implement bubble sort using list

```
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n - i - 1):
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]

# Example usage
if __name__ == "__main__":
    user_input = input("Enter numbers separated by spaces: ")
```

```
# Converting input string to a list of integers
numbers = list(map(int, user_input.split()))

print("Original list:", numbers)

bubble_sort(numbers)

print("Sorted list:", numbers)
```

Write a java program to check whether given file is hidden or not. If not then display its path, otherwise display appropriate message

```
import java.io.File;
import java.util.Scanner;

public class HiddenFileChecker {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the path of the file: ");
        String filePath = scanner.nextLine();

        File file = new File(filePath);

        if (file.exists()) {
            if (file.isHidden()) {
                System.out.println("The file is hidden.");
            } else {
                System.out.println("The file is not hidden.");
                System.out.println("File path: " + file.getAbsolutePath());
            }
        } else {
            System.out.println("The file does not exist.");
        }

        scanner.close();
    }
}
```

Write a Python GUI program to create a label and change the label font style (font name, bold, size) using tkinter module.

```
import tkinter as tk
from tkinter import font

def update_label():
    selected_font_name = font_name_var.get()
    selected_font_size = font_size_var.get()
    bold_value = bold_var.get()

    if bold_value:
        current_font = (selected_font_name, selected_font_size, 'bold')
    else:
        current_font = (selected_font_name, selected_font_size)

    label.config(font=current_font)

root = tk.Tk()
```

```

root.title("Change Label Font")

label = tk.Label(root, text="Sample Label", font=("Helvetica", 12))
label.pack(pady=20)

font_name_var = tk.StringVar(value="Helvetica")
font_size_var = tk.StringVar(value="12")
bold_var = tk.BooleanVar(value=False)

font_names = ["Helvetica", "Arial", "Times New Roman", "Courier New"]

font_name_label = tk.Label(root, text="Select Font Name:")
font_name_label.pack()
font_name_menu = tk.OptionMenu(root, font_name_var, *font_names)
font_name_menu.pack()

font_size_label = tk.Label(root, text="Select Font Size:")
font_size_label.pack()
font_size_menu = tk.OptionMenu(root, font_size_var, "8", "10", "12", "14", "16",
"18", "20", "24", "28", "32")
font_size_menu.pack()

bold_checkbox = tk.Checkbutton(root, text="Bold", variable=bold_var)
bold_checkbox.pack()

apply_button = tk.Button(root, text="Apply Changes", command=update_label)
apply_button.pack(pady=10)

root.mainloop()

```

Create a class circles having members radius. Use operator overloading to add the radius of two circle objects. Also display the area of circle.

```

import math

class Circle:
    def __init__(self, radius):
        self.radius = radius

    def __add__(self, other):
        if isinstance(other, Circle):
            return Circle(self.radius + other.radius)
        return NotImplemented

    def area(self):
        return math.pi * (self.radius ** 2)

    def display(self):
        print(f"Radius: {self.radius:.2f}")
        print(f"Area: {self.area():.2f}")

if __name__ == "__main__":
    circle1 = Circle(5)
    circle2 = Circle(3)

    print("Circle 1:")
    circle1.display()

    print("\nCircle 2:")
    circle2.display()

```

```

circle3 = circle1 + circle2

print("\nCircle 3 (sum of Circle 1 and Circle 2):")
circle3.display()

```

Write a java program to display ASCII values of the characters from a file

```

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class ASCIIIDisplay {

    public static void main(String[] args) {
        String filename = "input.txt";
        try (BufferedReader br = new BufferedReader(new FileReader(filename))) {
            String line;
            while ((line = br.readLine()) != null) {
                for (char c : line.toCharArray()) {
                    System.out.println("Character: " + c + " |
ASCII Value: " + (int) c);
                }
            }
        } catch (IOException e) {
            System.out.println("An error occurred while reading the file: " +
e.getMessage());
        }
    }
}

```

Write a java program using applet to draw Temple

```

import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;

public class TempleApplet extends Applet {

    @Override
    public void init() {
        setBackground(Color.CYAN);
    }

    @Override
    public void paint(Graphics g) {
        g.setColor(Color.DARK_GRAY);
        g.fillRect(150, 250, 200, 100);
        g.setColor(Color.ORANGE);
        int[] xPoints = {140, 250, 360};
        int[] yPoints = {250, 150, 250};
        g.fillPolygon(xPoints, yPoints, 3);
        g.setColor(Color.BROWN);
        g.fillRect(230, 300, 40, 50);

        g.setColor(Color.LIGHT_GRAY);
        g.fillRect(170, 270, 40, 30);
        g.setColor(Color.BLACK);
        g.fillRect(290, 270, 40, 30);
        g.drawRect(170, 270, 40, 30);
        g.drawRect(290, 270, 40, 30);
        g.drawRect(230, 300, 40, 50);
        g.setColor(Color.BLACK);
    }
}

```

```
        g.drawString("Temple", 240, 320);    }  
}
```

Write an anonymous function to find area of square and rectangle.

```
area_of_square = lambda side: side ** 2  
  
area_of_rectangle = lambda length, width: length * width  
  
if __name__ == "__main__":  
    # Input for square  
    square_side = float(input("Enter the side length of the square: "))  
    square_area = area_of_square(square_side)  
    print(f"The area of the square is: {square_area}")  
  
    rectangle_length = float(input("Enter the length of the rectangle: "))  
    rectangle_width = float(input("Enter the width of the rectangle: "))  
    rectangle_area = area_of_rectangle(rectangle_length, rectangle_width)  
    print(f"The area of the rectangle is: {rectangle_area}")
```

Write Python GUI program which accepts a sentence from the user and alters it when a button is pressed. Every space should be replaced by *, case of all alphabets should be reversed, digits are replaced by?

```
import tkinter as tk  
  
def alter_sentence():  
    original_sentence = entry.get()    altered_sentence = ""  
  
    for char in original_sentence:  
        if char.isdigit(): # Replace digits with '?'  
            altered_sentence += '?'  
        elif char.isspace(): # Replace spaces with '*'  
            altered_sentence += '*'  
        elif char.isalpha(): # Reverse the case of alphabetic characters  
            if char.islower():  
                altered_sentence += char.upper()  
            else:  
                altered_sentence += char.lower()  
        else: # Keep other characters unchanged  
            altered_sentence += char  
  
    result_label.config(text=altered_sentence)  
root = tk.Tk()  
root.title("Sentence Alteration Tool")  
  
label = tk.Label(root, text="Enter a sentence:")  
label.pack(pady=10)  
  
entry = tk.Entry(root, width=50)  
entry.pack(pady=10)  
  
alter_button = tk.Button(root, text="Alter Sentence", command=alter_sentence)  
alter_button.pack(pady=10)  
  
result_label = tk.Label(root, text="", wraplength=400)  
result_label.pack(pady=20)
```

```
root.mainloop()
```

Write a java program to accept a number from user, If it is greater than 1000 then throw user defined exception "Number is out of Range" otherwise display the factors of that number. (Use static keyword)

```
import java.util.Scanner;

class NumberOutOfRangeException extends Exception {
    public NumberOutOfRangeException(String message) {
        super(message);
    }
}

public class FactorFinder {

    static void displayFactors(int number) {
        System.out.println("Factors of " + number + " are:");
        for (int i = 1; i <= number; i++) {
            if (number % i == 0) {
                System.out.print(i + " ");
            }
        }
        System.out.println();
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int userInput = scanner.nextInt();

        try {
            if (userInput > 1000) {
                throw new NumberOutOfRangeException("Number is out of Range");
            } else {
                displayFactors(userInput);
            }
        } catch (NumberOutOfRangeException e) {
            System.out.println(e.getMessage());
        } finally {
            scanner.close();
        }
    }
}
```

Write a java program to accept directory name in TextField and display list of files and subdirectories in List Control from that directory by clicking on Button

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.File;

public class DirectoryLister extends JFrame {
    private JTextField directoryTextField;
    private JButton listButton;
    private JList<String> fileList;
    private DefaultListModel<String> listModel;
```



```

public DirectoryLister() {
    setTitle("Directory Lister");
    setSize(400, 300);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLocationRelativeTo(null);
    directoryTextField = new JTextField(25);
    listButton = new JButton("List Files and Subdirectories");
    listModel = new DefaultListModel<>();
    fileList = new JList<>(listModel);
    JScrollPane scrollPane = new JScrollPane(fileList);

    JPanel panel = new JPanel();
    panel.setLayout(new FlowLayout());
    panel.add(new JLabel("Enter Directory:"));
    panel.add(directoryTextField);
    panel.add(listButton);
    add(panel, BorderLayout.NORTH);
    add(scrollPane, BorderLayout.CENTER);

    listButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            listFilesInDirectory();
        }
    });
}

private void listFilesInDirectory() {
    String directoryPath = directoryTextField.getText();
    File directory = new File(directoryPath);

    listModel.clear();

    if (directory.exists() && directory.isDirectory()) {
        File[] files = directory.listFiles();
        if (files != null) {
            for (File file : files) {
                listModel.addElement(file.getName());
            }
        }
    } else {
        JOptionPane.showMessageDialog(this, "Invalid directory path!",
"Error", JOptionPane.ERROR_MESSAGE);
    }
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
        DirectoryLister app = new DirectoryLister();
        app.setVisible(true);
    });
}
}

```

Write a Python program to unzip a list of tuples into individual lists.

```

tuple_list = [(1, 'apple'), (2, 'banana'), (3, 'cherry')]
numbers, fruits = zip(*tuple_list)
numbers_list = list(numbers)
fruits_list = list(fruits)
print("Numbers List:", numbers_list)

```

```
print("Fruits List:", fruits_list)
```

Write Python GUI program to accept a decimal number and convert and display it to binary, octal and hexadecimal number

```
import tkinter as tk
from tkinter import messagebox

def convert_number():
    try:
        decimal_number = int(entry.get())
        binary_number = bin(decimal_number)[2:]
        octal_number = oct(decimal_number)[2:]
        hexadecimal_number = hex(decimal_number)[2:].upper()
        result_var.set(f"Binary: {binary_number}\nOctal: {octal_number}\nHexadecimal: {hexadecimal_number}")
    except ValueError:
        messagebox.showerror("Input Error", "Please enter a valid decimal number.")

root = tk.Tk()
root.title("Decimal Number Converter")

label = tk.Label(root, text="Enter a decimal number:")
label.pack(pady=10)

entry = tk.Entry(root, width=30)
entry.pack(pady=10)

convert_button = tk.Button(root, text="Convert", command=convert_number)
convert_button.pack(pady=10)

result_var = tk.StringVar()
result_label = tk.Label(root, textvariable=result_var, justify=tk.LEFT)
result_label.pack(pady=20)

root.mainloop()
```

Write a java program to count the number of integers from a given list. (Use Command line arguments).

```
public class CountIntegers {

    public static void main(String[] args) {
        int count = 0;
        for (String arg : args) {
            try {
                Integer.parseInt(arg);
                count++; // Increment count if parsing is successful
            } catch (NumberFormatException e) {
            }
        }

        System.out.println("Number of integers: " + count);
    }
}
```

Write a java Program to accept the details of 5 employees (Eno, Ename, Salary) and display it onto the JTable.

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class EmployeeDetails extends JFrame {
    private JTextField enoField;
    private JTextField enameField;
    private JTextField salaryField;
    private DefaultTableModel tableModel;

    public EmployeeDetails() {

        setTitle("Employee Details");
        setSize(600, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        enoField = new JTextField(10);
        enameField = new JTextField(10);
        salaryField = new JTextField(10);
        JButton addButton = new JButton("Add Employee");

        tableModel = new DefaultTableModel(new String[]{"Employee
Number", "Employee Name", "Salary"}, 0);
        JTable employeeTable = new JTable(tableModel);
        JScrollPane scrollPane = new JScrollPane(employeeTable);
        JPanel inputPanel = new JPanel();
        inputPanel.add(new JLabel("Employee Number:"));
        inputPanel.add(enoField);
        inputPanel.add(new JLabel("Employee Name:"));
        inputPanel.add(enameField);
        inputPanel.add(new JLabel("Salary:"));
        inputPanel.add(salaryField);
        inputPanel.add(addButton);

        add(inputPanel, BorderLayout.NORTH);
        add(scrollPane, BorderLayout.CENTER);

        addButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                addEmployee();
            }
        });
    }

    private void addEmployee() {
        String eno = enoField.getText();
        String ename = enameField.getText();
        String salary = salaryField.getText();

        if (!eno.isEmpty() && !ename.isEmpty() && !salary.isEmpty()) {
            tableModel.addRow(new Object[]{eno, ename, salary});
            enoField.setText("");
            enameField.setText("");
            salaryField.setText("");
        } else {
            JOptionPane.showMessageDialog(this, "Please fill in all fields.",
```

```

"Input Error", JOptionPane.ERROR_MESSAGE);
    }
}

public static void main(String[] args) {
    // Create and display the application window
    SwingUtilities.invokeLater(() -> {
        EmployeeDetails app = new EmployeeDetails();
        app.setVisible(true);
    });
}
}

```

Write a Python GUI program to create a list of Computer Science Courses using Tkinter module (use Listbox).

```

import tkinter as tk

class CourseListApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Computer Science Courses")
        self.root.geometry("300x300") # Set the window size

        self.course_listbox = tk.Listbox(root, selectmode=tk.SINGLE)
        self.course_listbox.pack(pady=20, padx=20)

        self.courses = [
            "Data Structures",
            "Algorithms",
            "Computer Networks",
            "Database Management Systems",
            "Operating Systems",
            "Software Engineering",
            "Machine Learning",
            "Artificial Intelligence",
            "Web Development",
            "Mobile App Development"
        ]

        for course in self.courses:
            self.course_listbox.insert(tk.END, course)

        self.show_button = tk.Button(root, text="Show Selected Course",
command=self.show_selected_course)
        self.show_button.pack(pady=10)

        def show_selected_course(self):
            selected_index = self.course_listbox.curselection()
            if selected_index:
                selected_course = self.course_listbox.get(selected_index)
                tk.messagebox.showinfo("Selected Course", f"You selected:
{selected_course}")
            else:
                tk.messagebox.showwarning("Selection Error", "Please select a
course.")

if __name__ == "__main__":
    root = tk.Tk()
    app = CourseListApp(root)
    root.mainloop()

```

Write a Python program to accept two lists and merge the two lists into list of tuple

```
def merge_lists_to_tuples(list1, list2):
    return list(zip(list1, list2))

def main():
    list1 = input("Enter the first list of elements (comma-separated):")
    list1 = list(list1.split(','))
    list2 = input("Enter the second list of elements (comma-separated):")
    list2 = list(list2.split(','))

    list1 = [item.strip() for item in list1]
    list2 = [item.strip() for item in list2]
    merged_list = merge_lists_to_tuples(list1, list2)

    print("Merged list of tuples:")
    print(merged_list)

if __name__ == "__main__":
    main()
```

Write a java program to check whether given candidate is eligible for voting or not.
Handle user defined as well as system defined Exception.

```
import java.util.Scanner;

class InvalidAgeException extends Exception {
    public InvalidAgeException(String message) {
        super(message);
    }
}

public class VotingEligibility {
    public static void checkEligibility(int age) throws InvalidAgeException {
        if (age < 0) {
            throw new InvalidAgeException("Age cannot be negative.");
        } else if (age < 18) {
            throw new InvalidAgeException("Candidate must be at least 18 years old to vote.");
        } else {
            System.out.println("The candidate is eligible to vote.");
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.print("Enter the age of the candidate: ");
            int age = scanner.nextInt();

            checkEligibility(age);
        } catch (InvalidAgeException e) {
            System.out.println("User-defined Exception: " + e.getMessage());
        } catch (Exception e) {
            System.out.println("System-defined Exception: " + e.getMessage());
        }
    }
}
```

```

        System.out.println("System-defined Exception: Invalid input. Please
enter a valid age.");
    } finally {
        scanner.close();
    }
}
}

```

Write a java program using Applet for bouncing ball. Ball should change its color for each bounce

```

import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.Timer;

public class BouncingBallApplet extends Applet implements ActionListener {
    private int x, y;    private int diameter = 30;
    private int xDirection = 2;
    private int yDirection = 2;    private Color ballColor;
    private Timer timer;
    public void init() {
        // Initialize ball position
        x = 50;
        y = 50;
        ballColor = Color.RED;
        // Set up a timer for animation
        timer = new Timer(20, this);
        timer.start();
    }

    public void paint(Graphics g) {
        g.setColor(ballColor);
        g.fillOval(x, y, diameter, diameter);
    }

    public void actionPerformed(ActionEvent e) {
        // Update ball position
        x += xDirection;
        y += yDirection;

        // Check for bounce and change direction
        if (x <= 0 || x >= getWidth() - diameter) {
            xDirection = -xDirection;    changeColor();
        }
        if (y <= 0 || y >= getHeight() - diameter) {
            yDirection = -yDirection;    changeColor();
        }

        repaint();
    }

    private void changeColor() {
        ballColor = new Color((int)(Math.random() * 255), (int)
(Math.random() * 255), (int)(Math.random() * 255));
    }
}

```

) Write a Python GUI program to calculate volume of Sphere by accepting radius

```

as
input.

import tkinter as tk
from tkinter import messagebox
import math

class SphereVolumeCalculator:
    def __init__(self, root):
        self.root = root
        self.root.title("Sphere Volume Calculator")
        self.root.geometry("300x200")
        self.label = tk.Label(root, text="Enter the radius of the
sphere:")
        self.label.pack(pady=10)

        self.radius_entry = tk.Entry(root)
        self.radius_entry.pack(pady=10)

        self.calculate_button = tk.Button(root, text="Calculate Volume",
command=self.calculate_volume)
        self.calculate_button.pack(pady=10)

    def calculate_volume(self):
        try:
            radius = float(self.radius_entry.get())
            if radius < 0:
                raise ValueError("Radius cannot be negative.")

            # Calculate the volume
            volume = (4/3) * math.pi * (radius ** 3)

            messagebox.showinfo("Volume of Sphere", f"The volume of
the sphere is: {volume:.2f}")
        except ValueError as ve:
            messagebox.showerror("Input Error", f"Invalid input: {ve}")
        except Exception as e:
            messagebox.showerror("Error", f"An error occurred: {e}")

if __name__ == "__main__":
    root = tk.Tk()
    app = SphereVolumeCalculator(root)
    root.mainloop()

```

Write a java program to accept a number from a user, if it is zero then throw user defined Exception "Number is Zero" . If it is non-numeric then generate an error

```

import java.util.Scanner;

class ZeroNumberException extends Exception {
    public ZeroNumberException(String message) {
        super(message);
    }
}

public class NumberCheck {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

```

```

try {
    System.out.print("Enter a number: ");
    String input = scanner.nextLine();

    // Try to convert the input to an integer
    int number = Integer.parseInt(input);

        if (number == 0) {
            throw new ZeroNumberException("Number is Zero");
        } else {
            System.out.println("You entered: " + number);
        }
    } catch (ZeroNumberException e) {
        System.out.println("User-defined Exception: " + e.getMessage());
    } catch (NumberFormatException e) {
        System.out.println("Error: Invalid input. Please enter a
numeric value.");
    } catch (Exception e) {
        System.out.println("An unexpected error occurred: " +
e.getMessage());
    } finally {
        scanner.close();
    }
}

```

Write a Python GUI program to accept a string and a character from user and count the occurrences of a character in a string.

```

import tkinter as tk
from tkinter import messagebox

class CharacterCounter:
    def __init__(self, root):
        self.root = root
        self.root.title("Character Counter")
        self.root.geometry("300x200") # Set the window size

        self.label_string = tk.Label(root, text="Enter a string:")
        self.label_string.pack(pady=10)

        self.string_entry = tk.Entry(root, width=30)
        self.string_entry.pack(pady=5)

        self.label_char = tk.Label(root, text="Enter a character:")
        self.label_char.pack(pady=10)

        self.char_entry = tk.Entry(root, width=30)
        self.char_entry.pack(pady=5)

        self.count_button = tk.Button(root, text="Count Occurrences",
command=self.count_occurrences)
        self.count_button.pack(pady=10)

    def count_occurrences(self):
        input_string = self.string_entry.get()
        input_char = self.char_entry.get()
        if len(input_char) != 1:
            messagebox.showerror("Input Error", "Please enter exactly one
character.")

```



```

        return

        count = input_string.count(input_char)

        messagebox.showinfo("Character Count", f"The character
'{input_char}' occurs {count} times in the string.")

if __name__ == "__main__":
    root = tk.Tk()
    app = CharacterCounter(root)
    root.mainloop()

```

Python Program to Create a Class in which One Method Accepts a String from the User and Another method Prints it. Define a class named Country which has a method called print Nationality. Define subclass named state from Country which has a method called printState. Write a method to print state, country and nationality.

```

class Country:
    def __init__(self):
        self.nationality = ""

    def accept_nationality(self):
        self.nationality = input("Enter your nationality: ")

    def print_nationality(self):
        print(f"Nationality: {self.nationality}")

class State(Country):
    def __init__(self):
        super().__init__() # Initialize the parent class
        self.state_name = ""

    def accept_state(self):
        self.state_name = input("Enter the name of your state: ")

    def print_state(self):
        print(f"State: {self.state_name}")

    def print_details(self):
        self.print_state() # Print state
        print(f"Country: {self.__class__.__bases__[0].__name__}") # Print the
country class name
        self.print_nationality() # Print nationality

if __name__ == "__main__":
    state_obj = State()

    state_obj.accept_nationality()
    state_obj.accept_state()

    state_obj.print_details()

```
