Lokesh Slip 1 👍

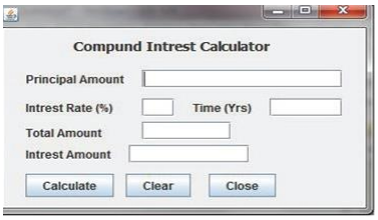
Java 1:A) Write a java program to count the frequency of each character in a given string.

```
public class lokesh1 {
    public static void main(String[] args) {
        String input = "example string";
        int[] frequency = new int[256]; // ASCII characters limit

        // Counting frequency of each character
        for (int i = 0; i < input.length(); i++) {
            frequency[input.charAt(i)]++;
        }

        // Displaying character frequencies
        for (int i = 0; i < 256; i++) {
            if (frequency[i] > 0) {
                System.out.println((char) i + ": " + frequency[i]);
            }
        }
    }
}
```

Java 2 :




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

public class lokesh2 {

    public static void main(String[] args) {
        // Create frame
        JFrame frame = new JFrame("Compound Interest Calculator");
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```



```
        frame.setLayout(null);

        // Create labels and text fields
        JLabel labelPrincipal = new JLabel("Principal Amount:");
        labelPrincipal.setBounds(30, 30, 150, 25);
        frame.add(labelPrincipal);
        JTextField textPrincipal = new JTextField();
        textPrincipal.setBounds(180, 30, 150, 25);
        frame.add(textPrincipal);

        JLabel labelRate = new JLabel("Interest Rate (%):");
        labelRate.setBounds(30, 70, 150, 25);
        frame.add(labelRate);
        JTextField textRate = new JTextField();
        textRate.setBounds(180, 70, 150, 25);
        frame.add(textRate);

        JLabel labelTime = new JLabel("Time (Years):");
        labelTime.setBounds(30, 110, 150, 25);
        frame.add(labelTime);
        JTextField textTime = new JTextField();
        textTime.setBounds(180, 110, 150, 25);
        frame.add(textTime);

        JLabel labelTotalAmount = new JLabel("Total Amount:");
        labelTotalAmount.setBounds(30, 150, 150, 25);
        frame.add(labelTotalAmount);
        JTextField textTotalAmount = new JTextField();
        textTotalAmount.setBounds(180, 150, 150, 25);
        textTotalAmount.setEditable(false);
        frame.add(textTotalAmount);

        JLabel labelInterestAmount = new JLabel("Interest Amount:");
        labelInterestAmount.setBounds(30, 190, 150, 25);
        frame.add(labelInterestAmount);
        JTextField textInterestAmount = new JTextField();
        textInterestAmount.setBounds(180, 190, 150, 25);
        textInterestAmount.setEditable(false);
        frame.add(textInterestAmount);

        // Create buttons
        JButton buttonCalculate = new JButton("Calculate");
        buttonCalculate.setBounds(30, 230, 100, 25);
        frame.add(buttonCalculate);

        JButton buttonClear = new JButton("Clear");
        buttonClear.setBounds(140, 230, 100, 25);
        frame.add(buttonClear);

        JButton buttonClose = new JButton("Close");
        buttonClose.setBounds(250, 230, 100, 25);
        frame.add(buttonClose);
```



```
        // Add action listeners
        buttonCalculate.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {
                    double principal = Double.parseDouble(textPrincipal.getText());
                    double rate = Double.parseDouble(textRate.getText());
                    double time = Double.parseDouble(textTime.getText());

                    // Compound Interest formula: A = P(1 + r/n)^(nt)
                    double totalAmount = principal * Math.pow((1 + (rate / 100)), time);
                    double interestAmount = totalAmount - principal;

                    // Display results
                    textTotalAmount.setText(String.format("%.2f", totalAmount));
                    textInterestAmount.setText(String.format("%.2f", interestAmount));

                } catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(frame, "Please enter valid numbers.", "Error",
JOptionPane.ERROR_MESSAGE);
                }
            }
        });

        buttonClear.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                textPrincipal.setText("");
                textRate.setText("");
                textTime.setText("");
                textTotalAmount.setText("");
                textInterestAmount.setText("");
            }
        });

        buttonClose.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                frame.dispose();
            }
        });

        // Display frame
        frame.setVisible(true);
    }
}
```


Part A) Python GUI program to display an alert message when a button is pressed.

```
import tkinter as tk
from tkinter import messagebox

# Function to display the alert message
def show_alert():
    messagebox.showinfo("Alert", "Button has been pressed!")

# Create main window
root = tk.Tk()
root.title("Alert Message Program")

# Create a button
button = tk.Button(root, text="Press Me", command=show_alert)
button.pack(pady=20)

# Run the application
root.mainloop()
```

Part B) Python class to find the validity of a string of parentheses.

```
class ParenthesesValidator:
    def is_valid(self, s: str) -> bool:
        stack = []
        # Dictionary to map closing brackets to their corresponding opening ones
        bracket_map = {')': '(', '}': '{', ']': '['}

        # Iterate over each character in the string
        for char in s:
            # If it's a closing bracket
            if char in bracket_map:
                # Pop the top of the stack or a dummy value if the stack is empty
                top_element = stack.pop() if stack else '#'

                # Check if the popped element is the corresponding opening bracket
                if bracket_map[char] != top_element:
                    return False
            else:
                # It's an opening bracket, push it onto the stack
                stack.append(char)

        # If the stack is empty, all brackets were matched correctly
        return not stack

# Test the class
validator = ParenthesesValidator()

# Examples
print(validator.is_valid("()"))       # Output: True
```

```
print(validator.is_valid("()[]{}"))     # Output: True
print(validator.is_valid("(]"))         # Output: False
print(validator.is_valid("({[}]"))      # Output: False
print(validator.is_valid("{{{"))        # Output: False
```


Slip @2:

Java

Part A) Java Program to accept 'n' numbers through the command line, store only Armstrong numbers into the array, and display that array.

```
import java.util.ArrayList;

public class ArmstrongNumbers {

    // Method to check if a number is an Armstrong number
    public static boolean isArmstrong(int number) {
        int originalNumber = number;
        int sum = 0;
        int digits = String.valueOf(number).length();

        while (number > 0) {
            int digit = number % 10;
            sum += Math.pow(digit, digits);
            number /= 10;
        }

        return sum == originalNumber;
    }

    public static void main(String[] args) {
        // ArrayList to store Armstrong numbers
        ArrayList<Integer> armstrongNumbers = new ArrayList<>();

        // Accept numbers from the command line
        for (String arg : args) {
            int num = Integer.parseInt(arg);
            if (isArmstrong(num)) {
                armstrongNumbers.add(num);
            }
        }

        // Display Armstrong numbers
        System.out.println("Armstrong numbers: " + armstrongNumbers);
    }
}
```


Part B) Java Program to define a class Product and calculate the total amount using an array of objects.

```
import java.util.Scanner;

class Product {
    // Fields of Product class
    int pid;
    String pname;
    double price;
    int qty;

    // Constructor to initialize Product details
    public Product(int pid, String pname, double price, int qty) {
        this.pid = pid;
        this.pname = pname;
        this.price = price;
        this.qty = qty;
    }

    // Method to display the product details
    public void displayProduct() {
        System.out.println("Product ID: " + pid);
        System.out.println("Product Name: " + pname);
        System.out.println("Price: " + price);
        System.out.println("Quantity: " + qty);
        System.out.println("Total Amount: " + calculateTotalAmount());
    }

    // Method to calculate the total amount (price * quantity)
    public double calculateTotalAmount() {
        return price * qty;
    }
}

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

        // Get the number of products
        System.out.print("Enter the number of products: ");
        int n = scanner.nextInt();

        // Array to store Product objects
        Product[] products = new Product[n];

        // Input details for each product
```



```
        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for product " + (i + 1) + ":");

            System.out.print("Product ID: ");
            int pid = scanner.nextInt();

            System.out.print("Product Name: ");
            String pname = scanner.next();

            System.out.print("Price: ");
            double price = scanner.nextDouble();

            System.out.print("Quantity: ");
            int qty = scanner.nextInt();

            // Create a new Product object and store it in the array
            products[i] = new Product(pid, pname, price, qty);
        }

        // Display details of each product
        System.out.println("\nProduct details:");
        for (Product product : products) {
            product.displayProduct();
            System.out.println();
        }

        scanner.close();
    }
}
```

Python:

Part A: Python GUI Program to Change Input String to Uppercase

```
import tkinter as tk

# Function to convert input text to uppercase
def convert_to_uppercase():
    input_text = entry.get()
    upper_text = input_text.upper()
    result_label.config(text=f"Uppercase: {upper_text}")

# Create the main window
root = tk.Tk()
root.title("Uppercase Converter")

# Create a label for instructions
label = tk.Label(root, text="Enter text:")
label.pack()

# Create an entry widget for text input
entry = tk.Entry(root)
entry.pack()

# Create a button to trigger the uppercase conversion
convert_button = tk.Button(root, text="Convert to Uppercase", command=convert_to_uppercase)
convert_button.pack()

# Create a label to display the result
result_label = tk.Label(root, text="")
result_label.pack()

# Start the GUI event loop
root.mainloop()
```


2. **Part B: Date Class with Validation and Exception Handling [B) Define a class Date (Day, Month, Year) with functions to accept and display it. Accept date from user. Throw user defined exception “invalid Date Exception” if the date is invalid.]**

```
class InvalidDateException(Exception):
    """Custom exception for invalid dates."""
    def __init__(self, message="Invalid Date"):
        self.message = message
        super().__init__(self.message)

class Date:
    def __init__(self):
        self.day = 0
        self.month = 0
        self.year = 0

    def accept(self):
        """Accepts the date from the user and validates it."""
        try:
            self.day = int(input("Enter day: "))
            self.month = int(input("Enter month: "))
            self.year = int(input("Enter year: "))
            if not self.is_valid_date():
                raise InvalidDateException("The entered date is invalid.")
        except ValueError:
            raise InvalidDateException("Invalid input! Please enter integers for day, month, and year.")

    def is_valid_date(self):
        """Validates the date."""
        # Check for valid month
        if self.month < 1 or self.month > 12:
            return False

        # Days in each month
        days_in_month = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]

        # Check for leap year
        if self.year % 4 == 0 and (self.year % 100 != 0 or self.year % 400 == 0):
            days_in_month[1] = 29  # February in a leap year

        # Check for valid day
        if self.day < 1 or self.day > days_in_month[self.month - 1]:
            return False

        return True

    def display(self):
```



```
        """Displays the date in DD/MM/YYYY format."""
        print(f"Date: {self.day:02}/{self.month:02}/{self.year}")

# Usage
try:
    date = Date()
    date.accept()
    date.display()
except InvalidDateException as e:
    print(e)
```