OOP lab exam problem statements.

1. **Implement a program to manage a list of students using the Iterator interface in Java. The program should allow the user to perform the following operations:**

**Add a student to the list.**

**Remove a student from the list.**

**Display all students in the list.**

**Replace the grade of any student**

**Each student should have attributes like name, roll number, and grade. Implement a Student class to represent a student, and a StudentManager class to manage the list of students using iterators.**

1. Write a Java program to manage employees in a company. The company has different types of employees, including **full-time employees** and **part-time employees**. Each employee has a **name, employee ID**, and **hourly rate**.

Full-time employees have additional attributes such as **salary** and **benefits**,

(Use super keyword)

while part-time employees have attributes for the **number of hours worked** and **overtime rate**.

(Use super keyword)

Your task is to implement a Java program using inheritance to represent these different types of employees and their attributes. Additionally, the program should provide functionality to calculate the monthly salary for each type of employee.

1. You are required to design a system for managing different types of shapes. The system should be able to calculate the area and perimeter of various shapes such as circles, rectangles, and triangles. Each shape has different methods to calculate its area and perimeter. (Use abstract methods-calculateArea(), calculatePerimeter(), displayDetails())

Your task is to implement a Java program using polymorphism to represent these different types of shapes and calculate their area and perimeter. Additionally, the program should provide functionality to display the details of each shape.

1. Implement a program for a temperature converter. The program should convert temperatures between Celsius and Fahrenheit. However, there are certain constraints:

I. The temperature value should be within a valid range (-273.15°C to 1000°C).

Ii. The conversion should only be performed if the input temperature is within the valid range.

Your task is to implement a Java program that handles these constraints using custom exceptions. Define a custom exception class **InvalidTemperatureException** to handle the case when the input temperature is outside the valid range.

1. Implement a Java program for calculating the area of geometric shapes. The program should support calculating the area of a rectangle, a square, and a circle. Each shape has a different method to calculate its area. Your task is to implement a Java program that demonstrates method overloading by providing multiple versions of the **calculateArea()** method to calculate the area of each shape. (Use method overloading)
2. implement a Java program to represent a simple calculator. The calculator should have the following features:
3. Addition of two numbers
4. Subtraction of two numbers
5. Multiplication of two numbers
6. Division of two numbers

Additionally, the program should initialize a constant value for the value of PI (3.14) and print a welcome message when an instance of the calculator is created. (static block)

Print “New calculator instance” using the instance block.

Your task is to implement a Java program that demonstrates the use of constructors, static block, and instance block to achieve these features.

1. Implement a Java program to represent a shape hierarchy. The program should have interfaces for different types of shapes, such as **Drawable** for shapes that can be drawn and **Resizable** for shapes that can be resized.

Interface **Drawable** has draw() method

Interface **Resizable** has resize(double factor) method

Implement classes for specific shapes such as circle, rectangle and demonstrate multiple inheritance by implementing both **Drawable** and **Resizable** interfaces in appropriate classes.

Circle implements both interfaces and implements both methods

Rectangle implements only **Drawable** interface and implements draw() method

1. Implement a Java program to perform basic operations on arrays. The program should support the following operations:
2. Initialize an array with given elements.
3. Find the sum of all elements in the array.
4. Find the maximum element in the array.
5. Find the minimum element in the array.
6. Sort the elements of the array in ascending order.
7. Reverse the elements of the array.

9. Implement a Java program to perform basic operations on strings. The program should support the following operations:

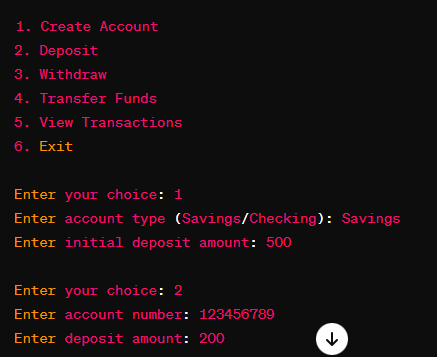
1. Concatenate two strings.
2. Find the length of a string.
3. Convert a string to uppercase.
4. Convert a string to lowercase.
5. Check if a string contains a specific substring.
6. Replace a substring with another substring in a string.

1) Implement a Student class with attributes such as ID, name, age, gender, grade, and contact information **using ArrayList** .Create a menu-driven program with an administrator login feature that allows authorized users to:

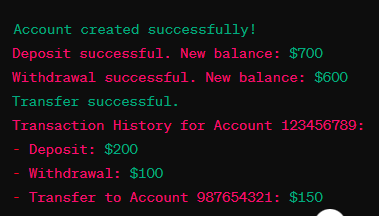
1. Add a new student to the database.
2. Remove a student from the database by their ID.
3. Update student information (name, age, grade, etc.).
4. Display all students in the database.
5. Search for a student by their ID, name, grade, or any other criteria.
6. Implement error handling and validation for user inputs, such as invalid IDs, duplicate entries, etc.

2) Develop a Banking System in Java with **abstraction and interfaces** for security and flexibility. Implement SavingsAccount and CheckingAccount classes adhering to the BankingAccount interface. Allow users to create accounts, deposit, withdraw, transfer funds, and view transactions.

Sample INput:



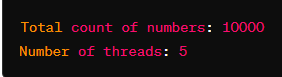
Sample Output:



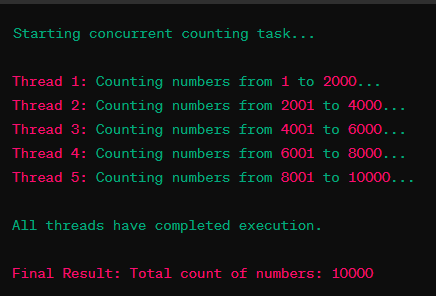
3) Develop a Java program demonstrating **custom exception handling** for bank account withdrawals. Define a WithdrawalException class to handle insufficient funds scenarios. Implement a BankAccount class with a withdraw method, throwing WithdrawalException when necessary. Create a user interface to input account details and withdrawal amount, catching and displaying custom exception messages.

4) write Java program demonstrates multithreading by concurrently counting numbers using multiple threads. The user specifies the total count of numbers and the number of threads to use. Each thread is responsible for counting a portion of the total numbers, ensuring efficient parallelism.

Sample Input:



Sample Output:



**1.** Create a class called Time, which has three private instance variables – hour, min and sec. It contains a method called add( ) which takes one Time object as parameter and prints the added value of the calling Time object and passes the Time object. In the main method, declare two Time objects and assign values using the constructor and call the add() method.

**2.** Write a Java program using try and catch to generate NegativeArrayIndex Exception and Arithmetic Exception.

**3.**Write a program to create a class named Vehicle having protected instance variables regnNumber, speed, color, ownerName and a method showData ( ) to show “This is a vehicle class”. Inherit the Vehicle class into subclasses named Bus and Car having individual private instance variables routeNumber in Bus and manufacturerName in Car and both of them having showData ( ) method showing all details of Bus and Car respectively with content of the super class’s showData ( ) method.

**4.**Design an abstract class having two methods. Create Rectangle and Triangle classes by inheriting the shape class and override the above methods to suitably implement for Rectangle and Triangle class.

**5.**Develop a BankApplication that manages deposit and withdrawal. use dynamic method dispatch and abstract class concepts.

**6**. Write a Java program to create a class called "Library" with a collection of books and methods to add and remove books.

7. **Develop a Library Management System in Java. The system should allow users to perform various operations such as adding books to the library, displaying available books, searching for books, checking out books, and checking in books.**

1. Book Class:
   * Create a Book class with the following attributes:
     + title (String): Title of the book.
     + author (String): Author of the book.
     + available (boolean): Indicates whether the book is available for checkout.
   * Implement appropriate constructors and methods to access and modify these attributes.
   * Override the toString() method to provide a meaningful string representation of a book.
2. Library Class:
   * Create a Library class to manage the book inventory.
   * Implement methods to:
     + Add a new book to the library.
     + Display all books in the library.
     + Search for a book by its title.
     + Check out a book (mark it as unavailable).
     + Check in a book (mark it as available).
3. Main Class (LibraryManagementSystem):
   * Implement the main class to interact with users through the console.
   * Display a menu with options to perform different operations such as:
     + Display all books in the library.
     + Add a new book.
     + Search for a book by title.
     + Check out a book.
     + Check in a book.
     + Exit the program.

**8.Develop an Online Movie Ticket Booking System in Java. The system should allow users to browse available movies, book tickets, and manage their bookings.**

#### **Movie Class:**

1. Create a Movie class with the following attributes:
   * title (String): Title of the movie.
   * genre (String): Genre of the movie.
   * duration (int): Duration of the movie in minutes.
   * rating (double): Rating of the movie.
2. Implement appropriate constructors and methods to access and modify these attributes.
3. Override the toString() method to provide a meaningful string representation of a movie.

#### **Theater Class:**

1. Create a Theater class to manage movie screenings and bookings.
2. Implement methods to:
   * Add a new movie screening.
   * Display available movies.
   * Allow users to book tickets for a specific movie screening.
   * Display user's booked tickets.
   * Cancel a booked ticket.

#### **Main Class (OnlineTicketBookingSystem):**

1. Implement the main class to interact with users through the console.
2. Display a menu with options to perform different operations such as:
   * Display available movies.
   * Book tickets for a movie.
   * View booked tickets.
   * Cancel a booked ticket.
   * Exit the program.

9. **Develop a Word Counter program in Java. The program should count the occurrences of each word in a given text and display the results.**

#### **Requirements:**

1. Implement a class named WordCounter.
2. Create a method countWords that takes a String text as input and returns a Map<String, Integer> containing the count of each word in the text.
3. Implement a main method in a separate class (e.g., WordCounterApp) to interact with users through the console.
4. In the main method, prompt the user to enter a text.
5. Use the WordCounter class to count the occurrences of each word in the entered text and display the results.

10. **You are required to implement a program that manipulates a list of integers in Java.**

Implement a Java program with the following functionalities:

* + Add an integer to the end of the list.
  + Insert an integer at a specified index in the list.
  + Remove an integer from the list based on its value.
  + Remove an integer at a specified index in the list.
  + Search for the index of a given integer in the list.
  + Display all integers in the list.

1. Create a Java class named Vehicle with properties like make, model, and year. Create another class named Car that inherits from Vehicle and adds specific properties like number of doors. Write a program to demonstrate the inheritance relationship and access properties from both classes.

2. Build an online shopping cart system where users can add, remove, and purchase items. Define an interface named Purchasable with methods like addItem(), removeItem(), and calculateTotalPrice(). Implement this interface in classes representing different types of items such as Electronics, Clothing, and Books. Develop a shopping cart manager class to manage user interactions and handle the purchase process.

3. Develop a vehicle rental system where customers can rent different types of vehicles. Define an interface called Rentable with methods like rent(), returnVehicle(), and calculateRentalCost(). Implement this interface in classes representing different types of vehicles such as Car, Motorcycle, and Bicycle. Create a rental manager class to handle the renting and returning of vehicles.

4. Design a system to manage a library's collection of books using collections. Create classes to represent books with attributes like title, author, ISBN, and availability status. Implement functionalities such as adding new books, borrowing books, returning books, searching for books by title or author, and displaying available books.

5. Develop a program to manage a student database using collections. Implement functionalities such as adding new students, removing existing students, searching for students by ID or name, and displaying all student details. Utilize a collection type like ArrayList or HashMap to store and manage student records efficiently.

6.Create a program that prints numbers from 1 to 10 using multiple threads. Each thread should print a subset of numbers sequentially. Ensure proper synchronization to maintain the order of numbers.

7.Build a flight reservation system that allows users to search for flights, book tickets, and cancel reservations. Implement exception handling to handle situations such as seat availability, invalid departure/arrival dates, or unexpected errors during booking confirmation.

8.Build a Vehicle class representing vehicles with attributes like make, model, year, and mileage. Implement a constructor to initialize these attributes using user-provided values. Validate input parameters to ensure data consistency (e.g., positive mileage value, valid year). Test the constructor by creating vehicle objects with various parameters.

9.Develop a system to manage online courses with different types of courses such as Programming, Mathematics, and Literature. Design base classes and derived classes to represent these course types, inheriting common attributes and behaviors such as title, instructor, duration, and enrollment fees. Utilize inheritance to model the relationships between course types and implement methods for course-specific functionalities.

10. Build a simulation of the animal kingdom with various types of animals such as Mammals, Birds, and Fish. Create an abstract class called Animal with abstract methods for behaviors like eat, sleep, and move. Develop concrete subclasses for each animal type, extending the Animal class and implementing the behavior methods with species-specific actions.

1. You are tasked with creating a Java program that counts the number of unique words in a given text using a HashSet.

Requirements:

WordCounter Class:

Create a WordCounter class that includes the following:

A method countUniqueWords(String text) that takes a text as input and returns the count

of unique words.

Use a HashSet to store unique words.

Consider a word as any sequence of characters separated by whitespace.

Main Application: Implement a main application that demonstrates the functionality of the WordCounter class.Allow the user to input a text string.

Use the WordCounter class to count and display the number of unique words in the input

text.

Sample Output :

Enter a text string: This is a simple Java program. Java is powerful and simple.

Number of unique words: 8

2. a. Check that given number is Armstrong or not.Eg-153=1 3 +5 3 +3 3 =1+225+27=153

b. Write a Java program to check whether two strings are anagram or not?

RACE and CARE are anagram strings.

c. Take two DOB in string format from user .Compare it and display the

messages as “Younger”, ”Elder” or “Same age”.

d. Check that given no. is prime or not.

3. a. Create arrayList, add the integer elements in arrayList using asList().Remove

the duplicate values and return an arrayList containing unique values. Implement the

logic inside removeDuplicates() method. Test the functionalities using the main ()

method of the Tester class. Sample Input and Output---------10, 20, 10, 15,40,15,40 ---

10,20,15,40

b. Create any arraylist and perform following operations-

i. check if an element exists in ArrayList?

ii.add element at particular index of ArrayList?

iii. remove element at particular index of ArrayList?

iv.sort a given array list.(collections.sort())

v. reverse elements in an array list.

vi. compare two array lists.

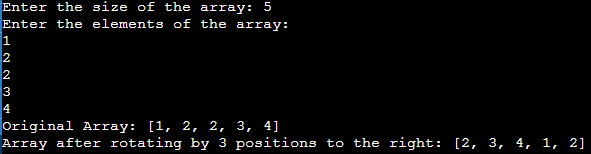
vii.. find first and last occurrence of repeated elements.

4. Generate a random array of integers with a specified size and range of values.

Implement the following array manipulation operations as separate methods:

1. Rotate the elements of the array to the right by a specified number of position.
2. Remove duplicate elements from the array while preserving the original order.
3. Rearrange the elements of the array such that all even numbers appear before all odd numbers, preserving the relative order of even and odd numbers.

Sample output:





5. You are given two integer arrays, nums1 and nums2, where nums1 is guaranteed to be the same length or shorter than nums2. Write a method to return an array representing their intersection. Each element in the result should appear as many times as it shows in both arrays, and you may return the result in any order. You can implement method like: **public static int[] intersection(int[] nums1, int[] nums2).You can use array concept or collection framework classes .**

**Sample o/p:**

**nums1 = {1,3, 2, 1};**

**nums2 = {3, 2};**

**nums3 = {4, 9, 5};**

**nums4 = {9, 4, 9, 8, 4};**

****

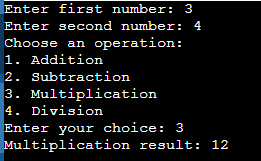
6. Design a Java program that should support functionalities such as employee registration, salary calculation, and employee evaluation.The program should have the following features:

* Employee Hierarchy:
  + Design an abstract superclass called Employee.
  + Implement subclasses FullTimeEmployee and PartTimeEmployee that inherit from Employee.
  + Each subclass should have instance variables to store employee details such as name, employee ID, and salary information (e.g., hourly rate for part-time employees, monthly salary for full-time employees).
  + Each subclass should override methods such as calculateSalary() to perform salary calculation specific to the employee type and evaluatePerformance() to evaluate employee performance.
* Salary Calculation:
  + Implement methods in each subclass to calculate the salary based on the employee's work hours or performance metrics.
  + Ensure that salaries are calculated accurately and take into account factors such as overtime pay or performance bonuses.
  + Performance evaluation not applicable for part-time employee
* Input/Output:
  + Implement a user interface to interact with the employee management system.
  + Allow the user to register new employees with unique employee IDs and provide their relevant details.
  + Provide options to calculate and display the salary for each employee.
  + Implement functionality to evaluate the performance of employees and display performance ratings.

7. design a Java program that simulates a **multithreaded calculator.** The program should create multiple threads, each responsible for performing a specific arithmetic operation on two numbers.

The program should have the following features:

1. Thread Creation:
   * Create separate threads for addition, subtraction, multiplication, and division operations.
2. Input:
   * Prompt the user to enter two numbers and choose an arithmetic operation.(addition,subtraction,multiplication and division)
3. Output:
   * Display the result of the arithmetic operation performed by each thread.
4. Concurrency:
   * Ensure that the threads execute concurrently, with each thread performing its designated operation independently of the others.
5. Error Handling:
   * Validate user input to ensure that numeric values entered are valid.
   * Handle cases where division by zero occurs.



8. Write a Java program for following operations: You can write menu driven program -

1. Create 5 threads.
2. Two Synchronized methods -one will print 2’s table and another will prints 5’s table upto 5 places.
3. Program that shows demonstration for daemon thread .

9. Write a Java program to find the length of the longest substring without repeating characters in a given string.

The program should have the following features:

1. Input:
   * Prompt the user to enter a string.
2. Output:
   * Display the length of the longest substring without repeating characters.
3. Substring Calculation:
   * Implement a method to calculate the length of the longest substring without repeating characters.
   * A substring is a contiguous sequence of characters within the string.

Sample o/p:

1. **Enter the string: abcabcbb**

**Length of the longest substring without repeating characters: 3**

1. **Enter the string: abcdefghhhhh**

**Length of the longest substring without repeating characters: 8**

10. Write a Java program for user defined excision for valid email address.

Program should check that the user has entered a valid email address which should contain @ and . otherwise it has to raise user defined exception **InvalidEmailException.**checks if the email contains '@' and '.', and if '@' comes before '.'. If these conditions are met, it considers the email address valid; otherwise, it's considered invalid.

