OOP Problem Statement

1) Create the CaesarCipher class with the following parts:

- Private fields for the alphabet and shiftedAlphabet
- Write a constructor CaesarCipher that has one int parameter key. This method should initialize all the private fields of the class.
- Write an encrypt method that has one String parameter named input. This method returns a String that is the input encrypted using shiftedAlphabet.
- Write a decrypt method that has one String parameter named input. This method returns a String that is the encrypted String decrypted using the key associated with this CaesarCipher object. One way to do this is to create another private field mainKey, which is initialized to be the value of key. Then you can create a CaesarCipher object within decrypt: CaesarCipher cc = new CaesarCipher(26 - mainKey); and call cc.encrypt(input).
- Create the TestCaesarCipher class with the following parts:
- Create a CaesarCipher object with key 18, encrypt the String read in using the Scanner class object, print the encrypted String, and decrypt the encrypted String using the decrypt method.

2) Create the URLFinder class with the following parts:

· Private fields for the url

· Write a constructor URLFinder that has one int parameter url. This method should initialize all the private fields of the class.

- Write an urlChecker method that has one String parameter named inputUrl. This method returns a Boolean “true” for valid url.
- Create the TestURLFinder class with the following parts:
- create a URLFinder object with String read in using the Scanner class

3) Perform the following operations on given matrix

M =

1	2	3
4	5	6
7	8	9

1. Transpose
2. Determinant
3. Inverse

4) A) Write a Program to Print the Pascal's and Number Triangle in Java

```
Input : N = 5
Output:
          1
        1   1
      1   2   1
    1   3   3   1
  1   4   6   4   1
1   5   10   10   5   1
```

4) B) Write a Java program to print the number triangle

```
     1
    2 2
   3 3 3
  4 4 4 4
 5 5 5 5 5
6 6 6 6 6 6
```

5)

A. An ArrayList consists of 1-25 numbers. Write a Java program to remove prime numbers from an ArrayList using an iterator.

B. Write a Java program to

   a. Create and traverse (or iterate) an ArrayList using a for-loop, iterator, and advance for-loop.

   b. Check if the element(value) exists in the ArrayList?

   C. Add element at the particular index of the ArrayList?

6)

- Write a Java program to find the largest and smallest elements in an array of integers.
- Implement a function to reverse an array in place.
- Given two arrays, write a method to merge them into a single sorted array.

7)

- Write a program to check if a given string is a palindrome.
- Implement a function to count the occurrences of a specific character in a string.
- Write a program to remove all whitespace from a string.

8) Design a class BankAccount with methods for deposit, withdraw, and check balance. Implement exception handling for insufficient funds during withdrawal.

9) Write a program to read data from a text file using IO Stream class and display the number of characters and words on the console.

10)

A. Write a program to find the greatest common divisor (GCD) of two numbers.
B. Write a program to convert a decimal number to binary.

11) Create the CarAssembly class which implements Runnable interface with the following parts:

- Private fields for the componentName(String) and timeToPrepare(int)
- Write a constructor CarAssembly that has two parameters componentName and timeToPrepare . This method should initialize all the private fields of the class.
- Write an run method that has sleep method which takes timeToPrepare parameter. The sleep method is invoveked between two print statements componentName is preparing & componentName is ready.
- Components names and their preparation times are as follows
- Engine-3000, Body-4000, Wheels-5000

Create three threads namely engineThread, bodyThread, wheelThread and use Join method for Sysnchronization.

12) Design a Java program to manage an ArrayList of integers that supports dynamic insertion at any position, deletion, updating values, and efficiently computing the sum of elements between two given indices after each modification.

Sample Input	Sample Output
insert 0 5	[5]
insert 1 10	[5, 10]
insert 1 15	[5, 15, 10]
update 2 20	[5, 15, 20]
sum 0 2	25
delete 1	[5, 20]
sum 0 1	25

13) Write a Java program using HashMap to add, remove, and track the frequency of words. You also need to find the most frequent word, and if there's a tie, return the smallest word alphabetically. The program should handle up to 10^5 operations efficiently.

Sample Input	Sample Output
add apple	Frequency of 'apple': 1
add banana	Most frequent word: banana
add apple	
remove apple	
query apple	
mostFrequent	

14) Design and implement a multi-threaded banking system in Java that simulates multiple users performing concurrent transactions on shared bank accounts. Each transaction can be a deposit, withdrawal, or transfer between accounts.

15)Design a shopping cart program where users can add items, apply discount codes, and check out. Use custom exceptions to handle scenarios like invalid coupon codes, out-of-stock items, and negative quantity inputs. (Exception handling)

16) Design an interface Vehicle with methods startRide(), endRide(), and calculateFare(int distance). Implement classes like Bike, Auto, and Cab, where each vehicle calculates fares differently. Use a PricingStrategy interface to dynamically adjust fares based on conditions like peak hours and holidays.

17) Design a University Staff Management System using a base class Staff and derived classes Professor, AdministrativeStaff, and MaintenanceStaff.
Override methods like displayDetails() and calculateBonus() differently in each subclass using polymorphism.
Use a list of base class pointers or references to manage multiple staff objects and demonstrate runtime polymorphism.
(Advanced) Implement a promote() method with different behaviors in each subclass.

18)

1. Create a class Rectangle with attributes length and breadth. Write a constructor that uses this keyword to initialize the attributes. Also, create a method compareArea(Rectangle r) that compares the area of two rectangles.
2. Create a class MathOperations with:
    - A static method square(int n) that returns the square of a number.
    - An instance method cube(int n) that returns the cube of a number. Write a Java program that demonstrates calling both static and instance methodsproperly.

19) Create a Java program for a seating system in a cinema hall represented by a 2D array (rows × columns).
    - Mark all seats as available (e.g., 0) initially.
    - Allow the user to book seats by marking them as booked (e.g., 1).
    - Display the current seat map (matrix form).
    - Add functionality to check if a given seat is available before booking.

20)

Develop a Java program that takes a paragraph input from the user and:

- Remove all vowels (a, e, i, o, u) from the paragraph using StringBuilder.
- Efficiently update the paragraph after each deletion.
- Finally, display the transformed paragraph along with the count of charactersremoved.

21) Create a base class `Employee` with attributes: name, id, and basicSalary, along with methods `displayDetails()` and `calculateSalary()`. Derive two subclasses: `Manager` with an additional bonus attribute and `Developer` with a projectAllowance attribute. Override the `calculateSalary()` method in both subclasses to include their respective additional amounts. In the main method, create objects of Manager and Developer, and call `displayDetails()` for each. Demonstrate polymorphism by invoking `calculateSalary()` using base class references and display the total salary.

22)

Create a class BankAccount with the following attributes and methods:

Attributes:

accountNumber (String)

balance (double)

Methods:

A constructor that initializes the accountNumber and balance.

A method withdraw(double amount) that:

Throws an ArithmeticException if the withdrawal amount is greater than the current balance.

Throws an IllegalArgumentException if the withdrawal amount is less than or equal to zero.

In the main() method:

Create an object of the BankAccount class with an initial balance.

Use try-catch blocks to handle the following scenarios:

Catch the ArithmeticException and display a message "Insufficient funds for withdrawal."

Catch the IllegalArgumentException and display a message "Invalid withdrawal amount."

After handling the exception, allow the program to continue running.

Display the current balance after each operation.

23)

Create a class Printer with a method printNumbers() that prints the numbers from 1 to 10 with a small delay between each number (use Thread.sleep(500) to simulate the delay).

Create two threads:

One thread will call the printNumbers() method and print numbers from 1 to 10.

The second thread will call the printNumbers() method and also print numbers from 1 to 10.

In the main() method:

Create two Printer objects.

Create two threads and start them to execute the printNumbers() method concurrently.

Ensure that the numbers from both threads are printed without any interruption.

- Additional attribute: department (String)

24) Create a Java program to demonstrate access modifiers (private, public, protected, and default).

1. Class Person:
    - Attributes:
        - name (private), age (public), address (protected), phoneNumber (default)
    - Methods:
        - Constructor to initialize all attributes.
        - displayDetails() (public) to display name, age, and address.
        - updatePhoneNumber() (public) to update phoneNumber.
2. Class Employee (extends Person):
    - Additional attribute: employeeId (public)
    - Override displayDetails() to include employeeId.
3. In the main() method:
    - Create an Employee object and demonstrate access to attributes and methods using different access modifiers.

25)

1) Create an application for employee management with the following classes:

a) Create an Employee class with following attributes and behaviors:

i) int empId
ii) String empName
iii) String email
iv) String gender
v) float salary
vi) void GetEmployeeDetails() -> prints employee details

b) Create one more class EmployeeDB with the following attributes and behaviors:
i) ArrayList list;
ii) boolean addEmployee(Employee e) -> adds the employee object to the collection
iii) boolean deleteEmployee(int empId) -> delete the employee object from the collection with the given empid
iv) String showPaySlip(int empId) -> returns the payslip of the employee with the given empId

---

26)

You are given a sorted integer array nums in non-decreasing order. Your task is to remove the duplicate elements in-place such that each element appears only once, and return the new length of the modified array.

You must not allocate extra space for another array; you must do this by modifying the input array in-place with O(1) extra memory.

After removing the duplicates, the first part of the array should contain the unique elements, and the remaining elements can be left as any value (underscores _ or any arbitrary values).

Example:

Input: nums = [0,0,1,1,1,2,2,3,3,4]
Output: 5, nums = [0,1,2,3,4,_,_,_,_,_]

---

27)

Write a class MathOperation which accepts 5 integers through the command line. Create an array using these parameters. Loop through the array and obtain the sum and average of all the elements and display the result.

Handle various exceptions that may arise such as:

- ArithmeticException
- NumberFormatException
- and other relevant exceptions.

---

28)

Create a base class named Fruit with the following attributes:

• name	(String)
• taste	(String)
• size	(String)

Define a method eat() in the Fruit class that prints the name and taste of the fruit.

Now, create two subclasses, Apple and Orange, that inherit from the Fruit class. Override the eat() method in each subclass to display the specific taste of that fruit.

---

29)
Given a number N, the task is to count the number of unique digits in the given number.
Examples:

Input: N = 22342   Output: 2

Explanation: The digits 3 and 4 occurs only once. Hence, the output is 2.

Input: N = 99677 Output: 1
Explanation: The digit 6 occurs only once. Hence, the output is 1.

---

30)

Given an array of positive integers nums and a positive integer target, return the minimal length of a subarray whose sum is greater than or equal to target. If there is no such subarray, return 0 instead.

Example 1: Input: target = 7, nums = [2,3,1,2,4,3] Output: 2

Example 2: Input: target = 11, nums = [1,1,1,1,1,1,1,1] Output: 0

---

31)

Write a Java program to receive an integer number as a command-line argument, and print the binary, octal, and hexadecimal equivalents of the given number.

```
Given Number : 20
Binary equivalent : 10100
Octal equivalent : 24
Hexadecimal equivalent : 14
```

---

32) Develop an online payment system that supports different payment methods.

Requirements:

1. Create an abstract class Payment with:
   - Attributes: amount, transactionID.
   - Abstract method processPayment().
   - Concrete method showTransactionDetails().
2. Create subclasses:
   - CreditCardPayment (cardNumber, CVV, expiryDate).

    - PayPalPayment (email, password).
    - UPIPayment (UPI ID).
3. Implement the processPayment() method in each subclass to handle payments uniquely.
4. Create a PaymentGateway class to process transactions dynamically.

33) Design a simple system to calculate the area of different 2D shapes using interfaces in Java.

Define an interface named Shape with a method:
Create the following classes that implement the Shape interface:

- Circle with a field radius
- Rectangle with fields length and width
- Triangle with fields base and height

Each class must implement the calculateArea() method according to the respective formula:

- Circle: $\pi \times \text{radius}^2$
- Rectangle: length × width
- Triangle: 0.5 × base × height

In the main() method, use polymorphism to create an array of Shape references and call calculateArea() on each.

34). simple banking system that handles user withdrawals, including proper use of exception handling and custom exceptions.

Requirements:

1. Create a class BankAccount with the following:
    - Field: double balance
    - Constructor to initialize balance
    - Method:

      If the withdrawal amount is greater than the balance, throw a custom exception named InsufficientFundsException.

    - Otherwise, deduct the amount from the balance.

2. Define a custom exception class:
    - Include a constructor that accepts a custom error message.

    In the main() method:

    - Create a BankAccount object with an initial balance.
    - Try to withdraw different amounts (some valid, some invalid).
    - Catch the exception and display appropriate error messages.

35)
The Citizen class should have following attributes name, id, country, sex, maritalStatus, anualIncome, and economyStatus. Validate the fields if the age is below 18 and country is not 'India' throw NonEligibleException and give proper message. Use toString method to display the citizen object in proper format. Use separate packages for Exception and application classes

36)
A. Write a Java program to remove prime numbers between 1 to 25 from ArrayList using an iterator.
B. Write a Java program to

a. create and traverse (or iterate) ArrayList using for-loop, iterator, and advance for-loop.

b. check if element(value) exists in ArrayList?

c. add element at particular index of ArrayList?

37).

Write a Java program that handles various types of exceptions while performing different operations. The application should read data from a file specified by the user, handling potential `FileNotFoundException` and `IOException`. It should also allow the user to input values for arithmetic operations and handle division by zero using `ArithmeticException`. Additionally, implement exception handling for `InputMismatchException` when the user provides invalid input, `ArrayIndexOutOfBoundsException` for accessing invalid indices in arrays, and `NullPointerException` when performing operations on `null` values. The program should provide user-friendly error messages and ensure smooth execution even when exceptions occur.

38)
Create an abstract class Person with:

- Fields: name, age
- Constructor to initialize the fields
- Abstract method:
- Create a class Student that inherits from Person:
  - Additional fields: rollNumber, course
  - Override the displayDetails() method to print all student details
- Create another class Teacher that also extends Person:
  - Additional fields: employeeId, subject
  - Override the displayDetails() method to print all teacher details
- Demonstrate encapsulation by making all fields private and using getter and setter methods.
- In the main() method:
  - Create an array of Person references (use polymorphism).
  - Store both Student and Teacher objects.
  - Call the displayDetails() method for each object using a loop.

39)

B) Develop a JAVA program to create a class named shape. Create three sub classes namely: circle, triangle and square, each class has two member functions named draw () and erase (). Demonstrate polymorphism concepts by developing suitable methods, defining member data and main program.

40) Develop a JAVA program to create an abstract class Shape with abstract methods calculateArea() and calculatePerimeter(). Create subclasses Circle and Triangle that extend the Shape class and implement the respective methods to calculate the area and perimeter of each shape.

41) A) Write a java program to Move all zeroes to end of array

Input: arr[] = {1, 2, 0, 4, 3, 0, 5, 0};

Output: arr[] = {1, 2, 4, 3, 5, 0, 0, 0};

B) Write a Java program to create an interface Sortable with a method sort() that sorts an array of integers in ascending order. Create two classes BubbleSort and SelectionSort that implement the Sortable interface and provide their own implementations of the sort() method.

42) A) Write a Java program to create a class called "Book" with instance variables title, author, and price. Implement a default constructor and two parameterized constructors:

One constructor takes the title and author as parameters.
The other constructor takes title, author, and price as parameters.
Print the values of the variables for each constructor.

B) Write a Java program to create a class called "TrafficLight" with attributes for color and duration, and methods to change the color and check for red or green.