1. Implement a program for a movie theatre that manages seat reservations. Add appropriate methods
    a. To display a seating chart
    b. To allow customers to choose and reserve seats.
    c. To quit the application
2. Design a hotel room booking system with the following functionalities:
    a. Implement a method bookRoom(int roomId) to check if a specific room is available.
    b. If the room is unavailable, throw a RoomUnavailableException.
    c. Catch the exception and provide the user with a list of alternative available rooms, if any.
3. Ronak wants to develop an app for his movie theatre that manages seat reservations to help his villagers book tickets. The theatre has a limited number of seats, and customers can choose and reserve seats based on availability. The program should display a seating chart, allow seat reservations, and provide an option to quit the application once the customer is done.
   Customers should be able to choose the menu provided in the system and proceed.
4. Create a system to handle various types of products in an e-commerce application, such as electronics, clothing, and groceries. All products should share attributes like name, price, and description. Category-specific attributes might include warranty period for electronics, size for clothing, and expiration for groceries. Develop a structure to represent these product types while maintaining their shared characteristics.
5. A weather station collects temperature data every hour throughout the day. A scientist wants to analyse this temperature data to determine the hottest, and coldest hours, the average temperatures of the day and the number of hours where the temperature exceeded the threshold of 30 degrees Celsius. Analyse this scenario and implement the same using Java and implement the same using Java
6. Imagine an e-commerce platform like Amazon, where customers can shop for a wide range of items, including electronics, clothing, groceries, and cosmetics. Each of these categories has specific details that need to be captured, such as brand and warranty for electronics, size and material for clothing, weight and expiration date for groceries, and brand and type for cosmetics. To efficiently manage the addition of items to the shopping cart, the platform's backend system uses method overloading to handle each product category individually, ensuring that the right information is captured for each type of item.
7. Develop a class "SortingClass" that has three overloaded methods to sort data of int, float, and String types. Demonstrate its working by passing an array of all the above three types. Use any sorting technique.
8. Demonstrate the concept of class and its members by writing a Java program to define a class Lamp. It can be in an on or off-state. You can turn on and turn off the lamp. Display whether the Lamp is in on or off-state
9. Smart Home Lighting System- Imagine you are designing a smart home lighting system where each room in the house is equipped with smart lamps that can be controlled programmatically. The system allows users to turn lamps on or off and check their status via a smart home app or a central control panel.
10. Design a system to manage different types of employees in an organization, such as full-time, part-time, and contract employees. All employees should share attributes like name, id, and salary. Category-specific features might include benefits for full-time employees, hourlyRate for part-time employees, and contractDuration for contract employees. Create a structure that accommodates these distinctions while maintaining shared functionality.
11. Write a program to check whether a given string is a palindrome. A string is considered a palindrome if it reads the same forward and backward, ignoring case and spaces.

12. In a banking application, users are limited to a daily withdrawal limit of $500. If a withdrawal request exceeds this limit, throw an ExceedLimitException.
13. Construct a Robot that declares three methods start(), work(), and stop(). Create a Disposal class that implements the methods in the Robot and adds the findWaste() method which returns true if the waste is wet, false otherwise. Write the associated driver class.
14. Write a Java program to represent a Complex number.
    Include member methods/constructors to:
    - Initialize a complex number to a default value of zero
    - Initialize a complex number to a user-defined value
    - Add two complex numbers by passing them as parameters and return the result.
    - Subtract two complex numbers by passing them as parameters and return the result.
    - Display a complex number.
15. Demonstrate the working of the LinkedList class in the Collection framework by performing the following tasks:
    a. Add < ‘x’, ‘y’, ‘z’> elements to the list named ‘list1’ and display the contents
    b. Add the element < ‘b’> at the end and the element < ‘w’> at the start of list1
    c. Add an element ‘a’ to list1 at an index 4
    d. Create a new list list2 that contains the sublist of list1 with index starting from 1 to3 and display list2
    e. Remove the elements of list1 that are in list2 and display list1 after removal
16. Imagine you are tasked with implementing an employee management system for a company. Describe a scenario where inheritance can be employed to model different types of employees. Create a base class Employee with attributes like name and salary, and derive two classes, Manager and Developer, each with additional attributes and methods. Discover how using inheritance enhances code maintainability and scalability in this context.
17. In a university enrollment system, students are enrolled in multiple courses. Each course can have multiple students. Design a data structure using Java collections to represent this relationship. Consider scenarios like quickly finding all students enrolled in a specific course or listing all courses a particular student is taking.
18. Write a Java program to implement a lambda expression to check if a given string is empty.
19. Create a music player system using interfaces. Design two interfaces: MusicPlayer and Playlist. The MusicPlayer interface should have methods for playing, pausing, and stopping music, while the Playlist interface should have methods for adding and removing songs from a playlist. Implement a class MyMusicPlayer that implements both interfaces. Use the interfaces to ensure that the music player supports basic playback functionality and playlist management. Explian the use of the music player by creating an instance of MyMusicPlayer and interacting with it
20. Write a Java program to implement a lambda expression to create a lambda expression to check if a number is prime.
21. Create a Calculator class with methods add, subtract, multiply, and divide. Each method should take two parameters (operands) and perform the corresponding arithmetic operation. Implement exception handling to catch and handle potential exceptions, such as ArithmeticException for division by zero or NumberFormatException for invalid input.
22. Write a Java program to implement a lambda expression to convert a list of strings to uppercase and lowercase.
23. Write a Java program that takes the value from a user and passes it into a method that checks whether it is odd, then throws a user-defined exception OddNumberException if it is else throws EvenNumber Exception. Implement user-defined exceptions and handle these exceptions carefully.
24. Implement a Java class ArraySum which includes

a) a method sumA( ) to find the sum of the elements in an array ‘a’ with elements 1,2,3,4,5 initialized by a default constructor

b) a show ( ) method to display the array elements and the sum

25. Design a simple banking application with a class BankAccount that has a method withdraw to allow users to withdraw money. Create a custom exception class InsufficientFundsException that is thrown when a withdrawal results in a negative balance. Modify the withdraw method to throw this custom exception when needed. Implement a try-catch block in a demo program to handle this custom exception and display a user-friendly message.
26. Develop a Java program that creates a simple book database. Each book is represented with an ID, title, author (First Name & last name), Genre (category – technical, Sci-Fi, fiction, Comedy etc) and Publisher's name. Define methods to perform the following tasks:
    a. Given a title, returns a status to indicate whether or not the book exists in the database.
    b. Given a string “str”, list the details of all the books whose title contains str.
    c. Given a genre, list publishers who have published books in that genre.
27. A reputed Hospital has its headquarter in Delhi. This headquarter keeps a record of the number of patients admitted to its subsidiaries like HealthIndia, IVY and Apollo hospitals which have different numbers of patients. Analyze the above scenario and design a class that displays the total number of patients admitted to the hospitals.
28. Write a Java program to find the frequency of each character in the word