Core Java Assignment 3.1

1. Design a Java program that will accept an array of integers and perform various operations on it. The program should have the following features:
    a. Accept an array of integers of size n from the user.
    b. Display the contents of the array on the console.
    c. Calculate and display the sum of all the elements in the array.
    d. Calculate and display the average of all the elements in the array.
    e. Find and display the largest element in the array.
    f. Find and display the smallest element in the array.
    g. Sort the array in ascending order and display the sorted array on the console.

    You should use static methods to implement the above features.

    Example Input:

    Array: {5, 10, 15, 20, 25}

    Example Output:

    Contents of Array: 5 10 15 20 25
    Sum of Array: 75
    Average of Array: 15
    Largest element in Array: 25
    Smallest element in Array: 5
    Sorted Array: 5 10 15 20 25

2. create a program that helps a restaurant manage their menu. The restaurant has 10 dishes on their menu, and they want to be able to add, remove, and modify dishes easily.

    Your task is to create a Java program that uses arrays to store the menu items and their corresponding prices. The program should have a menu that allows the user to perform the following actions:
    a. View the current menu with prices
    b. Add a new dish to the menu
    c. Remove a dish from the menu
    d. Modify the price of a dish on the menu
    e. Exit the program

    When the user chooses to add a new dish, the program should prompt the user for the name of the dish and its price. The new dish should be added to the end of the menu array.

    When the user chooses to remove a dish, the program should prompt the user for the index of the dish they want to remove. The dish at that index should be removed from the menu array.

When the user chooses to modify the price of a dish, the program should prompt the user for the index of the dish they want to modify and the new price. The price of the dish at that index should be updated in the menu array.
After each action, the program should display the updated menu with prices. When the user chooses to exit the program, the program should display a farewell message and terminate.

3. Design an airline reservation system using Java Arrays. The system should allow users to make reservations for flights, view existing reservations, and cancel reservations.
   The system should have the following functionalities:
   a. **User registration**: The system should allow users to register with their name, email, and contact number.
   b. **Flight availability**: The system should have a pre-defined list of flights and their available seats. The user should be able to view the list of flights and their availability.
   c. **Seat reservation**: The user should be able to select a flight and reserve a seat. The system should allocate the seat to the user and update the availability of seats for the selected flight.
   d. **View reservations**: The user should be able to view their existing reservations along with the flight details.
   e. **Cancel reservations**: The user should be able to cancel their reservation. The system should release the reserved seat and update the availability of seats for the selected flight.

   The system should be designed using Java Arrays and should include proper error handling and exception handling mechanisms to handle cases such as invalid inputs, seat availability, etc.

4. Design a reservation system for a movie theater that uses Java arrays to keep track of available seats and sold seats. The system should allow customers to select seats and make reservations, as well as allow the theater staff to view the current seating arrangement and generate reports on ticket sales.

   Requirements:
   a. The system should display a seating chart with available and sold seats.
   b. Customers should be able to select seats by row and column numbers.

c. The system should prevent customers from selecting already sold seats.
d. Customers should be able to purchase multiple seats in a single transaction.
e. The system should generate a unique reservation number for each transaction.
f. The theater staff should be able to view a report of ticket sales, including the number of seats sold and the revenue generated.
g. The theater staff should be able to view a report of available seats.
h. The system should provide error messages for invalid user input, such as selecting a non-existent seat.

5. Develop a Java program to manage inventory items in a warehouse. The warehouse has a fixed number of storage shelves, and each shelf can store a maximum of 10 items. You need to create a program that allows the user to add and remove items from the shelves, and also display the current contents of each shelf.

   Requirements:

   The program should use a static array to represent the shelves in the warehouse. The program should have a menu-driven interface that allows the user to choose from the following options:

   a. Add item to shelf
   b. Remove item from shelf
   c. Display contents of all shelves
   d. Exit

   When the user chooses to add an item to a shelf, the program should prompt the user to enter the name of the item and the shelf number (1-10).

   When the user chooses to remove an item from a shelf, the program should prompt the user to enter the shelf number (1-10) and remove the item from that shelf.

   When the user chooses to display the contents of all shelves, the program should print out the contents of each shelf in a tabular format, showing the shelf number and the items stored on that shelf.

   The program should handle invalid user input gracefully and display appropriate error messages.

Example Output:

Welcome to the Warehouse Inventory Management System!

Please select an option from the following menu:

Add item to shelf

Remove item from shelf

Display contents of all shelves

Exit

Option: 1

Please enter the name of the item: Widget

Please enter the shelf number (1-10): 3

Item 'Widget' added to shelf 3.

Please select an option from the following menu:

Add item to shelf

Remove item from shelf

Display contents of all shelves

Exit

Option: 3

Current contents of all shelves:

Shelf 1: [empty]

Shelf 2: [empty]

Shelf 3: Widget

Shelf 4: [empty]

Shelf 5: [empty]

Shelf 6: [empty]

Shelf 7: [empty]

Shelf 8: [empty]

Shelf 9: [empty]

Shelf 10: [empty]

Thank you for using the Warehouse Inventory Management System!

6. Create a sales management program that will keep track of salespersons and their sales figures. The program should allow the user to add new salespersons, update existing ones, and view the total sales for each salesperson.

The program should be implemented using Java arrays to store the salespersons' data. Each salesperson's data should include their name, ID, and an array of their sales figures for the past 12 months.

The program should have the following functionality:

a. **Add a new salesperson**: The user should be prompted to enter the salesperson's name, ID, and sales figures for each of the past 12 months. The program should create a new salesperson object and add it to the array of salespersons.
b. **Update an existing salesperson**: The user should be prompted to enter the ID of the salesperson they wish to update. The program should then allow the user to update the sales figures for each of the past 12 months for that salesperson.
c. **View total sales for a salesperson**: The user should be prompted to enter the ID of the salesperson they wish to view the total sales for. The program should then calculate the total sales for that salesperson and display it to the user.
d. **View total sales for all salespersons**: The program should calculate the total sales for each salesperson and display them in a table.
e. **Exit the program**: The user should be able to exit the program at any time.

7. Create a Java program to perform various operations on matrices. The program should prompt the user to enter the number of rows and columns for two matrices, and then allow the user to input the elements of the matrices. The program should then display a menu of options for various matrix operations that the user can perform.

   The available operations should include:

   a. Addition of two matrices
   b. Subtraction of two matrices
   c. Multiplication of two matrices
   d. Transpose of a matrix

   After the user selects an operation, the program should perform the operation and display the result. If the operation is not possible, the program should display an appropriate error message.