Assignment 3

Learning:

- Basics of Class in Java
- Handling exceptions
- ArrayList, HashSet

1. Write a Java program to build a calculator for basic arithmetic operations. The calculator accepts only integer inputs. In certain cases, other characters may be entered as input. Implement the calculator such that if the input entered is not a digit[0-9], then it should continue with the appropriate exception handling. Also, handle the incorrect operation “division by zero” in your code.

Input format:

3 lines of input.

1st line is the operator. 2nd line 1st argument. 3rd line 2nd argument.

Output format:

Print result if operation is valid. Else, show the appropriate exception.

Sample Input and Output:

Input:

+ 5 7

Output:

12

Input:

/ 10 0

Output:

ArithmeticException: Division by zero handled. Enter a valid argument.

2. Create a class “Queue” with two operations enqueue() and dequeue(). Use enqueue() to insert elements into a queue and dequeue() to delete elements from the queue. During execution, handle the underflow exception condition by printing “EmptyQueue”.

Input Format :

One line input:

● If the operation is enqueue() then add the given element to the queue list.

● If the operation is dequeue() then delete the element from the queue list.

Output Format :

One line output:

● If the operation is enqueue(), print “Success”.

● If the operation is dequeue(), print the element, if an exception occurs then prin“EmptyQueue”.

Sample Input and Output :

Note: E indicates enqueue and D indicates dequeue

Input:

E 10 E 34 E 60 D

D D D

Output:

Success Success Success 10
34
60 EmptyQueue

3. Create a class ‘Book’ that represents a book in a library. The class should have the following properties:
• title (String)
• author (String)
• price (double)
• yearPublished (int)
Your class should have:
1. A constructor that initializes all the properties of the class.
2. A method displayDetails() that prints the details of the book.
A main method in another class named ‘Library’ where you create three different Book objects using the constructor and then call the displayDetails() method for each object. Create a Java program for this scenario.
Sample Input:
Title: The Great Gatsby
Author: F. Scott Fitzgerald
Price: $10.99
Year Published: 1925
Title: To Kill a Mockingbird
Author: Harper Lee
Price: $7.99
Year Published: 1960
Title: 1984
Author: George Orwell
Price: $8.99
Year Published: 1949
Sample output:
Title: The Great Gatsby
Author: F. Scott Fitzgerald
Price: $10.99
Year Published: 1925
Title: To Kill a Mockingbird
Author: Harper Lee
Price: $7.99
Year Published: 1960
Title: 1984
Author: George Orwell
Price: $8.99
Year Published: 1949

4. Write a Java program to develop a small library management system for a local library. Your task is to implement a feature that tracks and manages the books in the library using an **ArrayList**. Each book has a title, an author, and a unique identifier (ID). The library should be able to perform the following operations:
addBook(): Add a new book to the library.
removeBook(): Remove a book from the library based on its ID.
listAllBook(): Display all the books currently in the library.
findBook(): Search for a book by its title and display its details.
exit(): Exit from the program
Create a Book class with the following attributes:
id(unique identifier, integer)
title(string)
author (String)
Create a Library class that uses an ArrayList<Book> to manage the collection of books. Implement methods for the operations mentioned above.

5. Write a java program to manage a list of unique employee IDs which utilizes a **HashSet** to ensure that each employee ID is unique and to provide functionalities to add, remove, and display all employee IDs. Employee Manager should be able to perform the following operations:
addEmployeeId(): Uses HashSet's add method which returns true if the ID was added successfully and false if it was a duplicate.
removeEmployeeId(): Uses HashSet's remove method which returns true if the ID was successfully removed and false if it did not exist.
displayEmployeeIds(): Iterates through the HashSet to print each ID. It also handles the case when the set is empty.
exit( ): Exit from the program
In the main method, create an instance of your EmployeeManager class and demonstrate the use of the methods by performing the following operation which is mentioned above.