Lab 07 – Interfaces and Exception Handling

1. Illustrate the usage of multiple inheritance in JAVA using interfaces.
2. Illustrate the usage of exception handling using **try-catch-finally** blocks in JAVA.
3. Illustrate the usage of user defined exceptions using **throws** and **throw**.
4. Interfaces and Exception handling for Stack:
    a. Package: MyStack
        i. Design an interface **Stack** with 2 methods: push() and pop().
        ii. Design two custom exception classes for **StackUnderFlow** and **StackOverFlow**
        iii. Design a class **FixedStack** which implements a fixed length version of Stack.
        iv. Design a class **DynamicStack** which implements a growable version of the stack.
    b. Import the necessary classes from MyStack package and utilize them in the main method to create a fixed and a dynamic stack object with appropriate exception handling methods.
5. Design a class **Employee** with the following instructions:
    a. Instance Variables - String name, int age, double grossSalary, float takeHomeSalary, char grade
    b. Methods - input(), display()
    c. Exception Handling – **IO Exception**
    d. **Wrapper classes** to read keyboard inputs and parse them into their types

6. Design a class **Student** and provide a custom exception **SeatsFilledException**, which is thrown when the student registration number (always a unique number) is > XX25 (XX is the last two digits of the year of joining).

7. Design an interface **Series** with the following methods:

    a. getNext (returns the next number in series)

    b. reset (restart the series)

    c. setStart (to set the value from which the series should start)

    Design a class **ByTwos** that will implement the methods of the interface series such that it generates a series of numbers, each two greater than the previous one.

8. Design a program by the following steps:

    a. **Student** class

        i. Getters and setters for rollNo, marks

    b. **Sports** interface

        i. Getter for grade

    c. **Result** class

        i. **Inherits** Student class

        ii. **Implements** Sports

        iii. Calculates final marks based on sports + student superclass