Programming Assignment: Object-Oriented Programming in C++

Instructions:

- Implement each program in C++.
- Follow proper object-oriented design principles.
- Use meaningful variable names and comments for better understanding.
- Test each program with multiple cases.

Exception Handling

1. **Basic Exception Handling** Write a C++ program that takes two integers as input and performs division. Handle exceptions for division by zero and invalid input types.
2. **Custom Exception Handling** Create a class `AgeException` and throw an exception if the user enters an age less than 18. Handle this exception and display an appropriate message.
3. **Multiple Catch Blocks** Write a program that takes an integer from the user and throws different exceptions for negative, zero, and positive numbers. Catch the exceptions separately and display corresponding messages.
4. **Exception Handling in Constructors** Create a class `Student` with attributes `name` and `marks`. If `marks` is negative or greater than 100, throw an exception. Handle the exception appropriately.

File Handling

5. **Writing to a File** Write a program to take student details (name, roll number, marks) as input and store them in a file named "students.txt".
6. **Reading from a File** Modify the above program to read student details from "students.txt" and display them on the console.
7. **Appending Data to a File** Extend the previous program to allow appending new student records to "students.txt" without overwriting existing data.
8. **File Copy Program** Write a C++ program that copies the contents of one file into another. Handle exceptions if the source file does not exist.

Templates

9. **Function Template** Write a function template `findMax()` that takes two parameters of any data type (int, float, double, char) and returns the larger value.
10. **Class Template** Implement a class template `Array` that allows the user to create an array of any data type. Provide functions to insert, display, and find the maximum element in the array.

Programming Assignment: Student Record Management System

Objective:

- To implement **Exception Handling** to manage runtime errors.
- To use **File Handling** for storing and retrieving student records.
- To utilize **Templates** to allow flexibility in handling different data types.

Problem Statement:

Design a **Student Record Management System** that allows users to:

1. **Add student records** (Name, Roll Number, Marks).
2. **Display all records** from a file.
3. **Search for a student record** by Roll Number.
4. **Handle errors** using **Exception Handling** (e.g., file not found, invalid input, division by zero).
5. **Use Templates** for handling student marks in different data types (int, float).

Tasks to Implement:

1. Define a Template Class `Student`

- Implement a **template class** `Student<T>` that stores:
  - `string name` (Student Name)
  - `int rollNo` (Roll Number)
  - `T marks` (Marks: can be `int` or `float`)
- Implement functions:
  - **`void getData()`** → To take input from the user.
  - **`void showData()`** → To display student details.

2. File Handling Operations

- Implement functions to **write** student data into a file (`students.txt`).
- Implement functions to **read** student records from the file.
- Implement a **search function** to find a student by Roll Number.

3. Exception Handling

- Handle **file errors** (e.g., file not opening).
- Handle **invalid input** (e.g., entering non-numeric data for marks).
- Handle **division by zero** (optional: if any percentage calculations are included).