Programming Paradigms Lab Assignment (CS 2273)

Assignment 3 (Part 1): Class and Object concepts using C++

Time 2 weeks

Develop the below mentioned programs using C++. Apply following features wherever applicable.

- **All C++ concepts as mentioned in "Assignment 2" and the following**
- Access Specifier
- Constant data member
- Default Constructor
- Overloaded Constructor
- Copy Constructor
- 'this' pointer
- Assignment(=) operator
- Static member/member function
- Operator overloading : addition(+) operator
- Class Template

Problems

- Write a program to take input of N number of students information such as Name, Age, Department and Year. The student information should be shored in array of `Student` **Class**. Print those information in the console. Develop member functions of `Student` Class such as `ReadStudentData(...)`, `PrintStudentData(...)` for this purpose.

- Develop a program that supports functionalities of a variable length stack. Let us call it GrowingStack. The GrowingStack is supposed to be initialized with some predefined initial size to store integer elements. If the space in GrowingStack is exhausted by populating elements, it inflates by adding extra space in it. However, GrowingStack should have some upper bound of its size and once that is reached, GrowingStack cannot inflate any more. Note that linklist shouldn't be used in this program.

- Create a GrowingStack class with required data members and member functions. Note that each array can have different maximum size based on initialization.
- Develop the following stack routines as member functions of the class -
    - `push(...)` : Push element(s) into a particular stack
    - `pop(...)` : Pop an element from the stack
    - `getMaxSize(...)` : Should notify the maximum number of elements the stack can store
    - `currentSize(...)` : Should notify the current number of elements in the stack
    - `isEmpty(...)` : Should notify if the stack is empty
    - `inflate(...)`: Inflates the stack size by adding extra space in it.
    - Any other suitable function(s) you think relevant
- Demonstrate the basic stack functionality using above routines
- Demonstrate that multiple stack can be instantiated and can co-exists independently
- Let's assume there are two stacks as stack1 and stack2, and initially, those are populated with some elements. Create another stack3 and populate elements from stack1 and stack2 as long as those are not empty. In the odd count, the element will be populated from stack1, and in the even count, the element will be populated from stack2. If any stack gets empty first, then the remaining elements from another stack will be popped and pushed in stack3.
- Demonstrate that a new GrowingStack can be created from an existing class using "copy constructor"
- Demonstrate that an existing GrowingStack content can be updated from the content of another Stack using "=" operator.

3. Modify the Problem 2 to implement a generic GrowingStack to store any kind of data types such as `int, short, float, double` or `struct` using class template concept.

4. Implement problem number 3 or 8 (based on whichever you have done earlier) from assignment number 2 using the concept of class.