

Programming Paradigms Lab Assignment (CS453)

Assignment Sheet 3 : Class and Object concepts using C++

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

- All C++ concepts as mentioned in “Assignment Sheet 2”
- 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

- 1 Write a program to take input of N number of students information such as Name, Age, Department and Year. The student information should be stored in array of `Student` Class. Print those information in the console. Develop member functions of `Student` Class such as `ReadStudentData(...)`, `PrintStudentData(...)` for this purpose.
- 2 Develop a program of finite(limited) stack where elements to be stored is `int`.
 - Create a Stack class named `MyStack` with required data members and member functions. Note that each stack can have different maximum size based on initialization.
 - Develop the below stack routines as member function of the class
 - `Push(...)` : Push element(s) into a particular stack
 - `Pop(...)` : Pop an element from the stack
 - `MaxSize(...)` : 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
 - `Display(...)` : Should display the current snapshot of the stack content
 - Demonstrate the basic stack functionality using above routines

- Demonstrate that multiple stack can be instantiated and can co-exists independently
-
- Demonstrate that a new stack can be created from an existing class using “copy constructor”
- Demonstrate that an existing stack content can be updated from the content of another stack using “=” operator.
-

3 Modify the Problem 2 to implement a generic stack to store any kind of data types such as int, short, float, double or struct.

\

4 Modify the Problem 2 with below support -

- Provide a mechanism to maintain the count of stacks created
- Provide a mechanism to update a stack by adding two existing stacks using “+ operator”.

Example code :

```
MyStack aStackA, aStackB, aStackC;
```

```
...
```

```
aStackC = aStackA + aStackB;
```

- Provide a mechanism to restrict that maximum N stacks can be created
-

5. Design class for Linked list and use it in your program to add two different degree of polynomial expressions.

6. Design a class 'Set' to implement following operations.

- i. Union
- ii. Intersection
- iii. Complement
- iv. Set difference
- v. Test equal sets
- vi. Print Cartesian product (for two sets and more than two sets)
- vii. Display power set
- viii. Test if a set is distinct or not.
- ix. For a set with numeric values, find mean, variance, standard deviation, and median.

7. Design a class 'Matrix' to implement following operations.

- i. Addition
- ii. Multiplication
- iii. Test equal matrices
- iv. Transpose
- v. Create sub matrix depending on user's choice
- vi. Inverse

8. Design a class 'Graph' to implement following operations. Use vertices and edges as objects of set class (in Question No. 6) .

- i. Create sub graph depending on user's choice
- ii. Graph union
- iii. Graph intersection
- iv. Check for a disconnected vertex
- v. Find degree of any node
- vi. At least one path from given two vertices.

9. A screen information is nothing but an image made of thousands of dots known as pixels. For example resolution of 800x600 means the device prepares screen information using $800 \times 600 = 4,80,000$ dots. Create a class 'screen' that contains a 2D array that represents pixels on screen and to be displayed on terminal. Provide the following facilities

- i. A 2D array as data member that stores screen pixels.
- ii. A member function that erases screen.
- iii. User can draw a Line, Pattern or any graphic symbol on the screen such that logic for any graphic symbol or pattern belongs to separate class.

Attempt this when above questions are solved:

iv. On supplying adjacency matrix, user can see graph displayed on the screen. Consider a vertex as a rectangular area of dots and an edge as a straight line. Try to use vertical and horizontal lines as many as possible. For labeling numbers on vertices, use vertical and horizontal straight lines as shown in figure.

