FAST School of Computing

Object Oriented Programming – Spring 2023

Cyber Security Department

LAB 08

Classes in C++

Learning Outcomes

In this lab you are expected to learn the following:

- ✗ Pointer data members of Classes
- ✗ Arrays and Dynamic Allocation of Class Objects
- ✗ Friend Functions

Note: Plagiarism(from some else or internet) in any 1 question will lead to zero marks in the whole lab task.

Problem 1:

The absence of array bounds checking in C++ is a source of potential hazard. Write a class which will perform bounds checking on integer array.

Write a class **IntegerList** with private member variables as:

int *list; // A pointer to an int . This member points to the dynamically allocated array of integers (which you will be allocating in constructor).

int numElements; // An integer that holds the number of elements in the dynamically allocated array.

int size;// Size of the list

And **public member functions**

1. **IntegerList(int)**// The class constructor accepts an int argument that is the number of elements to allocate for the array. The array is allocated, and all elements are set to zero.
2. **IntegerList(IntegerList &copy); //** Make a copy constructor
3. **bool isValid(int);**// This function validates a subscript into the array. It accepts a subscript value as an argument and returns Boolean true if the subscript is in the range 0 through numElements - 1. If the value is outside that range, Boolean false is returned.
4. **void setElement(int, int**); // The setElement member function sets a specific element of the list array to a value. The first argument is the element subscript, and the second argument is the value to be stored in that element. The function uses **isValid()** to validate the subscript. If subscript is valid, value is stored at that index, if an invalid subscript is passed to the function, the program aborts.
5. **int getElement(int);**// The getElement member function retrieves a value from a specific element in the list array. The argument is the subscript of the element whose value is to be retireved The function should use **isValid()** function to validate the subscript. If the subscript is valid, the value is returned. If the subscript is invalid, the program aborts. To abort, you can use exit**(EXIT_FAILURE)** function. **(include library <cstdlib>** for it**).**

Problem 2:

Write a definition of class **FaceBook** profile to implement basic properties of facebook profile. Your class should have the member variables:-

- **name: A string that represents user's name**
- **email: user email address**
- **gender: A characters that represents user's gender**
- **contact: char array that represents user's contact number**

Write the **member functions** for the member variables:

- Default and parameterized constructors
- Setters and getters of the member variables
- Write a function that updates the contact number

Write a **global function GenderCoun**t function that will receive the **array of object** and return the count of gender.

Problem 3:

Design a class Complex for handling Complex numbers and include the following **private** data members:

- **real:** a double
- **imaginary:** a double

The class has the following **member functions.**

1. A **constructor** initializing the number with **default parameters.**
2. Overloaded Constructors.
   - Complex(double r, double i)
   - Complex(Complex & copy) // copy constructor

3. **Getters** and **Setters** of the class data members as given below
   - void setReal(double r)
   - double getReal()
   - void setImaginary(double i)

- double getImaginary()

4. Write the following **Friend Functions** to perform the following operations:
    - **friend Complex addComplex(Complex &c1,Complex &c2)**

      It adds both complex numbers and returns newly generated complex number.
    - **friend Complex subComplex(Complex &c1)**

      It subtracts both complex numbers and returns newly generated complex number.
    - **friend Complex mulComplex(Complex &c1)**

      It multiplies both complex numbers and returns newly generated complex number. (a+bi)(c+di) = (ac−bd) + (ad+bc)i

Submission Details:

2. Save single .cpp file with your roll no and lab number e.g. i22-XXXX_Lab8.cpp
3. Take screen shot of running test cases of tasks.
4. Zip the .cpp file and screen shots (Do not create .rar file) with roll no and lab no. e.g. i22-XXXX_Lab8.zip.
5. Submit the zip file on google class room.