**Lab1 Topic: A structure, its member functions, and its data members.**

**Concept/Rule 1**: All members of a structure are public by default.

**Concept/Rule 2**: A variable/object of a structure declared outside of the structure

definition can access all of the structure members.

**Concept/Rule 3:** A structure can have more than one constructor.

A constructor is a special member function that has no return

type and its name is the same as the structure name.

A constructor should be used to initialize the data members of

the structure.

**Concept/Rule 4:** The default constructor does not have any parameter.

Define the default constructor to initialize data members of a

structure.

**Concept/Rule 5:** The default constructor is called when an object/variable of the

structure is created in the following syntax:

StructureTagName objectName;

**Your Tasks:**

1. **Compile** the following program, lab1Program.cpp, and run it to see the output. Capture a screenshot of the program output.

#include <iostream>

#include <string>

using namespace std;

struct EmailAddressBook

{

string name; //data member

string emailAddress; // data member

EmailAddressBook( )

{ cout << "The default constructor is called " << endl;

name = "Blank ";

emailAddress = "Blank ";

}

EmailAddressBook(string n, string a )

{ cout <<"In the overloaded constructor. "<< endl;

name = n;

emailAddress = a;

}

void Display( )

{

cout << "Name = " << name << "Email address = " << emailAddress << endl;

}

};

int main( )

{

EmailAddressBook myAssociate1; // LINE 1

myAssociate1.Display(); // LINE 2

cout << " In the main ,Name1 = " << myAssociate1.name<< "Email address 1= "<< myAssociate1.emailAddress << endl; // LINE 3

EmailAddressBook myAssociate2("Jack Smith", "jsmith@msn.com"); // LINE 4

myAssociate2.Display(); // LINE 5

cout << " In the main ,Name 2 = " << myAssociate2.name<< "Email address 2= "<< myAssociate2.emailAddress << endl; // LINE 6

EmailAddressBook myAssociateArray[3]; // LINE 7

for (int i = 0 ; i < 3 ; i++) // LINE 8

{ cout << "Enter your associate name and his/her email address # " << i + 1 << endl; // LINE 9

cin >> myAssociateArray[i].name >>myAssociateArray[i].emailAddress ; // LINE 10

}// LINE 11

cin.get( ); // Wait for any character to be entered.

}

1. Add the statement **private:** above the statement **string name;** . As a result, all members have become private. Run the program to see errors. Copy/paste error messages.

Error 1 error C2248: 'EmailAddressBook::EmailAddressBook' : cannot access private member declared in class 'EmailAddressBook' f:\c++ class 162\the lab work\trytolab1\lab1program.cpp 30 TrytoLAb1

Error 2 error C2248: 'EmailAddressBook::Display' : cannot access private member declared in class 'EmailAddressBook' f:\c++ class 162\the lab work\trytolab1\lab1program.cpp 31 TrytoLAb1

Error 3 error C2248: 'EmailAddressBook::name' : cannot access private member declared in class 'EmailAddressBook' f:\c++ class 162\the lab work\trytolab1\lab1program.cpp 32 TrytoLAb1

Error 4 error C2248: 'EmailAddressBook::emailAddress' : cannot access private member declared in class 'EmailAddressBook' f:\c++ class 162\the lab work\trytolab1\lab1program.cpp 32 TrytoLAb1

Error 5 error C2248: 'EmailAddressBook::EmailAddressBook' : cannot access private member declared in class 'EmailAddressBook' f:\c++ class 162\the lab work\trytolab1\lab1program.cpp 33 TrytoLAb1

Error 6 error C2248: 'EmailAddressBook::Display' : cannot access private member declared in class 'EmailAddressBook' f:\c++ class 162\the lab work\trytolab1\lab1program.cpp 34 TrytoLAb1

Error 7 error C2248: 'EmailAddressBook::name' : cannot access private member declared in class 'EmailAddressBook' f:\c++ class 162\the lab work\trytolab1\lab1program.cpp 35 TrytoLAb1

Error 8 error C2248: 'EmailAddressBook::emailAddress' : cannot access private member declared in class 'EmailAddressBook' f:\c++ class 162\the lab work\trytolab1\lab1program.cpp 35 TrytoLAb1

Error 9 error C2248: 'EmailAddressBook::EmailAddressBook' : cannot access private member declared in class 'EmailAddressBook' f:\c++ class 162\the lab work\trytolab1\lab1program.cpp 36 TrytoLAb1

Error 10 error C2248: 'EmailAddressBook::name' : cannot access private member declared in class 'EmailAddressBook' f:\c++ class 162\the lab work\trytolab1\lab1program.cpp 39 TrytoLAb1

Error 11 error C2248: 'EmailAddressBook::emailAddress' : cannot access private member declared in class 'EmailAddressBook' f:\c++ class 162\the lab work\trytolab1\lab1program.cpp 39 TrytoLAb1

Answer the following question:

**Can the myAssociate1 object declared in the main( ) access private members?**

No.

1. Remove the statement **private:** and compile the program again. Capture a screenshot of the program output.
2. Comment out LINE 2 thru LINE 11 in the main( ) function. Compile and run the programto see what is called.
3. Uncomment LINE 2. Compile and run the programto see what is called.
4. Uncomment LINE 3. Compile and run the programto see what is called.
5. Uncomment LINE 4. Compile and run the programto see what is called.
6. Uncomment LINE 5 and LINE 6. Compile and run the programto see what is called.
7. Uncomment LINE 7. Compile and run the programto see what is called.
8. Add a for loop in the main to display three sets of values in myAssociateArray. Save your program as **Lab1ProgramYourname.cpp.** Capture a screenshot of the program output.
9. **Analyze the codes and assess your understandings.**

Answer the following questions:

* 1. What does the program do when LINE 1 is executed?

*The structure EmailAddressBook is accessed to get a member named myAssociate1.*

* 1. What does the program do when LINE 4 is executed?

*The structure EmailAddressBook is accessed to get a member named*

*myAssociate2.*

* 1. What values are resided inside myAssociate1 when the program is run?

*“Blank” until user input is entered*

* 1. What values are resided inside myAssociate2 when the program is run?

“*Blank” until user input is entered a second time*

* 1. How many objects are there in myAssociateArray?

*3 objects*

* 1. Is it true that each structure object has a copy of data members? If yes, what makes you agree with the statement? *Yes: The compiler puts that data in memory for access.*
  2. Identify line(s) of code that prove each Concept/Rule.

*Line 7 for number of objects in the array; Line 2 and 5 for values of MyAssociate1 and 2, respectively.*

1. **Do the following Lab 1 Assessment Programming Project:**

Design a structure, HouseItems, that contains 2 member functions, two constructors

(default constructor and an overloaded constructor), and three data members.

It is your option to come up with desired members of the structure. Define each

member function the way you wish. Test the structure in a main ( ) function. **Save the**

**program as Assessment1ProjectYourname.cpp**. Run the program and capture a

screenshot of the program output.

**What To Turn In**

Submit :

**Lab1ResultYourname.doc, Lab1ProgramYourname.cpp**, and

**Assessment1ProjectYourname.cpp**.

Grading Rubric: (Possible points: 30)

|  |  |
| --- | --- |
| Lab 1 Completion of Tasks #1-#9 | 9 points |
| Lab 1 Completion of Task #10 | 2 points |
| Lab 1 Completion of Task #11 | 7 points |
| Lab1 Assessment Programming Project done in Task 12:   * The program runs. * The program proves Lab 1 Concept/Rule 1. * The program proves Lab 1 Concept/Rule 2. * The program proves Lab 1 Concept/Rule 3. * The program proves Lab 1 Concept/Rule 4. * The program proves Lab 1 Concept/Rule 5. | 2 points  2 points  2 points  2 points  2 points  2 points |