**National University of Computer and Emerging Sciences**

Logo

Description automatically generated with medium confidence

**Lab Manual 08**

**Object Oriented Programming**

|  |  |
| --- | --- |
| Course Instructor | Miss Abeeda |
| Lab Instructor (s) | Miss Siddiqua Nayyer  Mr. Dilawar Shabbir |
| Section | A |
| Semester | Spring 2021 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

## Objectives

After performing this lab, students shall be able to:

* Operator Overloading
* Non-member functions ( + Revision of static data members)
* Stream insertion and extraction  (<<, >>)
* Class Relationships (Dependency, Association, Aggregation)

**TASK 1:**

We implemented a class named **Complex** in the previous class.

Overload << and >> operators this time for the same class and show its functionality inside main.

**Sample Run:**

|  |  |
| --- | --- |
| **Driver.cpp** | **Output** |
| int main()  {  Complex C1;  Complex C2(5,6);  Complex C3;  cout<<"Input a complex number"<<endl;  cin>>C3;  cout<<C1<<C2<<C3;  if(C1==C2)  cout<<"C1 == C2"<<endl;  else  cout<<"C1 != C2"<<endl;  if(C1!=C3)  cout<<"C1 != C3"<<endl;  else  cout<<"C1==C3"<<endl;  Complex C4= C2 - C3;  cout<<C4;  system("pause");  return 0;  } | A picture containing text  Description automatically generated |

**TASK 2:**

Implement a class named **Month**.

The class should have the following private members:

* **name** A string object that holds the name of a month, such as “January,” “February,” etc.
* **monthNumber** An integer variable that holds the number of the month. For example, January would be 1, February would be 2, etc. Valid values for this variable are 1 through 12.

In addition, provide the following member functions:

• A **default constructor** that sets monthNumber to 1 and name to “January.”

• A **constructor** that accepts the name of the month as an argument. It should set name to the value passed as the argument and set monthNumber to the correct value.

• **Prefix and postfix overloaded ++** operator functions that increment monthNumber and set name to the name of next month. If monthNumber is set to 12 when these functions execute, they should set monthNumber to 1 and name to “January.”

• **Prefix and postfix overloaded − −** operator functions that decrement monthNumber and set name to the name of previous month. If monthNumber is set to 1 when these functions execute, they should set monthNumber to 12 and name to “December.”

**TASK 3:**

Implement a class called **Box**. The **Box** class will have three data members:

* double length; // Length of a box
* double breadth; // Breadth of a box
* double height; // Height of a box

You need to implement the following:

* A default constructor.
* An overloaded constructor.
* All setters for length, breadth, height.
* All getters for length, breadth, height.
* Print Function

There should be a static data member

* static int objectCount; // Increases every time object is created

Write member functions as follow:

* + 1. static int getCount();
    2. double Volume();
    3. double Area();

1. Write a suitable main() function to test the functionality of the static members and functions.
2. Write non-member function print\_surface\_area(). 🡺 SA=2lw+2lh+2hw
3. Another non-member function double\_data\_members(length, breadth, height).
4. Write a suitable main() function to test all the functions of the **Box** class and to test the functionality of the non-member function.

**TASK 4:**

Implement a class **Teacher** that has following members:

int EmployeeID

char\* Name

Create another class **Student** with these data members:

Char \* roll number

Char \* name

Teacher\* TeacherList; //List of all teachers who are teaching this student

* Create constructors, destructors.
* A display function of both the classes such that it shows Teachers are associated with Students.

**Note:**

* Deallocate all dynamically allocated memory, if any.
* Make student.h, student.cpp, teacher.h, teacher.cpp and driver file.
* Do not use any string class built-in functions except for strlen(), if required.
* Follow all the code indentation, naming conventions and code commenting guidelines.