**National University of Computer & Emerging Sciences, Karachi**

**Computer Science Department**

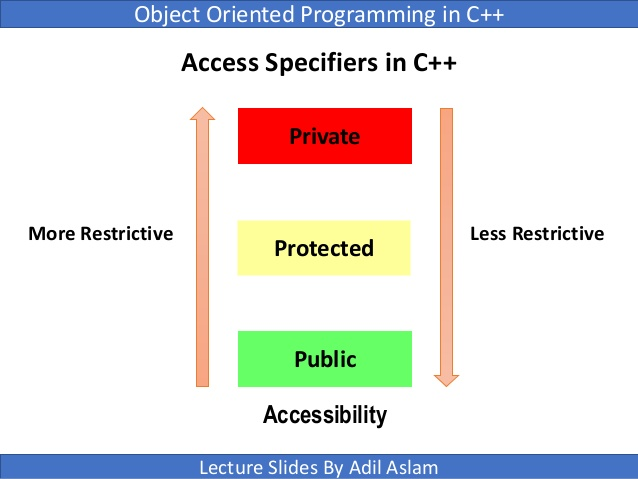
**Fall 2022, Lab Manual - 05**

|  |  |
| --- | --- |
| **Course Code: CL-1004** | **Course : Object Oriented Programming Lab** |
| **Instructor(s) :** | **Shahroz Bakht** |

# Contents:

* Access Modifiers
* Static Keyword
* Constant Keyword
* Member initializer list

**Introduction to Access Modifiers in C++:**



Access modifiers is the techniques that is applied to members of class to restrict their access beyond the class.

In C++, access modifiers can be achieved by using three keywords:

**Public**

**Private**

**Protected**

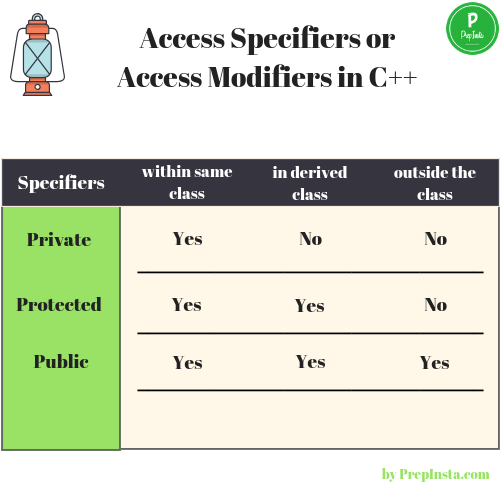
**Types of Access Modifiers in C++**

There are three types of Access Modifiers in C++

1. Public
2. Private
3. Protected

**Public members** can be accessed anywhere i.e. inside or outside the class but within the program only,

**Private members** can be accessed inside the class only,

**Protected members** are like the private they can be accessed in the child class/derived class. Let’s look at these modifiers with examples:

**Public:**

|  |
| --- |
| // C++ program to demonstrate public access modifier  #include<iostream>  using namespace std;  // class definition  class Circle  {    public:          double radius;          double  compute\_area()          {            return 3.14\*radius\*radius;       }       };  int main ()  {   Circle obj; |

As there are no restrictions in public modifier, we can use the (.)dot operator directly accesses member functions and data.

|  |
| --- |
| // accessing public data member outside class      obj.radius = 5.5;      cout << "Radius is: " << obj.radius << "\n";      cout << "Area is: " << obj.compute\_area();      return 0; } |

**Output:**

Radius is: 5.5

Area is: 94.985

**Private:**

Only the member functions or the [friend functions](https://www.geeksforgeeks.org/friend-class-function-cpp/) are allowed to access the private data members of a class.

The example below has an error let’s find out:



// C++ program to demonstrate private

// access modifier

#include<iostream>

using namespace std;

class Circle

{

// private data member

private:

double radius;

// public member function

public:

double compute\_area()

{ // member function can access private

// data member radius

return 3.14\*radius\*radius;

}

};

// main function

int main()

{

// creating object of the class

Circle obj;

// trying to access private data member

// directly outside the class

obj.radius = 1.5;

cout << "Area is:" << obj.compute\_area();

return 0;

}

// C++ program to demonstrate private

// access modifier

#include<iostream>

using namespace std;

class Circle

{

// private data member

private:

double radius;

// public member function

public:

void compute\_area(double r)

{ // member function can access private

// data member radius

radius = r;

double area = 3.14\*radius\*radius;

cout << "Radius is: " << radius << endl;

cout << "Area is: " << area;

}

};

// main function

int main()

{ // creating object of the class

Circle obj;

// trying to access private data member

// directly outside the class

obj.compute\_area(1.5);

return 0; }

**Output**:

Radius is: 1.5

Area is: 7.065

**Protected:**

#include <bits/stdc++.h>

using namespace std;

// base class

class Parent

{ // protected data members

protected:

int id\_protected;

};

// sub class or derived class from public base class

class Child : public Parent

{ public:

void setId(int id)

{ // Child class is able to access the inherited protected data members of base class

id\_protected = id;

}

void displayId()

{

cout << "id\_protected is: " << id\_protected << endl;

}

};

// main function

int main() {

Child obj1;

// member function of the derived class can access the protected data members of the base class

obj1.setId(81);

obj1.displayId();

return 0;

}

**Output:**

id\_protected is: 81

const**Keyword in C++**

Constant is something that doesn't change. In C language and C++ we use the keyword const to make program elements constant. **const** keyword can be used in many contexts in a C++ program. It can be used with:

1. Variables
2. Function arguments and return types
3. Class Data members
4. Class Member functions
5. Objects

**Constant Variables in C++**

int main

{

const int i = 10;

const int j = i + 10; // works fine

i++; // this leads to Compile time error

}

If you make any variable as constant, using const keyword, you cannot change its value. Also, the constant variables must be initialized while they are declared.

In the above code we have made i as constant, hence if we try to change its value, we will get compile time error. Though we can use it for substitution for other variables.

**Static Keyword in C++**

Static is a keyword in C++ used to give special characteristics to an element. Static elements are allocated storage only once in a program lifetime in static storage area. And they have a scope till the program lifetime. Static Keyword can be used with following,

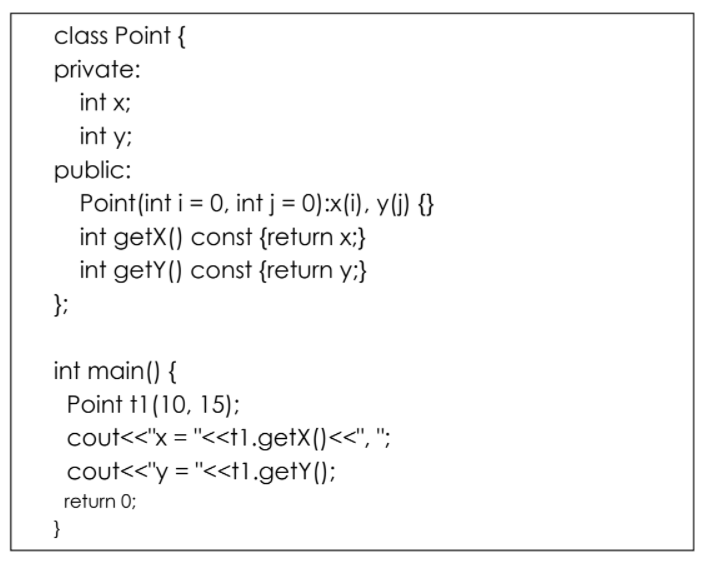
1. Static variable in functions
2. Static Class Objects
3. Static member Variable in class
4. Static Methods in class

**MEMBER INITIALIZATION LIST**

Initializer List is used in initializing the data members of a class. The list of members to be initialized is indicated with constructor as a comma-separated list followed by a colon. Following is an example that uses the initializer list to initialize x and y of Point class.

**Uses:**

* For initialization of non-static const data members.
* For initialization of reference members.
* For initialization of member objects which do not have default constructor.
* For initialization of base class members.
* When constructor’s parameter name is same as data member.
* For Performance reasons.



**LAB ACTIVITIES**

**Task # 1**

Write a program that contains variables to hold employee data like; employeeCode,

employeeName and date Of Joining. Write a function that assigns the user defined values to

these variables. Write another function that asks the user to enter current date and then checks

if the employee tenure is more than three years or not. Call the functions in main.

Write a program that contains variables to hold employee data like; employeeCode,

employeeName and date Of Joining. Write a function that assigns the user defined values to

these variables. Write another function that asks the user to enter current date and then checks

if the employee tenure is more than three years or not. Call the functions in main.

Write a program that contains variables to hold employee data like; employeeCode,

employeeName and date Of Joining. Write a function that assigns the user defined values to

these variables. Write another function that asks the user to enter current date and then checks

if the employee tenure is more than three years or not. Call the functions in main.

Write a program that contains a class name Employee which contains the following private data members:

1. employeeCode
2. employeeName
3. dateOfJoining

Perform the following operations

1. Write a function that assigns the user defined values to these variables.
2. Create 4 objects of the class Employee and pass the user defined values through a parameterized constructor.
3. Write a function that asks the user to enter current date and then checks if the employee tenure is more than three years or not with employee details. Also display the number of employees that have tenure more than 3 years.

**Task # 2**

Rectangle is an object that represents a specific shape. Analyze the concept and identify the data

members and methods that should be included in Rectangle class

Rectangle, Triangle, Square and Circle are objects that represents a specific shape. Analyze the concept and identify the data members and methods that should be included in Rectangle, Triangle, Square and Circle class respectively.

**Task # 3**

Write a program to implement the matrix using a class. Create two objects of class matrix. Each matrix should have 2 rows and two columns. The operations to be performed on the matrices are:

1. Addition of matrix
2. Subtraction of matrix
3. Multiplication of a matrix
4. Display all the results

**Task # 4**

Create a class for a new ice cream vendor called LeCream. The management of LeCream has decided that they are going to sell their ice cream in 4 different flavors namely

1. chocolate
2. vanilla
3. strawberry
4. mango

Carefully design the program by observing the following rules.

1. LeCream is charging Rs 100 for two scoops and Rs 150 for three scoops. Hence you will need a function to determine the number of scoops and based on that the price. If a user enters more than three scoops your program should display invalid input and it should exit.
2. LeCream allows its customers to purchase a vanilla wafer with their ice cream. If the customer wants to purchase the wafer he will have to pay an additional Rs 10. This amount should be added to the total amount payable by the user.
3. The program should show a menu that asks the customer for his requirements and then displays the final payable amount with full details about the flavor, number of scoops and wafer.

**Task # 05**

Write a C++ program that creates a class called laptop. The data members of the class are private which are brand (string), model (string), serial (int), color (string), price (float), processor speed (float), RAM (int), screen size(float). Create member function that will set the individual values. Since the RAM can be upgraded therefore create a function that allows you to upgrade the RAM only. In the end, create a function that will display all the data members.

**Task # 06**

Write a program that creates a class called number. Your class will have two data members namely num (float) and result (int). To find the factorial of the entered number you will need to design three functions as follows:

* Function to determine if a number is a whole number or not
* Function to determine if the number is positive or not
* Function to find the actual factorial
* Function to display the number and its factorial

Remember that to find the factorial the number must of positive and a whole number. So if any of these conditions are not met then you cannot determine the factorial.