Dr. D. Y. Patil Educational Federation's
Dr. D. Y. Patil College of Engineering & Innovation
Approved by AICTE, Affilated to SPPU Pune

Department of Computer Engineering

LAB MANUAL
OOP and Computer Graphics Laboratory
(SE)
Semester I

Prepared By:
Prof. Snehal R. Mangale

OOP & Computer Graphics Laboratory

Course Code	Course Name	Teaching Scheme (Hrs./ Week)	Credits
210247	OOP and Computer Graphics	4	2

Course Objectives:

To understand basics of Computer Graphics, apply various methods and techniques for implementing line-circle drawing, projections, animation, shading, illumination and lighting using concepts of Object Oriented Programming.

Course Outcomes:

On completion of the course, learner will be able to–

CO1: U**nderstand** and **apply** the concepts like inheritance, polymorphism, exception handling and generic structures for implementing reusable programming codes.

CO2: Analyze the concept of file and **apply** it while storing and retrieving the data from secondary storages.

CO3: Analyze and **apply** computer graphics algorithms for line-circle drawing, scan conversion and filling with the help of object oriented programming concepts.

CO4: Understand the concept of windowing and clipping and apply various algorithms to fill and clip polygons.

CO5: Apply logic to implement, curves, fractals, animation and gaming programs.

Operating System recommended: 64-bit Open source Linux or its derivative

Programming tools recommended: Use suitable programming language/Tool for implementation

Table of Contents

6	Write C++ program using STL for sorting and searching user defined records such as personal records (Name, DOB, Telephone number etc) using vector container. OR Write C++ program using STL for sorting and searching user defined records such as Item records (Item code, name, cost, quantity etc) using vector container.	CO1, CO2	24-25
7	Write a program in C++ to use map associative container. The keys will be the names of states and the values will be the populations of the states. When the program runs, the user is prompted to type the name of a state. The program then looks in the map, using the state name as an index and returns the population of the state.	CO1, CO2	27-28

Lab Assignment No.	1
Title	Implement a class Complex which represents the Complex Number data type. Implement the following 1. Constructor (including a default constructor which creates the complex number 0+0i). 2. Overload operator+ to add two complex numbers. 3. Overload operator* to multiply two complex numbers. 4. Overload operators << and >> to print and read Complex Numbers.
Roll No.	
Class	SE
Date of Completion	
Subject	OOP & CG Lab
Assessment Marks	
Assessor's Sign	

LAB ASSIGNMENT 1

Problem Statement

Implement a class Complex which represents the Complex Number data type. Implement the following operations:

1. Constructor (including a default constructor which creates the complex number 0+0i).

2. Overloaded operator+ to add two complex numbers.

3. Overloaded operator* to multiply two complex numbers.

4. Overloaded << and >> to print and read Complex Numbers.

Objectives:

1. Understand the concept of constructors and how to design it.
2. Understand the OOP concept of operator overloading.

Outcomes:

1. Will be able to design constructors for classes designed in programming.
2. Will be able to program using operator functions.

Software Requirements:

1. 64-bit Open source Linux or its derivative
2. Open Source C++ Programming tool like G++/GCC.

Theory:

C++ Class

A class is the collection of related data and function under a single name. A C++ program can have any number of classes. When related data and functions are kept under a class, it helps to visualize the complex problem efficiently and effectively.

Class is defined in C++ programming using keyword class followed by identifier(name of class). Body of class is defined inside curly brackets an terminated by semicolon at the end in similar way as structure.

```
class class_name
  {
  // some data
  // some functions
  };
```

Keyword private makes data and functions private and keyword public makes data and functions public. Private data and functions are accessible inside that class only whereas, public data and functions are accessible both inside and outside the class. This feature in is known as data hiding.

C++ Objects

When class is defined, only specification for the object is defined. Object has same relationship to class as variable has with the data type. Objects can be defined in similary way as structure is defined.

class_name variable name;

Data Member & Data Functions

The data within the class is known as data member. The function defined within the class is known as member function.Data members and member functions can be accessed in similar way the member of structure is accessed using member operator(.).

object_name.data_memeber;

Constructors:

Constructors are the special type of member function that initialises the object automatically when it is created Compiler identifies that the given member function is a constructor by its name and return type. Constructor has same name as that of class and it does not have any return type.

Constructor can be overloaded in similar way as function overloading. Overloaded constructors have same name(name of the class) but different number of argument passed. Depending upon the number and type of argument passed, specific constructor is called. Since, constructor are called when object is created. Argument to the constructor also should be passed while creating object.

A object can be initialized with another object of same type.- Default Copy Constructor. If you want to initialise a object A3 so that it contains same value as A2.

```
Area A3(A2);     /* Copies the content of A2 to A3 */
                        OR,
Area A3=A2;      /* Copies the content of A2 to A3 */
```

Operator Overloading:

The meaning of operators are already defined and fixed for basic types like: int, float, double etc in C++ language. For example: If you want to add two integers then, + operator is used. But, for user-defined types(like: objects), you can define the meaning of operator, i.e, you can redefine the way that operator works. For example: If there are two objects of a class that contain string as its data member, you can use + operator to concatenate two strings. Suppose, instead of strings if that class contains integer data member, then you can use + operator to add integers. This feature in C++ programming that allows programmer to redefine the meaning of operator when they operate on class objects is known as operator overloading.

To overload a operator, a operator function is defined inside a class as:

```
class class_name
{
........ ...... ......
public:
   return_type operator sign (argument/s)
   {
     ...... ..... ......
   }
........ ...... ......
};
```

The return type comes first which is followed by keyword **operator**, followed by operator sign,i.e., the operator you want to overload like: +, <, ++ etc. and finally the arguments is passed. Then, inside the body of you want perform the task you want when this operator function is called.This operator function is called when, the operator(sign) operates on the object of that class class_name.

Algorithm

1. Design a class Complex which represents the Complex Number data type.
2. Create a custructor which creates the complex number 0+0i

Complex(): real(0), imag(0){ }

3. Use operator overloading to overload operator+ to add two complex numbers.

Complex operator + (Complex c2)

4. Use operator overloading to overload operator* to multiply two complex numbers.

Complex operator * (Complex c3)

5. Use operator overloading to overload operator<< and >> to print and read Complex Numbers.

friend ostream& operator << (ostream &out, Complex &c); //overloading '<<' operator

friend istream& operator >> (istream &in, Complex &c); //overloading '>>' operator

Conclusion:

Lab Assignment No.	2
Title	Develop a program in C++ to create a database of student's information system containing the following information: Name, Roll number, Class, Division, Date of Birth, Blood group, Contact address, Telephone number, Driving license no. and other. Construct the database with suitable member functions. Make use of constructor, default constructor, copy constructor, destructor, static member functions, friend class, this pointer, inline code and dynamic memory allocation operators-new and delete as well as exception handling.
Roll No.	
Class	SE
Date of Completion	
Subject	OOP & CG Lab
Assessment Marks	
Assessor's Sign	

LAB ASSIGNMENT 2

Problem Statement

Develop an object oriented program in C++ to create a database of student information system containing the following information: Name, Roll number, Class, division, Date of Birth, Blood group, Contact address, telephone number, driving license no. etc Construct the database with suitable member functions for initializing and destroying the data viz constructor, default constructor, Copy constructor, destructor, static member functions, friend class, this pointer, inline code and dynamic memory allocation operators-new and delete.

Objectives:

1. Understand the concept of constructor and its use
2. Understand the concept friend function
3. Understand the concept of copy constructor.
4. Understand the use of static and inline function
5. Understand the concept of exception handling

Outcomes:

1. Will be to make use of constructors
2. Will be able to design a program using friend functions.
3. Will be able to design a program using exception handling.

Software Requirements:

1. 64-bit Open source Linux or its derivative
2. Open Source C++ Programming tool like G++/GCC.

Theory:

Constructors:

A special method of the class that will be automatically invoked when an instance of the class is created is called as constructor. Following are the most useful features of constructor.

1) Constructor is used for Initializing the values to the data members of the Class.

2) Constructor is that whose name is same as name of class.

3) Constructor gets Automatically called when an object of class is created.

4) Constructors never have a Return Type even void.

5) Constructor is of Default, Parameterized and Copy Constructors.

The various types of Constructor are as follows:-

Constructors can be classified into 3 types

1. Default Constructor

2. Parameterized Constructor

3. Copy Constructor

1. Default Constructor:- Default Constructor is also called as Empty Constructor which has no arguments and It is Automatically called when we creates the object of class but Remember name of Constructor is same as name of class and Constructor never declared with the help of Return Type. Means we can‟t declare a Constructor with the help of void Return Type. , if we never Pass or declare any Arguments then this called as the Copy Constructors.

2. Parameterized Constructor: - This is another type constructor which has some Arguments and same name as class name but it uses some Arguments So For this We have to create object of Class by passing some Arguments at the time of creating object with the name of class. When we pass some Arguments to the Constructor then this will automatically pass the Arguments to the Constructor and the values will retrieve by the Respective Data Members of the Class.

3. Copy Constructor: - This is also another type of Constructor. In this Constructor we pass the object of class into the Another Object of Same Class. As name Suggests you Copy, means Copy the values of one Object into the another Object of Class .This is used for Copying the values of class object into an another object of class So we call them as Copy Constructor and For Copying the values We have to pass the name of object whose values we wants to Copying and When we are using or passing an Object to a Constructor then we must have to use the & Ampersand or Address Operator.

Destructor: As we know that Constructor is that which is used for Assigning Some Values to data Members and For Assigning Some Values this May also used Some Memory so that to free up the Memory which is Allocated by Constructor, destructor is used which gets Automatically Called at the End of Program and we doesn‟t have to Explicitly Call a Destructor and Destructor Cant be Parameterized or a Copy This can be only one Means Default

Destructor which Have no Arguments. For Declaring a Destructor we have to use ~tiled Symbol in front of Destructor

Friend Class A friend class can access private and protected members of other class in which it is declared as friend. It is sometimes useful to allow a particular class to access private members of other class. For example, a LinkedList class may be allowed to access private members of Node.

A static member function can access only the names of static members, enumerators, and nested types of the class in which it is declared. Suppose a static member function f() is a member of class X . The static member function f() cannot access the nonstatic members X or the nonstatic members of a base class of X
Inline function is one of the important feature of C++. So, let's first understand why inline functions are used and what is the purpose of inline function?

When the program executes the function call instruction the CPU stores the memory address of the instruction following the function call, copies the arguments of the function on the stack and finally transfers control to the specified function. The CPU then executes the function code, stores the function return value in a predefined memory location/register and returns control to the calling function. This can become overhead if the execution time of function is less than the switching time from the caller function to called function (callee). For functions that are large and/or perform complex tasks, the overhead of the function call is usually insignificant compared to the amount of time the function takes to run. However, for small, commonly-used functions, the time needed to make the function call is often a lot more than the time needed to actually execute the function's code. This overhead occurs for small functions because execution time of small function is less than the switching time.

C++ provides an inline functions to reduce the function call overhead. Inline function is a function that is expanded in line when it is called. When the inline function is called whole code of the inline function gets inserted or substituted at the point of inline function call. This substitution is performed by the C++ compiler at compile time. Inline function may increase efficiency if it is small.
The syntax for defining the function inline is:

```
inline return-type function-name(parameters)
{
   // function code
}
```

Conclusion:

Lab Assignment No.	3
Title	Imagine a publishing company which does marketing for book and audio cassette versions. Create a class publication that stores the title (a string) and price (type float) of publications. From this class derive two classes: book which adds a page count (type int) and tape which adds a playing time in minutes (type float). Write a program that instantiates the book and tape class, allows user to enter data and displays the data members. If an exception is caught, replace all the data member values with zero values.
Roll No.	
Class	SE
Date of Completion	
Subject	OOP & CG Lab
Assessment Marks	
Assessor's Sign	

LAB ASSIGNMENT 3

Problem Statement

Imagine a publishing company that markets both book and audiocassette versions of its works. Create a class publication that stores the title (a string) and price (type float) of a publication. From this class derive two classes: book, which adds a page count (type int), and tape, which adds a playing time in minutes (type float). Each of these three classes should have a getdata() function to get its data from the user at the keyboard, and a putdata() function to display its data. Write a main() program to test the book and tape classes by creating instances of them, asking the user to fill in data with getdata(), and then displaying the data with putdata().

Objectives:

To learn the concept of Class, Object,Data Hiding , Inheritance

Outcomes:

The performer will be able to design a program in c++ using concept Class, Object,Data Hiding , Inheritance

Software Requirements:

1. 64-bit Open source Linux or its derivative
2. Open Source C++ Programming tool like G++/GCC.

Theory:

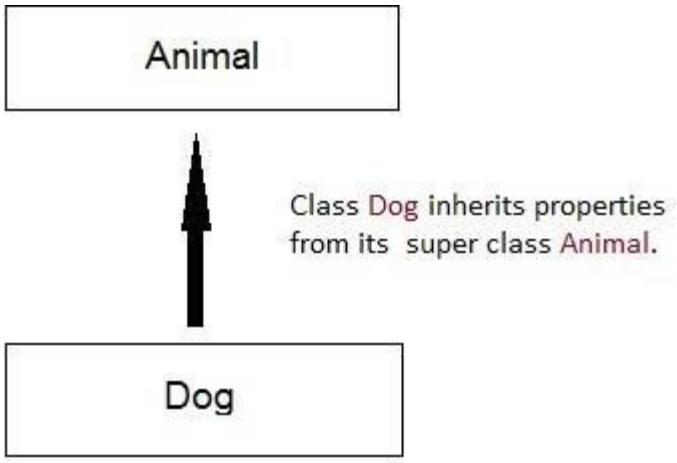
Inheritance

Inheritance is the capability of one class to acquire properties and characteristics from another class. The class whose properties are inherited by other class is called the **Parent** or **Base** or **Super** class. And, the class which inherits properties of other class is called **Child** or **Derived** or **Sub** class. Inheritance makes the code reusable. When we inherit an existing class, all its methods and fields become available in the new class, hence code is reused.

class Subclass_name : access_mode Superclass_name

While defining a subclass like this, the super class must be already defined or atleast declared before the subclass declaration.Access Mode is used to specify, the mode in which the properties of superclass will be inherited into

subclass, public, privtate orprotected.



Depending on Access modifier used while inheritance, the availability of class members of Super class in the sub class changes. It can either be private, protected or public.

Base class	Derived Class		
	Public Mode	**Private Mode**	**Protected Mode**
Protected	Protected	Private	Protected

Algorithm:

1. Design a Base Class publication which contains fiels to store title and price..
2. Design two derived classes to hold properties of Tape and Book respectively.
3. Create a functions for getting data and putting data in the base class as well as in the derived class.
4. Create main() program to test the book and tape classes by creating instances of them, asking the user to fill in data with getdata(), and then displaying the data with putdata().

Conclusion:

Lab Assignment No.	4
Title	Write a C++ program that creates an output file, writes information to it, closes the file, open it again as an input file and read the information from the file.
Roll No.	
Class	SE
Date of Completion	
Subject	OOP & CG Lab
Assessment Marks	
Assessor's Sign	

LAB ASSIGNMENT 4

Problem Statement

Write a C++ program that creates an output file, writes information to it, closes the file and open it again as an input file and read the information from the file.

Objectives:

To learn the concept of File Handling in c++ programming

Outcomes:

The performer will be able to design a program in c++ using concept File Handling

Software Requirements:

1. 64-bit Open source Linux or its derivative
2. Open Source C++ Programming tool like G++/GCC.

Theory:

A file must be opened before you can read from it or write to it. Either the **ofstream** or **fstream** object may be used to open a file for writing and ifstream object is used to open a file for reading purpose only.

```
void open(const char *filename, ios::openmode mode);
```

Here, the first argument specifies the name and location of the file to be opened and the second argument of the **open()** member function defines the mode in which the file should be opened.

When a C++ program terminates it automatically closes flushes all the streams, release all the allocated memory and close all the opened files. But it is always a good practice that a programmer should close all the opened files before program termination.

```
void close();
```

While doing C++ programming, you write information to a file from your program using the stream insertion operator (<<) just as you use that operator to output information to the screen. The only difference is that you use an **ofstream** or **fstream** object instead of the **cout** object.

You read information from a file into your program using the stream extraction operator (>>) just as you use that operator to input information from the keyboard. The only difference is that you use an **ifstream** or **fstream** object instead of the **cin** object.

Algorithm:

1. Create a file "telelphone.txt"
2. Read input from the user .
3. Write the data to the file using oftsream.
4. Read the input from the file using ifstream
5. Display the contents

Conclusion:

Lab Assignment No.	5
Title	Write a function template for selection sort that inputs, sorts and outputs an integer array and a float array.
Roll No.	
Class	SE
Date of Completion	
Subject	OOP & CG Lab
Assessment Marks	
Assessor's Sign	

LAB ASSIGNMENT 5

Problem Statement

Write a function template selectionSort. Write a program that inputs, sorts and outputs an int array and a float array.

Objectives:

To learn the use of Templates and sorting in c++ programming

Outcomes:

The performer will be able to design a program in c++ using Templates and Sorting .

Software Requirements:

1. 64-bit Open source Linux or its derivative
2. Open Source C++ Programming tool like G++/GCC.

Theory:

Templates

Templates are the foundation of generic programming, which involves writing code in a way that is independent of any particular type.

A template is a blueprint or formula for creating a generic class or a function. The library containers like iterators and algorithms are examples of generic programming and have been developed using template concept.

The general form of a template function definition is shown here:

```
template <class type> ret-type func-name(parameter list)
{
  // body of function
}
```

Here, type is a placeholder name for a data type used by the function. This name can be used within the function

definition.

Just as we can define function templates, we can also define class templates. The general form of a generic class declaration is shown here:

template <class type> class class-name {

Here, type is the placeholder type name, which will be specified when a class is instantiated. You can define more than one generic data type by using a comma-separated list.

Algorithm:

1. Create Template template <typename T>
2. Create a function to perform sorting.

void Sort(T* const array,int size){

3. Create a function to display array

void printarray(T* const array,int size)

Conclusion:

Lab Assignment No.	6
Title	Write C++ program using STL for sorting and searching user defined records such as personal records (Name, DOB, Telephone number etc) using vector container. OR Write C++ program using STL for sorting and searching user defined records such as Item records (Item code, name, cost, quantity etc) using vector container.
Roll No.	
Class	SE
Date of Completion	
Subject	OOP & CG Lab
Assessment Marks	
Assessor's Sign	

LAB ASSIGNMENT 6

Problem Statement

Write C++ program using STL for Sorting and searching with user-defined records such as Person Record (Name, birth date, telephone no), item record (item code, item name, quantity and cost)

Objectives:

To learn the use of standard template library (STL) in c++ programming

Outcomes:

The performer will be able to design a program in c++ using standard template library (STL) .

Software Requirements:

1. 64-bit Open source Linux or its derivative
2. Open Source C++ Programming tool like G++/GCC.

Theory:

STL Container List:

Array and Vector are contiguous containers, i.e they store their data on continuous memory, thus the insert operation at the middle of vector/array is very costly (in terms of number of operaton and process time) because we have to shift all the elements, linked list overcome this problem. Linked list can be implemented by using the list container.

list<T> lst;

Construct an empty list lst which can hold values of type T.

list<T> lst(inIterBegin, inIterEnd);
Construct lst containing values from the range[inIterBegin,inIterEnd) in another (not necessarily list) container, but whose component type is the same as the component type of lst.

lst.begin()
Return an iterator (or const_iterator) pointing to the first component of lst.

lst.end()
Return an iterator (or const_iterator) pointing to one-past-the-last component oflst.

lst.rbegin()
Return a reverse_iterator (or const_reverse_iterator) pointing to the last component of lst.

lst.rend()
Return a reverse_iterator (or const_reverse_iterator) pointing to one-before-the-first component of lst.

lst.size()
Return a value of type size_type giving the number of values currently in lst.

lst.empty()
Return true if lst is empty (contains zero values); otherwise return false.

lst.insert(iter, val)
Insert val into lst immediately before the value pointed to by iter, and return aniterator pointing at the new component with value val.

lst.unique()
Remove all but one of any group of consecutive components containing duplicate items.

lst.merge(otherList)
Merge the list contained in otherList with the invoking list and clear otherList. (Both lists are assumed to be ordered and the result is also ordered.)

lst.reverse()
Reverse the order of all components of a list.

lst.sort()
Sort the components of a list into ascending order.

Conclusion:

Lab Assignment No.	7
Title	Write a program in C++ to use map associative container. The keys will be the names of states and the values will be the populations of the states. When the program runs, the user is prompted to type the name of a state. The program then looks in the map, using the state name as an index and returns the population of the state.
Roll No.	
Class	SE
Date of Completion	
Subject	OOP & CG Lab
Assessment Marks	
Assessor's Sign	

LAB ASSIGNMENT 7

Problem Statement:

Write a program in C++ to use map associative container. The keys will be the names of states and the values will be the populations of the states. When the program runs, the user is prompted to type the name of a state. The program then looks in the map, using the state name as an index and returns the population of the state.

Objectives:

1. Understand the concept of map in associate container
2. Understand the concept friend function

Outcomes:

1. Will be to make use of constructors
2. Will be able to design a program using friend functions.
3. Will be able to design a program using exception handling.

Software Requirements:

3. 64-bit Open source Linux or its derivative
4. Open Source C++ Programming tool like G++/GCC.

Theory:

Maps are associative containers that store elements in a mapped fashion. Each element has a key value and a mapped value. No two mapped values can have same key values.

Some basic functions associated with Map:
begin() – Returns an iterator to the first element in the map
end() – Returns an iterator to the theoretical element that follows last element in the map
size() – Returns the number of elements in the map
max_size() – Returns the maximum number of elements that the map can hold
empty() – Returns whether the map is empty
pair insert(keyvalue, mapvalue) – Adds a new element to the map
erase(iterator position) – Removes the element at the position pointed by the iterator
erase(const g)– Removes the key value ‘g’ from the map
clear() – Removes all the elements from the map

Creating a Map in C++ STL

Maps can easily be created using the following statement :

map<key_type, value_tye> map_name;

Copy
This will create a map with key of type Key_type and value of type value_type. One thing which is to remembered is that key of a map and corresponding values are always inserted as a pair, you cannot insert only key or just a value in a map.

Conclusion: