# Practical No.1

**Aim :** Write a program on data encapsulation

**INLAB AIM :** Define a class to represent a bank account.Include name of depositor,account number,type of account,balance amount in the account as data members and member functions to assign initial values,to deposit an amount,to withdraw an amount after checking the balance,to display name and balance.Write a main program to perform above operation on bank account.

**OBJECTIVES:**

- To analyze the importance of data encapsulation in object oriented programming
- To demonstrate the use of class,object and data encapsulation in C++

**Class Diagram:**

**POSTLAB AIM :** Define a class calculator which includes member function for addition,subtraction,multiplication and division of two numbers.Write a main program to create four function calculator using switch case.

**OBJECTIVE:**

- To apply data encapsulation on object calculator
- To build calculator with basic arithmatic operation using the feature of data encapsulation in C++

**Class Diagram:**

**AIM :** Write a program on data encapsulation

**INLAB**

**AIM** : Define a class to represent a bank account.Include name of depositor,account number,type of account,balance amount in the account as data members and member functions to assign initial values,to deposit an amount,to withdraw an amount after checking the balance,to display name and balance.Write a main program to perform above operation on bank account.

**OBJECTIVES:**

- To analyze the importance of data encapsulation in object oriented programming
- To demonstrate the use of class,object and data encapsulation in C++

**THEORY :** In Object Oriented Programming(OOP), data is treated as a critical element and does not allow it to flow freely. It bounds data closely to the functions that operate on it and protects it from accidental modification from outside functions. Encapsulation is the process of combining data and functions into a single unit called class. Classes provide definitions and instance that is objects are created from class.Using the method of encapsulation, the programmer cannot directly access the data. Data is only accessible through the functions existing inside the class. Data encapsulation led to the important concept of data hiding. Data hiding is the implementation details of a class that are hidden from the user.

The key strength behind Data Encapsulation in C++ is that the keywords or the access specifiers can be placed in the class declaration as public, protected or private. A class placed after the keyword public is accessible to all the users of the class. The elements placed after the keyword private are accessible only to the methods of the class. In between the public and the private access specifiers, there exists the protected access specifier. Elements placed after the keyword protected are accessible only to the methods of the class or classes derived from that class.

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 What is mean by data encapsulation?
- Q 2 What is the use of data encapsulation in object oriented programming?
- Q 3 What are the advantages of data encapsulation?
- Q 4 What are the different access specifiers?
- Q 5 How to implement data encapsulation in C++?

**REFERENCE:**


- *https://www.cse.iitb.ac.in/~rkj/cs101/lect13.pdf*
- *Object Oriented Programming in C++ by Robert Lafore*


## POSTLAB

**AIM**: Define a class calculator which includes member function for addition,subtraction,multiplication and division of two numbers.Write a main program to create four function calculator using switch case.

**OBJECTIVES:**

- To apply data encapsulation on object calculator
- To build calculator with basic arithmatic operation using the feature of data encapsulation in C++

**THEORY :**In an object-oriented language, code and data may be combined in such a way that a self-contained "black box" is created. When code and data are linked together in this fashion, an object is created. In other words, an object is the device that supports encapsulation.

For example:

```
class Calculator{
private: float a,b,ans;
public:
void addition();
void subtraction();
void division();
void multiplication();
};
```

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 What is mean by class?
- Q 2 What is mean by object?
- Q 3 What is the difference between private and public access specifier?
- Q 4 What is the difference between structure and class in c++?
- Q 5 What is namespace?

**REFERENCE:**


- *The Complete reference C++ by Herbert Schildt*
- *Object Oriented Programming in C++ by Robert Lafore*

# Practical No.2

**Aim :** Write a program using constructor

**INLAB AIM :** Write a program using constructor

**OBJECTIVES:**

- To demonstrate the use of constructor in C++
- To build stack using the concept of data encapsulation and constructor

**Class Diagram:**

**AIM :** Write a program using constructor

**INLAB**

**AIM** : Define a class stack.Include member function push,pop for addition,deletion of element of a stack and a constructor to initialize top of the stack.Write a main program to perform push and pop operation on a stack.

**OBJECTIVES:**

- To demonstrate the use of constructor in C++
- To build stack using the concept of data encapsulation and constructor

**THEORY :** Automatic initialization is carried out using a special member function called a constructor. A constructor is a member function that is executed automatically whenever an object is created.As a general rule, an object's constructor is called when the object comes into existence, and an object's destructor is called when the object is destroyed. A local object's constructor is executed when the object's declaration statement is encountered. Global objects have their constructors execute before main( ) begins execution. Global constructors are executed in order of their declaration, within the same file.A constructor is declared and defined as :

```
class Sample
{
int a, b;
public:
Sample(){
a=0; b=0;
}
};
```

When a class contains a constructor like the defined one above, it is guaranteed that an object created by class will initialized automatically

For example:

```
Sample S;
```

It will not only creates the object S of type Sample but also initializes its data members a & b to zero. A constructor with no parameter is called as the default constructor. If no constructor is defined then the complier supplies a default constructor.

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 What is mean by constructor?
- Q 2 When the constructor get executed?
- Q 3 What is the use of constructor?
- Q 4 What are the characteristics of constructor?
- Q 5 What is mean by default constructor?

**REFERENCE:**


- *http://www.it.iitb.ac.in/frg/wiki/images/1/1a/CS101x_S445_Default_and_Copy_Constructors_IIT_Bombay.pdf*
- *Object Oriented Programming in C++ by Robert Lafore*

# Practical No.3

**Aim :** Write a program using array of object

**INLAB AIM :** Write a program using array of object

**OBJECTIVES:**

- To construct array of object in C++
- To demonstrate the use of array of object in C++

**Class Diagram:**

**AIM :** Write a program using array of object

**INLAB**

**AIM** : Create a class employee which include member function for getting and displaying details of an employee.Write a main program to get and display the data of atleast 5 employee using the concept of array of object.

**OBJECTIVES:**

- To construct array of object in C++
- To demonstrate the use of array of object in C++

**THEORY :** In C++, it is possible to have arrays of objects. The syntax for declaring and using an object array is exactly the same as it is for any other type of array.For example stack s[5]; creates an array of class stack having size of an array to be 5. s[0],s[1],s[2],s[3],s[4] are the five objects present in array of object.

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 What is mean by array of object?
- Q 2 How to create array of object?
- Q 3 What is the use of array of object?
- Q 4 How to declare array of object?
- Q 5 Explain the importance of array of object.

**REFERENCE:**


- *The Complete reference C++ by Herbert Schildt*
- *Object Oriented Programming in C++ by Robert Lafore*

# Practical No.4

**Aim :** Write a program on operator overloading

**INLAB AIM :** Write a program to overload arithmetic operators for objects that has been allocated memory dynamically

**OBJECTIVES:**

- To determine the use of operator overloading and dynamic allocation operator in C++
- To create program by overloading existing C++ operator

**Class Diagram:**

**POSTLAB AIM :** Write a program to oveload prefix and postfix version of increment and decrement operator

**OBJECTIVE:**

- To demonstrate the working of increment and decrement operator using the concept of operator overloading
- To develop and analyze the difference while overloading postfix and prefix notation of increment and decrement operator

**Class Diagram:**

**AIM :** Write a program on operator overloading

**INLAB**

**AIM** : Write a program to overload arithmetic operators for objects that has been allocated memory dynamically

**OBJECTIVES:**

- To determine the use of operator overloading and dynamic allocation operator in C++
- To create program by overloading existing C++ operator

**THEORY :** In C++, one can overload most operators so that they perform special operations relative to classes that you create. For example, a class that maintains a stack might overload + to perform a push operation and – – to perform a pop. When an operator is overloaded, none of its original meanings are lost. Instead, the type of objects it can be applied to is expanded.The ability to overload operators is one of C++'s most powerful features. It allows the full integration of new class types into the programming environment. After overloading the appropriate operators, one can use objects in expressions in just the same way that you use C++'s built-in data types. Operator overloading also forms the basis of C++'s approach to I/O.

C++ provides two dynamic allocation operators: new and delete. These operators are used to allocate and free memory at run time. Dynamic allocation is an important part of almost all real-world programs. As explained in Part One, C++ also supports dynamic memory allocation functions, called malloc( ) and free( ). These are included for the sake of compatibility with C. However, for C++ code, you should use the new and delete operators because they have several advantages.

The new operator allocates memory and returns a pointer to the start of it. The delete operator frees memory previously allocated using new. The general forms of new and delete are shown here:

p_var = new type;

delete p_var;

Here, p_var is a pointer variable that receives a pointer to memory that is large enough to hold an item of type *type*.Since the heap is finite, it can become exhausted. If there is insufficient availablememory to fill an allocation request, then new will fail and a bad_alloc exception will be generated. This exception is defined in the header <new>.

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 What is mean by operator overloading?
- Q 2 What are the dynamic allocation operators?
- Q 3 What is the difference between malloc() and new?

- Q 4 How to free dynamically allocated memory?
- Q 5 What is the difference between static allocation and dynamic allocation?

**REFERENCE:**


- *https://www.cse.iitb.ac.in/~rkj/cs101/lect22.pdf*
- *The Complete reference C++ by Herbert Schildt*


## POSTLAB

**AIM**: Write a program to oveload prefix and postfix version of increment and decrement operator

**OBJECTIVES:**

The objectives and expected learning outcomes of this practical are:

- To demonstrate the working of increment and decrement operator using the concept of operator overloading
- To develop and analyze the difference while overloading postfix and prefix notation of increment and decrement operator

**THEORY :**Standard C++ allows one to explicitly create separate prefix and postfix versions of the increment or decrement operators. To accomplish this, one must define two versions of the operator++( ) function.

Here are the general forms for the prefix

and postfix ++ and – – operator functions.

```
// Prefix increment
type operator++( ) {
// body of prefix operator
}
ob1 = ob2 = ob3; // multiple assignment
// Postfix increment
type operator++(int x) {
// body of postfix operator
}
// Prefix decrement
type operator– –( ) {
// body of prefix operator
}
```

```
// Postfix decrement
type operator– –(int x) {
// body of postfix operator
}
```

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 What is syntax of overloading prefix version of increment operator?
- Q 2 What is syntax of overloading postfix version of increment operator?
- Q 3 What is the use of int in parenthesis of postfix version of increment operator?
- Q 4 What is mean by unary operator?
- Q 5 What is difference between prefix and postfix version of increment and decrement operator?

**REFERENCE:**


- *The Complete reference C++ by Herbert Schildt*
- *Object Oriented Programming in C++ by Robert Lafore*

# Practical No.5

**Aim :** Write a program on data conversion

**INLAB AIM :** Design classes such that they support the following statements:

Rupee r1,r2; Dollar d1,d2;

d1=r1;

r2=d2;

Write a program which does such conversions.

**OBJECTIVES:**

- To demonstrate the use of conversion routine
- To build program on conversion from user defined data type to user defined data type

**Class Diagram:**

**POSTLAB AIM :** Design classes such that they support the following statements:

Radian r1,r2; Degree d1,d2;

d1=r1;

r2=d2;

Write a program which does such conversions.

**OBJECTIVE:**

- To demonstrate the use of conversion routine
- To build program on conversion from user defined data type to user defined data type

**Class Diagram:**

**AIM :** Write a program on data conversion

**INLAB**

**AIM** : Design classes such that they support the following statements:

Rupee r1,r2; Dollar d1,d2;

d1=r1;

r2=d2;

Write a program which does such conversions.

**OBJECTIVES:**

- To demonstrate the use of conversion routine
- To build program on conversion from user defined data type to user defined data type

**THEORY :** In C++,some conversions take place between user-defined types and basic types. Two approaches are used in such conversions: A one-argument constructor changes a basic type to a user-defined type, and a conversion operator converts a user-defined type to a basic type. When one user-defined type is converted to another, either approach can be used.

| | Routine In Destination | Routine In Source |
|---|---|---|
| Basic to basic | (Built-In Conversion Operators) | |
| Basic to class | Constructor | NA |
| Class to basic | NA | Conversion operator |
| Class to class | Constructor | Conversion operator |

Table 5.1 Type Conversions

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 What is mean by data conversion?
- Q 2 How to convert one basic data type into another basic data type?
- Q 3 How to convert user defined data type into basic data type?
- Q 4 How to convert basic data type into user defined data type?
- Q 5 What is the use of conversion constructor?

**REFERENCE:**

- *The Complete reference C++ by Herbert Schildt*
- *Object Oriented Programming in C++ by Robert Lafore*

**POSTLAB**

**AIM**: Design classes such that they support the following statements:

Radian r1,r2; Degree d1,d2;

d1=r1;

r2=d2;

Write a program which does such conversions.

**OBJECTIVES:**

- To demonstrate the use of operator function
- To build program on conversion from user defined data type to user defined data type

**THEORY :** The conversion from used defined data type to user defined data type can be achieved by using operator function.Below is the syntax of overloaded casting operator

```
operator type()
{
   .
   .
   .
}
```

The 'operator' is the keyword that has to be used followed by basic data type. For example If you want to overload the float operator. You will overload it like operator float() {}. There are three conditions that need to be satisfied for an overloaded casting operator.

1. Overloaded casting operator does not have any return type.
2. It cannot take any parameters or in other words no arguments can be passed to an overloaded casting operator.
3. Finally, it has to be defined inside a class definition. The class definition will be the user defined data type that we want to convert into a basic type whose casting operator has been overloaded.

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 What is use of operator function?
- Q 2  Explain the syntax of operator function?
- Q 3 Explain the syntax of conversion constructor?
- Q 4 What is the difference between implicit and explicit conversion?
- Q 5 What is the use of conversion constructor?

**REFERENCE:**


- *The Complete reference C++ by Herbert Schildt*
- *Object Oriented Programming in C++ by Robert Lafore*

# Practical No.6

**Aim :** Write a program on inheritance

**INLAB AIM :** Create employee bio-data using following classes i) Personal record ii))Professional record iii) Academic record Assume appropriate data members and member function to accept required data & print bio-data. Create bio-data using multiple inheritance using C++.

**OBJECTIVES:**

- To analyze the importance of inheritance for code reusability
- To create program on multiple inheritance

**Class Diagram:**

**POSTLAB AIM :** Create a class hierarchy of employee as a parent class and manager,scientist,loborer class as child class.Assume suitable data members and members functions to implement this relationship.

**OBJECTIVE:**

- To analyze and create program using inheritance
- To bulid program on class hierarchy

**Class Diagram:**

**AIM :** Write a program on inheritance

**INLAB**

**AIM** : Create employee bio-data using following classes i) Personal record ii))Professional record iii) Academic record Assume appropriate data members and member function to accept required data & print bio-data. Create bio-data using multiple inheritance using C++.

**OBJECTIVES:**

- To analyze the importance of inheritance for code reusability
- To create program on multiple inheritance

**THEORY :** A class can be derived from more than one base class. This is called multiple inheritance. A class can also be contained within another class.Inheritance permits the reusability of software: Derived classes can extend the capabilities of base classes with no need to modify—or even access the source code of—the base class. This leads to new flexibility in the software development process, and to a wider range of roles for software developers.Multilevel inheritance can also be implemented if a derived class serves as a base class for another derived class.

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 What is mean by inheritance?
- Q 2 What is mean by multiple inheritance?
- Q 3 What is mean by multilevel inheritance?
- Q 4 What is mean by public inheritance?
- Q 5 What is mean by private inheritance?

**REFERENCE:**


- *The Complete reference C++ by Herbert Schildt*
- *Object Oriented Programming in C++ by Robert Lafore*

**POSTLAB**

**AIM**: Create a class hierarchy of employee as a parent class and manager,scientist,loborer class as child class.Assume suitable data members and members functions to implement this relationship.

**OBJECTIVES:**

The objectives and expected learning outcomes of this practical are:

- To analyze and create program using inheritance
- To bulid program on class hierarchy

**THEORY :** Inheritance is one of the cornerstones of OOP because it allows the creation of hierarchical classifications. Using inheritance, you can create a general class that defines traits common to a set of related items. This class may then be inherited by other, more specific classes, each adding only those things that are unique to the inheriting class.

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 What is mean by class hierarchy?
- Q 2 What is mean by multiple inheritance?
- Q 3 What is mean by multilevel inheritance?
- Q 4 What is mean by public inheritance?
- Q 5 What is mean by private inheritance?

**REFERENCE:**


- *The Complete reference C++ by Herbert Schildt*
- *Object Oriented Programming in C++ by Robert Lafore*

# Practical No.7

**Aim :** Write a program on containership

**INLAB AIM :** Write a program to implement containership of book class inside a library class.

**OBJECTIVES:**

- To demonstrate the use of containership
- To build program using the concept of containership

**Class Diagram:**

**AIM :** Write a program on containership

**INLAB**

**AIM** : Write a program to implement containership of book class inside a library class.

**OBJECTIVES:**

- To demonstrate the use of containership
- To build program using the concept of containership

**THEORY :** When a class contains objects of another class or its members, this kind of relationship is called containership or nesting and the class which contains objects of another class as its members is called as container class.
Syntax for the declaration of another class is:

```
Class class_name1
{
_____
};
Class class_name2
{
_____
};
Class class_name3
{
Class_name1 obj1; // object of class_name1
Class_name2 obj2;// object of class_name2
_______
};
```

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 What is mean by containership?
- Q 2 How to implement containership?
- Q 3 What is mean by aggregation?
- Q 4 Explain containership with an example?
- Q 5 Given a scenario explain how implement containership.

**REFERENCE:**


- *The Complete reference C++ by Herbert Schildt*
- *Object Oriented Programming in C++ by Robert Lafore*

# Practical No.8

**Aim :** Write a program to implement run time polymorphism in C++

**INLAB AIM :** Write a program to implement run time polymorphism in C++

**OBJECTIVES:**

- To demonstrate the use of run time polymorphism in C++
- To build program on run time polymorphism

**Class Diagram:**

**AIM :** Write a program to implement run time polymorphism in C++

**INLAB**

**AIM** : Write a program to implement run time polymorphism in C++

**OBJECTIVES:**

- To demonstrate the use of run time polymorphism in C++
- To build program on run time polymorphism

**THEORY :** Run-time polymorphism is accomplished by using inheritance and virtual functions. A virtual function is a member function that is declared within a base class and redefined by a derived class. To create a virtual function, precede the function's declaration in the base class with the keyword virtual. When a class containing a virtual function is inherited, the derived class redefines the virtual function to fit its own needs. In essence, virtual functions implement the "one interface, multiple methods" philosophy that underlies polymorphism. The virtual function within the base class defines the form of the interface to that function. Each redefinition of the virtual function by a derived class implements its operation as it relates specifically to the derived class. That is, the redefinition creates a specific method. When accessed "normally," virtual functions behave just like any other type of class member function. However, what makes virtual functions important and capable of supporting run-time polymorphism is how they behave when accessed via a pointer. A base-class pointer can be used to point to an object of any class derived from that base. When a base pointer points to a derived object that contains a virtual function, C++ determines which version of that function to call based upon the type of object pointed to by the pointer. And this determination is made at run time. Thus, when different objects are pointed to, different versions of the virtual function are executed. The same effect applies to base-class references.

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 What is mean by run time polymorphism?
- Q 2 What is the difference between compile time and run time polymorphism?

- Q 3 What is virtual function?
- Q 4 How to implement run time polymorphism?
- Q 5 What is the use of run time polymorphism?

**REFERENCE:**


- *http://script.spoken-tutorial.org/index.php/Advanced-C++*
- *Object Oriented Programming in C++ by Robert Lafore*

# Practical No.9

**Aim :** Write a program on abstract class

**INLAB AIM :** Write a program having student class as an abstract class and create many derived classes such as engineering,medical,science etc from student class.Create their object and process them.

**OBJECTIVES:**

- To construct abstract class
- To build program using abstract class

**Class Diagram:**

**AIM :** Write a program on abstract class

**INLAB**

**AIM** : Write a program having student class as an abstract class and create many derived classes such as engineering,medical,science etc from student class.Create their object and process them.

**OBJECTIVES:**

- To construct abstract class
- To build program using abstract class

**THEORY :** A class that contains at least one pure virtual function is said to be *abstract*. Because an abstract class contains one or more functions for which there is no definition (that is, a pure virtual function), no objects of an abstract class may be created. Instead, an abstract class constitutes an incomplete type that is used as a foundation for derived classes.Although you cannot create objects of an abstract class, you can create pointers and references to an abstract class. This allows abstract classes to support run-time polymorphism, which relies upon base-class pointers and references to select the proper virtual function.

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 What is mean by abstract class?
- Q 2 How to create abstract class?
- Q 3 Explain pure virtual function?
- Q 4 What is the difference between virtual function and pure virtual function?
- Q 5 If one places only one pure virtual function in a class then will it became an abstract class?

**REFERENCE:**


- *Object Oriented Programming in C++ by Robert Lafore*
- *Mastering in C++ by Venugopal Verma*

# Practical No.10

**Aim :** Write a program using friend function

**INLAB AIM :** Design a Class 'Complex' with data members for real and imaginary part. Write a program to perform arithmetic operations of two complex numbers using friend function

**OBJECTIVES:**

- To demonstrate the use of friend function
- To construct program using friend function

**Class Diagram:**

**AIM :** Write a program using friend function

**INLAB**

**AIM** : Design a Class 'Complex' with data members for real and imaginary part. Write a program to perform arithmetic operations of two complex numbers using friend function.

**OBJECTIVES:**

- To demonstrate the use of friend function
- To construct program using friend function

**THEORY :** It is possible to grant a nonmember function access to the private members of a class by using a friend. A friend function has access to all private and protected members of the class for which it is a friend. To declare a friend function, include its prototype within the class, preceding it with the keyword friend.

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 What is friend function?
- Q 2 What is the requirement of friend function?
- Q 3 How to declare friend function?
- Q 4 Is it a member function of a class?
- Q 5 How to use friend function?

**REFERENCE:**


- *Object Oriented Programming in C++ by Robert Lafore*
- *http://www.it.iitb.ac.in/frg/wiki/images/e/ee/CS101x_S447_Friends_and_Static_Members_IIT_Bombay.pdf*

# Practical No.11

**Aim :** Write a program for input/output operation on file

**INLAB AIM :** Create a class person and perform following operations using a menu driven program.

'a'--add data for a person
'd'--display the data for a person
'w'--write all person's data to file
'r'--read all person's data from file
'u'-- update data for a person

**OBJECTIVES:**

- To make use of files in C++
- To construct program for performing various operation on file

**Class Diagram:**

**AIM :** Write a program for input/output operation on file

**INLAB**

**AIM** : Create a class person and perform following operations using a menu driven program.

'a'--add data for a person
'd'--display the data for a person
'w'--write all person's data to file
'r'--read all person's data from file
'u'-- update data for a person.

**OBJECTIVES:**

- To make use of files in C++
- To construct program for performing various operation on file

**THEORY :** To perform file I/O, you must include the header <fstream> in your program. It defines several classes, including ifstream, ofstream, and fstream. These classes are derived from istream, ostream, and iostream, respectively. Remember, istream,ostream, and iostream are derived from ios, so ifstream, ofstream, and fstream also have access to all operations defined by ios. Another class used by the file system is filebuf, which provides low-level facilities to manage a file stream. Usually, you don't use filebuf directly, but it is part of the other file classes.

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 Which is the stream class required for performing operations on file?
- Q 2 What are the different modes for opening the files?
- Q 3 How to open a file in append mode?
- Q 4 What are the file pointers?
- Q 5 What is the difference between get and put pointer?

**REFERENCE:**


- *Object Oriented Programming in C++ by Robert Lafore*
- *https://www.cse.iitb.ac.in/~cs101/2014.2/lectureslides/Recap_Quiz/CS101_File_handling_Recap_and_Quiz_IIT_Bombay.pdf*

# Practical No.12

**Aim :** Write a program to overload extraction and insertion operator

**INLAB AIM :** Write a program to overload extraction and insertion operator

**OBJECTIVES:**

- To overload insertion and extraction operator
- To build program on operator overloading

**Class Diagram:**

**AIM :** Write a program to overload extraction and insertion operator

**INLAB**

**AIM** : Write a program to overload extraction and insertion operator

**OBJECTIVES:**

- To overload insertion and extraction operator
- To build program on operator overloading

**THEORY :** Overloading extraction and insertion operator is a powerful feature of C ++ . It lets you treat I/O for user-defined data types in the same way as basic types like int and double . For example, if you have an object of class crawdad called cd1 , one can display it with the statement
cout << "\ncd1=" << cd1;
just as if it were a basic data type.
One can overload the extraction and insertion operators so they work with the display and keyboard ( cout and cin ) alone. With a little more care, one can also overload them so they work with disk files as well.

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 Which is the stream class required for performing operations on file?
- Q 2 What are the different modes for opening the files?
- Q 3 How to open a file in append mode?
- Q 4 What are the file pointers?
- Q 5 What is the difference between get and put pointer?

**REFERENCE:**


- *Object Oriented Programming in C++ by Robert Lafore*
- *https://www.cse.iitb.ac.in/~cs101/2014.2/lectureslides/CS101x_S446_Operator_Overloadi_IIT_Bombay.pdf*

# Practical No.13

**Aim :** Write a program using template

**INLAB AIM :** Implement a generic vector. Include following member functions:

To create the vector.

To modify the value of a given element
To multiply by a scalar value
To display the vector in the form (10,20,30,...)

**OBJECTIVES:**

- To demonstrate the use of template
- To build program using class and function template

**Class Diagram:**

**AIM :** Write a program using template

**INLAB**

**AIM** : Implement a generic vector. Include following member functions:

To create the vector.
To modify the value of a given element
To multiply by a scalar value
To display the vector in the form (10,20,30,...)

**OBJECTIVES:**

- To demonstrate the use of template
- To build program using class and function template

**THEORY :** Templates are a feature of the C++ programming language that allows functions and classes to operate with generic types. This allows a function or class to work on many different data types without being rewritten for each one. 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 basic syntax for declaring a templated class is as follows:

```
template <class a_type> class a_class {…};
```

The keyword 'class' above simply means that the identifier a_type will stand for a datatype.When defining a function as a member of a templated class, it is necessary to define it as a templated function:

```
template<class a_type> void a_class<a_type>::a_function(){…}
```

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 What is template?
- Q 2 How to create template class?
- Q 3 How to create template function?
- Q 4 How to instantiate object of template class?
- Q 5 What is the use of template?

**REFERENCE:**

- *Object Oriented Programming in C++ by Robert Lafore*
- *The Complete reference C++ by Herbert Schildt*

# Practical No.14

**Aim :** Write a program using exception handling

**INLAB AIM :** Crete User defined exception to check the following conditions and throw the exception if the criterion does not meet.
a.User has age between 18 and 55
b.User has income between Rs. 50,000 –Rs. 1,00,000 per month
c.User stays in Pune/ Mumbai/ Bangalore / Chennai
d.User has 4-wheeler
Accept age, Income, City,Vehicle from the user and check for the conditions mentioned above. If any of the condition not met then throw the exception.

**OBJECTIVES:**

- To demonstrate the use of exception
- To build program using exception handling

**Class Diagram:**

**AIM :** Write a program using exception handling

**INLAB**

**AIM** : Crete User defined exception to check the following conditions and throw the exception if the criterion does not meet.
a.User has age between 18 and 55
b.User has income between Rs. 50,000 –Rs. 1,00,000 per month
c.User stays in Pune/ Mumbai/ Bangalore / Chennai
d.User has 4-wheeler
Accept age, Income, City,Vehicle from the user and check for the conditions mentioned above. If any of the condition not met then throw the exception.

**OBJECTIVES:**

- To demonstrate the use of exception
- To build program using exception handling

**THEORY :** Exceptions provide a systematic, object-oriented approach to handling run-time errors generated by C++ classes.Exceptions are errors that occur at run time.They are caused by a wide variety exceptional circumstance, such as running out of memory, not being able to open a file, trying to initialize an object to an impossible value, or using an out-of-bounds index to a vector.The sequence of events when an exception occurs.

1.Code is executing normally outside a try block.

2.Control enters the try block.

3.A statement in the try block causes an error in a member function.

4.The member function throws an exception.

5.Control transfers to the exception handler (catch block) following the try block

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 Which is mean by exception?
- Q 2 What is the requirement of exception handling?
- Q 3 How to create user defined exception class?
- Q 4 What are the sequence of events when exceprion occurs?

- Q 5 Explain try,throw,catch?

**REFERENCE:**


- *Object Oriented Programming in C++ by Robert Lafore*
- *https://onlinecourses.nptel.ac.in*

# Practical No.15

**Aim :** Write a program using Standard template library

**INLAB AIM :** Write C++ program using STL for Double ended queue

**OBJECTIVES:**

- To demonstrate STL for implementation of dequeue operations
- To build program using STL

**Class Diagram:**

**AIM :** Write a program using Standard template library

**INLAB**

**AIM** : Write C++ program using STL for Double ended queue

**OBJECTIVES:**

- To demonstrate STL for implementation of deque operations
- To build program using STL

**THEORY :** The C++ STL (Standard Template Library) is a powerful set of C++ template classes to provides general-purpose templatized classes and functions that implement many popular and commonly used algorithms and data structures like vectors, lists, queues, and stacks.
At the core of the C++ Standard Template Library are following three well-structured components:
Containers:Containers are used to manage collections of objects of a certain kind. There are several different types of containers like deque, list, vector, map etc.
Algorithms:Algorithms act on containers. They provide the means by which you will perform initialization, sorting, searching, and transforming of the contents of containers.
Iterators:Iterators are used to step through the elements of collections of objects. These collections may be containers or subsets of containers.
Double ended queue : deque(usually pronounced like "deck") is an irregular acronym of double-ended queue. Double-ended queues are sequence containers with dynamic sizes that can be expanded or contracted on both ends (either its front or its back).

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 Which are the core components of STL?
- Q 2 What is the use of container?
- Q 3 What is the use of algorithm in STL?
- Q 4 What is the use of iterator in STL?
- Q 5 What are different types of container classes available in STL?

**REFERENCE:**


- *Object Oriented Programming in C++ by Robert Lafore*
- *The Complete reference C++ by Herbert Schildt*

# Practical No.16

**Aim :** Write a program using inheritance and files

**INLAB AIM :** Create a class hierarchy of employee as a parent class and manager,scientist,loborer class as child class.Assume suitable data members and members functions to implement this relationship and write an interactive program to write,read,display the data from file for any type of employee.

**OBJECTIVES:**

- To build program using multiple concepts of C++
- To build program using inheritance and files

**Class Diagram:**

**AIM :** Write a program using inheritance and files

**INLAB**

**AIM** : Create a class hierarchy of employee as a parent class and manager,scientist,loborer class as child class.Assume suitable data members and members functions to implement this relationship and write an interactive program to write,read,display the data from file for any type of employee.

**OBJECTIVES:**

- To build program using multiple concepts of C++
- To build program using multilevel inheritance and files

**THEORY :** Inheritance is one of the cornerstones of OOP because it allows the creation of hierarchical classifications. Using inheritance, you can create a general class that defines traits common to a set of related items. This class may then be inherited by other, more specific classes, each adding only those things that are unique to the inheriting class.To perform file I/O, you must include the header <fstream> in your program. It defines several classes, including ifstream, ofstream, and fstream. These classes are derived from istream, ostream, and iostream, respectively. Remember, istream,ostream, and iostream are derived from ios, so ifstream, ofstream, and fstream also have access to all operations defined by ios.
Another class used by the file system is filebuf, which provides low-level facilities to manage a file stream. Usually, you don't use filebuf directly, but it is part of the other file classes.

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 Which is the stream class required for performing operations on file?
- Q 2 What are the different modes for opening the files?
- Q 3 What is multilevel inheritance?
- Q 4 What is the difference between multilevel and multiple inheritance?
- Q 5 What is the difference between get and put pointer?

**REFERENCE:**


- *Object Oriented Programming in C++ by Robert Lafore*
- *http://www.it.iitb.ac.in/frg/wiki/images/d/d9/CS101x_Inheritance_IIT_Bombay_Transcript.pdf*

# Practical No.17

**Aim :** Write a program on virtual destructor

**INLAB AIM :** Write a program on virtual destructor

**OBJECTIVES:**

- To demonstrate the use of virtual destructor
- To build program using virtual destructor

**Class Diagram:**

**AIM :** Write a program on virtual destructor

**INLAB**

**AIM** : Write a program on virtual destructor

**OBJECTIVES:**

- To demonstrate the use of virtual destructor
- To build program using virtual destructor

**THEORY :**Destructors in the Base class can be Virtual. Whenever Upcasting is done, Destructors of the Base class must be made virtual for proper destrucstion of the object when the program exits.Deleting a derived class object using a pointer to a base class that has a non-virtual destructor results in undefined behavior. To correct this situation, the base class should be defined with a virtual destructor.

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 Which is virtual destructor?
- Q 2 What is the need for virtual destructor?
- Q 3 How to declare a virtual destructor?
- Q 4 How a virtual destructor get executed?
- Q 5 Explain virtual destructor.

**REFERENCE:**


- *Object Oriented Programming in C++ by Robert Lafore*
- *http://www.it.iitb.ac.in/frg/wiki/images/e/eb/CS101x_S459_Polymorphism_Virtual_Functions_IIT_Bombay.pdf*

# Practical No.18

**Aim :** Write a program on nested namespace

**INLAB AIM :** Write a program on nested namespace

**OBJECTIVES:**

- To demonstrate the use of nested namespace
- To build program using nested namespace

**Class Diagram:**

**AIM :** Write a program on nested namespace

**INLAB**

**AIM** : Write a program on nested namespace

**OBJECTIVES:**

- To demonstrate the use of nested namespace
- To build program using nested namespace

**THEORY :** Using namespace, one can define the context in which names are defined. In essence, a namespace defines a scope.
A namespace definition begins with the keyword namespace followed by the namespace name as follows

```
namespace namespace_name {
   // code declarations
}
```

Namespaces can be nested where you can define one namespace inside another name space as follows:

```
namespace namespace_name1 {
   // code declarations
   namespace namespace_name2 {
      // code declarations
   }
}
```

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 Which is mean by namespace?
- Q 2 What is mean by nested namespace?
- Q 3 How to declare nested namespace?
- Q 4 What is the requirement of nested namepace?
- Q 5 How to implement nested namespace?

**REFERENCE:**

- *Object Oriented Programming in C++ by Robert Lafore*
- *https://www.cse.iitb.ac.in/~cs101/2011.2/Lectures/lecture_slides/2011_08_03_Basic_features_of_CPP_slot5.pdf*

# Practical No.19

**Aim :** Write a program on nested classes

**INLAB AIM :** Write a program on nested classes

**OBJECTIVES:**

- To demonstrate the use of nested classes
- To build program using nested classes

**Class Diagram:**

**AIM :** Write a program on nested classes

**INLAB**

**AIM** : Write a program on nested classes

**OBJECTIVES:**

- To demonstrate the use of nested classes
- To build program using nested classes

**THEORY :** In C++,It is possible to define one class within another. Doing so creates a nested class. Since a class declaration does, in fact, define a scope, a nested class is valid only within the scope of the enclosing class.

**ALGORITHM:**

**CODE:**

**OUTPUT:**

**CONLUSION:**

**DISCUSSION AND VIVA VOCE:**

- Q 1 Which is nested class?
- Q 2 What is the use of nested class?
- Q 3 How to declare nested class?
- Q 4 Explain nested class.
- Q 5 How to create an object of nested class?

**REFERENCE:**


- *Object Oriented Programming in C++ by Robert Lafore*
- *http://www.geeksforgeeks.org/nested-classes-in-c/*