

Introduction to C++ Programs

- 1) WAP to illustrate the concept of OOP.
- 2) WAP to find the prime number in C++.
- 3) WAP to create class 'time' with data members days, hours, minutes and Second. Then add two 'time' object by taking object as argument and also Returning object as argument.

C++ Language Constructs

- a. WAP to add and subtract two numbers.
- b. WAP to add and subtract two numbers using function and display result inside function.
- c. WAP to add and subtract two numbers using function and display result from main.
- d. WAP to show function overloading.
- e. WAP to calculate area of circle(radius), rectangle(l,b), area of triangle (a, b, c) using function overloading.
- f. WAP to show default argument.
- g. WAP to calculate price of n items. Pass number of items and unit price to a function and return price total price. If unit price is not specified use Default price.
- h. WAP to show DMA.
- i. WAP to take n numbers from user and display their average.
Also create memory dynamically for array and delete it.
- j. WAP to show the use of inline function.
- k. WAP to show reference variable.
- l. WAP to show passing argument as reference.
- m. WAP to show return by reference.
- n. WAP to swap 2 values using function and reference variable. Display result from main.
- o. WAP to show how namespace works.
- p. WAP to create file named "programmerone.cpp" with function display.
Also create file named "programmetwo.cpp" with function named display.
Access both display from third program named "result.cpp". Use namespace whenever needed.

Classes And Objects

- 1) WAP to access public data from main function.
- 2) WAP to access the data from the function outside the class.
- 3) WAP to access the data from the function inside the class.
- 4) Create a class named "animal" with data and methods as follows:

Data: name, id_number, cost

Methods: void setdata(char, int, float)

and void showdata(); create two different objects of type "animals"

Use setdata() to set the data member values and use showdata() to display them.

- 5) WAP to create a class student. Ask user to enter name and marks obtained in 5 different subjects. The program should print the name and total marks obtained by students
- 6) WAP to show different type of constructor (default, parameterized and copy) and destructor.
- 7) Explain all constructor with and display appropriate message when object goes out of scope
- 8) Create a class named "Rectangle" with two data members-length and breadth and a function to calculate the area which is length*breadth.

The class has three constructor which are:

- i)-having no parameter-values of both length and breadth are assigned zero
- ii)-having two number as parameters the two numbers are assigned as length and breadth respectively.
- iii)-having one number as parameter both length and breath are assigned that number. Now create objects of "Rectangle" class having none, one and two parameters and print their areas using member function area.

- 9) WAP to print the name of students by creating a student class. If no name is passed while creating an object of the student class, then the name should be "unknown" otherwise the name should be equal to the string value passed while creating object of the student class
- 10) WAP to show how friend function works.
- 11) Explain friend class with example.
- 12) WAP to create classes ABC with data member feet and XYZ with data member inch. Add members of class ABC and XYZ using friend function
- 13) WAP to add distances D1(feet, inch) and D2(feet, inch). Display result by returning values using friend function
- 14) WAP to show passing single argument as an object.
- 15) WAP to show returning object by normal way and also with nameless object.
- 16) WAP to add two complex numbers using member function add(). Pass single argument to function, return from function and display result.
- 17) WAP to add distances D1(feet, inch) and D2(feet, inch). Display result by returning values with nameless object.
- 18) WAP to add distances D1(feet, inch) and D2(feet, inch). Display result inside function
- 19) WAP to swap the numbers by using reference variable.
- 20) WAP to show static variable and static function.
- 21) WAP to show the order of creation and destruction using static data members.
There must be static keyword in your program to display the value of static data.

Operator Overloading

- 1) WAP to show conversion from Basic to User Defined.
- 2) WAP to show conversion from User Defined to Basic.
- 3) WAP to show conversion between User Defined to User Defined both destination and source conversion.
- 4) WAP to overload unary operator(++/--)
- 5) WAP to overload predecrement and post decrement operator.
- 6) WAP to overload binary operator + using member function.
- 7) WAP to overload binary operator + using friend function.
- 8) WAP to add and subtract 2 distances d1 and d2 (feet,inch) using operator overloading + and - using member function.
- 9) WAP to add and subtract 2 distances d1 and d2 (feet,inch) using operator overloading + and - using friend function.
- 10) WAP to overload >,<== operator to compare two distances(feet, inch) using concept of operator overloading using member function.
- 11) WAP to overload >,<== operator to compare two distances(feet, inch) using concept of operator overloading using friend function.
- 12) WAP to overload >,<== operator to compare two amount using the concept of overloading.
- 13) WAP to overload insertion and extraction operator.

Inheritance

SINGLE INHERITANCE

- 1) WAP to show single inheritance with default constructor.
- 2) WAP to show single inheritance with main function access.
- 3) WAP to show single inheritance parameterize constructor.
- 4) WAP to show single inheritance parameterized only in child class.
- 5) WAP to show single inheritance with same function name of base and derived class.
- 6) WAP to show parent function call inside child class.
- 7) WAP to show protected data inside base class.

8) WAP to show parent function call inside child class having same member function name.

MULTIPLE INHERTANCE

- 1) WAP to show multiple inheritance constructor destructor.
- 2) WAP to show multiple inheritance with different function name.
- 3) WAP to show the ambiguity in multiple inheritance.

MULTI LEVEL INHERTANCE

- 1) WAP to show multi level inheritance.

MULTI PATH INHERTANCE

- 1) WAP to show multi path inheritance.

IMPORTANT:

- 1) Create a class called cricketer with member variables to represent name, age, and number of matches played. From this derive two classes Bowler and Batsmen. Bowler class has number of wickets as member variable and Batsmen class has number of runs and centuries as member variables. Use appropriate member functions in all classes to read and display respective data.
- 2) Create a class student with data members student ID and name. Create class internal exam with marks first as data member and external exam with marks second. Create class result by inheriting class internal exam and external exam that will add marks of class internal exam and external exam.
- 3) WAP to show order of constructor(default, parameterized) and destructor call in multiple inheritance. Base class one ename and class two have employee id and child class result.
- 4) WAP to convert object from class that represent weight of gold in Nepal, tola to object of a class that represent internal gold measurement of weight in gram scale.
- 5) WAP with three classes students, test, and result by using multi level inheritance. Assume necessary member and function yourself and program with input information input data and calculate marks total and display result.

Polymorphism

- 1) WAP to show virtual function mechanism.
- 2) WAP to show pure virtual function mechanism and abstract class.

- 3) WAP to show virtual destructor.
- 4) WAP to show dynamic cast operator.
- 5) WAP to show type id operator.
- 6) A bookshop in Kathmandu valley sells books and DVD. Two classes Book and DVD are inherited from the abstract base class Media. The Media class has common data members such as title and publication. The class Book has data members such as number_of_pages and that of DVD has data member such as duration. Each class members has member functions such as read data() and showdata(). WAP for modeling the class hierarchy for book shop and process objects of these classes using pointers to the base class.
- 7) Design a class that represents paper. Create class book, newspaper, copy derived from class paper. Add necessary member function to make your program meaningful and also show constructor and virtual destructor. Create object Dynamically using pointer to the base class.

File Handling

- 1) WAP to open a file using constructor.
- 2) WAP to open a file using member function.
- 3) WAP to use r/w file using f stream.
- 4) WAP using fin and fout.
- 5) WAP to show use of manipulators.
- 6) WAP to store book database in a file. Design a book class with members ISBN number, name of author, title, price. The program should be able to store book object to a file, find a book in the database and display according to ISBN number. Also display all records.
 1. getInput()
 2. display()
 3. add_record
 4. show_record
 5. search_record_by isbn number & modify also.

TEMPLATE

FUNCTION TEMPLATE:

- 1) WAP to show more than one template mechanism.
- 2) WAP to show function template overloading with function.
- 3) WAP to show overloading template with template and normal function.

CLASS TEMPLATE:

- 1) WAP to find MAX value of array with default argument with class template.
- 2) WAP to show use of multiple template.
- 3) WAP to show template and non template argument.
- 4) WAP to find maximum and minimum value of an array using class template.
- 5) WAP to demonstrate different functions of vector and reverse() which is algorithm.
- 6) WAP to compare 2 values using function template and display largest value. Check your program for char, int and float type data

EXCEPTION HANDLING

- 1) WAP to show simple try catch throw.
- 2) WAP to show throwing inside function.
- 3) WAP to show multiple catch statements.
- 4) WAP to show catching all exceptions.
- 5) WAP to show specifying exceptions.
- 6) WAP to show re throwing exception.
- 7) WAP to show try catch throw in user defined *classes* type.
- 8) WAP to compute square root of a number. The input value must be tested for validity. If it is negative an exception must be raised.
- 9) WAP to read name, roll number and marks that catches multiple exception using exception class if roll number is negative and marks greater than full marks.
- 10) WAP to show how do you throw *only specified* exception from a function.
- 11) WAP to throw multiple exception. Using exception class and each exception is handled by *separate* handler.