**1.A super class Product has been defined to store the details of a product sold by a wholesaler to a retailer. Define a sub class Sales to compute the total amount paid by the retailer with or without fine along with service tax.   
Some of the members of both the classes are given below:  
Class name : Product  
Data member/instance variable:  
name : stores the name of the product  
code : integer to store the product code  
amount : stores the total sale amount of the product (in decimals)  
Member functions/methods:  
Product(String n, int c, double p) : parameterized constructor to assign data members name=n, code=c and amount = p  
void show( ) : displays the details of the data members  
Class name: Sales  
Data member/instance variable:  
day : stores number of days taken to pay the sale amount  
tax : to store the service tax (in decimals)  
totamt : to store the total amount (in decimals)  
Member functions/methods:  
Sales(…) : parameterized constructor to assign values to data members of both the classes  
void compute( ) : calculates the service tax @ 12·4% of the actual sale amount calculates the fine @ 2·5% of the actual sale amount only if the amount paid by the retailer to the wholesaler exceeds 30 days calculates the total amount paid by the retailer as (actual sale amount + service tax + fine)  
void show( ) : displays the data members of super class and the total amount  
Assume that the super class Product has been defined. Using the concept of inheritance, specify the class Sales giving the details of the constructor(…), void compute( ) and void show( ).**

**2.Write a C++ program to calculate the percentage of a student using multi-level inheritance. Accept the marks of three subjects in base class. A class will derived from the above mentioned class which includes a function to find the total marks obtained and another class derived from this class which calculates and displays the percentage of student.**

**3.Write class declarations and member function definitions for a C++ base class to represent an Employee (emp-code, name).  
  
Derive two classes as Fulltime (daily rate, number of days, salary) and Parttime (number of working hours, hourly rate, salary).  
  
Write a menu driven program to:  
  
1. Accept the details of ‘n’ employees.  
2. Display the details of ‘n’ employees.  
3. Search a given Employee by emp-code.**

Q4.Create two classes named Mammals and MarineAnimals. Create another class named BlueWhale which inherits both the above classes. Now, create a function in each of these classes which prints "I am mammal", "I am a marine animal" and "I belong to both the categories: Mammals as well as Marine Animals" respectively. Now, create an object for each of the above class and try calling  
1 - function of Mammals by the object of Mammal  
2 - function of MarineAnimal by the object of MarineAnimal  
3 - function of BlueWhale by the object of BlueWhale  
4 - function of each of its parent by the object of BlueWhale

Q5.Make a class named Fruit with a data member to calculate the number of fruits in a basket. Create two other class named Apples and Mangoes to calculate the number of apples and mangoes in the basket. Print the number of fruits of each type and the total number of fruits in the basket.

Q6,We want to calculate the total marks of each student of a class in Physics,Chemistry and Mathematics and the average marks of the class. The number of students in the class are entered by the user. Create a class named Marks with data members for roll number, name and marks. Create three other classes inheriting the Marks class, namely Physics, Chemistry and Mathematics, which are used to define marks in individual subject of each student. Roll number of each student will be generated automatically.

Q7.We want to store the information of different vehicles. Create a class named Vehicle with two data member named mileage and price. Create its two subclasses  
\*Car with data members to store ownership cost, warranty (by years), seating capacity and fuel type (diesel or petrol).  
\*Bike with data members to store the number of cylinders, number of gears, cooling type(air, liquid or oil), wheel type(alloys or spokes) and fuel tank size(in inches)  
Make another two subclasses Audi and Ford of Car, each having a data member to store the model type. Next, make two subclasses Bajaj and TVS, each having a data member to store the make-type.  
Now, store and print the information of an Audi and a Ford car (i.e. model type, ownership cost, warranty, seating capacity, fuel type, mileage and price.) Do the same for a Bajaj and a TVS bike.

Q8.Create a Base class that consists of private, protected and public data members and member functions. Try using different access modifiers for inheriting Base class to the Derived class and create a table that summarizes the above three modes (when derived in public, protected and private modes) and shows the access specifier of the members of base class in the Derived class.

Q9.You are given three classes A, B and C. All three classes implement their own version of func. In class A, func multiplies the value passed as a parameter by 2. In class B, func multiplies the value passed as a parameter by 3. In class C, func multiplies the value passed as a parameter by 5.You are given class D such that You need to modify the class D and implement the function update\_val which sets D's val to new\_val by manipulating the value by only calling the func defined in classes A, B and C.It is guaranteed that new\_val has only 2, 3 and 5 as its prime factors. Implement class D's function update\_val. This function should update D's val only by calling A, B and C's func.

**Sample Input**

new\_val = 30

**Sample Output**

A's func called 1 times

B's func called 1 times

C's func called 1 times

Q10.Create a class called Student that contains the data members like age, name, enroll\_no, marks. Create another class called Faculty that contains data members like facultyName, facultyCode, salary,deptt, age, experience, gender. Create the function display() in both the classes to display the respective information. The derived Class Person demonstrates multiple inheritance. The program should be able to call both the base classes and displays their information. Remove the ambiguity (When we have exactly same variables or same methods in both the base classes, which one will becalled?) by proper mechanism

Q11.Implement a real case scenario by a proper C++ code to provide the solution to Diamond Problem in C++