RAISONI GROUP
— a vision beyond —

Assignment No 1

Title : Write a program to compute the area of triangle and circle by overloading area function

Problem statement : Implement a C++ program to understand concept of overloading.

Objective :
1] Understand basic principles of object oriented programming.
2] Implement concepts of overloading.
3] Develop programs using object oriented concepts.

Theory :

Object oriented programming is a computer programming model that organizes software design around data, or objects, rather than functions and logic. An object can be defined as data field that has unique attributes and behaviour.

Basic concepts of object oriented programming are :
- Inheritance

- Encapsulation
- Polymorphism
- Data abstraction
- Class and object
- Constructor and Destructor
- Exception Handling

Polymorphism :

Polymorphism is one of the core concept of object oriented programming and describe the situations in which something occurs in several different forms

In function overloading a function can take different forms based on number of arguments and their datatypes.

All the functions will have the same name, but they will differ in their arguments.

RAISONI GROUP
—a vision beyond—

Algorithm:

Step 1: Start the program.

Step 2: Declare function area() to find area of triangle with two int arguments.

Step 3: Declare function area() to find area of circle with one int arguments.

Step 4: Read value of base and height of triangle from user.

Step 5: Call function area with above two values as arguments.

Step 6: Read value of radius of a circle from user.

Step 7: Call function area with above one value as arguments.

Step 8: End of program.

Conclusion: Hence I have successfully implemented program to understand concept of overloading.

RAISONI GROUP
— a vision beyond —

Assignment 2

Title : Define a class to represent a bank account, include following members.

Data members : Name of depositor, Account number, Type of account, balance amount in the account.

Member functions : To assign initial values, to deposite an amount, to withdraw an amount after checking balance, to display name and balance.
Write a main program to test program using class and object.

Problem Statement : Implement C++ program which a class having data members and member function and object to access members of class.

Objective :
1] Understand the basic principles of object oriented programming.

2] Develop program using object oriented concepts.

RAISONI GROUP
— a vision beyond —

Theory:

Object Oriented Programming:

Object Oriented programming style that is associated with concept of class, object and various other concepts revoluing around these two, like Inheritance, Polymorphism, Abstraction Encapsulation etc.

Objects:

Objects are basic unit of OOP. They are instances of class, which have data members and use various member function to perform task.

Class:

It is similar to structures in C language, class can also be defined as user defined data type but it also contains function in it. So class is basically a blueprint for object.

Abstraction:

Abstraction refers to showing only the essential features of application and hiding the details.

Encapsulation:

It can also be called or said data binding. Encapsulation is all about binding data variables and functions together in class.

Inheritance:

Inheritance is a way to reuse once written code again and again. The class which is inherited is called Base class and class which inherits is called Derived class.

Polymorphism:

Polymorphism is one of core concept of object oriented programming and describe situations in which something occurs in several different forms.

Algorithm:

Step 1: Declare class BankAccount with data members Name of depositor, Account number, type of account, Balance amount and member functions - To assign initial value to deposit an amount, to withdraw an amount after

checking balance, to display name and balance.

Step 2: Define parameterized constructor to assign value to data members.

Step 3: Define member function withdraw() to withdraw an amount from account.

Step 4: Define member function deposite() to deposite an amount in account.

Step 5: Define member function display() to display Account No, Account type and Balance.

Step 6: Write main() function and accept values of acc no, name, acc type and balance from user.

Step 7: Create object of class BankAccount

Step 8: Call all member function one by one

Step 9: End of program.

Conclusion: Hence, successfully implemented C++ program having member and member function and objects to access members of class.

RAISONI GROUP
a vision beyond

Assignment No 3

Title: Create two classes DM and DB which stores values of distances. DM stores distances in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use friend function to carry out addition operation.

Problem Statement: Implement a C++ program to understand concept of Friend Function.

Objective:
1] Understand basic principles of object oriend programming.
2] Apply the concepts of inheritance and Friend function.
3] Develop program using object concepts.

Theory:

A friend function of a class is defined outside the class scope but it has right of access all private and protected

members of class. Such a function need not be a member of any of these classes. To make an outside function friendly to class, we have to simply declare this function as a friend of the class as shown below:

```
classAB
C
 {
   public:
   ......
   ......
   friend void xyz(void): //declaration
 };
```

A friend function possesses certain special characteristics:

1] It is not scope of class to which it has been declared asfriend.

2] Since it is not in scope of the class, it cannot be called using object of that class.

3] Unlike member functions, it can not access member names directly and has to use a object name and dot membership operator with each member name.

4] It can be declared either in public or private part of class without affecting its meaning.

5] Usually, it has object as arguments.

Algorithm:

Step 1: Start the Program.

Step 2: Declare class DM with data members meter, centimeter and member function accept(), display().

Step 3: Declare DB with data members feet, inches and member functions accept(), display().

Step 4: Declare friend function add() in both the classes.

Step 5: Define function accept() to take distance values.

Step 6: Define function display to display distance values.

Step 7: Define friend function add() to calculate addition of distances by adding

two objects.

Step 8: Create objects of class DB and DM.

Step 9: Call member functions accept() and display().

Step 10: Call add function and display addition of distances.

Step 11: End of program.

Conclusion: Hence, successfully implemented C++ program to understand concept of friend function.

RAISONI GROUP
a vision beyond

Assignment No. 4

Title:

Create a class MAT of size m*n. Define all possible matrix operations for MAT type object.

Problem Statement:

Implement a C++ program to understand concept matrix operations.

Objective:

1) Understand the basic principles of object oriented programming.

2) Apply the concepts of overloading inheritance polymorphism.

3) Develop program using object oriented concepts.

Theory:

If we create two or more members having same name but different in number or type of parameter,

is known as C++ overloading.

In C++ we can overload:
methods
constructors
Indexed properties

It is because these members have parameters only. Types of overloading in C++ are:

Function overloading
Operator overloading

Operator Overloading:

Operator overloading is an important concept in C++. It is a type of polymorphism in which an operator is overloaded to give user defined meaning to it.

Overloaded operator is used to perform operation on user defined data type.

For example '+' operator can be overloaded to perform addition on various data type like for Integer, String (concatination) etc.

Implementing Operator Overloading :

Operator overloading can be done by implementing a function which can be:

1] Member function
2] Non-member function
3] Friend function

Operator Overloading Function can be a member function if the left operand is an object of that class, but if Left operand is different, then operator overloading function must be non-member function.

Operator Overloading function can be made friend function if it needs access to private and protected members of class.

The operator keyword declares function specifying what operator-symbol means when applied to instances of a class. This gives operator more than one meaning or overloads it.

Algorithm:

Step 1: Declare class Matrix with data member as two dimensional array and member functions accept(), display(), operator (+), operator -() and operator *()

Step 2: Define all member functions.

Step 3: Create three objects of class Matrix.

Step 4: Ask user to enter elements of first matrix and accept it.

Step 5: Ask user to enter elements of first matrix and accept it.

Step 6: Assign M3 = m + M2 and display result.

Step 7: Assign M3 = M + M2 and display result.

Step 8: Assign M3 = M * M2 and display result.

Step 9: End of program.

Conclusion:

Hence, I have successfully implemented C++ program to know concept about matrix operation.

RAISONI GROUP
—a vision beyond—

Assignment 5

Title: Create stud class to display student information using constructor and destructor (Default Constructor, Multiple Constructor, Copy Constructor, Overloaded Constructor)

Problem statement: Implement a C++ program to understand concept of constructor and Destructor.

Objective:

1] Understand basic principle of object oriented programming.

2] Apply the concepts of class, overloading Inheritance and polymorphism

3] Develop program using object oriented concepts

Theory:

C++ Constructor:

In C++ constructor is a special method which is invoked automatically at time of object creation. It is used to intialize the data members of new object generally. The Constructor in C++ has same name as class or structure.

There are two types of constructors in C++
- Default Constructor
- Parameterized Constructor

C++ Default Constructor:
A constructor which has parameters is called parameterized constructor. It has used to provide different values to distinct objects.

Multiple Constructor:
In C++ we can have more than one constructor in class with same name, as long as each has different list of arguments. This concept is known as Constructor overloading and is quite similar to function overloading.

Copy Constructor:
The Copy Constructor is Constructor which creates an object by initializing it with an object of same class, which has been created previously.

RAISONI GROUP
— a vision beyond —

C++ Destructor:

A destructor works opposite to constructor, it destructs the objects of classes. It can be defined only once in a class
Like constructor, it is invoked automatically.
A destructor is defined like constructor. It must have same name as class. But it is prefixed with a title design (~).

Algorithm:

Step 1: Start the program.

Step 2: Declare class stud with data members name, address and roll no and member functions read(), display() and construction stud() and distructor stud().

Step 3: Define Constructor function stu().

RAISONI GROUP
— a vision beyond —

Step 4 : Define member function read()
and display()

Step 5 : Define destructor function ~sty()

Step 6 : Create object of class sty.

Step 7 : Call all member functions.

Step 8 : End of program.

Conclusion :

Hence I have successfully implemented C++ program to understand concept of constructor and destructor.

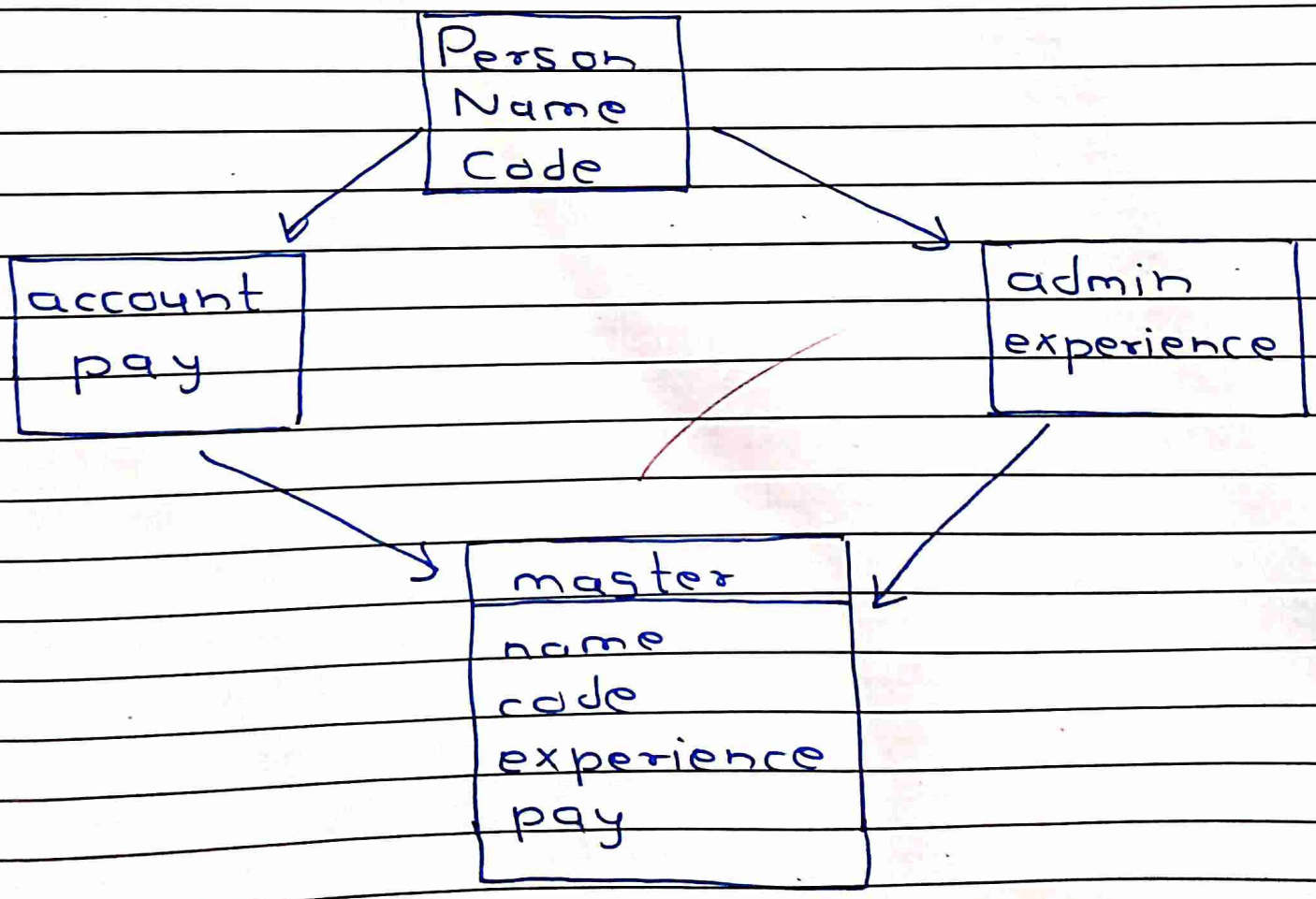
RAISONI GROUP
— a vision beyond —

Assignment No. 6

Title :

Consider class network of given figure. The class Master derives information from both account and admin classes which in turn derive information from class person.

Define all four classes and write a program to create, update and display information contained in master objects.



RAISONI GROUP
a vision beyond

Problem Statement:

Implement a C++ program to understand concepts of multiple and multilevel inheritance using virtual function.

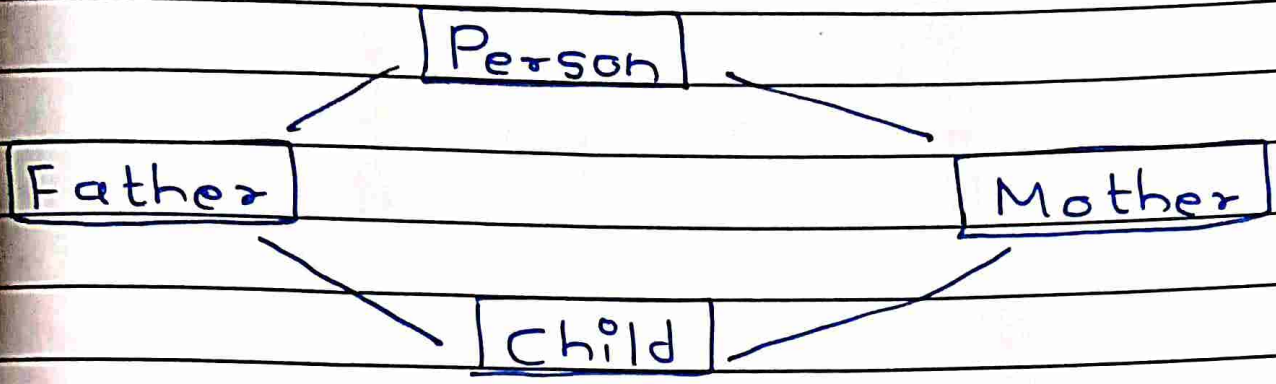
Objective:

- Understand the basic principle of object oriented programming.
- Apply the concepts of Inheritance.
- Develop programs using object oriented concept.

Theory:

The Diamond problem occurs when a child class inherits from two parents classes, who both share a common grandparent class. This is illustrated in diagram.

Here, we have a class child, inheriting from classes father and mother.



As shown in figure class child inherits traits of class person twice once from father and again from mother. This give rise to ambiguity since compiler fails to understand which way to go.

This scenario gives rise to diamond shaped inheritance graph and is famously called diamond problem.

The solution to diamond problem is to use virtual keyword. We make two parents classes into virtual classes in order to avoid two copies of grandparent class in child class.

Here we have used virtual keyword when class father and mother inherited person class. This is

usually called virtual inheritance, which guarantee that only a single instance of inherited class is passed on.

In other words, the child class will have single instance of person class, shared by both father and mother classes by having single instance of person class ambiguity is resolved.

- Algorithm :-

Step 1: Start the program.

Step 2: Declare class person with data members as name and code.

Step 3: Declare class account which is sub class of class person, with data member as pay.

Step 4: Declare class admin which is also sub class of class person with data member experience.

Step 5: Declare class master which is sub class of class account and class admin.

Step 6: Create object of class master and access member of class person, admin and account.

Step 7: End of the program.

Conclusion:

Hence, I have successfully implemented C++ program to understand concept of multiple and multilevel inheritance using virtual keyword.

RAISONI GROUP
— a vision beyond —

Assignment No 7

Title :

A book shop shell both books and video tapes Create a class media that stores title and price of publication. Create two derived classes. one for storing number of pages in book and another for storing playing time of tape.

A function display() must be defined in all classes to display class content Write a program using polymorphism and virtual function.

- Problem Statement

Implement a C++ program to understand concept of polymorhism and virtual function.

- Objective

1] Understand basic principles of object oriented programming.

2] Apply concept of overloading inheritance, polymorphism.

2) Develop programs using object oriented concepts.

Theory:

Static and Dynamic Binding:-

Polymorphism means "one name" multiple forms. The overloaded member function are selected for invoking by matching arguments both type and number.

This information is known to the compiler at compile time and compiler is able to select the appropriate function for a particular call at compile time itself.

It would be nice if appropriate member function could be selected while program is running.

This is known as runtime polymorphism. C++ supports a mechanism known as virtual function to achieve run time polymorphism.

RAISONI GROUP
— a vision beyond —

Virtual function:

Polymorphism refers to property by which objects belonging to different classes are able to respond to same message but different forms. An essential requirement of polymorphism is therefore the ability to refer to objects without any regard to their classes.

When we use same function name in both base and derived classes the function in base class is declared as virtual using keyword virtual preceeding its normal declaration.

Rules for Virtual Function:

When virtual function are created for implementing late binding, observe some basic rules that satisfy compiler requirements.

1] The virtual function must be member of same class.

2] They cannot be static member.

3] They are accessed by using object pointers.

4] A virtual function can be a friend of another class.

5] A virtual function in a base class must be defined, even though it may not be used.

6] The prototypes of base class version of virtual function and all derived class version must be identical.

7] We cannot have virtual constructors but we can have virtual destructors.

8] While a base pointer points to any type of derived object to reverse is not true.

9] When a base pointer points to a derived class incrementing

or decrementing it will not make it to point to next object of derived class.

10] If a virtual function is defined in base class it need not be necessarily redefined in derived class.

· Algorithm

Step 1: Start the program

Step 2: Declare class media with data member title and price add member function display() in class.

Step 3: Declare class book subclass of class media with data member pages and member function display().

Step 4: Declare class type, subclass of class media with data members time and member function display().

Step 5: Define display() function of class media, book and tape.

Step 6: Accept book details from user.

Step 7: Create object of class book and initialize its data member.

Step 8: Accept tape details from user.

Step 9: Create object of class book and intitialize its data member.

Step 10: Create object of class media.

Step 11: Call display function of class book and class tape of display book details and tape details respectively with help of object of class media.

Step 12: End of program.

Conclusion:

Hence I have successfully implemented C++ program to understand concept of polymorphism and virtual function.

RAISONI GROUP
—a vision beyond—

Assignment No. 8

Title: Write a program containing a possible exception. Use a try block to throw it and catch block to handle it properly.

Problem Statement : Implement a C++ program to understand concept of error handling.

Objective :

1) Understand the basic principles of object oriented programming.
2) Implement memory allocation technique and usage of exception handling, generic programming.
3) Develop program using object oriented concepts.

Theory :

Difference between errors and exceptions

Exceptions are those which can be handled at run time whereas errors cannot be handled. An exception

is a object of a type deriving from System.Exception class. SystemException is thrown by CLR when errors occur that are nonfatal and recoverable by user programs. It is meant to give you an opportunity to do something with thrown statement to transfer control to a catch clause in tryblock.

Exception handling

Exceptions are runtime anomalies or unusual conditions that a program may encounter while executing Anomalies might include conditions such as division by zero, accessing an array outside of its bounds or running out of memory or disk space.

Exceptions provide a way to transfer control from one part of a program to another C++ exception handling is built upon three keywords:

- try
- catch
- throw

Types of exceptions:

1) Synchronous exceptions
2) Asynchronous exceptions

1) Synchronous exceptions: Error such as out of range index and overflow are synchronous exceptions.

2) Asynchronous exceptions: The errors that are generated by any event beyond the control of program are called asynchronous exception.

Exception Handling Mechanism:

An exception is said to be thrown at place where some error or abnormal condition is detected. The throwing will cause normal program flow to be aborted in raised exception.

In handled exceptions, execution of program will resume at a designated block of code, called catch block which encloses point of throwing in terms of program execution.

C++ exception handling is built upon three keywords : try, catch and throw. Try is used to preface block of statement which may generate exceptions

Algorithm :

Step 1 : Create a class with two double type values.

Step 2 : Input X and Y

Step 3 : Within try block check
Y == 0 or not.
Division = X / Y
if Y == 0 throws(y) & catch exception
else
Print Division

Step 4 : End of program.

Conclusion :
Hence I have implemented a program containing possible exceptions.