**Experiment No 6**

**Aim:** Implementing the concept of inheritance in CPP

## Theory:

## Inheritance

In C++, it is possible to inherit attributes and methods from one class to another. We group the "inheritance concept" into two categories:

* **derived class** (child) - the class that inherits from another class
* **base class** (parent) - the class being inherited from

To inherit from a class, use the : symbol.

## Multilevel Inheritance

A class can also be derived from one class, which is already derived from another class.

In the following example, MyGrandChild is derived from class MyChild (which is derived from MyClass).

### Example

// Base class (parent)  
class MyClass {  
  public:  
    void myFunction() {  
      cout << "Some content in parent class." ;  
    }  
};  
  
// Derived class (child)  
class MyChild: public MyClass {  
};  
  
// Derived class (grandchild)  
class MyGrandChild: public MyChild {  
};  
  
int main() {  
  MyGrandChild myObj;  
  myObj.myFunction();  
  return 0;  
}

## Multiple Inheritance

A class can also be derived from more than one base class, using a **comma-separated list:**

### Example

// Base class  
class MyClass {  
  public:  
    void myFunction() {  
      cout << "Some content in parent class." ;  
    }  
};  
  
// Another base class  
class MyOtherClass {  
  public:  
    void myOtherFunction() {  
      cout << "Some content in another class." ;  
    }  
};  
  
// Derived class  
**class MyChildClass: public MyClass, public MyOtherClass** {  
};  
  
int main() {  
  MyChildClass myObj;  
  myObj.myFunction();  
  myObj.myOtherFunction();  
  return 0;  
}

## Access Specifiers

You learned from the [Access Specifiers](https://www.w3schools.com/CPP/cpp_access_specifiers.asp) chapter that there are three specifiers available in C++. Until now, we have only used public (members of a class are accessible from outside the class) and private (members can only be accessed within the class). The third specifier, protected, is similar to private, but it can also be accessed in the **inherited** class:

### Example

// Base class  
class Employee {  
  **protected: // Protected access specifier**  
    int salary;  
};  
  
// Derived class  
class Programmer: public Employee {  
  public:  
    int bonus;  
    void setSalary(int s) {  
      salary = s;  
    }  
    int getSalary() {  
      return salary;  
    }  
};  
  
int main() {  
  Programmer myObj;  
  myObj.setSalary(50000);  
  myObj.bonus = 15000;  
  cout << "Salary: " << myObj.getSalary() << "\n";  
  cout << "Bonus: " << myObj.bonus << "\n";  
  return 0;  
}

**Practical Related Questions:**

1.What isinheritance? Explain its uses.

2.List the types of inheritance and Explain them..

3.What a note on access specifiers in inheritance.

**Programs:-**

1 Write a program to implement multilevel inheritance from the figure. Accept and display data for one student.

Class:Student

Data Members: Roll\_no, name

Class: Test

Data Members: Marks1,marks2

Class: Result

Data Members: Total

2. Write a program to implement inheritance as shown in the figure. Assume Suitable member functions :

Class:Student

Data Members: Roll\_no, name

Class: Engg\_Student

Data Members: Sub\_ name

Class: Diploma\_Student

Data Members: Result

3. Write a program to declare a class Student having data members roll\_no and names. Declare one more class Teacher having data members name and subject. Inherit a new class Info form these two base classes. Accept and diplay this data of a Student and Teacher for one object of a class Info .

**Conclusion :-**

Hence, we learnt to implement the concept of inheritance in CPP.

**Practical 7**

**Q1 WAP to implement multilevel inheritance from following figure. Accept and display data for one student**.

Class:Student

Data Members: Roll\_no, name

Class: Test

Data Members: Marks1,marks2

Class: Result

Data Members: Total

#include<iostream.h>

#include<conio.h>

class Student

{

int roll\_no;

char name[30];

public:  
 void gets(){

cout<<”\nEnter roll no and name :”;

cin>>roll\_no>>name;

}

void puts(){

cout<<”\nRoll number : “;

cout<<”\nName of Student : “;

}

};

class Test : public student{

int marks1,marks2;

public:

void getT(){

cout<<”\nEnter marks of First and Second Test : “;

cin>>marks1>>marks2;

}

void putT(){

cout<”\nMarks of Test 1 : “<<marks1;

cout<<”\nMarks of Test 2 : “<<marks2;

}

};

class Result : public Test{

int total;

public:  
 void putR{

cout<<”\nTotal marks : “<<(marks1+marks2);

}

};

void main()

{

Result r;

r.gets();

r.getT();

r.puts();

r.putT();

r.putR();

getch();

}

**Q2. WAP to implement inheritance. Assume suitable member funtions.**

Class:Student

Data Members: Roll\_no, name

Class: Engg\_Student

Data Members: Sub\_ name

Class: Diploma\_Stud1 Write a program to implement multilevel inheritance from the figure. Accept and display data for one student.

2. Write a program to implement inheritance as shown in the figure. Assume Suitable member functions :

3. Write a program to declare a class Student having data members roll\_no and names. Declare one more class Teacher having data members name and subject. Inherit a new class Info form these two base classes. Accept and diplay this data of a Student and Teacher for one object of a class Info .ent

Data Members: Result

#include<iostream.h>

#include<conio.h>

class Student{

int rollno;

char name[30];

public:

void get1(){

cout<<”\nEnter rollno and name :”;

cin>>rollno>>name;

}

void put1(){

cout<<”\nRollNo : “<<rollno;

cout<<”\nName : “<<name;

}

};

class Engg\_stud :: public Student{

char sub\_name[20];

public:

void get2(){

cout<<”\n Enter Subject name :”;

cin>>sub\_name;

}

void put2(){

cout<<”\nSubject Name : “<<sub\_name;

}

};

class Diploma\_stud :: public Student{

char result[20];

public:

void get3()

{

cout<<”\nEnter result : “;

cin>>result;

}

void put3(){

cout<<”\nresult : “<<result;

}

};

void main()

{

Engg\_stud es;

Diploma\_stud ds;

es.get1();

es.get2();

ds.get3();

es.put1();

es.put2();

ds.put3();

getch(); }

**Q3.Write a program to declare class ‘Student’ having data members roll\_no and names. Declare one more class ‘Teacher’ having data members name and subject. Inherit a new class ‘Info’ from these two base classes. Accept and display data of ‘Student’ and ‘Teacher’ for one object of class ‘Info’.**

#include<iostream.h>

#include<conio.h>

class Student{

public:

int roll\_no;

char name[30];

void getS(){

cout<<”\nEnter student roll number and name : “;

cin>>roll\_no>>name;

}

void putS(){

cout<<”\nStudent Roll number = “<<roll\_no;

cout<<”\nStudent name = “<<name;

}

};

class Teacher {

pubic:

char name[30], subject[30];

void get(){

cout<<”\nEnter Teacher name and subject = “;

cin>>name>>subject;

}

void putT(){

cout<<”\nTeacher Name = “<<name;

cout<<”\nTeacher Subject = “<<subject;

}

};

class Info :: public Student, public Teacher{

public:

void putI(){

cout<<”\nFrom Info Class”;

}

};

void main(){

Info i;

clrscr();

i.getS();

i.getT();

i.putS();

i.putT();

i.putI();

getch();

}