Experiment 5

**Aim:** Write a program to Implement

**(a) Single Inheritance**

#include <iostream>

using namespace std;

class Student

{

public:

string name,roll;

student();

void input()

{

cout<<"Enter name and roll number of student \n";

cin>>name>>roll;

}

};

class Percentage : public Student

{

public :

float m1,m2,m3;

void input1()

{

cout<<"Enter marks of student out of 100 \n";

cin>>m1>>m2>>m3;

}

void calculate()

{

float percentage = (m1+m2+m3)/300 \* 100;

cout<<name<<" "<<roll<<" "<<percentage<<" \n";

}

};

int main()

{

cout<<"Code for SINGLE INHERITANCE \n ";

cout<<"Enter the number of students \n";

int n;

cin>>n;

Percentage result[n];

for (int i = 0; i < n; ++i)

{

result[i].input();

result[i].input1();

result[i].calculate();

}

return 0;

}

**(b) Multiple Inheritance**

#include <iostream>

using namespace std;

class A

{

public:

A()

{

cout<<"A constructor is called \n";

}

};

class B

{

public:

B()

{

cout<<"B constructor is called \n";

}

};

class C : public A, public B

{

public:

C()

{

cout<<"C constructor is called \n";

}

};

int main()

{

cout<<"Code for MULTIPLE INHERITANCE \n";

C c;

}

**(c) Multilevel Inheritance**

#include <iostream>

using namespace std;

class Person

{

string name, gender;

int age;

public:

void input()

{

cout << "Name : "; cin >> name;

cout << "Gender : "; cin >> gender;

cout << "Age : "; cin >> age;

}

void output()

{

cout << "Name: " << name << endl;

cout << "Age: " << age << endl;

cout << "Gender: " << gender << endl;

}

};

class Employee : public Person

{

string company;

float salary;

public:

void input()

{

Person :: input();

cout << "Company : "; cin >> company;

cout << "Salary : "; cin >> salary;

}

void output()

{

Person :: output();

cout << "Name of Company: " << company << endl;

cout << "Salary: Rs." << salary << endl;

}

};

class Programmer : public Employee

{

int n;

public:

void input()

{

Employee :: input();

cout << "Number of programming languages known : "; cin >> n;

}

void output()

{

Employee :: output();

cout << "Number of programming language known: " << n;

}

};

int main()

{

Programmer p;

cout<<"Enter data : \n";

p.input();

cout<<"Displaying data : \n";

p.output();

return 0;

}

**(d) Heirarchial Inheritance**

#include <iostream>

using namespace std;

class Person

{

string name, gender;

int age;

public:

void input()

{

cout << "Name : "; cin >> name;

cout << "Gender : "; cin >> gender;

cout << "Age : "; cin >> age;

}

void output()

{

cout << "Name: " << name << endl;

cout << "Age: " << age << endl;

cout << "Gender: " << gender << endl;

}

};

class Student : public Person

{

string institute,degree;

public:

void input()

{

Person :: input();

cout << "Name of institute : "; cin >> institute;

cout << "Degree : "; cin >> degree;

}

void output()

{

Person :: output();

cout<<"institute : "<<institute<<"\n";

cout<<"Degree : "<<degree<<"\n";

}

};

class Employee : public Person

{

string company;

float salary;

public:

void input()

{

Person :: input();

cout << "Name of company : "; cin >> company;

cout << "Salary : "; cin >> salary;

}

void output()

{

Person :: output();

cout << "Name of company : "<< company<<"\n";

cout << "Salary : "<<salary<<"\n";

}

};

int main()

{

cout<<"Student \n"<<"Enter data : \n";

Student s;

s.input();

s.output();

cout<<"Employee \n"<<"Enter data : \n";

Employee e;

e.input();

e.output();

return 0;

}

**(e) Hybrid Inheritance**

#include<iostream.h>

#include<conio.h>

class arithmetic

{

protected:

int num1, num2;

public:

void getdata()

{

cout<<"For Addition:";

cout<<"\nEnter the first number: ";

cin>>num1;

cout<<"\nEnter the second number: ";

cin>>num2;

}

};

class plus:public arithmetic

{

protected:

int sum;

public:

void add()

{

sum=num1+num2;

}

};

class minus

{

protected:

int n1,n2,diff;

public:

void sub()

{

cout<<"\nFor Subtraction:";

cout<<"\nEnter the first number: ";

cin>>n1;

cout<<"\nEnter the second number: ";

cin>>n2;

diff=n1-n2;

}

};

class result:public plus, public minus

{

public:

void display()

{

cout<<"\nSum of "<<num1<<" and "<<num2<<"= "<<sum;

cout<<"\nDifference of "<<n1<<" and "<<n2<<"= "<<diff;

}

};

void main()

{

clrscr();

result z;

z.getdata();

z.add();

z.sub();

z.display();

getch();

}