**Practical No.5**

**Inheritance**

**Program 5(a):**

Design a class for single level inheritance using public and private type derivation.

**Example:**

Write a program to add two numbers using single inheritance such that the base class function must accept the two numbers from the user and the derived class function must add these numbers and display the sum.

**Coding:**

#include<iostream.h>

#include<conio.h>

class data

{

protected:

int a,b;

public:

void read()

{

cout<<"Enter two numbers:";

cin>>a>>b;

}

};

class sum:private data

{

private:

int sum;

public:

void add()

{

read();

sum=a+b;

}

void display()

{

cout<<"The sum is "<<sum;

}

};

void main()

{

clrscr();

sum s;

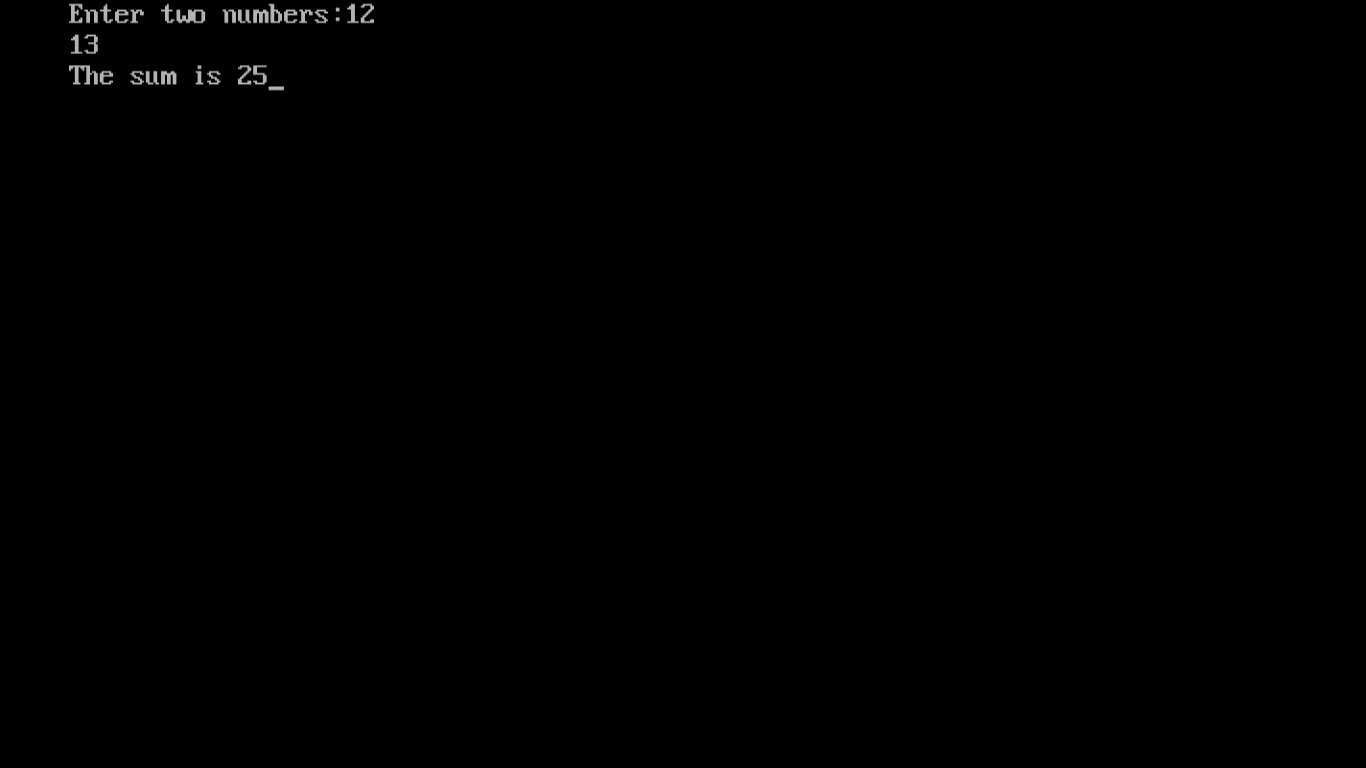
s.add();

s.display();

getch();

}

**Output:**



**Program 5(b):**

Design a class for multiple inheritance.

**Example:**

Write a program to define the following relationship using multiple inheritance.

**Coding:**

#include<iostream.h>

#include<conio.h>

class polygon

{

protected:

int height,width;

public:

void read(int a,int b)

{

height=a;

width=b;

}

};

class output

{

public:

void showData(int x)

{

cout<<"Area is "<<x;

}

};

class rectangle:public polygon, public output

{

public:

int area()

{

return(height\*width);

}

};

class triangle:public polygon, public output

{

public:

int area()

{

return(height\*width/2);

}

};

void main()

{

clrscr();

int h,w,choice,a;

cout<<"1.Area of rectangle\n2.Area of triangle\nEnter your choice:";

cin>>choice;

cout<<"Enter height and width:";

cin>>h>>w;

switch(choice)

{

case 1:rectangle r;

r.read(h,w);

a=r.area();

r.showData(a);

break;

case 2:triangle t;

t.read(h,w);

a=t.area();

t.showData(a);

break;

default:cout<<"Invalid choice";

}

getch();

}

**Output:**



**Program 5(c):**

Implement the hierarchical inheritance.

**Example:**

Write a program to define the following inheritance relationship.

Staff

Code name

Teacher Typist officer

Subject experience speed experience grade department

Regular casual

Salary daily wages

**Coding:**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

class staff

{

protected:

char name[20];

int code;

};

class teacher:public staff

{

private:

char subject[20];

int experience;

public:

void read()

{

cout<<"Enter name, code, subject and experience of the teacher:";

gets(name);

cin>>code;

gets(subject);

cin>>experience;

}

void display()

{

cout<<"Teacher details:\nName:"<<name<<"\nCode:"<<code<<"\nSubject:"

<<subject<<"\nExperience:"<<experience;

}

};

class office:public staff

{

private:

char dept[20];

int grade;

public:

void read()

{

cout<<"Enter name, code, department and grade of the officer:";

gets(name);

cin>>code;

gets(dept);

cin>>grade;

}

void display()

{

cout<<"Officer details:\nName:"<<name<<"\nCode:"<<code<<"\nDepartment:"<<dept

<<"\nGrade:"<<grade;

}

};

class typist:public staff

{

protected:

int speed, experience;

};

class regular:public typist

{

private:

int salary;

public:

void read()

{

cout<<"Enter name, code, speed, experience and salary of the regular typist:";

gets(name);

cin>>code>>speed>>experience>>salary;

}

void display()

{

cout<<"Regular typist details:\nName:"<<name<<"\nCode:"<<code<<"\nSpeed:"

<<speed<<"\nExperience:"<<experience<<"\nSalary:"<<salary;

}

};

class casual:public typist

{

private:

int dailywages;

public:

void read()

{

cout<<"Enter name, code, speed, experience and dailywages of the casual typist:";

gets(name);

cin>>code>>speed>>experience>>dailywages;

}

void display()

{

cout<<"Casual typist details:\nName:"<<name<<"\nCode:"<<code<<"\nSpeed:"<<speed<<"\nExperience:"<<experience<<"\nDailywages:"<<dailywages;

}

};

void main()

{

clrscr();

int choice;

cout<<"1.Teacher\n2.Officer\n3.Regular typist\n4.Casual typist\nEnter the choice, whose details you want to enter:";

cin>>choice;

switch(choice)

{

case 1:teacher t;

t.read();

t.display();

break;

case 2:office o;

o.read();

o.display();

break;

case 3:regular r;

r.read();

r.display();

break;

case 4:casual c;

c.read();

c.display();

break;

default:cout<<"Invalid choice";

}

getch();

}

**Output:**

