**Practical : 1**

**Aim :** WAP that defines a shape class with a constructor that gives value to width and height. The define two sub-classes triangle and rectangle, that calculate the area of the shape area (). In the main, define two variables a triangle and a rectangle and then call the area() function in this two varibles.

**Program :**

#include <iostream>

using namespace std;

class Shape

{

private:

double width, height;

public:

Shape(double w, double h)

{

this->width=w;

this->height=h;

}

double getWidth()

{

return width;

}

double getHeight()

{

return height;

}

};

class Rectangle: public Shape

{

public:

Rectangle(double width, double height):Shape(width,height)

{

}

double area()

{

return (getWidth()\*getHeight());

}

};

class Triangle: public Shape

{

public:

Triangle(double base, double height): Shape(base,height)

{

}

double area()

{

return (getWidth()\*getHeight())/2.0;

}

};

int main ()

{

Rectangle rectangle(5.0,3.0);

Triangle triangle(2.0,5.0);

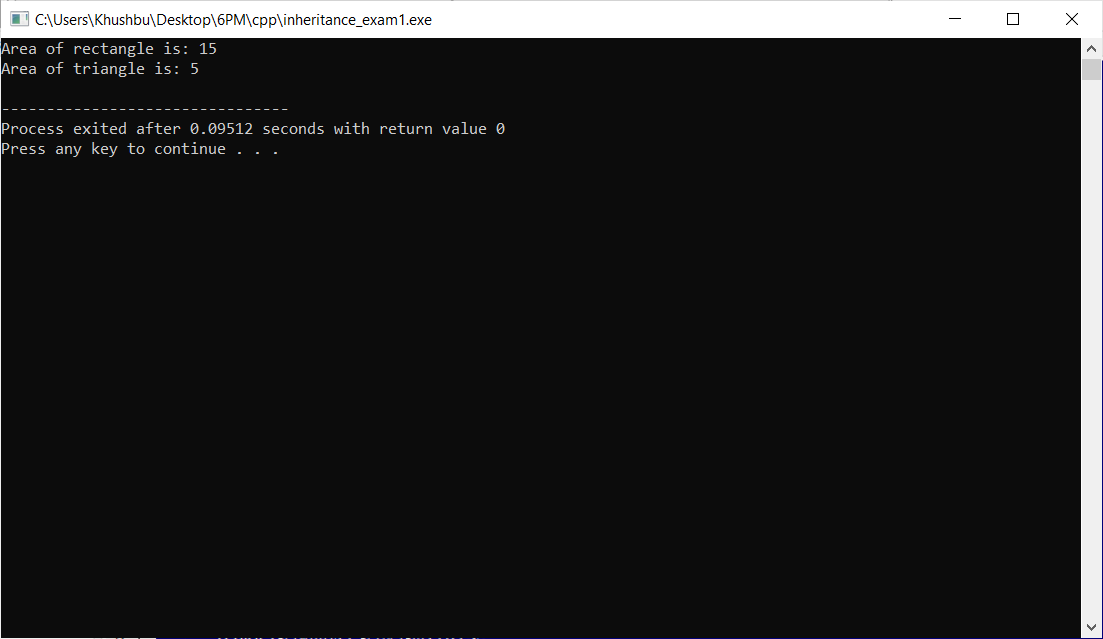
cout <<"Area of rectangle is: "<< rectangle.area() << endl;

cout <<"Area of triangle is: "<< triangle.area() << endl;

return 0;

}

**Output :**

****

**Practical : 2**

**Aim :**  WAP with a mother class and an inherited daugther class.Both of them should have a method void display ()that prints a message (different for mother and daugther). In the main define a daughter and call the display() method on it.

**Program :**

#include<iostream>

using namespace std;

class Mother

{

public :

char name[100];

int age ;

void m\_setData()

{

cout<<endl<<"enter MOTHER name :";

cin>>this->name;

cout<<endl<<"enter MOTHER age :";

cin>>this->age;

}

void m\_display()

{

cout<<endl<<"MOTHER name :"<<this->name;

cout<<endl<<"MOTHER age :"<<this->age;

}

};

class Daughter : public Mother

{

public :

int age;

char name[100];

void d\_setData()

{

cout<<endl<<"enter DAUGHTER name :";

cin>>this->name;

cout<<endl<<"enter DAUGHTER age :";

cin>>this->age;

}

void d\_display()

{

cout<<endl<<"DAUGHTER name :"<<this->name;

cout<<endl<<"DAUGHTER age :"<<this->age;

}

};

int main()

{

Daughter d;

int n,i;

cout<<endl<<"how many mother and daughter enter :";

cin>>n;

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

{

d.m\_setData();

d.d\_setData();

}

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

{

d.m\_display();

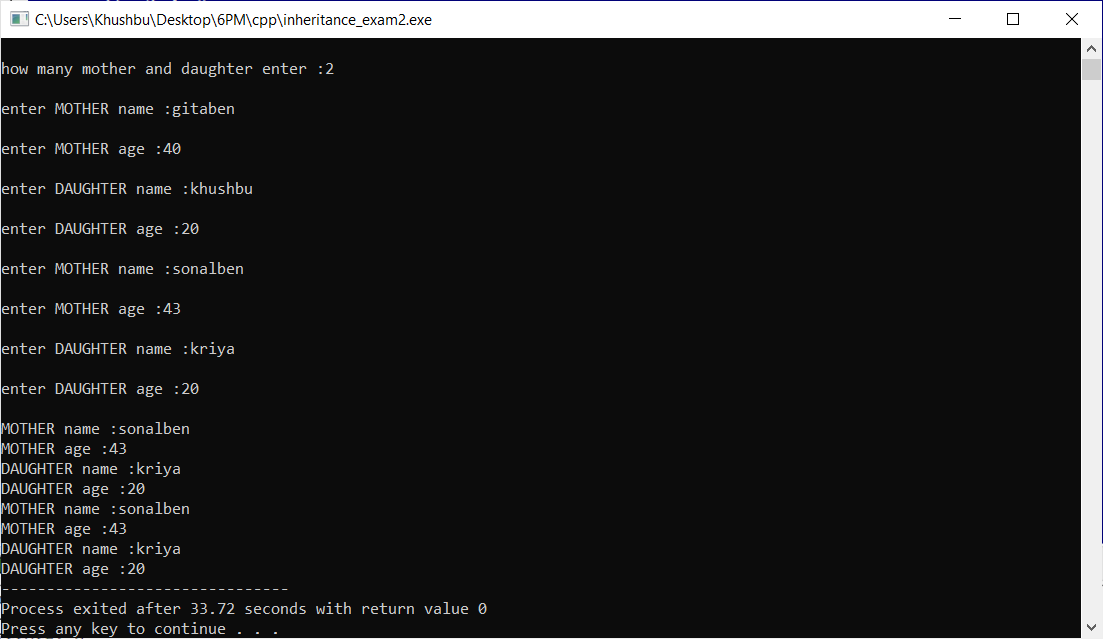
d.d\_display();

}

return 0;

}

**Output :**

****

**Practical : 3**

**Aim :** WAP with a mother class animal. Inside it define a name and an age variables, and set\_value() function. Then create two bases variables Zebra and Dolphin which write a message telling the age, the name and giving some extra information (e.g. place of origin).

**Program :**

#include <iostream>

#include <string.h>

using namespace std;

class Animal

{

protected:

int age;

char name[10];

public:

void set\_data ()

{

age = this->age;

strcpy(name,this->name);

}

};

class Zebra:public Animal

{

public:

void message\_zebra() {

age = 20;

strcpy(name,"abc");

cout<< "The zebra named "<< name <<" is "<< age << " years old. The zebra comes from Africa. \n";

}

};

class Dolphin: public Animal

{

public:

void message\_dolphin() {

age = 10;

strcpy(name,"def");

cout<< "The dolphin named "<< name << " is "<<age << " years old. The dolphin comes from New Zeland.\n";

}

};

int main ()

{

Zebra zeb;

Dolphin dol;

zeb.set\_data();

dol.set\_data();

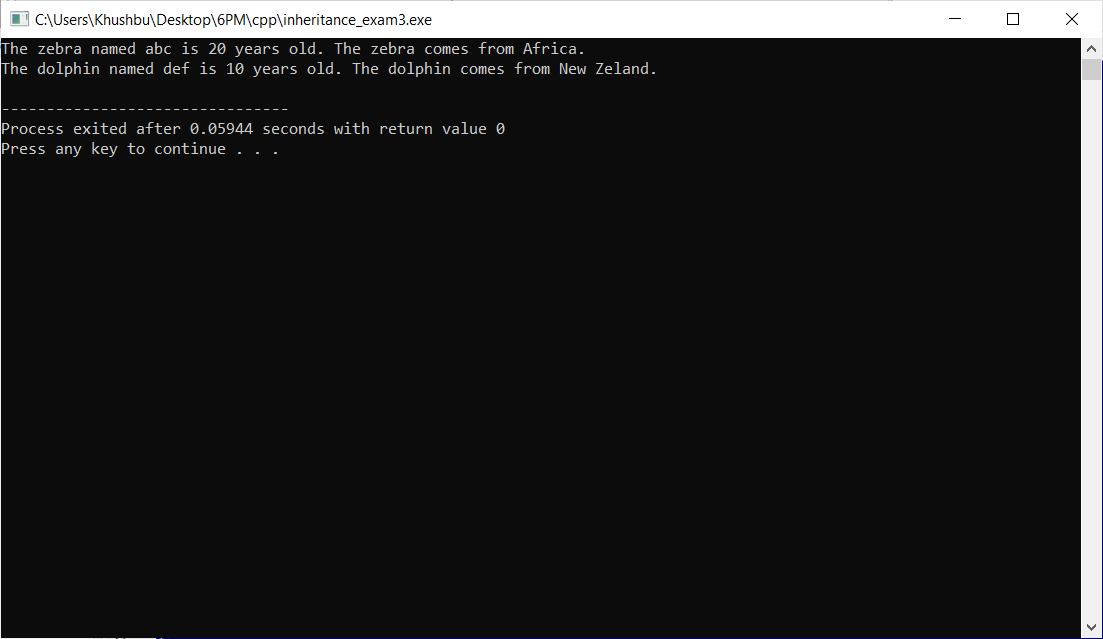
zeb.message\_zebra() ;

dol.message\_dolphin() ;

return 0;

}

**Output :**

****

**Practical : 4**

**Aim :** WAP to read and print employee information using multiple inheritance.

**Program :**

#include<iostream>

using namespace std;

class Emp

{

public :

int e\_id;

char e\_name[100];

char e\_designation[100];

char e\_age[100];

void setData()

{

cout<<endl<<"enter employee id :";

cin>>this->e\_id;

cout<<endl<<"enter employee name :";

cin>>this->e\_name;

cout<<endl<<"enter employee designation :";

cin>>this->e\_designation;

cout<<endl<<"enter employee age :";

cin>>this->e\_age;

}

void getData()

{

cout<<endl<<"employee id :"<<this->e\_id;

cout<<endl<<"employee name :"<<this->e\_name;

cout<<endl<<"employee designation :"<<this->e\_designation;

cout<<endl<<"employee age :"<<this->e\_age;

}

};

class Salary

{

public :

float monthly\_salary;

void setSalary()

{

cout<<"enter monthly salary :";

cin>>this->monthly\_salary;

}

void getSalary()

{

cout<<"monthly salary :"<<this->monthly\_salary;

}

};

class Hour : public Emp,public Salary

{

public :

float hourlyworked,hourlywages,earning;

void setHour()

{

cout<<endl<<"entern hourly worked :";

cin>>this->hourlyworked;

cout<<endl<<"entern hourly wages :";

cin>>this->hourlywages;

}

void getHour()

{

cout<<endl<<"hourly worked :"<<this->hourlyworked;

cout<<endl<<"hourly wages :"<<this->hourlywages;

}

};

int main()

{

Hour h;

h.setData();

h.setSalary();

h.setHour();

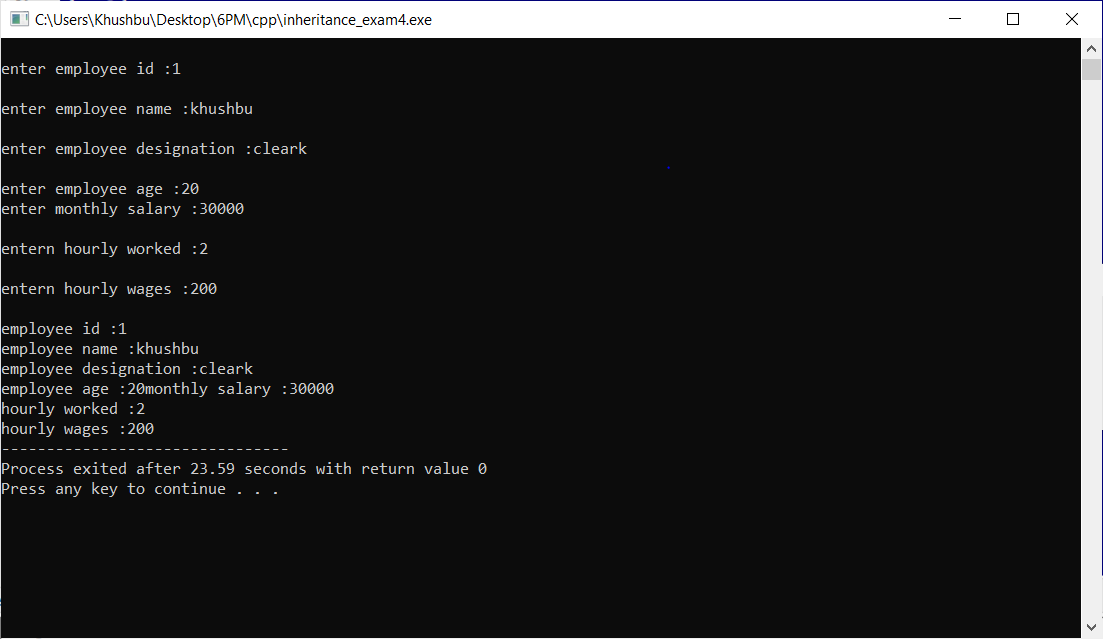
h.getData();

h.getSalary();

h.getHour();

}

**Output :**

****

**Practical : 5**

**Aim :** WAP to demonstrate example of hierarchical inheritance to get square and cube of a number.

**Program :**

#include<iostream>

using namespace std;

class square\_cube

{

public :

int n;

};

class square : public square\_cube

{

public :

int ans;

void setData()

{

cout<<"enter value of N sqaure:";

cin>>this->n;

}

void getSquare()

{

ans = n \* n;

cout << endl << "square is : "<<this->ans << endl;

}

};

class cube : public square\_cube

{

public :

int ans;

void setData()

{

cout<<"enter value of N cube:";

cin>>this->n;

}

void getcube()

{

ans = n \* n \* n;

cout << endl << "cube is : "<<this->ans << endl;

}

};

int main()

{

square s;

cube c;

s.setData();

c.setData();

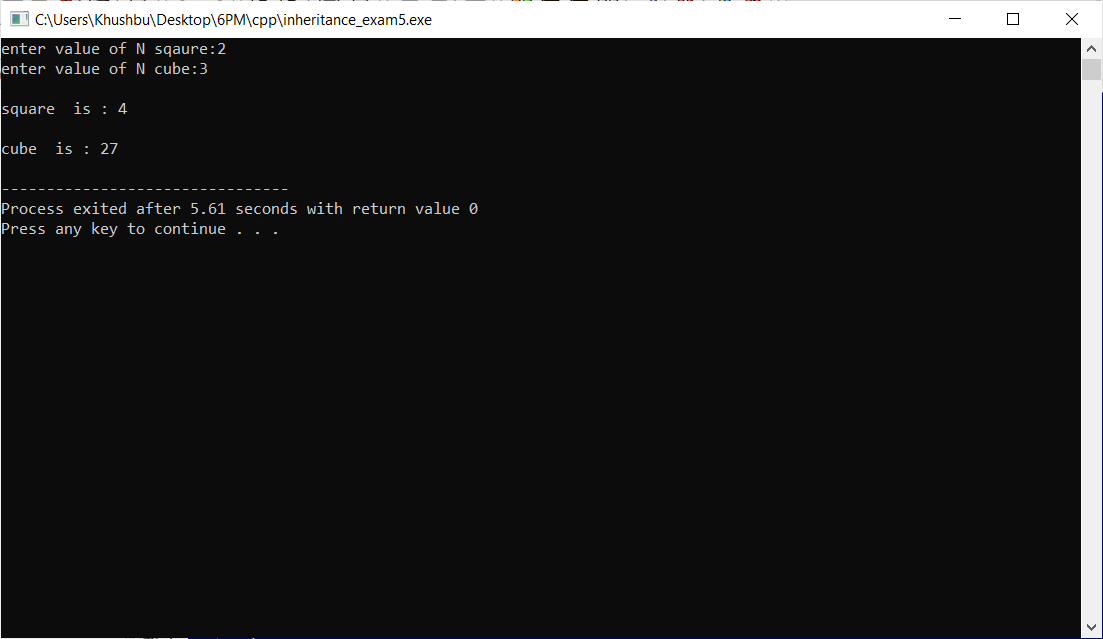
s.getSquare();

c.getcube();

return 0;

}

**Output :**

****

**Practical : 6**

**Aim :** WAP to read and print employee information with use of multilevel inheritance. (as like in below image)

**Program :**

#include<iostream>

using namespace std;

class Emp

{

public :

int e\_id;

char e\_name[100];

char e\_designation[100];

char e\_age[100];

void setData()

{

cout<<endl<<"enter employee id :";

cin>>this->e\_id;

cout<<endl<<"enter employee name :";

cin>>this->e\_name;

cout<<endl<<"enter employee designation :";

cin>>this->e\_designation;

cout<<endl<<"enter employee age :";

cin>>this->e\_age;

}

void getData()

{

cout<<endl<<"employee id :"<<this->e\_id;

cout<<endl<<"employee name :"<<this->e\_name;

cout<<endl<<"employee designation :"<<this->e\_designation;

cout<<endl<<"employee age :"<<this->e\_age;

}

};

class Salary : public Emp

{

public :

float monthly\_salary;

int exp;

void setSalary()

{

cout<<"enter monthly salary :";

cin>>this->monthly\_salary;

cout<<"enter experience :";

cin>>this->exp;

}

void getSalary()

{

cout<<"monthly salary :"<<this->monthly\_salary;

cout<<"experience :"<<this->exp;

}

};

class company : public Salary

{

public :

char companyname[100];

char address[100];

void setCompnay()

{

cout<<endl<<"enter company name :";

cin>>this->companyname;

cout<<endl<<"enter address :";

cin>>this->address;

}

void getCompnay()

{

cout<<endl<<"company name :"<<this->companyname;

cout<<endl<<"address :"<<this->address;

}

};

class Email : public company

{

public :

char email[100];

int contactno;

void setemail()

{

cout<<endl<<"enter email :";

cin>>this->email;

cout<<endl<<"enter contact no :";

cin>>this->contactno;

}

void getemail()

{

cout<<endl<<"email :"<<this->email;

cout<<endl<<"contact no :"<<this->contactno;

}

};

int main()

{

Email e;

e.setData();

e.setSalary();

e.setCompnay();

e.setemail();

e.getData();

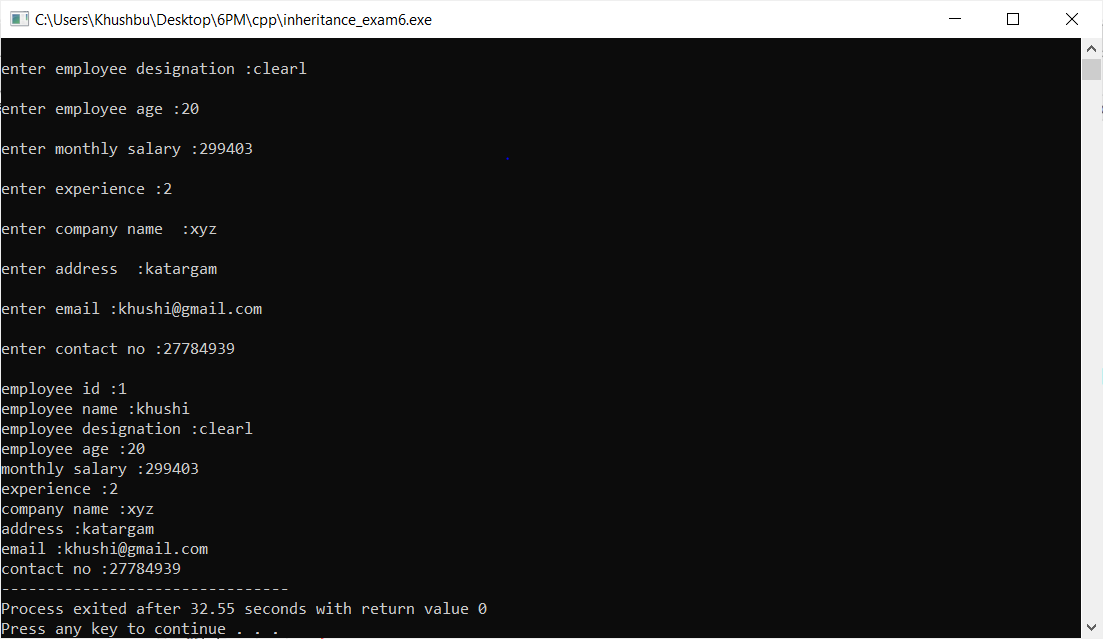
e.getSalary();

e.getCompnay();

e.getemail();

}

**Output :**

****