**Practice-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

{

public:

int width;

int height;

int area;

public:

Shape()

{

cout << "Enter value of height :- "; cin >> height;

cout << "Enter value of width :- "; cin >> width;

cout << "---------------------------------" << endl<<endl;

}

};

class Rectangle : public Shape

{

public:

void Areaofrectangle()

{

area = width \* height;

cout <<endl<<"Area of rectangle :- " << area;

}

};

class Triangle : public Shape

{

public:

void Areaoftriangle()

{

area = (width \* height)/2;

cout <<endl<< "Area of Triangle :- " << area;

}

};

int main()

{

Rectangle r1;

Triangle t1;

shape s1;

r1.Areaofrectangle();

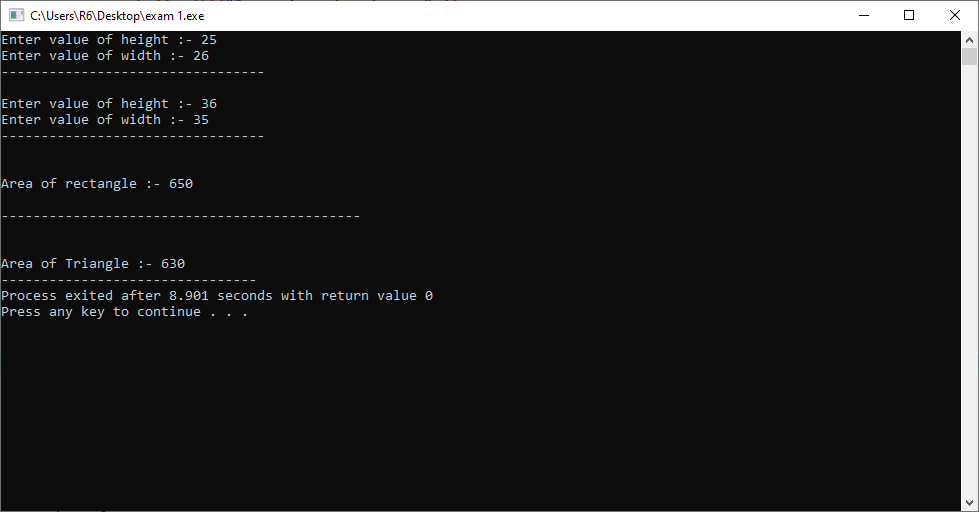
cout << endl << endl << "---------------------------------------------"<< endl << endl ;

t1.Areaoftriangle();

return 0;

}

**Output:**

****

**Practice-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 :

void Mdisplay()

{

cout << "PARENT class is :- MOTHER"<<endl;

}

};

class Daughter : public Mother

{

public:

void Ddislay()

{

cout << "CHILD class is :- DAUGHTER"<<endl;

}

};

int main()

{

Daughter d;

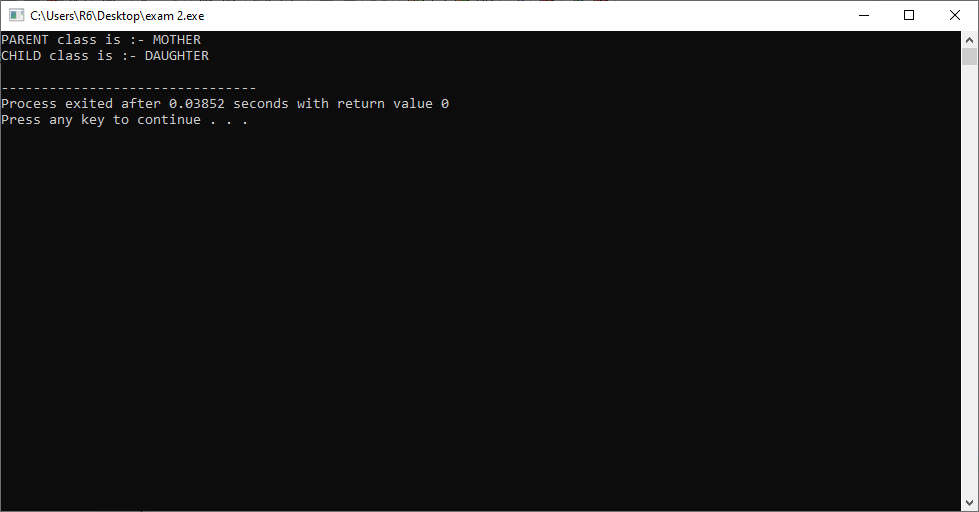
d.Mdisplay();

d.Ddislay();

return 0;

}

**Output:**

****

**Practice-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

{

public:

int age;

char name[100];

char origin[100];

};

class Zebra : public Animal

{

public:

void getdetailofzebra()

{

cout << "Enter Zebra's name is :- "; cin>>name;

cout << "Enter Zebra's age is :- "; cin>>age;

cout << "Enter Zebra come from :- "; cin>>origin;

}

void printdetailofzebra()

{

cout << "Zebra's name is :- "<<name<<endl;

cout << "Zebra's age is :- "<<age<<endl;

cout << "Zebra come from :- "<<origin<<endl;

}

};

class Dolphin : public Animal

{

public :

void getdetailofDolphin()

{

cout << "Enter Dolphin's name is :- "; cin>>name;

cout << "Enter Dolphin's age is :- "; cin>>age;

cout << "Enter Dolphin come from :- "; cin>>origin;

}

void printdetailofDolphin()

{

cout << "Dolphin's name is :- "<<name<<endl;

cout << "Dolphin's age is :- "<<age<<endl;

cout << "Dolphin come from :- "<<origin<<endl;

}

};

int main()

{

Zebra z1;

Dolphin d1;

z1.getdetailofzebra();

cout << endl<<endl;

d1.getdetailofDolphin();

cout << endl << endl << "----------------------------------------------------------" << endl << endl ;

z1.printdetailofzebra();

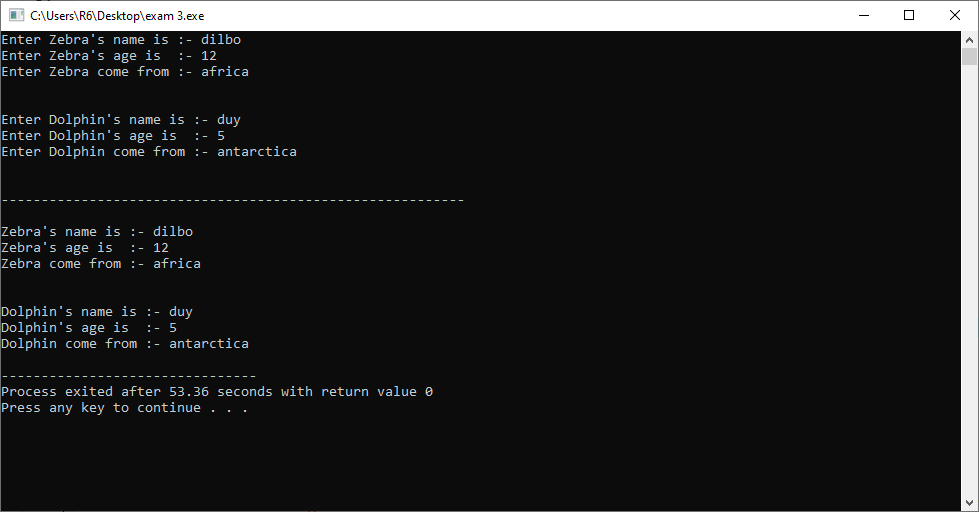
cout << endl<<endl;

d1.printdetailofDolphin();

return 0;

}

**Output:**

****

**Practice-4**

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

**Program:**

#include<iostream>

using namespace std;

class privateinfo

{

public :

int emp\_id; char emp\_name[100];

int emp\_age;

public:

void getprivateinfo()

{

cout << "Enter Employees Id:- "; cin >>this->emp\_id;

cout << "Enter Employees Name:- "; cin >>this->emp\_name;

cout << "Enter Employees Age:- "; cin >>this->emp\_age;

}

};

class Departmentinfo

{

public:

char emp\_role[100]; int emp\_experience;

char emp\_email[100];

char emp\_company\_name[100];

public:

void getdepartmentinfo()

{

cout << "Enter Employees Role:- "; cin >>this->emp\_role;

cout << "Enter Employees Email:- "; cin >>this->emp\_email;

cout << "Enter Employees Expericence:- "; cin >>this->emp\_experience;

cout << "Enter Employees Company name:-"; cin >>this->emp\_company\_name;

}

};

class Employee : public privateinfo,public Departmentinfo

{

public:

void getemployeedetails()

{

cout << "\*\*\*\*\*\*\*\*:Enter Emloyees details:\*\*\*\*\*\*\*\* " << endl ;

getprivateinfo();

getdepartmentinfo();

}

void printemployeedetails()

{

cout << endl << "========:Employees Information is:======== " << endl << endl ;

cout << "Id: " << this->emp\_id << endl;

cout << "Name: " << this->emp\_name << endl ;

cout << "Age: " << this->emp\_age << endl;

cout << "Role: " << this->emp\_role << endl;

cout << "Email: " << this->emp\_email << endl;

cout << "Expericence: " << this->emp\_experience << endl;

cout << "Company name: " << this->emp\_company\_name << endl;

}

};

int main()

{

Employee e[100];

int i,n;

cout << "How Many Employees:: "; cin >> n;

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

{

e[i].getemployeedetails();

}

cout << endl << endl <<"---------------------------------------------------------"<<endl<<endl;

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

{

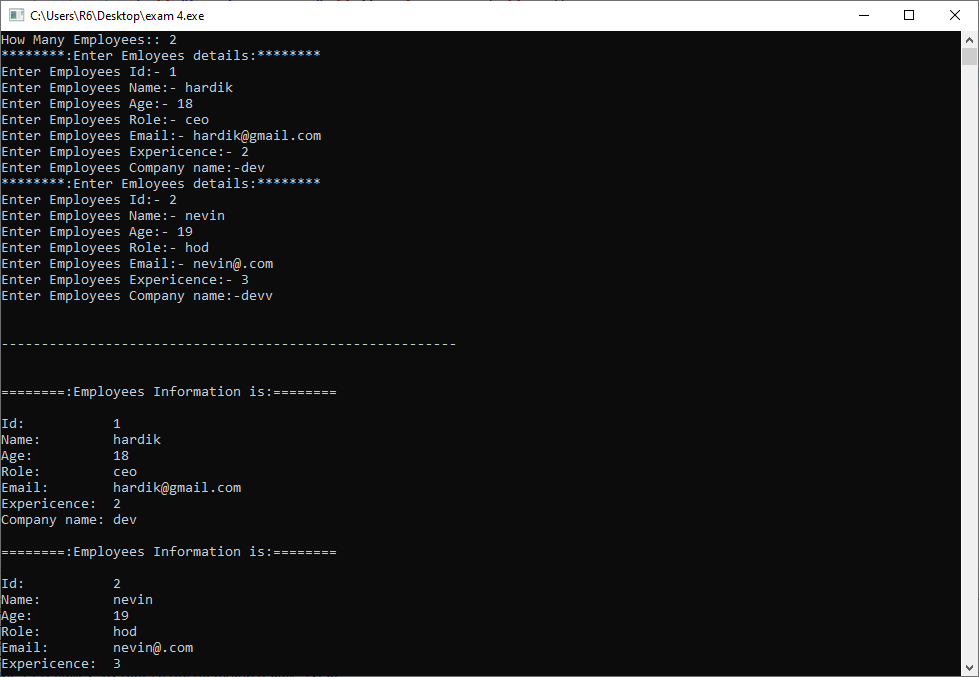
e[i].printemployeedetails();

}

return 0;

}

**Output:**

****

**Practice-5**

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

**Program:**

#include<iostream>

using namespace std;

class A

{

public:

int a;

int b;

int x;

int y;

void getnumber()

{

cout << "Enter a Value :- " ; cin >> a;

}

};

class B : public A

{

public :

void square()

{

getnumber();

x=a;

x=a\*a;

cout << endl << "Square of " << a << " is : => " << x;

}

};

class C : public A

{

public :

void cube()

{

getnumber();

y=a;

y=a\*a\*a;

cout << endl << "Cube of " << a << " is : => " << y;

}

};

int main()

{

B b1;

C c1;

b1.square();

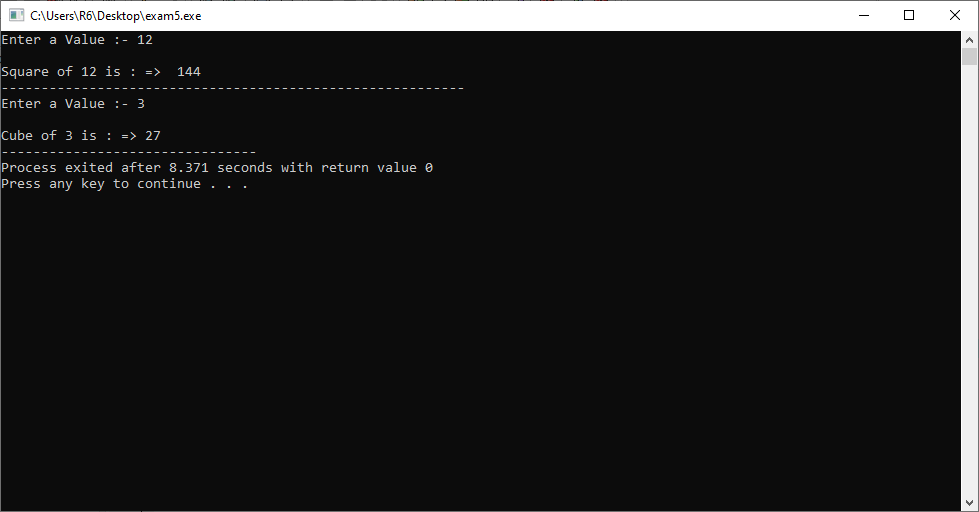
cout << endl << "----------------------------------------------------------"<<endl;

c1.cube();

return 0;

}

**Output:**

****

**Practice-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 A

{

public :

int emp\_id; char emp\_name[100];

int emp\_experience; char emp\_role[100];

int emp\_salary; char emp\_email[100];

int emp\_contact; char emp\_address[100];

char emp\_company\_name[100];

public:

void setprivateinfo()

{

cout << "Enter Employee Id:- "; cin >>this->emp\_id;

cout << "Enter Employee Name:- "; cin >>this->emp\_name;

cout << "Enter Employee Role:- "; cin >>this->emp\_role;

}

};

class B : public A

{

public:

void setdepartmentinfo()

{

cout << "Enter Employees Salary: "; cin >>this->emp\_salary;

cout << "Enter Employees Experience: "; cin >>this->emp\_experience;

}

};

class C : public B

{

public :

void setinfo()

{

cout << "Enter Employees Company name: "; cin >>this->emp\_company\_name;

cout << "Enter Employees Address: "; cin >>this->emp\_address;

}

void getinfo()

{

cout << "NAME :- " << this->emp\_name;

cout << "ROLE :- " << this->emp\_role;

cout << "SALARY :-" << this->emp\_salary;

}

};

class D : public C

{

public:

void employeedetails()

{

cout << "Enter Emloyees Email: "; cin >>this->emp\_email;

cout << "Enter Emloyees Contact: "; cin >>this->emp\_contact;

}

void getemployeedetails()

{

cout << endl << "========:Enter Emloyees details:======== " << endl << endl;

setprivateinfo();

setdepartmentinfo();

setinfo();

employeedetails();

}

void printemployeedetails()

{

cout << endl << "Employees Information is: " << endl << endl ;

cout << "Id: " << this->emp\_id << endl;

cout << "Name: " << this->emp\_name << endl;

cout << "Role: " << this->emp\_role << endl;

cout << "Salary: " << this->emp\_salary << endl;

cout << "experience: " << this->emp\_experience << endl;

cout << "Company name: " << this->emp\_company\_name << endl;

cout << "Address: " << this->emp\_address << endl;

cout << "Email: " << this->emp\_email << endl;

cout << "Contact: " << this->emp\_contact;

}

};

int main()

{

D e[100];

int i,n;

cout << "How Many Employees:: ";

cin >> n;

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

{

e[i].getemployeedetails();

}

cout << endl << endl << "--------------------------------------------------------"<<endl<<endl;

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

{

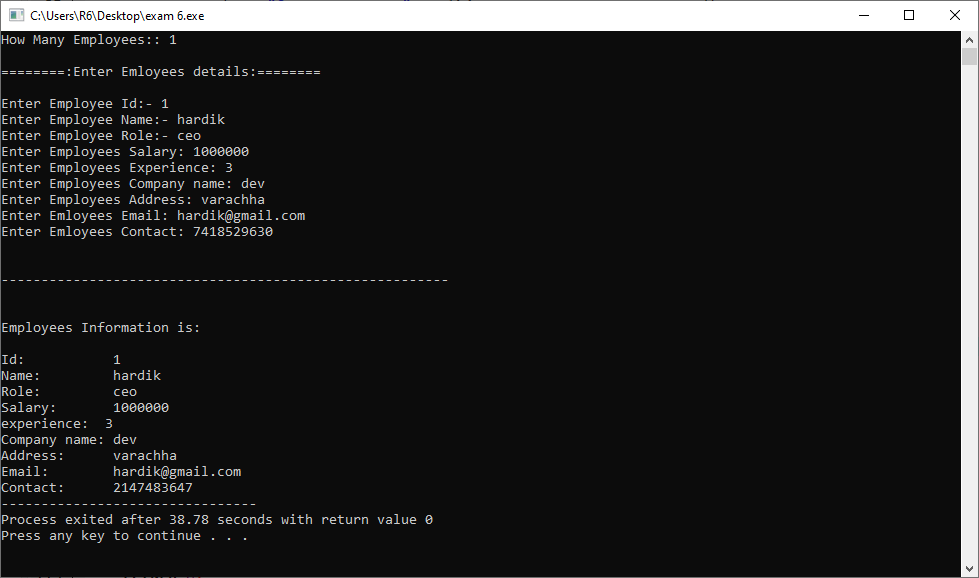
e[i].printemployeedetails();

}

return 0;

}

**Output:**

****