**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

{

public:

int width;

int height;

int area;

public:

Shape()

{

cout << "\* Enter width :- "; cin >> width;

cout << "\* Enter height :- "; cin >> height;

cout << endl;

}

};

class Rectangle : public Shape

{

public:

void RectangleArea()

{

area = width \* height;

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

}

};

class Triangle : public Shape

{

public:

void TriangleArea()

{

area = (width \* height)/2;

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

}

};

int main()

{

Rectangle r1;

Triangle t1;

r1.RectangleArea();

t1.TriangleArea();

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>

#include<string.h>

using namespace std;

class Mother

{

public :

void Motherdisplay()

{

cout << "\* This is a mother class. " <<endl;

}

};

class Daughter : public Mother

{

public:

void Daughterdisplay()

{

cout << "\* This is a daughter class. " <<endl;

}

};

int main()

{

Daughter d1;

d1.Daughterdisplay();

d1.Motherdisplay();

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

{

public:

int age;

char name[100];

char place[100];

};

class Zebra : public Animal

{

public:

void setData()

{

cout << "\* Enter zebra name : "; cin >>name;

cout << "\* Ente zebra age : "; cin >>age;

cout << "\* Enter zebra place : "; cin >>place;

}

void getData()

{

cout << endl <<"=> The name of zebra is "<< name << endl

<< "=> The sge of Zebra is "<< age << endl

<< "=> Zebra come from "<< place << endl;

}

};

class Dolphin : public Animal

{

public:

void setData()

{

cout << endl << "\* Enter dolphin name : "; cin >>name;

cout << "\* Ente dolphin age : "; cin >>age;

cout << "\* Enter dolphin place : "; cin >>place;

}

void getData()

{

cout << endl << "=> The name of dolphin is "<< name << endl

<< "=> The age of dolphin is "<< age << endl

<< "=> Dolphin come from "<< place << endl;

}

};

int main()

{

Zebra z1;

Dolphin d1;

z1.setData();

d1.setData();

cout << endl;

z1.getData();

d1.getData();

return 0;

}

**Output:**

****

**Practical-4**

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

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class A

{

public:

int emp\_id;

char emp\_name[100];

int emp\_age;

static char emp\_company\_name[100];

public:

void setA()

{

cout<<"\* Enter ID : "; cin >> emp\_id;

cout<<"\* Enter name : "; cin >> emp\_name;

cout<<"\* Enter age : "; cin >> emp\_age;

}

};

class B

{

public:

char emp\_role[100];

int emp\_salary;

char emp\_email[100];

char emp\_city[100];

int emp\_experience;

public:

void setB()

{

cout<<"\* Enter Salary : "; cin >> emp\_salary;

cout<<"\* Enter Role : "; cin >> emp\_role;

cout<<"\* Enter Email : "; cin >> emp\_email;

cout<<"\* Enter City : "; cin >> emp\_city;

cout<<"\* Enter Experience : "; cin >> emp\_experience;

}

};

class Employee : public A ,public B

{

public:

void getAllData()

{

cout <<"=> ID: "<< emp\_id <<endl

<<"=> Name: "<< emp\_name <<endl

<<"=> Age: "<< emp\_age <<endl

<<"=> Salary: "<< emp\_salary <<endl

<<"=> Role: "<< emp\_role <<endl

<<"=> Email: "<< emp\_email <<endl

<<"=> City: "<< emp\_city <<endl

<<"=> Experience: "<< emp\_experience <<endl;

}

};

char A :: emp\_company\_name[100] = "IT hub";

int main()

{

Employee emp[100];

int i,n;

cout <<"\* How many employee: ";

cin >> n;

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

{

emp[i].setA();

emp[i].setB();

}

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

{

emp[i].getAllData();

}

return 0;

}

**Output:**



**Practical-5**

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

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Number

{

public:

int n;

public:

void getNumber()

{

cout << endl;

cout << "\* Enter any number : ";

cin >> n;

}

};

class Square : public Number

{

public:

int square;

public:

void getSquare()

{

square = n\*n;

cout << "=> Square of this number :- "<< square <<endl;

}

};

class Cube : public Number

{

public:

int cube;

public:

void getCube()

{

cube = n\*n\*n;

cout << "=> Cube of this number :- "<< cube <<endl;

}

};

int main()

{

Square s;

Cube c;

s.getNumber();

s.getSquare();

c.getNumber();

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>

#include<string.h>

using namespace std;

class A

{

public :

int emp\_id;

char emp\_name[100];

int emp\_age;

int emp\_salary;

int emp\_experience;

char emp\_role[100];

char emp\_email[100];

static char emp\_company\_name[100];

};

class B : public A

{

public :

void setB()

{

cout << "\* Enter employee details :- " << endl;

cout << "\* Enter employee id :- ";

cin >> this->emp\_id;

cout << "\* Enter employee name :- ";

cin >> this->emp\_name;

cout << "\* Enter employee age :- ";

cin >> this->emp\_age;

}

};

class C : public B

{

public :

void setC()

{

cout << "\* Enter employee salary :- ";

cin >> this->emp\_salary;

cout << "\* Enter employee experience :- ";

cin >> this->emp\_experience;

cout << "\* Enter employee role :- ";

cin >> this->emp\_role;

cout << "\* Enter employee email :- ";

cin >> this->emp\_email;

cout << endl;

}

};

class D : public C

{

public :

void getAllData()

{

cout << endl << endl << "=> Employee details :- " << endl;

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

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

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

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

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

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

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

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

}

};

char A :: emp\_company\_name[100] = "IT hub";

int main()

{

D s[100];

int i,n;

cout <<"=> How many employee :- "; cin >> n;

cout << endl;

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

{

s[i].setB();

s[i].setC();

}

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

{

s[i].getAllData();

}

return 0;

}

**Output:**

****