**Task 1:-**

**Code:**

#include <iostream>

using namespace std;

class Shape

{

protected:

int width, height;

public:

Shape( int a = 0, int b = 0)

{

width = a;

height = b;

}

virtual int area()

{

cout << "Parent class area :" <<endl;

return 0;

}

};

class Rectangle: public Shape

{

public:

Rectangle( int a = 0, int b = 0):Shape(a, b) { }

int area ()

{

cout << "Rectangle class area :";

return (width \* height);

}

};

class Triangle: public Shape

{

public:

Triangle( int a = 0, int b = 0):Shape(a, b) { }

int area ()

{

cout << "Triangle class area :";

return (width \* height / 2);

}

};

//square class as described in lab task

class Square: public Shape

{

public:

Square( int a = 0, int b = 0):Shape(a, b) { }

int area ()

{

cout << "Square class area :";

return ((2\*width) + (2\*height));

}

};

// Main function for the program

int main()

{

Shape \*shape;

Rectangle rec(10,7);

Triangle tri(10,5);

// store the address of Rectangle

shape = &rec;

// call rectangle area.

cout<<shape->area()<<endl;

// store the address of Triangle

shape = &tri;

// call triangle area.

cout<<shape->area()<<endl;

cout<<endl<<"----Lab task----"<<endl;

//lab task starts

Square sq(4,4);

// store the address of Square

shape = &sq;

// call Square area.

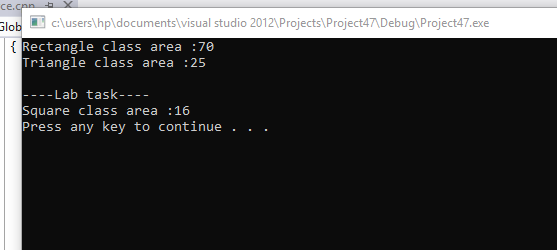
cout<<shape->area()<<endl;

system("pause");

return 0;

}

**Output:**



**Task 2:-**

**Code:**

#include <iostream>

using namespace std;

class Person

{

public:

virtual void introduce()

{

cout<<"Hey from person "<<endl;

}

};

class Student: public Person

{

public:

void introduce()

{

cout<<"Hey from student "<<endl;

}

};

class Farmer: public Person

{

public:

void introduce()

{

cout<<"Hey from farmer "<<endl;

}

};

void WhosThis(Person &p)

{

p.introduce();

}

int main()

{

Person p;

Student s;

Farmer f;

WhosThis(p);

WhosThis(s);

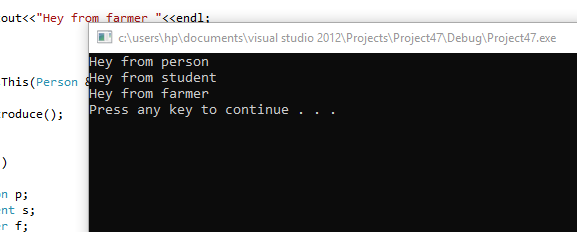
WhosThis(f);

system("pause");

return 0;

}

**Output:**



**Task 3:-**

**Code:**

#include<iostream>

#include<string>

using namespace std;

class Worker

{

private:

char workerName;

char \*ptr;

string SSN;

public:

Worker(): SSN("")

{

ptr=&workerName;

ptr= new char(0);

}

Worker(string x): SSN(x)

{

ptr=&workerName;

ptr= new char[SSN.length()+1];

}

~Worker (){

delete [] ptr;

}

void setWorkerName()

{

cout<<"Enter worker's name : "<<endl;

if(workerName!=0)

{

cin>>ptr;

}

}

void setSSN()

{

cout<<"Enter Social Security Number in xxx-xx-xxxx format : ";

cin>>SSN;

}

void getWorkerName()

{

cout<<ptr;

}

string getSSN()

{

return SSN;

}

virtual void printValues()

{

cout<<"Worker Name: ";

getWorkerName();

cout<<"Social Security Number: "<<getSSN();

}

};

class PayWorker: public Worker

{

private:

int dependents;

double payRate;

public:

PayWorker(string x): Worker(x)

{

}

void setPayRate()

{

cout<<"Enter worker's Pay Rate : ";

cin>>payRate;

}

void setDependents()

{

cout<<"Enter worker's dependents : ";

cin>>dependents;

}

double getPayRate()

{

return payRate;

}

int getDependents()

{

return dependents;

}

void printValues()

{

cout<<"In base class Worker workerName is : ";

getWorkerName();

cout<<"In base class Worker SSN is : "<<getSSN()<<endl;

cout<<"In derived class PayWorker payRate is : "<<getPayRate()<<endl;

cout<<"In derived class PayWorker dependents is : "<<getDependents()<<endl;

cout<<"Worker Name: ";

getWorkerName();

cout<<"Social Security Number: "<<getSSN()<<endl;

cout<<"Pay Rate: $"<<getPayRate()<<endl;

cout<<"Number of Dependents: "<<getDependents()<<endl;

}

};

int main()

{

string a="xxx-xx-xxxx";

PayWorker obj(a);

obj.setWorkerName();

obj.setSSN();

obj.setPayRate();

obj.setDependents();

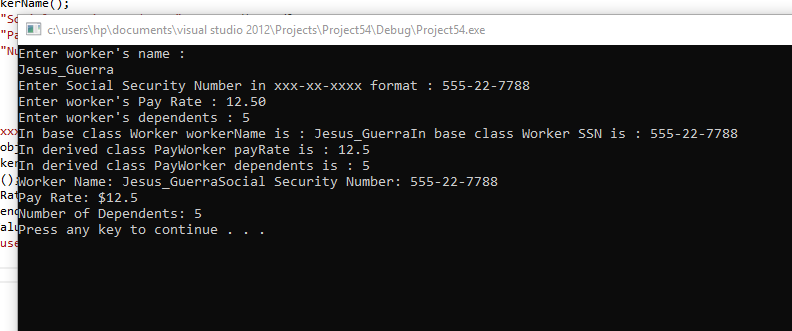
obj.printValues();

system("pause");

return 0;

}

**Output:**



-----------------------------------------------------------------------------------------