**Question 1:**

#include<iostream>

#include<string>

using namespace std;

class ship {

string name, year;

public :

ship()

{

name = "\0";

year = "\0";

}

void setname(string name) {

this->name = name;

}

string getname()

{

return name;

}

void setyear(string year) {

this->year = year;

}

string getyear()

{

return year;

}

virtual void print() {

cout << "THE NAME OF THE SHIP IS = "<<name << endl;

cout << "THE BUILT YEAR OF THIS SHIP IS = "<<year << endl;

}

};

class cruiseship :public ship

{

double passengers;

public :

void setpassengers(double passengers) {

this->passengers = passengers;

}

double getpassengers()

{

return passengers;

}

void print()

{

ship::print();

cout << "THE MAXIMUM NO OF PSSENGERS ARE = "<<passengers << endl;

}

};

class cargoship :public ship

{

int cargo\_capacity;

public:

void setcargo(int cargo\_capacity) {

this->cargo\_capacity=cargo\_capacity;

}

int getcargo()

{

return cargo\_capacity;

}

void print()

{

cout << "THE NAME OF THE SHIP IS = ";

cout << ship::getname() << endl;

cout << "THE CARGO CAPACITY = " << cargo\_capacity << endl;

}

};

int main()

{

ship arham;

arham.setname("titanic");

arham.setyear("1800");

cruiseship arham2;

arham2.setname("titantic 2");

arham2.setyear("1801");

arham2.setpassengers(1000);

cargoship arham3;

arham3.setname("titantic 3");

arham3.setyear("1802");

arham3.setcargo(500);

ship\* baseclass;

baseclass = &arham;

cout << "---------------------THE OUTPUT OF CLASS SHIP-----------------------" << endl;

baseclass->print();

baseclass = &arham2;

cout << "---------------------THE OUTPUT OF CLASS CRUSIE-----------------------" << endl;

baseclass->print();

baseclass = &arham3;

cout << "---------------------THE OUTPUT OF CLASS CARGO-----------------------" << endl;

baseclass->print();

}

**Header file :**

#pragma once

#include<iostream>

#include<string>

using namespace std;

class ship {

string name, year;

public:

ship();

void setname(string name);

string getname();

void setyear(string year);

string getyear();

virtual void print();

};

class cruiseship :public ship

{

double passengers;

public:

cruiseship();

void setpassengers(double passengers);

double getpassengers();

void print();

};

class cargoship :public ship

{

int cargo\_capacity;

public:

void setcargo(int cargo\_capacity);

int getcargo();

void print();

};

**Implementation file**

#include"Header.h"

ship::ship()

{

{

name = "\0";

year = "\0";

}

}

void ship::setname(string name)

{

this->name = name;

}

string ship::getname()

{

return name;

}

void ship::setyear(string year)

{

this->year = year;

}

string ship::getyear()

{

return year;

}

void ship::print()

{

{

cout << "THE NAME OF THE SHIP IS = " << name << endl;

cout << "THE BUILT YEAR OF THIS SHIP IS = " << year << endl;

}

}

cruiseship::cruiseship()

{

passengers = 0;

}

void cruiseship::setpassengers(double passengers)

{

this->passengers = passengers;

}

double cruiseship::getpassengers()

{

return passengers;

}

void cruiseship::print()

{

ship::print();

cout << "THE MAXIMUM NO OF PSSENGERS ARE = " << passengers << endl;

}

void cargoship::setcargo(int cargo\_capacity) {

this->cargo\_capacity = cargo\_capacity;

}

int cargoship::getcargo()

{

return cargo\_capacity;

}

void cargoship::print()

{

cout << "THE NAME OF THE SHIP IS = ";

cout << ship::getname() << endl;

cout << "THE CARGO CAPACITY = " << cargo\_capacity << endl;

}

**Source file**

#include"Header.h"

int main()

{

ship arham;

arham.setname("titanic");

arham.setyear("1800");

cruiseship arham2;

arham2.setname("titantic 2");

arham2.setyear("1801");

arham2.setpassengers(1000);

cargoship arham3;

arham3.setname("titantic 3");

arham3.setyear("1802");

arham3.setcargo(500);

ship\* baseclass;

baseclass = &arham;

cout << "---------------------THE OUTPUT OF CLASS SHIP-----------------------" << endl;

baseclass->print();

baseclass = &arham2;

cout << "---------------------THE OUTPUT OF CLASS CRUSIE-----------------------" << endl;

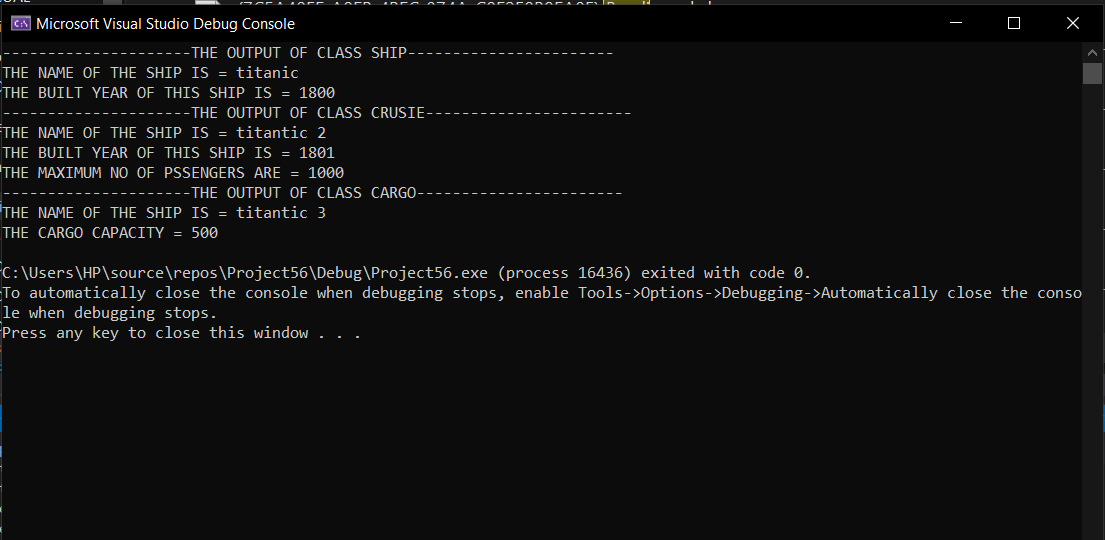
baseclass->print();

baseclass = &arham3;

cout << "---------------------THE OUTPUT OF CLASS CARGO-----------------------" << endl;

baseclass->print();

}



Question 2:

#include<iostream>

using namespace std;

class polygon

{

public :

float length;float width;

virtual float area()=0;

virtual float perimeter() = 0;

virtual void display() = 0;

};

class sqaure :public polygon {

public :

float area()

{

cout << "enter length = " << endl;

cin >> length;

return 4 \* length;

}

float perimeter() {

cout << endl;

cout << "enter length = " << endl;

cin >> length;

return 4 \* length;

}

void display() {

cout << "THE AREA IS = " << area() << endl;

cout << "THE PERIMETER IS = " << perimeter() << endl;

}

};

class reactangle :public polygon {

public:

float area()

{

cout << endl;

cout << "enter length = " << endl;

cin >> length;

cout << "enter width = " << endl;

cin >> width;

return 2\* (length\*width);

}

float perimeter() {

cout << endl;

cout << "enter length = " << endl;

cin >> length;

cout << "enter width = " << endl;

cin >> width;

return 2 \* (length + width);

}

void display() {

cout << "THE AREA IS = " << area() << endl;

cout << "THE PERIMETER IS = " << perimeter()<<endl;

}

};

class triangle :public polygon {

public:

float area()

{

cout << "enter length = " << endl;

cin >> length;

cout << "enter width = " << endl;

cin >> width;

return (length \* width)/2;

}

float perimeter() {

cout << endl;

cout << "enter length = " << endl;

cin >> length;

cout << "enter width = " << endl;

cin >> width;

return (length + width);

}

void display() {

cout << "THE AREA IS = " << area() << endl;

cout << "THE PERIMETER IS = " << perimeter()<<endl;

}

};

int main() {

polygon\* bptr;

sqaure arham;

reactangle arham1;

triangle arham2;

cout << "FOR CLASS SQUARE " << endl;

bptr = &arham;

bptr->display();

cout << "FOR CLASS REACTANGLE " << endl;

bptr = &arham1;

bptr->display();

cout << "FOR CLASS TRIANGLE " << endl;

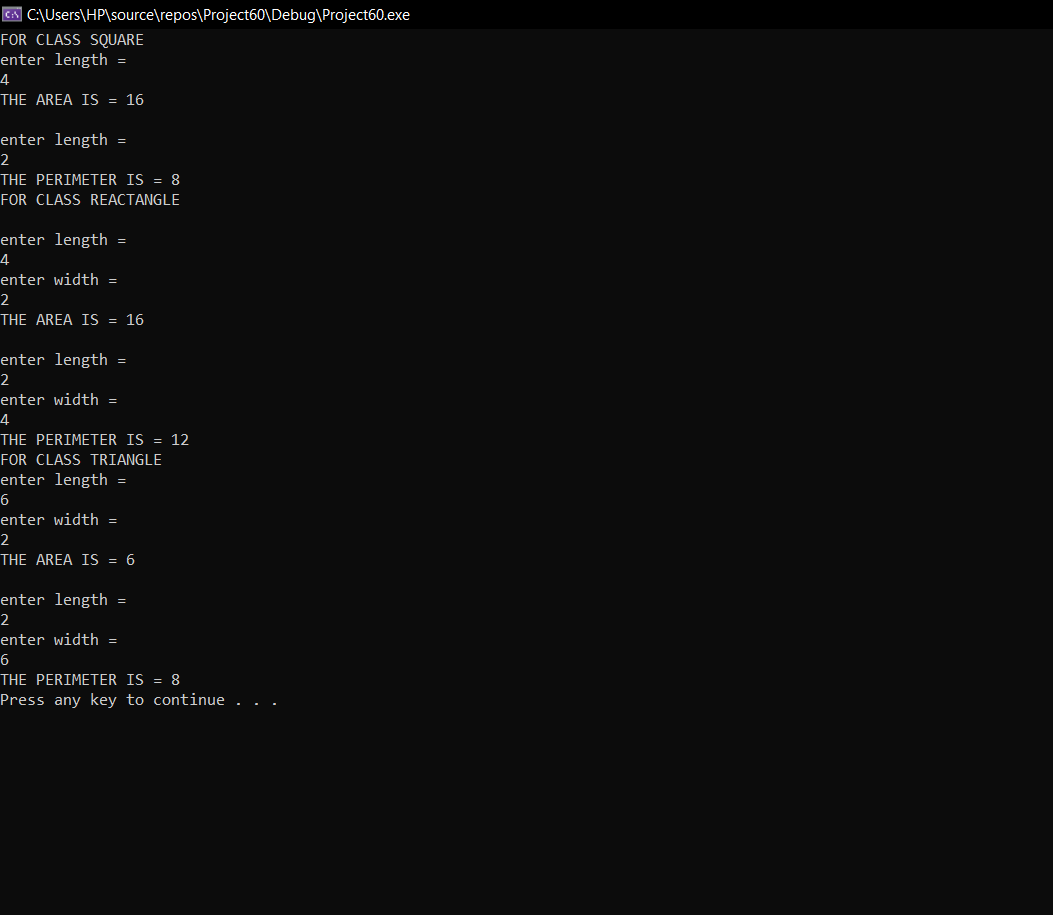
bptr = &arham2;

bptr->display();

system("pause");

return 0;

}



Question 3:

#include<iostream>

using namespace std;

class bill {

int units, per\_unit\_cost;

float calbill;

public :

virtual void monthlybill()

{

cout << "ENTER TOTAL UNITS= "<<endl;

cin >> units;

cout << "ENTER PER UNIT COST= " << endl;

cin >> per\_unit\_cost;

calbill = units \* per\_unit\_cost;

}

virtual void bill\_display()

{

cout << "THE TOTAL BILL IS = " << calbill << endl;

}

};

class billjan :public bill

{

public:

bill::monthlybill;

virtual ~billjan() {

cout << "virtual destructor called for bill jan " << endl;

}

};

class billfeb :public bill

{

public:

bill::monthlybill;

virtual ~billfeb() {

cout << "virtual destructor called for bill feb " << endl;

}

};

class billapril :public bill

{

public:

bill::monthlybill;

virtual ~billapril() {

cout << "virtual destructor called for bill april " << endl;

}

};

class billmay :public bill

{

public:

bill::monthlybill;

virtual ~billmay() {

cout << "virtual destructor called for bill may " << endl;

}

};

class billmarch :public bill

{

public:

bill::monthlybill;

virtual ~billmarch() {

cout << "virtual destructor called for bill march " << endl;

}

};

int main() {

cout << "for jan" << endl;

bill \*ok;

billjan arham;

ok = &arham;

ok->monthlybill();

cout << "for feb" << endl;

billfeb arham1;

ok = &arham1;

ok->monthlybill();

cout << "for marach = " << endl;

billmarch arham2;

ok = &arham2;

ok->monthlybill();

cout << "for april" << endl;

billapril arham3;

ok = &arham3;

ok->monthlybill();

cout << "for may"<< endl;

billmay arham4;

ok = &arham4;

ok->monthlybill();

cout << "ouput for jan " << endl;

bill \*arhamm;

arhamm = &arham;

arhamm->bill\_display();

cout << "ouput for feb " << endl;

arhamm = &arham1;

arhamm->bill\_display();

cout << "ouput for march " << endl;

arhamm = &arham2;

arhamm->bill\_display();

cout << "ouput for april " << endl;

arhamm = &arham3;

arhamm->bill\_display();

cout << "ouput for may " << endl;

arhamm = &arham4;

arhamm->bill\_display();

system("pause");

return 0;

}

