Task 5:

//header

#pragma once

//#ifndef CLASS5\_H

#include<string>

#include<iostream>

using namespace std;

#define CLASS5\_H

//#endif // !CLASS5.H

class wheels

{

int wheelsnum;

public:

//wheels() {};

void setwheels(int a);

//~wheels();

};

class engine

{

int enginenum;

public:

//engine() {};

void setengine(int w);

//~engine();

};

class window

{

int windownum;

public:

//window() {};

void setwindow(int w);

//~window();

};

class door

{

int doornum;

public:

//door() {};

void setdoor(int w);

//~door();

};

class car

{

protected:

int carnum;

engine obje;

window objwi;

wheels objw;

door objd;

public:

car(int a, wheels w, door d, window wi, engine e);

void set(int n);

int get();

~car();

};

//implementation

#include "interface5.h"

car::car(int a, wheels w, door d, window wi, engine e) : carnum(a), objw(w), objd(d), objwi(wi), obje(e)

{

cout << " successfully run ";

}

void car::set(int n)

{

carnum = n;

}

int car::get()

{

return carnum;

}

car::~car()

{

}void engine::setengine(int w)

{

enginenum = w;

}

void wheels::setwheels(int a)

{

wheelsnum = a;

}

void window::setwindow(int w)

{

windownum = w;

}

void door::setdoor(int w)

{

doornum = w;

}

#include<iostream>

#include"interface5.h"

using namespace std;

int main()

{

engine obje;

obje.setengine(1);

wheels objw;

objw.setwheels(4);

window objwi;

objwi.setwindow(8);

door objd;

objd.setdoor(6);

car objc(1, objw, objd, objwi, obje);

return 0;

}

Task 6:

//header

#pragma once

//#ifndef CLASS1\_H

#include<string>

#include<iostream>

using namespace std;

#define CLASS6\_H

//#endif // !CLASS1.H

//class person {

// string name;

//public:

// person() {}

// bool presence() { return true; }

// ~person()

// {

//

// }

//};

//class home {

//private:

// string address;

// person\* persons[3];

//public:

// home();

// person\* Getperson(int);

// bool presence(int);

// ~home()

// {

//

// }

//};

class Address {

private:

string streete;

string citya;

string statea;

string zipa;

public:

Address(string street, string city, string state,string zip) : streete(street), citya(city), statea(state), zipa(zip) {}

string getStreet() const;

string getCity() const;

string getState() const;

string getZip() const;

};

class Person {

public:

Person(string name, const Address& address) : namep(name), addressp(&address) {}

string getName() const;

const Address\* getAddress() const;

private:

string namep;

const Address\* addressp = nullptr;

};

//implementation

#include "interface6.h"

string Address:: getStreet() const

{

return streete;

}

string Address:: getCity() const

{

return citya;

}

string Address:: getState() const

{

return statea;

}

string Address:: getZip() const

{

return zipa;

}

string Person:: getName() const

{

return namep;

}

const Address\* Person:: getAddress() const

{

return addressp;

}

///driver

#include<iostream>

#include<string>

#include"interface6.h"

using namespace std;

//int main() {

// person ch1;

// home r1;

// r1.presence(1);

// return 0;

//

//}

int main() {

Address obj("street #6", "ghari shahu", "lahore", "5900");

Person fawwaz("fawwaz", obj);

Person ali("ali", obj);

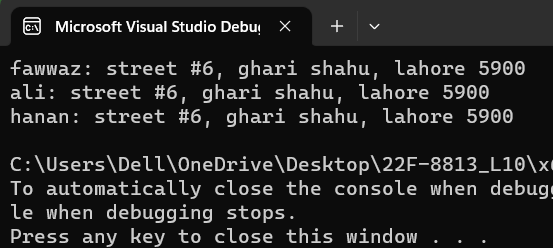
Person hanan("hanan", obj);

cout << fawwaz.getName() << ": " << fawwaz.getAddress()->getStreet() << ", "<< fawwaz.getAddress()->getCity() << ", " << fawwaz.getAddress()->getState() << " " << fawwaz.getAddress()->getZip() << endl;

cout << ali.getName() << ": " << ali.getAddress()->getStreet() << ", " << ali.getAddress()->getCity() << ", " << ali.getAddress()->getState() << " " << ali.getAddress()->getZip() << endl;

cout << hanan.getName() << ": " << hanan.getAddress()->getStreet() << ", " << hanan.getAddress()->getCity() << ", " << hanan.getAddress()->getState() << " " << hanan.getAddress()->getZip() << endl;

}



Task 7:

//header

#pragma once

//#ifndef CLASS1\_H

#include<string>

#include<iostream>

using namespace std;

#define CLASS7\_H

//#endif // !CLASS1.H

class Teacher;

class Person {

private:

string name;

const Teacher\* teacher = nullptr;

friend class Teacher;

public:

Person(string name) : name(name) {}

string getName() const;

};

class Teacher {

private:

Person\* person1 = nullptr;

public:

Teacher(Person& person) : person1(&person) {}

void setPerson(Person& person);

string getPersonName() const;

};

//implementation

#include "interface7.h"

string Person:: getName() const

{

return name;

}

void Teacher:: setPerson(Person& person)

{

person1 = &person;

}

string Teacher::getPersonName() const

{

return person1->getName();

}

///driver

#include<iostream>

#include<string>

#include"interface7.h"

using namespace std;

int main()

{

Person obj("fawwaz");

Teacher teacher(obj);

cout << "Person's name: " << teacher.getPersonName() << endl;

teacher.setPerson(obj);

}

