Lab 5 Tasks

Hamza Farhan

24K-0576

Task 1:

#include <iostream>

#include <string>

using namespace std;

class Car{

    int regNum;

    string model;

    string owner;

    public:

        Car (int regNum,string model,string owner){

            this->model=model;

            this->regNum=regNum;

            this->owner=owner;

            cout<<"registeration number is set to: "<<regNum<<endl;

            cout<<"model is set to: "<<model<<endl;

            cout<<"owner is set to: "<<owner<<endl;

        }

        Car(Car &c,string newOwn){

            regNum=c.regNum;

            model=c.model;

            owner=newOwn;

            cout<<"new owner is : "<<owner<<endl;

        }

        void displayDetails(){

            cout<<"registeration number is : "<<regNum<<endl;

            cout<<"model is : "<<model<<endl;

            cout<<"owner is : "<<owner<<endl;

        }

        ~Car(){}

};

int main (){

    Car c1(2211,"supra","hamza");

    Car c2(9090,"yaris","fuzail");

    Car c3(c1,"zaid");

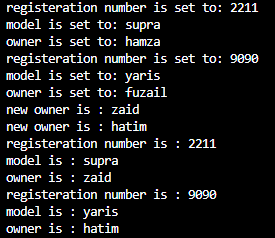
    Car c4(c2,"hatim");

    c3.displayDetails();

    c4.displayDetails();

}

Output:



Task 2:

#include <iostream>

#include <string>

using namespace std;

class Patient{

    int id;

    string name;

    float \*results;

    public:

        Patient(string name,int id,float\* results){

            this->id=id;

            this->name=name;

            this->results= new float[5];

            for (int i = 0; i < 5; i++)

            {

                this->results[i]=results[i];

            }

        }

        Patient(Patient &p){

            cout<<"copying details...\n";

            id=p.id;

            name=p.name;

            results = new float[5];

            for (int i = 0; i < 5; i++)

            {

                results[i]=p.results[i];

            }

        }

    void display(){

        cout<<"\ndisplaying results\n";

        cout<<"name "<<name<<endl;

        cout<<"ID "<<id<<endl;

        cout<<"test results:\n";

        for (int i = 0; i < 5; i++){

            cout<<results[i]<<endl;

        }

    }

    ~Patient(){

        delete[] results;

    }

};

int main (){

    float results[5]={22.2,56.9,50.0,30.3,41.7};

    Patient p1("hamza",0576,results);

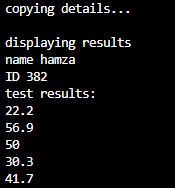
    Patient p2(p1);

    p2.display();

    return 0;

}

Output:



Task 3:

#include <iostream>

#include <string>

using namespace std;

class Professor{

    string name;

    string department;

    public:

        Professor() {

            name = "";

            department = "";

        }

        Professor(string name,string department){

            this->department=department;

            this->name=name;

        }

        void displaydetails(){

            cout<<"name: "<<name<<endl;

            cout<<"department: "<<department<<endl;

        }

};

class University{

    string name;

    Professor \*profs;

    int count;

    int capacity;

    public:

        University(string name,int cap){

            this->name=name;

            capacity=cap;

            count =0;

            profs = new Professor[capacity];

        }

        void addProf(string name,string department){

            if (count<capacity){

            profs[count] = Professor(name, department);

            count++;

            }else{

               cout<<"cannot add professor, maximum capacity reached\n";

            }

        }

        void displayDetails() {

            cout<<"university Name: " << name << endl;

            cout<<"professors:"<<endl;

            for (int i = 0; i < count; i++) {

                profs[i].displaydetails();

            }

        }

};

int main (){

    University u("FAST", 12);

    u.addProf("hamza","Computer Science");

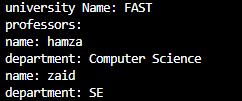
    u.addProf("zaid","SE");

    u.displayDetails();

    return 0;

}

Output:



Task 4:

#include <iostream>

#include <string>

using namespace std;

class Battery{

    int mAh;

    public:

        Battery(){}

        Battery(int mAh){

            this->mAh=mAh;

        }

        int getMah(){

            return mAh;

        }

};

class Smartphone{

    int storage,ram;

    string model;

    Battery b;

    public:

        Smartphone(string model,int ram,int storage,int mAh){

            this->model=model;

            this->ram=ram;

            this->storage=storage;

            b=Battery(mAh);

        }

    void details(){

        cout<<"smartphone details\n";

        cout<<"model: "<<model<<endl;

        cout<<"Storage: "<<storage<<endl;

        cout<<"RAM: "<<ram<<endl;

        cout<<"Battery in mAh: "<<b.getMah()<<endl;

    }

};

int main (){

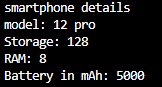
    Smartphone s("12 pro",8,128,5000);

    s.details();

    return 0;

}

Output:



Task 5:

#include <iostream>

#include <string>

using namespace std;

class TourGuide{

    string name;

    int experience;

    string specialization;

    public:

    TourGuide(){

        name="";

        specialization="";

        experience=0;

    }

    TourGuide(string name,int experience,string specialization){

        this->name=name;

        this->experience=experience;

        this->specialization=specialization;

    }

    void getinfo(){

        cout<<"name: "<<name<<endl;

        cout<<"experience: "<<experience<<" years"<<endl;

        cout<<"specialization: "<<specialization<<endl;

    }

};

class TravelAgency{

    string name;

    int count,cap;

    TourGuide \*guides;

    public:

        TravelAgency(string name,int cap){

            this->name=name;

            this->cap=cap;

            guides = new TourGuide[cap];

            count=0;

        }

        void addGuide(string name,int exp,string specialization){

            if(count<cap){

            guides[count]= TourGuide(name,exp,specialization);

            count++;

            }else{

                cout<<"maximum capacity reached. couldnt add tour guide\n";

            }

        }

        void display(){

            cout<<"tour guides available:"<<endl;

            for (int i = 0; i < count; i++)

            {

                guides[i].getinfo();

            }

        }

};

int main (){

    TravelAgency t("You Travel",4);

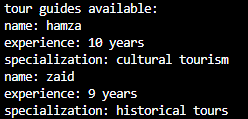
    t.addGuide("hamza",10,"cultural tourism");

    t.addGuide("zaid",9,"historical tours");

    t.display();

}

Output:



Task 6:

#include <iostream>

#include <string>

using namespace std;

class Movie{

    string director;

    string title;

    float duration;

    public:

        Movie(){}

        Movie(string title,string director,float duration){

            this->director=director;

            this->duration=duration;

            this->title=title;

        }

        void getInfo(){

            cout<<"title: "<<title<<"\ndirector: "<<director<<"\nduration: "<<duration<<" hours\n";

        }

};

class CinemaHall{

    Movie m[5];

    int count=0;

    string cinemaName;

    public:

        CinemaHall(string cinemaName){

            this->cinemaName=cinemaName;

        }

        void add(string title,string director,float duration){

            if(count<5){

                m[count]=Movie(title,director,duration);

                count++;

            }else {

                cout<<"max capacity of movies has reached\n";

            }

        }

        void display(){

            cout<<"cinema: "<<cinemaName<<endl;

            for (int i = 0; i < count; i++)

            {

                cout<<"---------------------\n";

                m[i].getInfo();

                cout<<"---------------------\n";

            }

        }

};

int main (){

    CinemaHall c1("cinema 1");

    CinemaHall c2("cinema 2");

    c1.add("interstellar","christopher nolan",2.9);

    c1.add("transformers","michael bay",2.5);

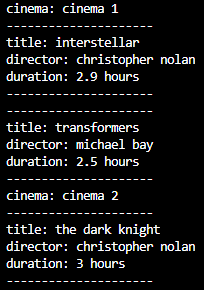
    c2.add("the dark knight","christopher nolan",3);

    c1.display();

    c2.display();

}

Output:



Task 7:

#include <iostream>

#include <string>

using namespace std;

class Product{

    int id;

    float price;

    string name;

    public:

        Product(){}

        Product(string name,int id,float price){

            this->price=price;

            this->id=id;

            this->name=name;

        }

        string getname(){

            return name;

        }

        float getprice(){

            return price;

        }

        int getid(){

            return id;

        }

        void getInfo(){

            cout<<"Product: "<<name<<"\nID: "<<id<<"\nPrice: $"<<price;

        }

};

class Inventory{

    Product p[100];

    int count;

    string name;

    public:

        Inventory(string name){

            count=0;

            this->name=name;

        }

        void add(string name,int id, float price){

            p[count]=Product(name,id,price);

            count++;

        }

        void sort(){

            for (int i = 0; i < count-1; i++)

            {

                for (int j = 0; j < count-1-i; j++)

                {

                    if (p[j].getprice() > p[j + 1].getprice()) {

                        Product temp = p[j];

                        p[j] = p[j + 1];

                        p[j + 1] = temp;

                    }

                }

            }

        }

        void search(string searchName){

            for (int i = 0; i < count; i++)

            {

                if(p[i].getname()==searchName){

                    cout<<"product found:\n";

                    p[i].getInfo();

                    break;

                }

            }

        }

        void display(){

            for (int i = 0; i < count; i++) {

                p[i].getInfo();

                cout << "\n--------------------\n";

            }

        }

};

int main(){

    Inventory i("inventory 1");

    i.add("iPhone 12 Pro",2211,999.9);

    i.add("MacBook",1122,1999.9);

    i.add("Lenovo ThinkBook",1212,500);

    i.add("Nike Mercurial",0111,249.9);

    i.add("Apple Watch Series 10",2112,399.9);

    cout<<"\ninventory\n";

    i.display();

    cout<<"\nsorted inventory\n";

    i.sort();

    i.display();

    i.search("MacBook");

}

Output:

