Lab Tasks 05

Name: Hamza Atif

Roll No:24K-0594

Task 01

#include <iostream>

#include <string>

using namespace std;

class Car{

int registrationNumber;

string\* model;

string\* ownerName;

public:

Car(int regNum, const string& Model, const string& ON){

registrationNumber = regNum;

model = new string(Model);

ownerName = new string(ON);

}

Car(const Car& car, const string& newOwner){

registrationNumber = car.registrationNumber;

model = new string(\*car.model);

ownerName = new string(newOwner);

}

~Car(){

delete model;

delete ownerName;

}

void display(){

cout << "Registration Number: " << registrationNumber << endl;

cout << "Model: " << \*model << endl;

cout << "Owner Name: " << \*ownerName << endl;

}

};

int main(){

Car car1(1012, "Lancer", "Ali");

Car car2(1013, "Honda", "Ahmed");

cout << "Original Cars:\n";

car1.display();

cout << endl;

car2.display();

cout << endl;

cout << "\nAfter New Owner Rents the Cars: " << endl;

Car car1NewOwner(car1, "Kashif");

car1NewOwner.display();

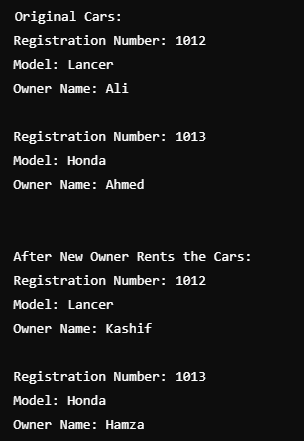
cout << endl;

Car car2NewOwner(car2, "Hamza");

car2NewOwner.display();

return 0;

}



Task 02

#include <iostream>

#include <string>

using namespace std;

class Patient{

string name;

int\* testResults;

int Count;

int id;

public:

Patient(int patientID, string& patientName, int\* results, int count){

id = patientID;

name = patientName;

Count = count;

testResults = new int[Count];

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

testResults[i] = results[i];

}

}

Patient(Patient& p){

id = p.id;

name = p.name;

Count = p.Count;

testResults = new int[Count];

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

testResults[i] = p.testResults[i];

}

}

~Patient(){

delete[] testResults;

}

void display(){

cout << "Patient ID: " << id << endl;

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

cout << "Test Results: ";

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

cout << testResults[i] << " ";

}

cout << endl;

}

};

int main(){

int results1[] = {85, 90, 78};

Patient p1(101, "Hamza Atif", results1, 3);

cout << "Original Patient: " << endl;

p1.display();

cout << endl;

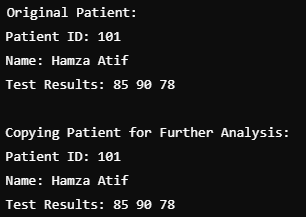
cout << "\nCopying Patient for Frther Analysis: " << endl;

Patient p2(p1);

p2.display();

return 0;

}



Task 03

#include <iostream>

#include <string>

#include<cstring>

using namespace std;

class Professor {

string department;

string name;

public:

Professor(const string& name, const string& department) : department(department), name(name){}

void display(){

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

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

}

};

class University{

string name;

int profCount;

Professor professors[10];

public:

University(const string& name) : name(name), profCount(0) {}

void addProfessor(const Professor& prof) {

if (profCount < 10) {

professors[profCount++] = prof;

} else {

cout << " limit exeeded for professors " << endl;

}

}

void displayDetails(){

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

cout << "Professors Working Here:" << endl;

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

professors[i].display();

}

}

};

int main(){

University uni("FAST University");

Professor p1("Ali", "Computer Science");

Professor p2("Hamza", "Mathematics");

Professor p3("Hammad", "Physics");

uni.addProfessor(p1);

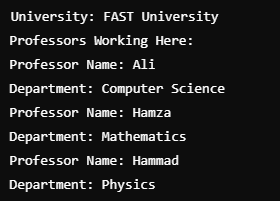
uni.addProfessor(p2);

uni.addProfessor(p3);

uni.displayDetails();

return 0;

}



Task 04

#include <iostream>

#include <string>

#include <cstring>

using namespace std;

class Battery {

int capacity;

string type;

public:

Battery(int capacity, const string& type) :capacity(capacity),type(type){}

void display() const {

cout << "Battery Capacity: " << capacity << "mAh" << endl;

cout << "Battery Type: " << type << endl;

}

};

class Smartphone {

string model;

Battery battery;

public:

Smartphone(const string& model, int batteryCapacity, const string& batteryType)

:battery(batteryCapacity, batteryType),model(model){}

void display(){

cout << "Smartphone Model: " << model << endl;

battery.display();

}

};

int main() {

Smartphone phone1("Samsung S21", 2500, "A");

Smartphone phone2("iPhone 13", 2200, "B");

cout << "Smartpone Details: " << endl;

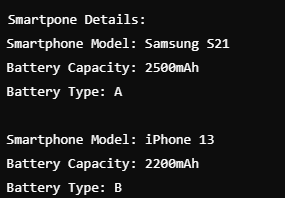
phone1.display();

cout << endl;

phone2.display();

return 0;

}



Task 05

#include <iostream>

#include <string>

#include <cstring>

using namespace std;

class TourGuide{

string specialization;

string name;

int yearsOfExperience;

public:

TourGuide(const string& name, int experience, const string& spec)

: name(name),yearsOfExperience(experience),specialization(spec){}

void display(){

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

count << " Exprience: " << yearsOfExperience << endl;

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

}

};

class TravelAgency{

string agencyName;

TourGuide tourGuides[10];

int guideCount;

public:

TravelAgency(const string& name):agencyName(name), guideCount(0){}

void addTourGuide(const TourGuide& guide){

if (guideCount < 10) {

tourGuides[guideCount++] = guide;

} else {

cout << "limit reached for tour guides " << endl;

}

}

void displayDetails(){

cout << "Travel Agency: " << agencyName << endl;

cout << "Tour Guides: " << endl;

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

tourGuides[i].display();

}

}

};

int main(){

TravelAgency agency("ABCDE Adventures");

TourGuide g1("Hamza", 5, "Historical Sites");

TourGuide g2("Atif", 8, "Mountain Trekking");

TourGuide g3("Hammad", 3, "City Tours");

agency.addTourGuide(g1);

agency.addTourGuide(g2);

agency.addTourGuide(g3);

agency.displayDetails();

return 0;

}



Task 06

#include<iostream>

#include<string>

#include<cstring>

using namespace std;

class Movie{

string title;

string director;

int duration;

public:

Movie(const string& title, const string& director, int duration)

: title(title),director(director),duration(duration){}

void display(){

cout << "Title: " << title << endl;

cout << "Director: " << director << endl;

cout<< "Duration: " << duration << " minutes" << endl;

}

};

class CinemaHall

{

string hallName;

Movie movies[10];

int movieCount;

public:

CinemaHall(const string& name):hallName(name),movieCount(0){}

void addMovie(const Movie& movie)

{

if (movieCount < 10)

{

movies[movieCount++] = movie;

}

}

void displayDetails(){

cout << "Cinema Hall: " << hallName << endl;

cout << "Currently Screening Movies: " << endl;

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

{

movies[i].display();

}

}

};

int main(){

CinemaHall cinema("ABCDE Theater");

Movie m1("Inception", "Christopher Nolan", 148);

Movie m2("Dunkirk", "Christopher Nolan", 149);

Movie m3("Interstellar", "Christopher Nolan", 169);

cinema.addMovie(m1);

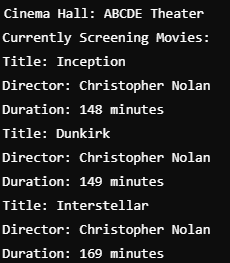
cinema.addMovie(m2);

cinema.addMovie(m3);

cinema.displayDetails();

return 0;

}



Task 07

include <iostream>

#include<cstring>

#include <string>

using namespace std;

class Product{

int id;

string name;

float price;

public:

Product(int id, const string& name, float price)

: id(id), name(name), price(price){}

int getId() const { return id; }

string getName() const { return name; }

float getPrice() const { return price; }

void display() const {

cout << "ID: " << id << ", Name: " << name << ", Price: " << price << endl;

}

};

class Store{

Product products[50];

int productCount;

public:

Store() : productCount(0){}

void addProduct(const Product& product){

if (productCount < 50) {

products[productCount++] = product;

}

}

void sortProductsByPrice(){

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

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

if (products[j].getPrice() > products[j + 1].getPrice()){

swap(products[j], products[j + 1]);

}

}

}

}

void searchByName(const string& name){

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

if (products[i].getName() == name){

products[i].display();

return;

}

}

cout << "Product not found." << endl;

}

void displayAllProducts() {

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

products[i].display();

}

}

};

int main() {

Store store;

store.addProduct(Product(101, "Shirt", 25.99));

store.addProduct(Product(102, "Pants", 40.50));

store.addProduct(Product(103, "Shoes", 60.00));

store.addProduct(Product(104, "Hat", 15.00));

cout << "Products sorted by price:" << endl;

store.sortProductsByPrice();

store.displayAllProducts();

cout << "\nSearch for product:" << endl;

store.searchByName("Shoes");

return 0;

}

