**Lab 05 Tasks**

**Fasiha Adnan**

**24K-0901**

**01:**

#include <iostream>

using namespace std;

class Apartment {

private:

int id;

const char\* address;

const char\* owner;

public:

Apartment(int id, const char\* address, const char\* owner) : id(id), address(address), owner(owner) {}

Apartment(const Apartment& other) : id(other.id), address(other.address), owner(other.owner) {}

void display() const {

cout << "-----------------------------" << endl;

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

cout << "Address: " << address << endl;

cout << "Owner: " << owner << endl;

}

};

int main() {

Apartment a1(201, "A-02 Street 7", "Ali");

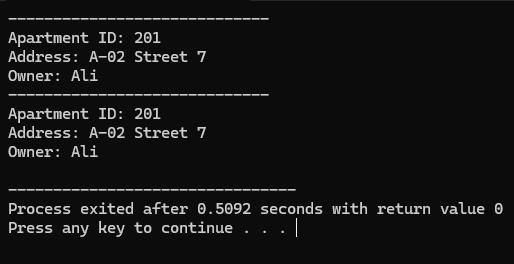
a1.display();

Apartment a2 = a1;

a2.display();

return 0;

}

****

**02:**

#include <iostream>

using namespace std;

class Student {

private:

int id;

const char\* name;

int\* examScores;

int numScores;

public:

Student(int id, const char\* name, int\* examScores, int numScores) : id(id), name(name), numScores(numScores) {

this->examScores = new int[numScores];

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

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

}

}

Student(const Student& other) : id(other.id), name(other.name), numScores(other.numScores) {

examScores = new int[numScores];

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

examScores[i] = other.examScores[i];

}

}

void display() const {

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

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

cout << "Scores: ";

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

cout << examScores[i] << " ";

}

cout << endl;

cout << "-------------------------" << endl;

}

~Student() {

delete[] examScores;

}

};

int main() {

int examScores1[] = {85, 90, 78};

int examScores2[] = {88, 76, 95};

Student s1(201, "Ali", examScores1, 1);

Student s2(202, "Sara", examScores2, 2);

s1.display();

s2.display();

Student s3 = s1;

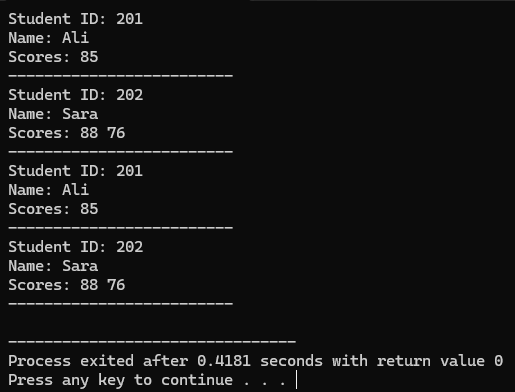
Student s4 = s2;

s3.display();

s4.display();

return 0;

}



**03:**

#include <iostream>

using namespace std;

class Employee {

public:

string name;

string designation;

Employee(string name, string designation) : name(name), designation(designation) {}

};

class Project {

public:

string title;

string deadline;

Employee\* employees[10];

int employeeCount;

Project(string title, string deadline) : title(title), deadline(deadline), employeeCount(0) {}

void addEmployee(Employee\* employee) {

if (employeeCount < 10) {

employees[employeeCount] = employee;

employeeCount++;

}

}

void display() {

cout << "Project: " << title << endl;

cout << "Deadline: " << deadline << endl;

cout << "Assigned Employees: " << endl;

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

cout << " - " << employees[i]->name << " (" << employees[i]->designation << ")" << endl;

}

cout << "-------------------------" << endl;

}

};

int main() {

// Creating employees

Employee emp1("Ali", "Software Engineer");

Employee emp2("Sara", "Project Manager");

Employee emp3("Alice", "Project Lead");

Project proj1("Content Analysis", "2025-04-02");

Project proj2("AI Research", "2025-08-04");

proj1.addEmployee(&emp1);

proj1.addEmployee(&emp2);

proj2.addEmployee(&emp2);

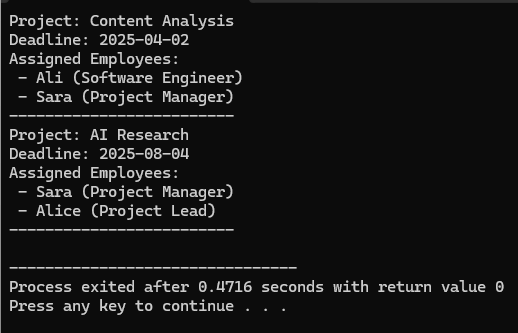
proj2.addEmployee(&emp3);

proj1.display();

proj2.display();

return 0;

}



**04:**

#include <iostream>

using namespace std;

class AlarmSystem {

public:

string securityLevel;

AlarmSystem(string level) : securityLevel(level) {}

void display() {

cout << "Alarm Security Level: " << securityLevel << endl;

}

};

class SmartHome {

private:

AlarmSystem alarm;

string homeName;

public:

SmartHome(string name, string securityLevel) : homeName(name), alarm(securityLevel) {}

void display() {

cout << "Smart Home: " << homeName << endl;

alarm.display();

cout << "-------------------------" << endl;

}

};

int main() {

SmartHome home1("Luxury Villa", "High");

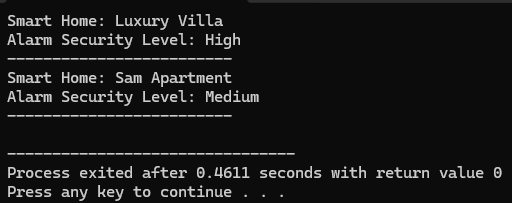
SmartHome home2("Sam Apartment", "Medium");

home1.display();

home2.display();

return 0;

}

****

**05:**

#include <iostream>

using namespace std;

class Doctor {

public:

string name;

string specialization;

int experience;

Doctor(string name, string specialization, int experience)

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

void display() {

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

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

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

}

};

class Hospital {

private:

string hospitalName;

Doctor\* doctors[10];

int doctorCount;

public:

Hospital(string name) : hospitalName(name), doctorCount(0) {}

void addDoctor(Doctor\* doctor) {

if (doctorCount < 10) {

doctors[doctorCount] = doctor;

doctorCount++;

}

}

void display() {

cout << "Hospital: " << hospitalName << endl;

cout << "\nDoctors currently working:" << "\n" << endl;

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

doctors[i]->display();

cout << "----------------------------------" << endl;

}

}

};

int main() {

Doctor doc1("Dr. Zara", "Cardiologist", 10);

Doctor doc2("Dr. Alisha", "Neurologist", 2);

Doctor doc3("Dr. Samad", "Orthopedic", 8);

Hospital h1("South City");

h1.addDoctor(&doc1);

h1.addDoctor(&doc2);

h1.addDoctor(&doc3);

h1.display();

return 0;

}

****

**06:**

#include <iostream>

using namespace std;

class Level {

public:

string levelName;

int difficulty;

Level(string name, int difficulty) : levelName(name), difficulty(difficulty) {}

void display() {

cout << "Level: " << levelName << ", Difficulty: " << difficulty << endl;

}

};

class VideoGame {

private:

string title;

string genre;

Level\* levels[10];

int levelCount;

public:

VideoGame(string title, string genre) : title(title), genre(genre), levelCount(0) {}

~VideoGame() {

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

delete levels[i];

}

}

void addLevel(string name, int difficulty) {

if (levelCount < 10) {

levels[levelCount] = new Level(name, difficulty);

levelCount++;

}

}

void display() {

cout << "Game: " << title << endl;

cout << "Genre: " << genre << endl;

cout << "Levels: " << endl;

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

levels[i]->display();

}

cout << "-------------------------" << endl;

}

};

int main() {

VideoGame game("Temple Run", "Adventure");

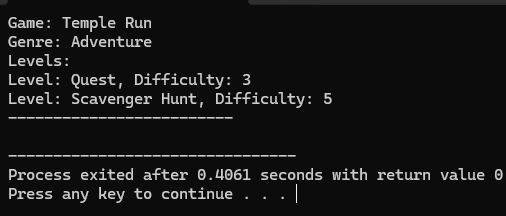
game.addLevel("Quest", 3);

game.addLevel("Scavenger Hunt", 5);

game.display();

return 0;

}



**07:**

#include <iostream>

using namespace std;

class Product {

public:

int id;

string name;

int quantity;

Product(int id, string name, int quantity) : id(id), name(name), quantity(quantity) {}

void display() {

cout << "Product ID: " << id << ", Name: " << name << ", Quantity: " << quantity << endl;

}

};

class Warehouse {

private:

Product\* inventory[10];

int productCount;

public:

Warehouse() : productCount(0) {}

~Warehouse() {

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

delete inventory[i];

}

}

void addProduct(int id, string name, int quantity) {

if (productCount < 10) {

inventory[productCount] = new Product(id, name, quantity);

productCount++;

}

}

void sortProducts() {

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

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

if (inventory[i]->name > inventory[j]->name) {

Product\* temp = inventory[i];

inventory[i] = inventory[j];

inventory[j] = temp;

}

}

}

}

void searchProduct(int id) {

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

if (inventory[i]->id == id) {

inventory[i]->display();

return;

}

}

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

}

void display() {

cout << "Warehouse Inventory:" << endl;

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

inventory[i]->display();

}

cout << "-------------------------" << endl;

}

};

int main() {

Warehouse warehouse;

warehouse.addProduct(101, "Rice", 5);

warehouse.addProduct(102, "Sugar", 10);

warehouse.addProduct(103, "Apple", 50);

cout << "Before Sorting:" << endl;

warehouse.display();

warehouse.sortProducts();

cout << "After Sorting:" << endl;

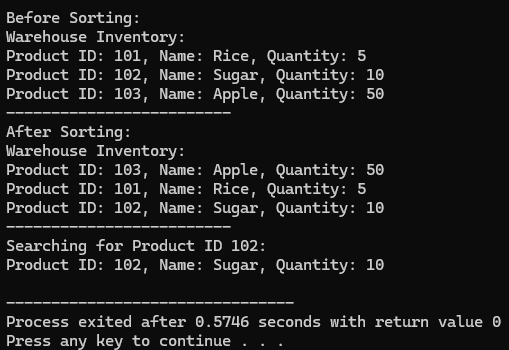
warehouse.display();

cout << "Searching for Product ID 102:" << endl;

warehouse.searchProduct(102);

return 0;

}

****