#include <iostream>

#include <string>

using namespace std;

// Definition of the Employee class

class Employee {

private:

// Private member variables to store employee details

int id; // Employee ID

string name; // Employee name

string position; // Employee position/job title

double salary; // Employee salary

public:

// Default constructor to initialize employee details

Employee() : id(0), name(""), position(""), salary(0.0) {}

// Method to set employee details

void setEmployeeDetails(int empId, const string &empName, const string &empPosition, double empSalary) {

id = empId; // Assign ID

name = empName; // Assign name

position = empPosition; // Assign position

salary = empSalary; // Assign salary

}

// Method to display employee details

void displayEmployeeDetails() const {

cout << "ID: " << id << ", Name: " << name

<< ", Position: " << position << ", Salary: $" << salary << endl;

}

};

// Main function to execute the program

int main() {

const int NUM\_EMPLOYEES = 10; // Constant for the number of employees

Employee employees[NUM\_EMPLOYEES]; // Array of Employee objects

cout << "Enter details for " << NUM\_EMPLOYEES << " employees:" << endl;

// Loop to input details for all employees

for (int i = 0; i < NUM\_EMPLOYEES; i++) {

int id; // Variable to store employee ID

string name; // Variable to store employee name

string position; // Variable to store employee position

double salary; // Variable to store employee salary

cout << "\nEmployee " << (i + 1) << " details:" << endl;

cout << "Enter ID: ";

cin >> id; // Read employee ID

cin.ignore(); // Ignore leftover newline character from previous input

cout << "Enter Name: ";

getline(cin, name); // Read employee name (allows spaces)

cout << "Enter Position: ";

getline(cin, position); // Read employee position (allows spaces)

cout << "Enter Salary: ";

cin >> salary; // Read employee salary

// Store the details in the current Employee object

employees[i].setEmployeeDetails(id, name, position, salary);

}

// Display the details of all employees

cout << "\nEmployee Details:" << endl;

for (int i = 0; i < NUM\_EMPLOYEES; i++) {

cout << "Employee " << (i + 1) << ": ";

employees[i].displayEmployeeDetails(); // Call the display method for each employee

}

return 0; // Return 0 to indicate successful program execution

}