Program 1:

#include <iostream>

#include <string>

#include <vector>

using namespace std;

class Account {

protected:

int accountNumber;

double balance;

public:

Account(int number, double bal) : accountNumber(number), balance(bal) {}

virtual void deposit(double amount) {

balance += amount;

cout << "Deposited " << amount << " successfully." << endl;

}

virtual void withdraw(double amount) {

if (balance >= amount) {

balance -= amount;

cout << "Withdrawn " << amount << " successfully." << endl;

} else {

cout << "Insufficient balance." << endl;

}

}

virtual void display() {

cout << "Account Number: " << accountNumber << endl;

cout << "Balance: " << balance << endl;

}

};

class Customer {

protected:

string name;

int age;

public:

Customer(string n, int a) : name(n), age(a) {}

void display() {

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

cout << "Age: " << age << endl;

}

};

class BankAccount : public Account, public Customer {

public:

BankAccount(string name, int age, int number, double bal) : Account(number, bal), Customer(name, age) {}

void display() override {

cout << "Account details:" << endl;

Account::display();

cout << "Customer details:" << endl;

Customer::display();

}

};

int main() {

BankAccount bankAccount("John Doe", 30, 123456, 5000);

bankAccount.display();

bankAccount.deposit(2000);

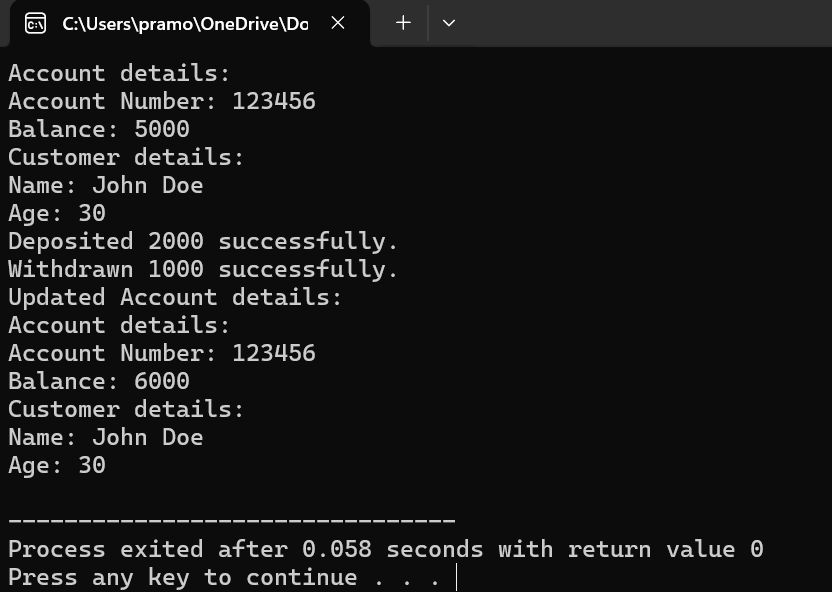
bankAccount.withdraw(1000);

cout << "Updated Account details:" << endl;

bankAccount.display();

return 0;

}



Program 2:

#include <iostream>

#include <string>

using namespace std;

class Student {

protected:

string name;

int rollNumber;

public:

Student(string n, int r) : name(n), rollNumber(r) {}

void display() {

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

cout << "Roll Number: " << rollNumber << endl;

}

};

class StudentReport : public Student {

private:

float marks;

public:

StudentReport(string n, int r, float m) : Student(n, r), marks(m) {}

void displayReport() {

display();

cout << "Marks: " << marks << endl;

}

};

int main() {

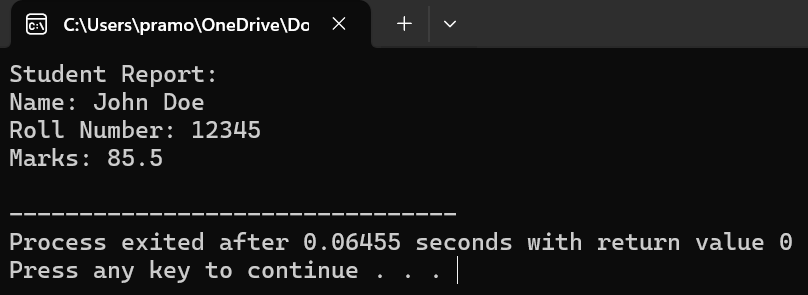
StudentReport student1("John Doe", 12345, 85.5);

cout << "Student Report:" << endl;

student1.displayReport();

return 0;

}



Program 3:

#include <iostream>

#include <string>

using namespace std;

class Employee {

protected:

string name;

int employeeID;

public:

Employee(string n, int id) : name(n), employeeID(id) {}

void display() {

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

cout << "Employee ID: " << employeeID << endl;

}

};

class RegularEmployee : public Employee {

protected:

float basicSalary;

public:

RegularEmployee(string n, int id, float salary) : Employee(n, id), basicSalary(salary) {}

float calculateSalary() {

return basicSalary;

}

};

class ContractualEmployee : public Employee {

protected:

float hourlyRate;

int hoursWorked;

public:

ContractualEmployee(string n, int id, float rate, int hours) : Employee(n, id), hourlyRate(rate), hoursWorked(hours) {}

float calculateSalary() {

return hourlyRate \* hoursWorked;

}

};

class Manager : public RegularEmployee, public ContractualEmployee {

public:

Manager(string n, int id, float salary, float rate, int hours) : RegularEmployee(n, id, salary), ContractualEmployee(n, id, rate, hours) {}

float calculateSalary() {

return RegularEmployee::calculateSalary() + ContractualEmployee::calculateSalary();

}

};

int main() {

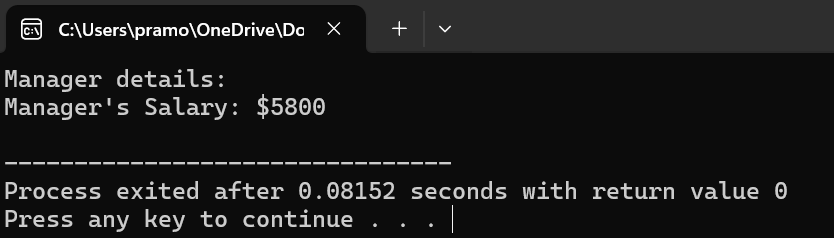
Manager manager("John Doe", 1001, 5000, 20, 40);

cout << "Manager details:" << endl;

cout << "Manager's Salary: $" << manager.calculateSalary() << endl;

return 0;

}



Program 4:

#include <iostream>

void swap(int\* a, int\* b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

void sortAscending(int\* arr, int size) {

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

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

if (\*(arr + j) > \*(arr + j + 1)) {

swap(arr + j, arr + j + 1);

}

}

}

}

void display(int\* arr, int size) {

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

std::cout << \*(arr + i) << " ";

}

std::cout << std::endl;

}

int main() {

const int size = 5;

int arr[size];

std::cout << "Enter " << size << " elements: ";

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

std::cin >> \*(arr + i);

}

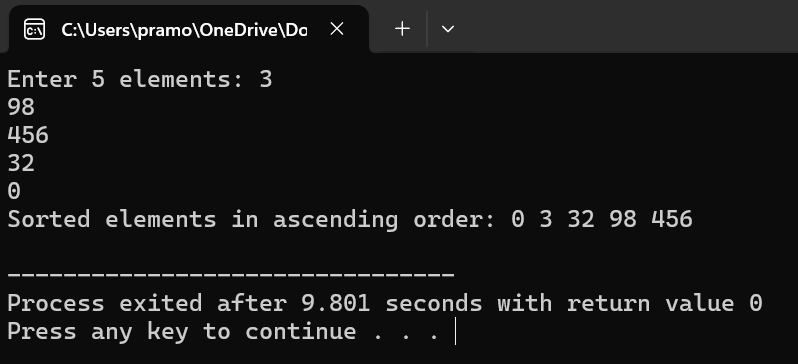
sortAscending(arr, size);

std::cout << "Sorted elements in ascending order: ";

display(arr, size);

return 0;

}



Program 5:

#include <iostream>

#include <string>

using namespace std;

class Patient {

protected:

string name;

int age;

public:

Patient(string n, int a) : name(n), age(a) {}

virtual void enterDetails() {

cout << "Enter patient name: ";

getline(cin, name);

cout << "Enter patient age: ";

cin >> age;

cin.ignore();

}

virtual void displayDetails() {

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

cout << "Age: " << age << endl;

}

};

class Inpatient : public Patient {

private:

int roomNumber;

public:

Inpatient(string n, int a, int room) : Patient(n, a), roomNumber(room) {}

void enterDetails() override {

Patient::enterDetails();

cout << "Enter room number: ";

cin >> roomNumber;

cin.ignore();

}

void displayDetails() override {

cout << "Inpatient details:" << endl;

Patient::displayDetails();

cout << "Room Number: " << roomNumber << endl;

}

};

int main() {

Inpatient patient("John Doe", 35, 101);

patient.enterDetails();

cout << "\nPatient details:" << endl;

patient.displayDetails();

return 0;

}

