OOP LAB 06

24k-0753

QUESTION 01

#include <iostream>

using namespace std;

class Employee

{

protected:

string name;

float salary;

public:

Employee(string n, float s)

{

name = n;

salary = s;

}

virtual void displayDetails()

{

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

cout << "Salary: " << salary << endl;

}

};

class Manager : public Employee

{

float bonus;

public:

Manager(string n, float s, float b) : Employee(n, s)

{

bonus = b;

}

void displayDetails() override

{

Employee::displayDetails();

cout << "Bonus: " << bonus << endl;

}

};

int main()

{

string name;

float salary, bonus;

cout << "Enter Manager's Name: ";

getline(cin, name);

cout << "Enter Salary: ";

cin >> salary;

cout << "Enter Bonus: ";

cin >> bonus;

Manager mgr(name, salary, bonus);

cout << "\nManager Details:\n";

mgr.displayDetails();

return 0;

}

QUESTION 02

#include <iostream>

using namespace std;

class Vehicle

{

protected:

string brand;

int speed;

public:

Vehicle(string b, int s)

{

brand = b;

speed = s;

}

virtual void displayDetails()

{

cout << "Brand: " << brand << endl;

cout << "Speed: " << speed << " km/h" << endl;

}

};

class Car : public Vehicle

{

protected:

int seats;

public:

Car(string b, int s, int seatCount) : Vehicle(b, s)

{

seats = seatCount;

}

void displayDetails() override

{

Vehicle::displayDetails();

cout << "Seats: " << seats << endl;

}

};

class ElectricCar : public Car

{

int batteryLife;

public:

ElectricCar(string b, int s, int seatCount, int battery) : Car(b, s, seatCount)

{

batteryLife = battery;

}

void displayDetails() override

{

Car::displayDetails();

cout << "Battery Life: " << batteryLife << " hours" << endl;

}

};

int main()

{

string brand;

int speed, seats, batteryLife;

cout << "Enter brand: ";

cin >> brand;

cout << "Enter speed (km/h): ";

cin >> speed;

cout << "Enter number of seats: ";

cin >> seats;

cout << "Enter battery life (hours): ";

cin >> batteryLife;

ElectricCar eCar(brand, speed, seats, batteryLife);

cout << "\nElectric Car Details:\n";

eCar.displayDetails();

return 0;

}

QUESTION 03

#include <iostream>

using namespace std;

class Person {

protected:

string name;

int age;

public:

Person(string n, int a) {

name = n;

age = a;

}

virtual void displayDetails() {

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

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

}

};

class Teacher : public Person {

protected:

string subject;

public:

Teacher(string n, int a, string sub) : Person(n, a) {

subject = sub;

}

void displayDetails() override {

Person::displayDetails();

cout << "Subject: " << subject << endl;

}

};

class Researcher : public Teacher {

protected:

string researchArea;

public:

Researcher(string n, int a, string sub, string research) : Teacher(n, a, sub) {

researchArea = research;

}

void displayDetails() override {

Teacher::displayDetails();

cout << "Research Area: " << researchArea << endl;

}

};

class Professor : public Researcher {

int publications;

public:

Professor(string n, int a, string sub, string research, int pub) : Researcher(n, a, sub, research) {

publications = pub;

}

void displayDetails() override {

Researcher::displayDetails();

cout << "Publications: " << publications << endl;

}

};

int main() {

string name, subject, researchArea;

int age, publications;

cout << "Enter name: ";

cin >> name;

cout << "Enter age: ";

cin >> age;

cout << "Enter subject: ";

cin >> subject;

cout << "Enter research area: ";

cin >> researchArea;

cout << "Enter number of publications: ";

cin >> publications;

Professor prof(name, age, subject, researchArea, publications);

cout << "\nProfessor Details:\n";

prof.displayDetails();

return 0;

}

QUESTION 04

#include <iostream>

using namespace std;

class Account

{

protected:

int accountNumber;

float balance;

public:

Account(int accNum, float bal)

{

accountNumber = accNum;

balance = bal;

}

virtual void displayDetails()

{

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

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

}

};

class SavingsAccount : public Account

{

protected:

float interestRate;

public:

SavingsAccount(int accNum, float bal, float rate) : Account(accNum, bal)

{

interestRate = rate;

}

void displayDetails() override

{

Account::displayDetails();

cout << "Interest Rate: " << interestRate << "%" << endl;

}

};

class CheckingAccount : public Account

{

protected:

float overdraftLimit;

public:

CheckingAccount(int accNum, float bal, float limit) : Account(accNum, bal)

{

overdraftLimit = limit;

}

void displayDetails() override

{

Account::displayDetails();

cout << "Overdraft Limit: $" << overdraftLimit << endl;

}

};

int main()

{

int accNum;

float balance, interestRate, overdraftLimit;

cout << "Enter Savings Account details:\n";

cout << "Account Number: ";

cin >> accNum;

cout << "Balance: ";

cin >> balance;

cout << "Interest Rate (%): ";

cin >> interestRate;

SavingsAccount savings(accNum, balance, interestRate);

cout << "\nEnter Checking Account details:\n";

cout << "Account Number: ";

cin >> accNum;

cout << "Balance: ";

cin >> balance;

cout << "Overdraft Limit: ";

cin >> overdraftLimit;

CheckingAccount checking(accNum, balance, overdraftLimit);

cout << "\nSavings Account Details:\n";

savings.displayDetails();

cout << "\nChecking Account Details:\n";

checking.displayDetails();

return 0;

}

QUESTION 05

#include <iostream>

using namespace std;

class Device

{

protected:

int deviceID;

bool status;

public:

Device(int id, bool s) : deviceID(id), status(s) {}

virtual void displayDetails()

{

cout << "Device ID: " << deviceID << endl;

cout << "Status: " << (status ? "On" : "Off") << endl;

}

};

class SmartPhone : virtual public Device

{

protected:

float screenSize;

public:

SmartPhone(int id, bool s, float size) : Device(id, s), screenSize(size) {}

void displayDetails() override

{

Device::displayDetails();

cout << "Screen Size: " << screenSize << " inches" << endl;

}

};

class SmartWatch : virtual public Device

{

protected:

bool heartRateMonitor;

public:

SmartWatch(int id, bool s, bool hr) : Device(id, s), heartRateMonitor(hr) {}

void displayDetails() override

{

Device::displayDetails();

cout << "Heart Rate Monitor: " << (heartRateMonitor ? "Yes" : "No") << endl;

}

};

class SmartWearable : public SmartPhone, public SmartWatch

{

private:

int stepCounter;

public:

SmartWearable(int id, bool s, float size, bool hr, int steps)

: Device(id, s), SmartPhone(id, s, size), SmartWatch(id, s, hr), stepCounter(steps) {}

void displayDetails() override

{

Device::displayDetails();

cout << "Screen Size: " << screenSize << " inches" << endl;

cout << "Heart Rate Monitor: " << (heartRateMonitor ? "Yes" : "No") << endl;

cout << "Step Counter: " << stepCounter << " steps" << endl;

}

};

int main()

{

int id, steps;

bool status, hrMonitor;

float screenSize;

cout << "Enter Device ID: ";

cin >> id;

cout << "Enter Status (1 for On, 0 for Off): ";

cin >> status;

cout << "Enter Screen Size: ";

cin >> screenSize;

cout << "Does it have a Heart Rate Monitor? (1 for Yes, 0 for No): ";

cin >> hrMonitor;

cout << "Enter Step Counter: ";

cin >> steps;

SmartWearable wearable(id, status, screenSize, hrMonitor, steps);

cout << "\nSmart Wearable Details:\n";

wearable.displayDetails();

return 0;

}