Lab 06 Tasks

Fasiha Adnan
24K-0901

01:

```
#include<iostream>
using namespace std;

class Employee {
	protected:
		string name;
		float salary;

	public:
		Employee (string name, float salary) : name(name), salary(salary){}

	void display(){
		cout << "Employee name: " << name << endl;
		cout << "Salary: " << salary << endl;
	}
};

class Manager : public Employee{
	protected:
		float bonus;
	public:
		Manager (string name, float salary, float bonus) : Employee(name,salary),
bonus(bonus){}

	void display(){
		cout << "Employee name: " << name << endl;
		cout << "Salary: " << salary << endl;
		cout << "Bonus: " << bonus << endl;
	}
};

int main(){
	Manager m1("Ali", 50000, 25000);
	m1.display();
}
```

```
Employee name: Ali
Salary: 50000
Bonus: 25000

--------------------------------
Process exited after 0.4929 seconds with return value 0
Press any key to continue . . .
```

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 display() {
        cout << "Brand: " << brand << endl;
        cout << "Speed: " << speed << " km/h" << endl;
    }
};

class Car : public Vehicle {
protected:
    int seats;

public:
    Car(string b, int s, int se) : Vehicle(b, s), seats(se) {}
    void display() override {
        Vehicle::display();
        cout << "Seats: " << seats << endl;
    }
};

class ElectricCar : public Car {
private:
    int batteryLife;

public:
    ElectricCar(string b, int s, int se, int bl) : Car(b, s, se), batteryLife(bl) {}
    void display() override {
        Car::display();
```

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

int main() {
        ElectricCar eCar("Tesla", 200, 5, 6);

    eCar.display();

    return 0;
}
```

```
Brand: Tesla
Speed: 200 km/h
Seats: 5
Battery Life: 6 hours

--------------------------------
Process exited after 0.4791 seconds with return value 0
Press any key to continue . . .
```

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 display(){
                        cout << "Name: " << name << endl;
                        cout << "Age: " << age << endl;
                }
};

class Teacher : public Person{
        protected:
                string subject;
        public:
                Teacher(string n, int a, string s) : Person (n, a), subject(s){}
        void display() override {
                Person::display();
```

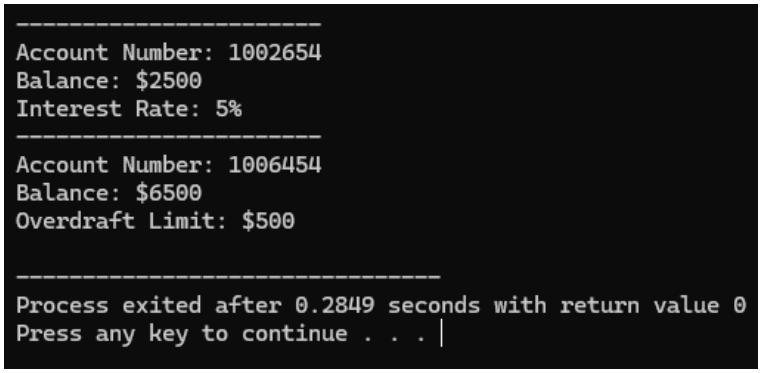
```
        cout << "Subject: " << subject << endl;
    }
};

class Researcher : public Teacher{
    protected:
        string researchArea;
    public:
        Researcher(string n, int a, string s, string rA) : Teacher (n, a, s),
researchArea(rA){}
   void display() override {
    Teacher::display();
    cout << "Research Area: " << researchArea << endl;
    }
};

class Professor : public Researcher{
    protected:
        int publications;
    public:
        Professor(string n, int a, string s, string rA, int p) : Researcher(n, a, s, rA),
publications(p){}
    void display() override{
        Researcher::display();
        cout << "Publications: " << publications << endl;
    }
};

int main(){
    Professor p1("Zaroon Shah", 55, "Chemistry", "Atom Theory", 5);
    p1.display();

    return 0;
}
```

```
Name: Zaroon Shah
Age: 55
Subject: Chemistry
Research Area: Atom Theory
Publications: 5

--------------------------------
Process exited after 0.3872 seconds with return value 0
Press any key to continue . . .
```

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 << "----------------------" << endl;
        cout << "Account Number: " << accountNumber << endl;
        cout << "Balance: $" << balance << endl;
    }
};

class SavingsAccount : public Account {
private:
    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 {
private:
    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() {
```

```
    SavingsAccount savings(1002654, 2500, 5);
    CheckingAccount checking(1006454, 6500, 500);
    savings.displayDetails();
    checking.displayDetails();

    return 0;
}
```



05:

```
#include <iostream>
using namespace std;

class Device {
protected:
    int deviceID;
    bool status;

public:
    Device(int id, bool stat) : deviceID(id), status(stat) {}
    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 stat, float screen) : Device(id, stat), screenSize(screen) {}
    void displayDetails() override {
        Device::displayDetails();
```

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

class SmartWatch : virtual public Device {
protected:
    bool heartRateMonitor;

public:
    SmartWatch(int id, bool stat, bool monitor) : Device(id, stat), heartRateMonitor(monitor) {}
    void displayDetails() override {
        cout << "Heart Rate Monitor: " << (heartRateMonitor ? "Yes" : "No") << endl;
    }
};

class SmartWearable : public SmartPhone, public SmartWatch {
private:
    int stepCounter;

public:
    SmartWearable(int id, bool stat, float screen, bool monitor, int steps)
        : Device(id, stat), SmartPhone(id, stat, screen), SmartWatch(id, stat, monitor),
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() {

    SmartWearable wearable(20052, 1, 6.7, 1, 5000);
    wearable.displayDetails();

    return 0;
}
```

```
Device ID: 20052
Status: On
Screen Size: 6.7 inches
Heart Rate Monitor: Yes
Step Counter: 5000 steps

--------------------------------
Process exited after 0.2518 seconds with return value 0
Press any key to continue . . .
```