# OOP Lab 06 tasks

Name : Hamza Atif

Roll No: 24K-0594

## Task 01:

#include<iostream>

#include<string>

using namespace std;

class Employee

{

public:

string name;

float salary;

Employee(string name , float salary){

this->name = name;

this->salary = salary;

}

void displayDetails(){

cout << "Employee name: " << name << endl;

cout << "Employee salary: " << salary << endl;

}

};

class Manager : public Employee{

public:

float bonus;

Manager(string name , float salary , float bonus) : Employee(name , salary){

this->bonus = bonus;

};

void displayDetails(){

Employee::displayDetails();

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

}

};

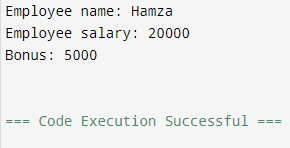
int main(){

Manager m1("Hamza" , 20000 , 5000);

m1.displayDetails();

return 0;

}



## Task 02:

#include<iostream>

#include<string>

using namespace std;

class Vehicle

{

public:

string brand;

int speed;

Vehicle(string brand , int speed){

this->brand = brand;

this->speed = speed;

}

void displayDetails(){

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

cout << "speed: " << speed << endl;

}

};

class Car : public Vehicle

{

public:

int seats;

Car(string brand , int speed , int seats) : Vehicle(brand , speed){

this->seats = seats;

}

void displayDetails(){

Vehicle::displayDetails();

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

}

};

class ElectricCar : public Car

{

public:

int batteryLife;

ElectricCar(string brand , int speed , int seats , int batteryLife) : Car(brand , speed , seats){

this->batteryLife = batteryLife;

}

void displayDetails(){

Car :: displayDetails();

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

}

};

int main(){

string brnd;

int spd;

int sts;

int bl;

cout << "Enter the brand of the car: ";

cin >> brnd;

cout << "Enter the speed of the car: ";

cin >> spd;

cout << "Enter the number of seats: ";

cin >> sts;

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

cin >> bl;;

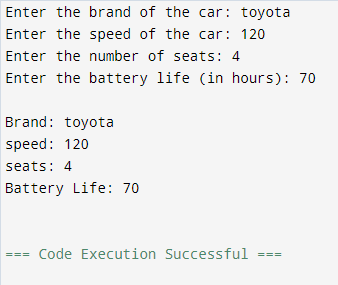
cout << endl;

ElectricCar car1(brnd , spd , sts , bl);

car1.displayDetails();

return 0;

}



## Task 03:

#include<iostream>

#include<string>

using namespace std;

class Person

{

public:

string name;

int age;

Person(string name , int age){

this->name = name;

this->age = age;

}

void displayDetails(){

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

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

}

};

class Teacher : public Person

{

public:

string subject;

Teacher(string name , int age , string subject):Person(name , age){

this->subject = subject;

}

void displayDetails(){

Person::displayDetails();

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

}

};

class Researcher:public Teacher

{

public:

string researchArea;

Researcher(string name , int age , string subject , string researchArea) : Teacher(name,age,subject){

this->researchArea = researchArea;

}

void displayDetails(){

Teacher::displayDetails();

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

}

};

class Professor:public Researcher

{

public:

int publication;

Professor(string name , int age , string subject , string researchArea , int publication) : Researcher(name , age , subject , researchArea){

this->publication = publication;

}

void displayDetails(){

Researcher::displayDetails();

cout << "Publications: " << publication << endl;

}

};

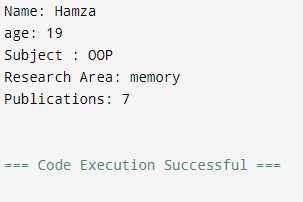
int main(){

Professor p1("Hamza" , 19 , "OOP" , "memory" , 7);

p1.displayDetails();

return 0;

}



## Task 04:

#include<iostream>

#include<string>

using namespace std;

class Account

{

public:

int accountNumber;

float balance;

Account(int accountNumber , float balance){

this->accountNumber = accountNumber;

this->balance = balance;

}

void displayDetails(){

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

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

}

};

class SavingsAccount : public Account

{

public:

float interestRate;

SavingsAccount(int accountNumber , float balance , float interestRate) : Account(accountNumber , balance){

this->interestRate = interestRate;

}

void displayDetails(){

Account::displayDetails();

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

}

};

class CheckingAccount : public Account

{

public:

float overdraftLimit;

CheckingAccount(int accountNumber , float balance , float overdraftLimit) : Account(accountNumber , balance){

this->overdraftLimit = overdraftLimit;

}

void displayDetails(){

Account::displayDetails();

cout << "Over Draft Limit : " << overdraftLimit << endl;

}

};

int main(){

int an;

float baln;

float odl;

float ir;

int choice;

cout << "Enter the Account number: " << endl;

cin >> an;

cout << "Enter the Balance: " << endl;

cin >> baln;

cout << "Enter the Over Draft Limit: " << endl;

cin >> odl;

cout << "Enter the Interest Rate: " << endl;

cin >> ir;

cout << "What Account do you want to view(1.Savings , 2.Checking)" << endl;

cin >> choice;

if(choice == 1){

SavingsAccount A1(an , baln , ir);

A1.displayDetails();

}

else if(choice == 2){

CheckingAccount C1(an , baln , odl);

C1.displayDetails();

}

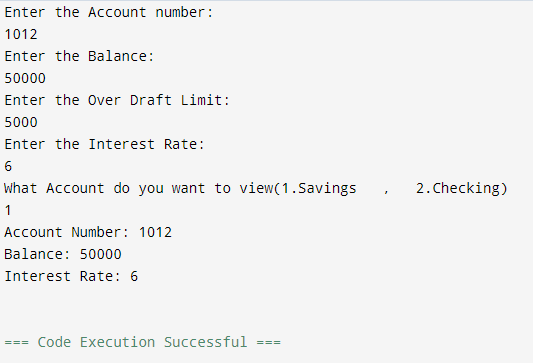
else{

cout << "Invalid Input\n";

}

return 0;

}



## Task 05:

#include<iostream>

#include<string>

using namespace std;

class Device

{

public:

int deviceID;

bool status;

Device(int deviceID , bool status){

this->deviceID = deviceID;

this->status = status;

}

void displayDetails(){

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

cout << "Status: " << status << endl;

}

};

class SmartPhone : public Device

{

public:

float screenSize;

SmartPhone(int deviceID , bool status , float screenSize) : Device(deviceID , status){

this->screenSize = screenSize;

}

void displayDetails(){

Device::displayDetails();

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

}

};

class SmartWatch : public SmartPhone

{

public:

bool heartRateMonitor;

SmartWatch(int deviceID , bool status , float screenSize , bool heartRateMonitor) : SmartPhone(deviceID , status , screenSize){

this->heartRateMonitor = heartRateMonitor;

}

void displayDetails(){

SmartPhone::displayDetails();

cout << "Heart Rate Monitor: " << heartRateMonitor << endl;

}

};

class SmartWearable : public SmartWatch

{

public:

int stepCounter;

SmartWearable(int deviceID , bool status , float screenSize , bool heartRateMonitor , int stepCounter) : SmartWatch(deviceID , status , screenSize , heartRateMonitor){

this->stepCounter = stepCounter;

}

void displayDetails(){

SmartWatch::displayDetails();

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

}

};

int main(){

SmartWearable S1(1123 , 1 , 23.4 , 1 , 40);

S1.displayDetails();

return 0;

}

