

**International University of Business Agriculture and Technology**

**Lab Report 7**

**Course Code: CSC 284**

**Course Name: Programming in C++ Lab**

**Submitted To:**

**Engr. A.S.M. Shakil Ahamed**

Senior Lecturer

Dept. of Computer Science and Engineering

International University of Business Agriculture and Technology

**Submitted By:**

Name: Md. Mahfujar Rahman

ID: 23303151

Section: C

**1.Code:**

#include <iostream>

#include <string>

using namespace std;

class Person

{

public:

    string name;

    Person(string n)

    {

        name = n;

    }

};

class Student : public Person

{

public:

    int studentID;

    Student(string name, int ID) : Person(name)

    {

        studentID = ID;

    }

};

int main()

{

    Student s1("Mahfujar Rahman", 23303151);

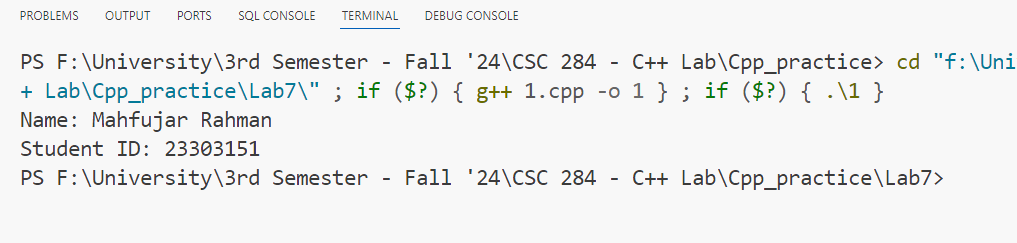
    cout << "Name: " << s1.name << endl;

    cout << "Student ID: " << s1.studentID << endl;

    return 0;

}

**Output:**

****

**2.Code:**

#include <iostream>

using namespace std;

class Shape

{

public:

    int width;

    int height;

    Shape(int w, int h)

    {

        width = w;

        height = h;

    }

};

class Rectangle : public Shape

{

public:

    Rectangle(int width, int height) : Shape(width, height)

    {

    }

    int calculateArea()

    {

        return width \* height;

    }

};

int main()

{

    Rectangle r1(5, 10);

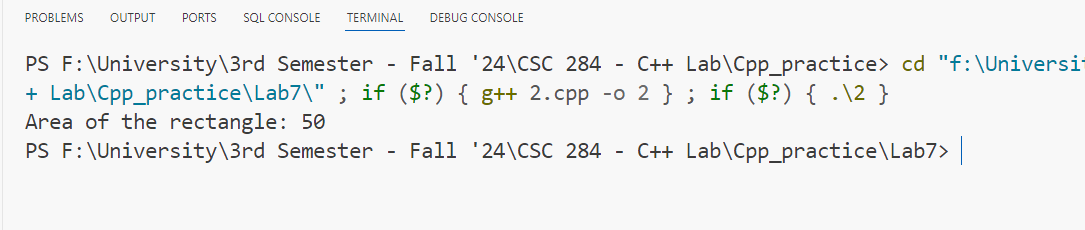
    int area = r1.calculateArea();

    cout << "Area of the rectangle: " << area << endl;

    return 0;

}

**Output:**

****

**3.Code:**

#include <iostream>

using namespace std;

class Vehicle

{

public:

    int numWheels;

    int topSpeed;

    Vehicle(int numWheels, int topSpeed)

    {

        this->numWheels = numWheels;

        this->topSpeed = topSpeed;

    }

};

class Car : public Vehicle

{

public:

    int numDoors;

    Car(int numWheels, int topSpeed, int numDoors) : Vehicle(numWheels, topSpeed)

    {

        this->numDoors = numDoors;

    }

};

int main()

{

    Car car(4, 180, 4);

    cout << "Number of wheels: " << car.numWheels << endl;

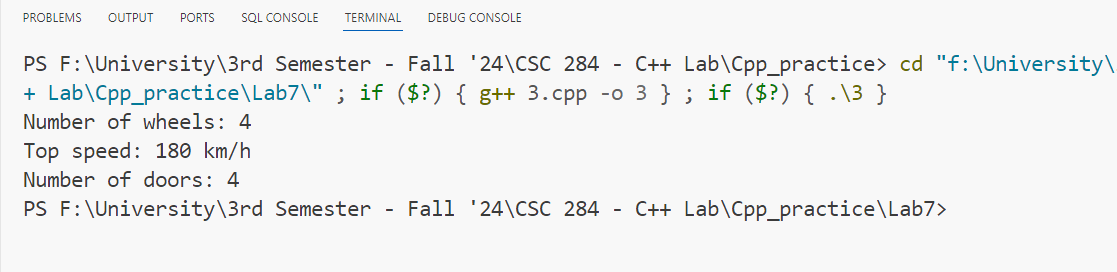
    cout << "Top speed: " << car.topSpeed << " km/h" << endl;

    cout << "Number of doors: " << car.numDoors << endl;

    return 0;

}

**Output:**



**4.Code:**

#include <iostream>

#include <string>

using namespace std;

class Animal

{

public:

    string name;

    Animal(string name)

    {

        this->name = name;

    }

};

class Cat : public Animal

{

public:

    int numLives;

    Cat(string name, int numLives) : Animal(name)

    {

        this->numLives = numLives;

    }

};

int main()

{

    Cat c1("Tiger", 9);

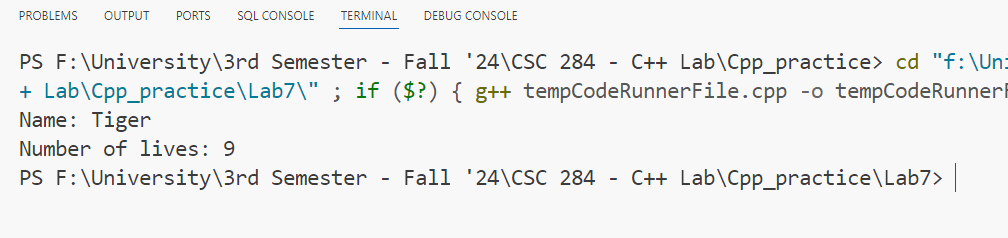
    cout << "Name: " << c1.name << endl;

    cout << "Number of lives: " << c1.numLives << endl;

    return 0;

}

**Output:**



**5.Code:**

#include <iostream>

using namespace std;

class BankAccount

{

public:

    double balance;

    BankAccount(double balance)

    {

        this->balance = balance;

    }

};

class SavingsAccount : public BankAccount

{

public:

    double interestRate;

    SavingsAccount(double balance, double interestRate) : BankAccount(balance)

    {

        this->interestRate = interestRate;

    }

};

int main()

{

    SavingsAccount sa1(1000.0, 0.05);

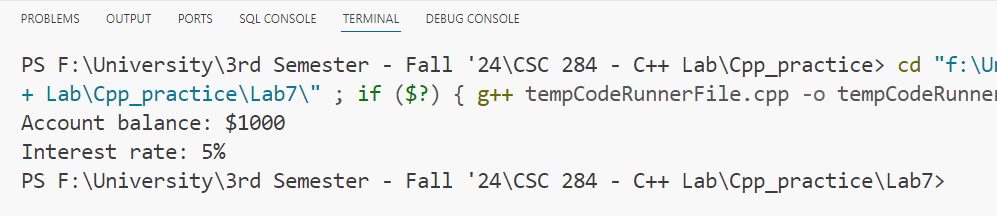
    cout << "Account balance: $" << sa1.balance << endl;

    cout << "Interest rate: " << sa1.interestRate \* 100 << "%" << endl;

    return 0;

}

**Output:**

****

**6.Code:**

#include <iostream>

using namespace std;

class Person

{

protected:

    string name;

    int age;

};

class Student

{

protected:

    int studentID;

    double GPA;

};

class StudentAthlete : protected Person, protected Student

{

public:

    string sportsPlayed;

    StudentAthlete()

    {

        cout << "Student Name: ";

        cin >> name;

        cout << "Student Age: ";

        cin >> age;

        cout << "Student ID: ";

        cin >> studentID;

        cout << "Student GPA: ";

        cin >> GPA;

        cout << "Sports Played: ";

        cin >> sportsPlayed;

    }

    ~StudentAthlete()

    {

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

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

        cout << "Student ID: " << studentID << endl;

        cout << "Student GPA: " << GPA << endl;

        cout << "Sports Played: " << sportsPlayed << endl;

    }

};

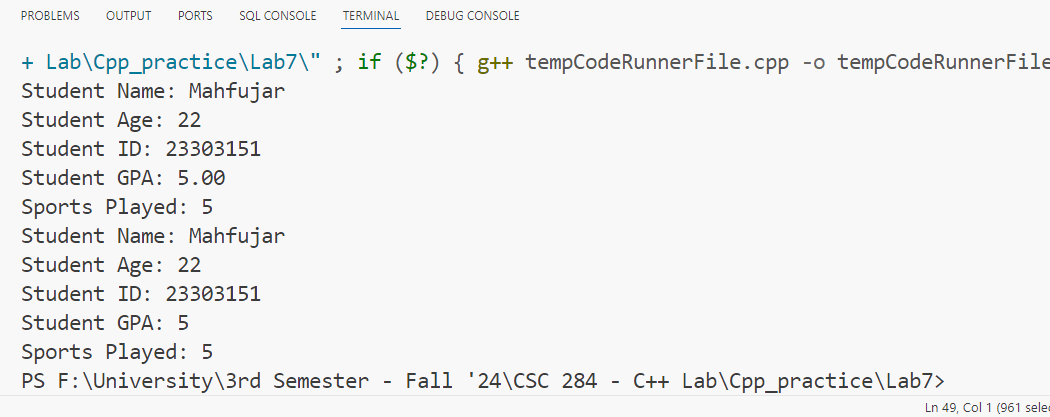
int main()

{

    StudentAthlete sa1;

}

**Output:**

****

**7.Code:**

#include <iostream>

#include <string>

using namespace std;

class Shape

{

public:

    int width;

    int height;

    Shape(int w, int h)

    {

        width = w;

        height = h;

    }

};

class Color

{

public:

    string color;

    Color(string colour)

    {

        color = colour;

    }

};

class ColoredShape : public Shape, public Color

{

public:

    ColoredShape(int width, int height, string color) : Shape(width, height), Color(color) {}

    void printInfo()

    {

        cout << "Width: " << width << endl;

        cout << "Height: " << height << endl;

        cout << "Color: " << color << endl;

    }

};

int main()

{

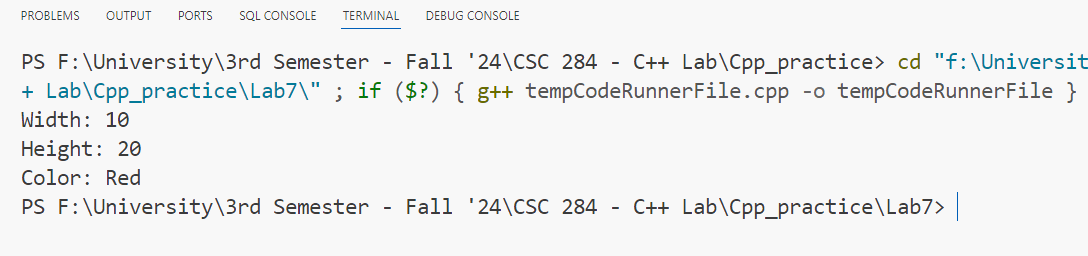
    ColoredShape shape(10, 20, "Red");

    shape.printInfo();

    return 0;

}

**Output:**

****

**8.Code:**

#include <iostream>

using namespace std;

class Vehicle

{

protected:

    int numWheels;

    int topSpeed;

public:

    Vehicle(int wheels, int speed) : numWheels(wheels), topSpeed(speed) {}

};

class Engine

{

protected:

    int horsepower;

public:

    Engine(int hp) : horsepower(hp) {}

};

class Car : public Vehicle, public Engine

{

private:

    int numDoors;

public:

    Car(int wheels, int speed, int hp, int doors)

        : Vehicle(wheels, speed), Engine(hp), numDoors(doors) {}

    void displayInfo()

    {

        cout << "Number of wheels: " << numWheels << endl;

        cout << "Top speed: " << topSpeed << " km/h" << endl;

        cout << "Horsepower: " << horsepower << " hp" << endl;

        cout << "Number of doors: " << numDoors << endl;

    }

};

int main()

{

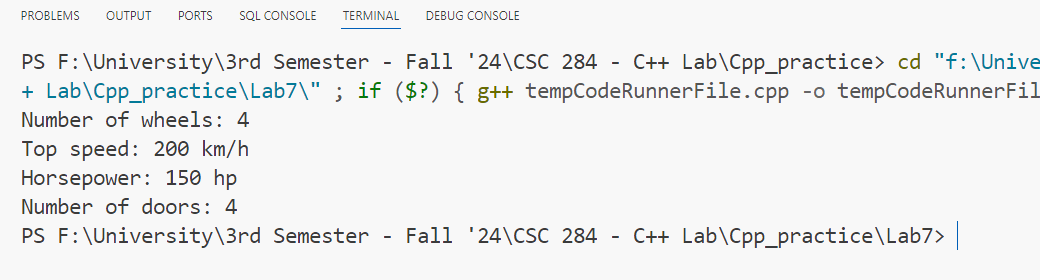
    Car myCar(4, 200, 150, 4);

    myCar.displayInfo();

    return 0;

}

**Output:**

****

**9.Code:**

#include <iostream>

#include <string>

using namespace std;

class Animal

{

protected:

    string name;

    int age;

public:

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

};

class Pet

{

protected:

    string owner;

public:

    Pet(string o) : owner(o) {}

};

class PetAnimal : public Animal, public Pet

{

private:

    string species;

public:

    PetAnimal(string n, int a, string o, string s)

        : Animal(n, a), Pet(o), species(s) {}

    void displayInfo()

    {

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

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

        cout << "Owner: " << owner << endl;

        cout << "Species: " << species << endl;

    }

};

int main()

{

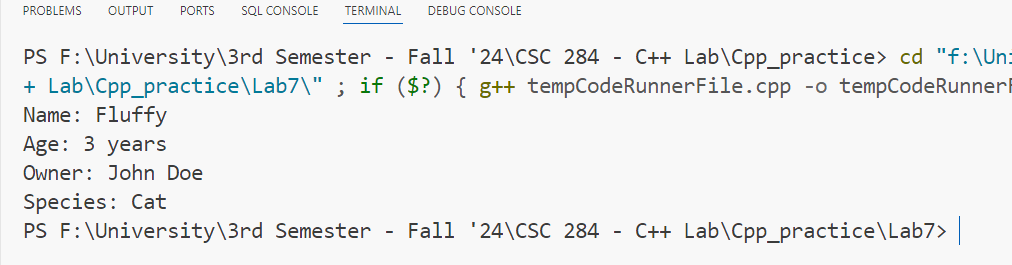
    PetAnimal myPet("Fluffy", 3, "John Doe", "Cat");

    myPet.displayInfo();

    return 0;

}

**Output:**

****

**10.Code:**

#include <iostream>

using namespace std;

class Mammals

{

public:

    void print()

    {

        cout << "I am a mammal" << endl;

    }

};

class MarineAnimals

{

public:

    void print()

    {

        cout << "I am a marine animal" << endl;

    }

};

class BlueWhale : public Mammals, public MarineAnimals

{

public:

    void print()

    {

        cout << "I belong to both the categories: Mammals as well as Marine Animals" << endl;

    }

};

int main()

{

    Mammals mammal;

    MarineAnimals marine;

    BlueWhale whale;

    mammal.print();

    marine.print();

    whale.print();

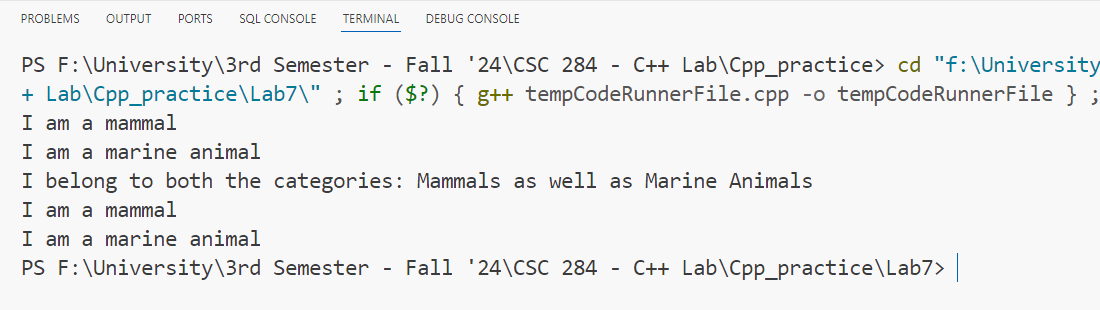
    whale.Mammals::print();

    whale.MarineAnimals::print();

    return 0;

}

**Output:**

****

**11.Code:**

#include <iostream>

using namespace std;

class Fruit

{

protected:

    int count;

public:

    Fruit() : count(0) {}

    void addFruit(int num)

    {

        count += num;

    }

    int getCount() const

    {

        return count;

    }

    virtual void printCount() const = 0;

};

class Apples : public Fruit

{

public:

    void printCount() const override

    {

        cout << "Number of Apples: " << getCount() << endl;

    }

};

class Mangoes : public Fruit

{

public:

    void printCount() const override

    {

        cout << "Number of Mangoes: " << getCount() << endl;

    }

};

class Basket

{

private:

    Apples apples;

    Mangoes mangoes;

public:

    void addApples(int num)

    {

        apples.addFruit(num);

    }

    void addMangoes(int num)

    {

        mangoes.addFruit(num);

    }

    void printFruitCounts() const

    {

        apples.printCount();

        mangoes.printCount();

        cout << "Total number of fruits: " << apples.getCount() + mangoes.getCount() << endl;

    }

};

int main()

{

    Basket basket;

    basket.addApples(5);

    basket.addMangoes(3);

    basket.addApples(2);

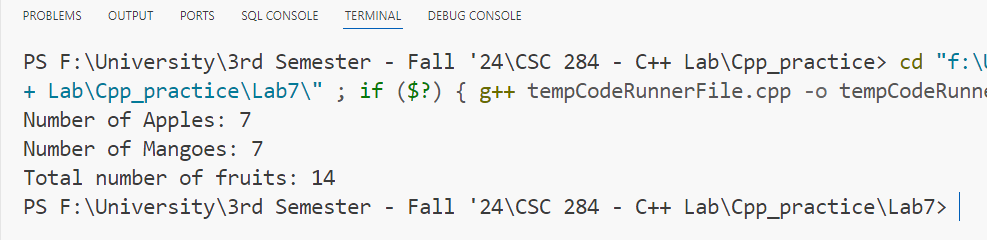
    basket.addMangoes(4);

    basket.printFruitCounts();

    return 0;

}

**Output:**

****

**12.Code:**

#include <iostream>

#include <string>

#include <vector>

using namespace std;

class Marks

{

protected:

    static int rollCounter;

    int rollNumber;

    string name;

    int marks;

public:

    Marks(const string &studentName) : name(studentName), marks(0)

    {

        rollNumber = ++rollCounter;

    }

    virtual void inputMarks() = 0; // Pure virtual function

    int getMarks() const { return marks; }

    int getRollNumber() const { return rollNumber; }

    string getName() const { return name; }

};

int Marks::rollCounter = 0;

class Physics : public Marks

{

public:

    Physics(const string &studentName) : Marks(studentName) {}

    void inputMarks() override

    {

        cout << "Enter Physics marks for " << name << ": ";

        cin >> marks;

    }

};

class Chemistry : public Marks

{

public:

    Chemistry(const string &studentName) : Marks(studentName) {}

    void inputMarks() override

    {

        cout << "Enter Chemistry marks for " << name << ": ";

        cin >> marks;

    }

};

class Mathematics : public Marks

{

public:

    Mathematics(const string &studentName) : Marks(studentName) {}

    void inputMarks() override

    {

        cout << "Enter Mathematics marks for " << name << ": ";

        cin >> marks;

    }

};

int main()

{

    int numStudents;

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

    cin >> numStudents;

    cin.ignore(); // To ignore the newline character after entering the number

    vector<Physics> physicsStudents;

    vector<Chemistry> chemistryStudents;

    vector<Mathematics> mathStudents;

    for (int i = 0; i < numStudents; ++i)

    {

        string name;

        cout << "Enter name of student " << i + 1 << ": ";

        getline(cin, name);

        Physics physicsStudent(name);

        Chemistry chemistryStudent(name);

        Mathematics mathStudent(name);

        physicsStudent.inputMarks();

        chemistryStudent.inputMarks();

        mathStudent.inputMarks();

        physicsStudents.push\_back(physicsStudent);

        chemistryStudents.push\_back(chemistryStudent);

        mathStudents.push\_back(mathStudent);

    }

    int totalMarks = 0;

    for (int i = 0; i < numStudents; ++i)

    {

        int studentTotal = physicsStudents[i].getMarks() + chemistryStudents[i].getMarks() + mathStudents[i].getMarks();

        totalMarks += studentTotal;

        cout << "Total marks for " << physicsStudents[i].getName() << " (Roll No: " << physicsStudents[i].getRollNumber() << "): " << studentTotal << endl;

    }

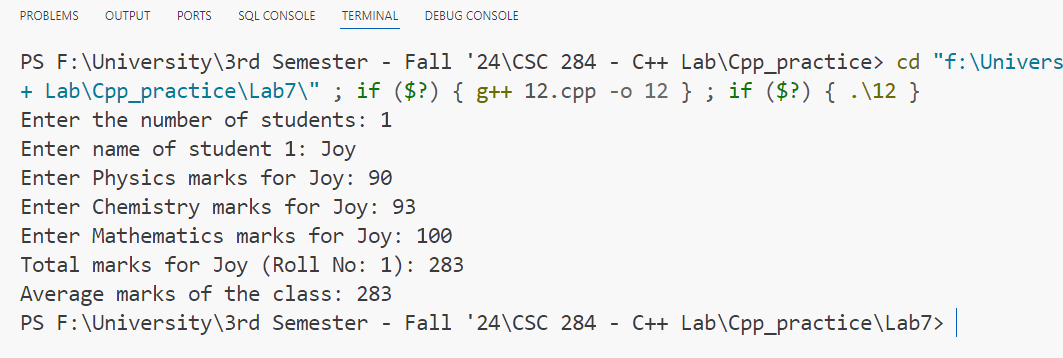
    double averageMarks = static\_cast<double>(totalMarks) / numStudents;

    cout << "Average marks of the class: " << averageMarks << endl;

    return 0;

}

**Output:**

****

**13.Code:**

#include <iostream>

#include <string>

using namespace std;

class Vehicle

{

protected:

    double mileage;

    double price;

public:

    Vehicle(double mileage, double price) : mileage(mileage), price(price) {}

    virtual void displayInfo() const = 0; // Pure virtual function

};

class Car : public Vehicle

{

protected:

    double ownershipCost;

    int warranty; // in years

    int seatingCapacity;

    string fuelType;

public:

    Car(double mileage, double price, double ownershipCost, int warranty, int seatingCapacity, const string &fuelType)

        : Vehicle(mileage, price), ownershipCost(ownershipCost), warranty(warranty), seatingCapacity(seatingCapacity), fuelType(fuelType) {}

};

class Bike : public Vehicle

{

protected:

    int numberOfCylinders;

    int numberOfGears;

    string coolingType;

    string wheelType;

    double fuelTankSize; // in inches

public:

    Bike(double mileage, double price, int numberOfCylinders, int numberOfGears, const string &coolingType, const string &wheelType, double fuelTankSize)

        : Vehicle(mileage, price), numberOfCylinders(numberOfCylinders), numberOfGears(numberOfGears), coolingType(coolingType), wheelType(wheelType), fuelTankSize(fuelTankSize) {}

};

class Audi : public Car

{

private:

    string modelType;

public:

    Audi(double mileage, double price, double ownershipCost, int warranty, int seatingCapacity, const string &fuelType, const string &modelType)

        : Car(mileage, price, ownershipCost, warranty, seatingCapacity, fuelType), modelType(modelType) {}

    void displayInfo() const override

    {

        cout << "Audi Model: " << modelType << endl;

        cout << "Ownership Cost: " << ownershipCost << endl;

        cout << "Warranty: " << warranty << " years" << endl;

        cout << "Seating Capacity: " << seatingCapacity << endl;

        cout << "Fuel Type: " << fuelType << endl;

        cout << "Mileage: " << mileage << endl;

        cout << "Price: " << price << endl;

    }

};

class Ford : public Car

{

private:

    string modelType;

public:

    Ford(double mileage, double price, double ownershipCost, int warranty, int seatingCapacity, const string &fuelType, const string &modelType)

        : Car(mileage, price, ownershipCost, warranty, seatingCapacity, fuelType), modelType(modelType) {}

    void displayInfo() const override

    {

        cout << "Ford Model: " << modelType << endl;

        cout << "Ownership Cost: " << ownershipCost << endl;

        cout << "Warranty: " << warranty << " years" << endl;

        cout << "Seating Capacity: " << seatingCapacity << endl;

        cout << "Fuel Type: " << fuelType << endl;

        cout << "Mileage: " << mileage << endl;

        cout << "Price: " << price << endl;

    }

};

class Bajaj : public Bike

{

private:

    string makeType;

public:

    Bajaj(double mileage, double price, int numberOfCylinders, int numberOfGears, const string &coolingType, const string &wheelType, double fuelTankSize, const string &makeType)

        : Bike(mileage, price, numberOfCylinders, numberOfGears, coolingType, wheelType, fuelTankSize), makeType(makeType) {}

    void displayInfo() const override

    {

        cout << "Bajaj Make: " << makeType << endl;

        cout << "Number of Cylinders: " << numberOfCylinders << endl;

        cout << "Number of Gears: " << numberOfGears << endl;

        cout << "Cooling Type: " << coolingType << endl;

        cout << "Wheel Type: " << wheelType << endl;

        cout << "Fuel Tank Size: " << fuelTankSize << " inches" << endl;

        cout << "Mileage: " << mileage << endl;

        cout << "Price: " << price << endl;

    }

};

class TVS : public Bike

{

private:

    string makeType;

public:

    TVS(double mileage, double price, int numberOfCylinders, int numberOfGears, const string &coolingType, const string &wheelType, double fuelTankSize, const string &makeType)

        : Bike(mileage, price, numberOfCylinders, numberOfGears, coolingType, wheelType, fuelTankSize), makeType(makeType) {}

    void displayInfo() const override

    {

        cout << "TVS Make: " << makeType << endl;

        cout << "Number of Cylinders: " << numberOfCylinders << endl;

        cout << "Number of Gears: " << numberOfGears << endl;

        cout << "Cooling Type: " << coolingType << endl;

        cout << "Wheel Type: " << wheelType << endl;

        cout << "Fuel Tank Size: " << fuelTankSize << " inches" << endl;

        cout << "Mileage: " << mileage << endl;

        cout << "Price: " << price << endl;

    }

};

int main()

{

    Audi audiCar(15.0, 50000, 2000, 3, 5, "Petrol", "A4");

    Ford fordCar(12.0, 40000, 1800, 2, 5, "Diesel", "Mustang");

    Bajaj bajajBike(40.0, 1500, 2, 5, "Air", "Alloys", 12.0, "Pulsar");

    TVS tvsBike(35.0, 1200, 1, 4, "Liquid", "Spokes", 10.0, "Apache");

    cout << "Audi Car Information:" << endl;

    audiCar.displayInfo();

    cout << endl;

    cout << "Ford Car Information:" << endl;

    fordCar.displayInfo();

    cout << endl;

    cout << "Bajaj Bike Information:" << endl;

    bajajBike.displayInfo();

    cout << endl;

    cout << "TVS Bike Information:" << endl;

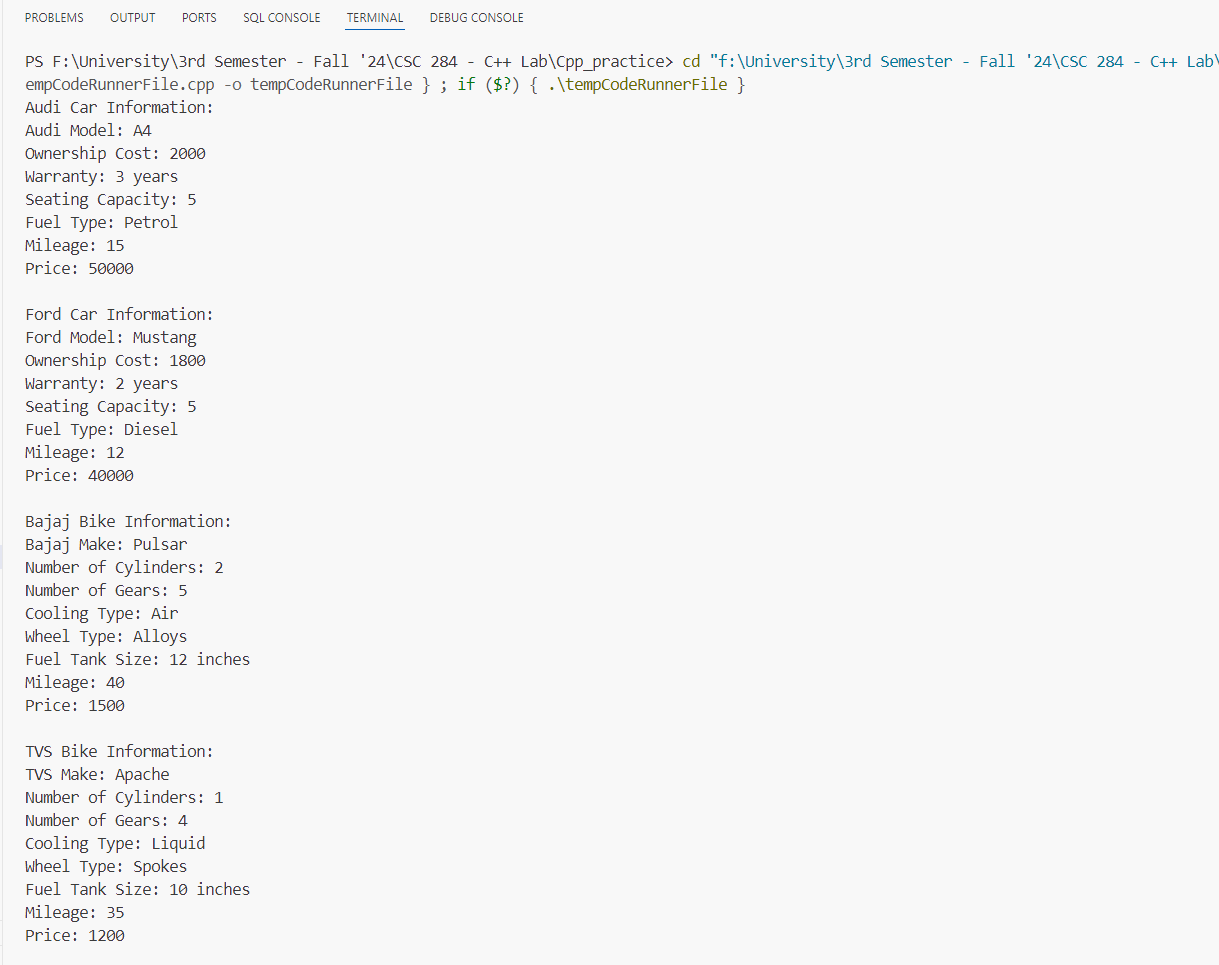
    tvsBike.displayInfo();

    cout << endl;

    return 0;

}

**Output:**

****

**14.Code:**

#include <iostream>

using namespace std;

class Shape

{

public:

    virtual void printMessage()

    {

        cout << "This is a shape." << endl;

    }

};

class Polygon : public Shape

{

public:

    void printMessage() override

    {

        cout << "Polygon is a shape." << endl;

    }

};

class Rectangle : public Polygon

{

public:

    void printMessage() override

    {

        cout << "Rectangle is a polygon." << endl;

    }

};

class Triangle : public Polygon

{

public:

    void printMessage() override

    {

        cout << "Triangle is a polygon." << endl;

    }

};

class Square : public Rectangle

{

public:

    void printMessage() override

    {

        cout << "Square is a rectangle." << endl;

    }

};

int main()

{

    Shape shape;

    Polygon polygon;

    Rectangle rectangle;

    Triangle triangle;

    Square square;

    shape.printMessage();

    polygon.printMessage();

    rectangle.printMessage();

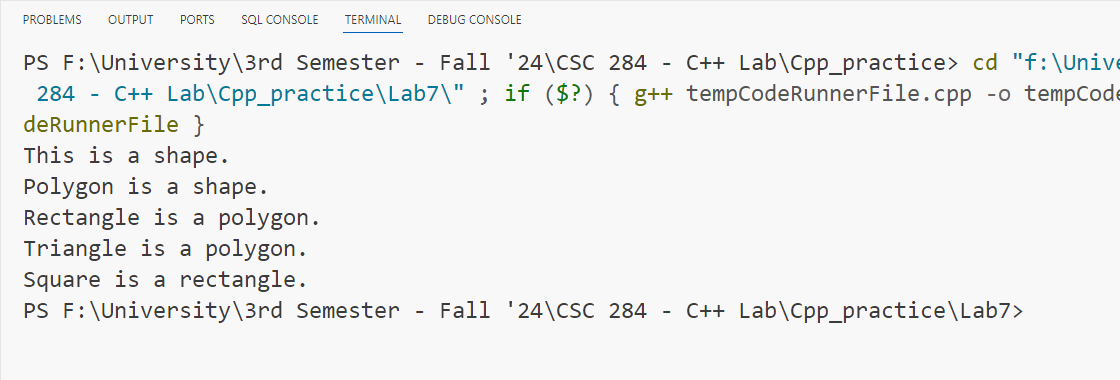
    triangle.printMessage();

    square.printMessage();

    return 0;

}

**Output:**

****

**15.Code:**

#include <iostream>

#include <string>

using namespace std;

class RBI

{

protected:

    double minInterestRate = 4.0;

    double minBalance = 1000;

    double maxWithdrawalLimit = 50000;

public:

    virtual void displayGuidelines()

    {

        cout << "RBI Guidelines:" << endl;

        cout << "Minimum Interest Rate: " << minInterestRate << "%" << endl;

        cout << "Minimum Balance: Rs. " << minBalance << endl;

        cout << "Maximum Withdrawal Limit: Rs. " << maxWithdrawalLimit << endl;

    }

};

class Bank : public RBI

{

protected:

    string bankName;

public:

    Bank(const string &name) : bankName(name) {}

    virtual void displayBankDetails()

    {

        cout << "Bank Name: " << bankName << endl;

        displayGuidelines();

    }

};

class SBI : public Bank

{

public:

    SBI() : Bank("State Bank of India") {}

    void displayBankDetails() override

    {

        cout << "Welcome to SBI!" << endl;

        Bank::displayBankDetails();

    }

};

class ICICI : public Bank

{

public:

    ICICI() : Bank("ICICI Bank") {}

    void displayBankDetails() override

    {

        cout << "Welcome to ICICI Bank!" << endl;

        Bank::displayBankDetails();

    }

};

class PNB : public Bank

{

public:

    PNB() : Bank("Punjab National Bank") {}

    void displayBankDetails() override

    {

        cout << "Welcome to PNB!" << endl;

        Bank::displayBankDetails();

    }

};

class Customer

{

private:

    string name;

    int accountNumber;

    double balance;

public:

    Customer(const string &customerName, int accNum, double initialBalance)

        : name(customerName), accountNumber(accNum), balance(initialBalance) {}

    void displayCustomerDetails()

    {

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

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

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

    }

};

int main()

{

    SBI sbi;

    ICICI icici;

    PNB pnb;

    Customer customer1("Alice", 123456, 2000.0);

    Customer customer2("Bob", 654321, 1500.0);

    cout << "--- SBI Details ---" << endl;

    sbi.displayBankDetails();

    cout << "\n--- ICICI Details ---" << endl;

    icici.displayBankDetails();

    cout << "\n--- PNB Details ---" << endl;

    pnb.displayBankDetails();

    cout << "\n--- Customer Details ---" << endl;

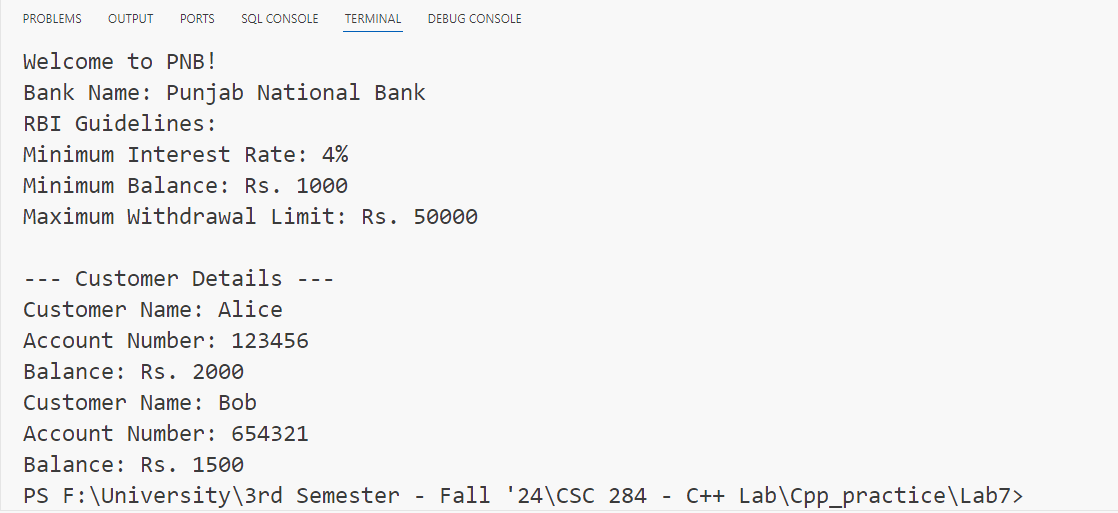
    customer1.displayCustomerDetails();

    customer2.displayCustomerDetails();

    return 0;

}

**Output:**

****

**16.Code:**

#include <iostream>

using namespace std;

class Number

{

protected:

    int num;

    int sqr;

    int cube;

public:

    Number(int n) : num(n), sqr(0), cube(0) {}

};

class Square : public Number

{

public:

    Square(int n) : Number(n)

    {

        sqr = num \* num;

    }

    void display()

    {

        cout << "Square of " << num << " is: " << sqr << endl;

    }

};

class Cube : public Number

{

public:

    Cube(int n) : Number(n)

    {

        cube = num \* num \* num;

    }

    void print()

    {

        cout << "Cube of " << num << " is: " << cube << endl;

    }

};

int main()

{

    int number;

    cout << "Enter a number: ";

    cin >> number;

    Square square(number);

    square.display();

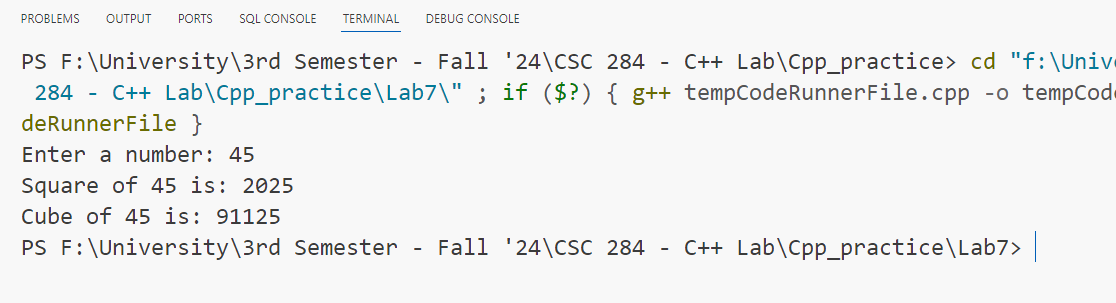
    Cube cube(number);

    cube.print();

    return 0;

}

**Output:**

****