Q1)

#include <iostream>

using namespace std;

class Shape {

protected:

double width;

double height;

public:

Shape(double w, double h) {

width = w;

height = h;

}

};

class Rectangle : public Shape {

public:

Rectangle(double w, double h) : Shape(w, h) {}

double getArea() {

return width \* height;

}

};

int main() {

double width, height;

cout << "Enter the width of the rectangle: ";

cin >> width;

cout << "Enter the height of the rectangle: ";

cin >> height;

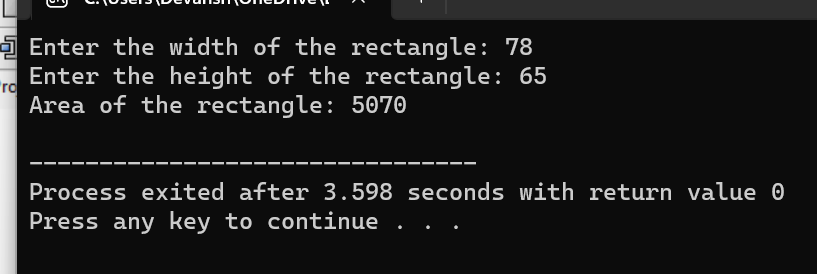
Rectangle rectangle(width, height);

double area = rectangle.getArea();

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

return 0;

}



Q2)

#include <iostream>

#include <string>

using namespace std;

class Person {

protected:

string name;

int age;

string gender;

public:

void setDetails(string n, int a, string g) {

name = n;

age = a;

gender = g;

}

void displayDetails() {

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

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

cout << "Gender: " << gender << endl;

}

};

class Student : public Person {

private:

int rollNumber;

string grade;

public:

void setStudentDetails(int roll, string g) {

rollNumber = roll;

grade = g;

}

void displayStudentDetails() {

displayDetails();

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

cout << "Grade: " << grade << endl;

}

};

int main() {

Student student;

string name, gender;

int age, rollNumber;

string grade;

cout << "Enter the student's name: ";

getline(cin, name);

cout << "Enter the student's age: ";

cin >> age;

cin.ignore();

cout << "Enter the student's gender: ";

getline(cin, gender);

cout << "Enter the student's roll number: ";

cin >> rollNumber;

cin.ignore();

cout << "Enter the student's grade: ";

getline(cin, grade);

student.setDetails(name, age, gender);

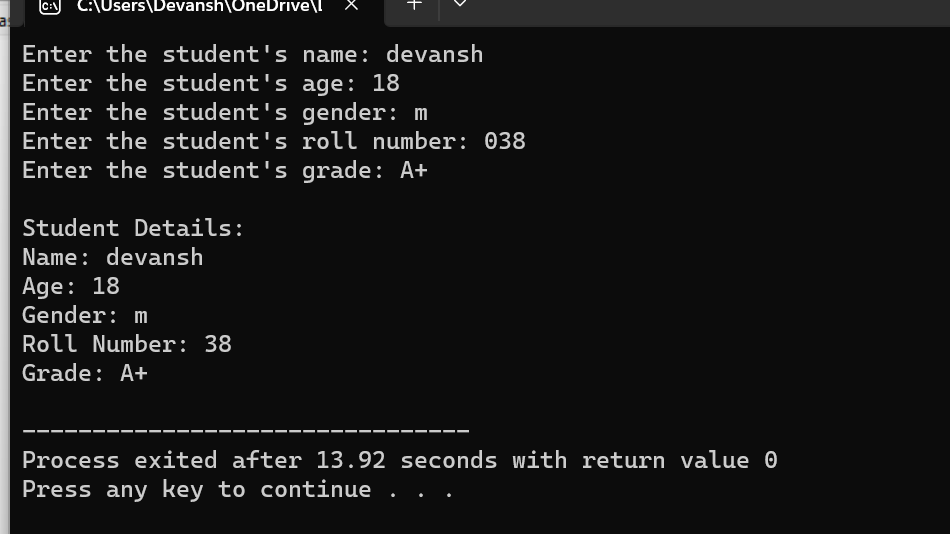
student.setStudentDetails(rollNumber, grade);

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

student.displayStudentDetails();

return 0;

}



Q3)

#include <iostream>

#include <string>

using namespace std;

class Person {

protected:

string name;

int age;

public:

void setPersonDetails(string n, int a) {

name = n;

age = a;

}

void displayPersonDetails() {

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

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

}

};

class Employee {

protected:

int empId;

float salary;

public:

void setEmployeeDetails(int id, float sal) {

empId = id;

salary = sal;

}

void displayEmployeeDetails() {

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

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

}

};

class Manager : public Person, public Employee {

private:

string department;

string designation;

public:

void setManagerDetails(string dept, string desig) {

department = dept;

designation = desig;

}

void displayManagerDetails() {

displayPersonDetails();

displayEmployeeDetails();

cout << "Department: " << department << endl;

cout << "Designation: " << designation << endl;

}

};

int main() {

Manager manager;

string name, department, designation;

int age, empId;

float salary;

cout << "Enter the manager's name: ";

getline(cin, name);

cout << "Enter the manager's age: ";

cin >> age;

cout << "Enter the manager's employee ID: ";

cin >> empId;

cout << "Enter the manager's salary: $";

cin >> salary;

cout << "Enter the manager's department: ";

getline(cin, department);

cout << "Enter the manager's designation: ";

getline(cin, designation);

manager.setPersonDetails(name, age);

manager.setEmployeeDetails(empId, salary);

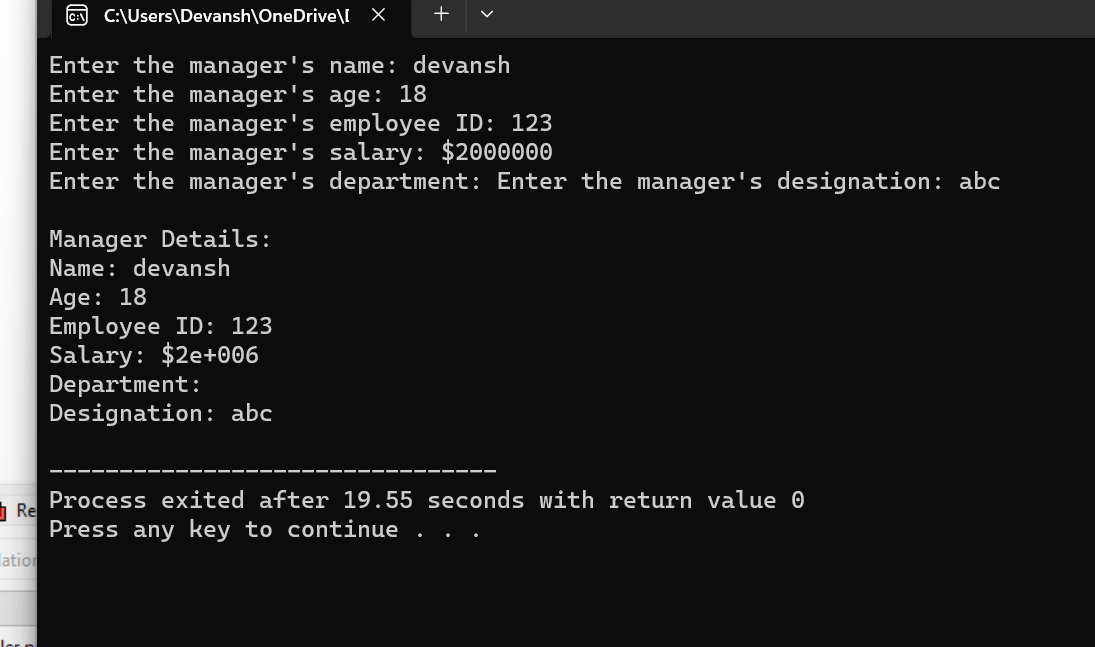
manager.setManagerDetails(department, designation);

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

manager.displayManagerDetails();

return 0;

}



Q4)

#include <iostream>

#include <string>

using namespace std;

class Shape {

protected:

float width;

float height;

public:

void setDimensions(float w, float h) {

width = w;

height = h;

}

void displayDimensions() {

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

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

}

};

class Color {

protected:

string color;

public:

void setColor(string c) {

color = c;

}

void displayColor() {

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

}

};

class Rectangle : public Shape, public Color {

public:

void calculateArea() {

float area = width \* height;

cout << "Area: " << area << " square units" << endl;

}

};

int main() {

Rectangle rectangle;

float width, height;

string color;

cout << "Enter the dimensions of the rectangle:\n";

cout << "Width: ";

cin >> width;

cout << "Height: ";

cin >> height;

cin.ignore();

cout << "Enter the color of the rectangle: ";

getline(cin, color);

rectangle.setDimensions(width, height);

rectangle.setColor(color);

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

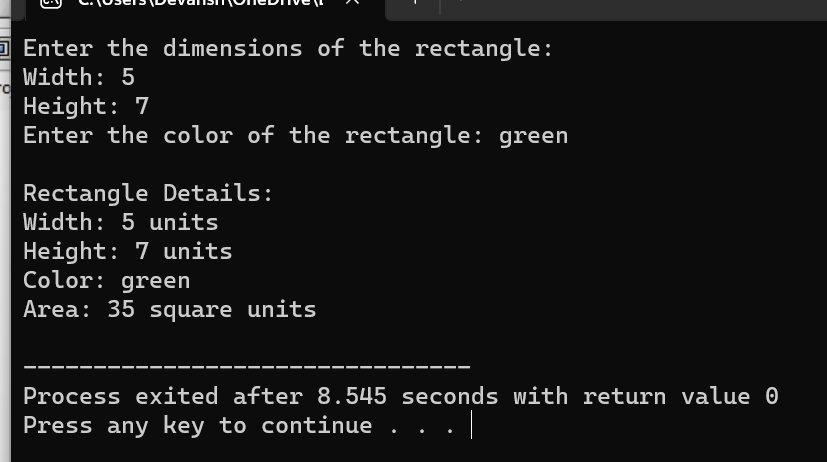
rectangle.displayDimensions();

rectangle.displayColor();

rectangle.calculateArea();

return 0;

}



Q5)

#include <iostream>

#include <string>

using namespace std;

class Vehicle {

protected:

string brand;

float price;

public:

void setDetails(string b, float p) {

brand = b;

price = p;

}

void displayDetails() {

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

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

}

};

class Car : public Vehicle {

protected:

string fuelType;

int numDoors;

public:

void setCarDetails(string ft, int nd) {

fuelType = ft;

numDoors = nd;

}

void displayCarDetails() {

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

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

}

};

class SportsCar : public Car {

private:

float topSpeed;

public:

void setSportsCarDetails(float ts) {

topSpeed = ts;

}

void displaySportsCarDetails() {

cout << "Top Speed: " << topSpeed << " mph" << endl;

}

};

int main() {

SportsCar sportsCar;

string brand, fuelType;

float price, topSpeed;

int numDoors;

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

cin >> brand;

cout << "Enter the price of the car: $";

cin >> price;

cout << "Enter the fuel type of the car: ";

cin >> fuelType;

cout << "Enter the number of doors of the car: ";

cin >> numDoors;

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

cin >> topSpeed;

sportsCar.setDetails(brand, price);

sportsCar.setCarDetails(fuelType, numDoors);

sportsCar.setSportsCarDetails(topSpeed);

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

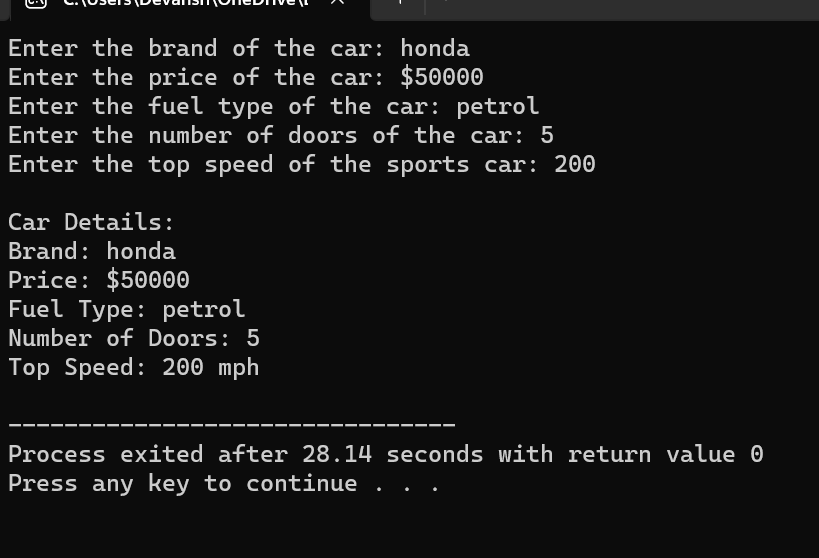
sportsCar.displayDetails();

sportsCar.displayCarDetails();

sportsCar.displaySportsCarDetails();

return 0;

}



Q6)

#include <iostream>

using namespace std;

class Animal {

public:

void eat() {

cout << "Animal is eating." << endl;

}

};

class Mammal : public Animal {

public:

void walk() {

cout << "Mammal is walking." << endl;

}

};

class Dog : public Mammal {

public:

void bark() {

cout << "Dog is barking." << endl;

}

};

int main() {

Dog dog;

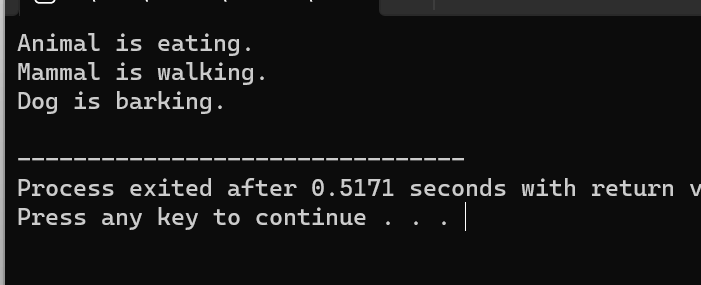
dog.eat();

dog.walk();

dog.bark();

return 0;

}



Q7)

#include <iostream>

using namespace std;

class Shape {

public:

virtual void calculateArea() {

cout << "Area calculation is not defined for the base shape." << endl;

}

};

class Rectangle : public Shape {

public:

void calculateArea() override {

cout << "Calculating area of the rectangle." << endl;

}

};

class Triangle : public Shape {

public:

void calculateArea() override {

cout << "Calculating area of the triangle." << endl;

}

};

class Color {

protected:

string color;

public:

void setColor(string c) {

color = c;

}

};

class ColoredRectangle : public Rectangle, public Color {

public:

void displayColor() {

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

}

};

int main() {

ColoredRectangle coloredRectangle;

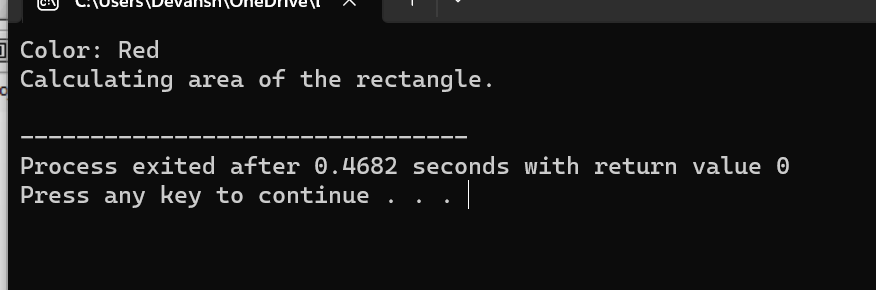
coloredRectangle.setColor("Red");

coloredRectangle.displayColor();

coloredRectangle.calculateArea();

return 0;

}



Q8)

#include <iostream>

using namespace std;

class Animal {

protected:

string name;

string type;

public:

void setDetails(string n, string t) {

name = n;

type = t;

}

void displayDetails() {

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

cout << "Type: " << type << endl;

}

};

class FlyingAnimal {

protected:

int flightSpeed;

public:

void setFlightSpeed(int speed) {

flightSpeed = speed;

}

void displayFlightSpeed() {

cout << "Flight Speed: " << flightSpeed << " km/h" << endl;

}

};

class Bird : public Animal, public FlyingAnimal {

public:

void displayBirdDetails() {

cout << "--- Bird Details ---" << endl;

displayDetails();

displayFlightSpeed();

}

};

int main() {

Bird bird;

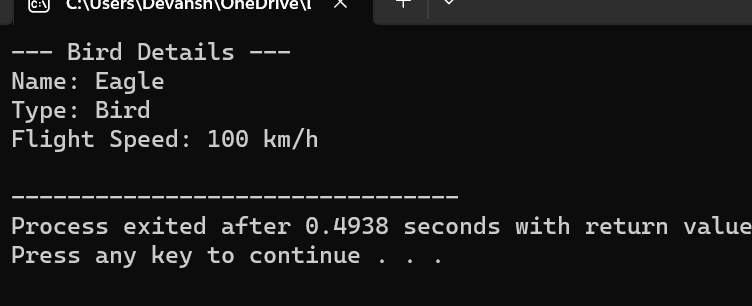
bird.setDetails("Eagle", "Bird");

bird.setFlightSpeed(100);

bird.displayBirdDetails();

return 0;

}



Q9)

#include <iostream>

using namespace std;

class Vehicle {

public:

void start() {

cout << "Vehicle started." << endl;

}

void stop() {

cout << "Vehicle stopped." << endl;

}

};

class Car : public Vehicle {

public:

void carSpecificFunction() {

cout << "Performing car-specific function." << endl;

}

};

class Motorcycle : public Vehicle {

public:

void motorcycleSpecificFunction() {

cout << "Performing motorcycle-specific function." << endl;

}

};

int main() {

Car myCar;

myCar.start();

myCar.carSpecificFunction();

myCar.stop();

cout << endl;

Motorcycle myMotorcycle;

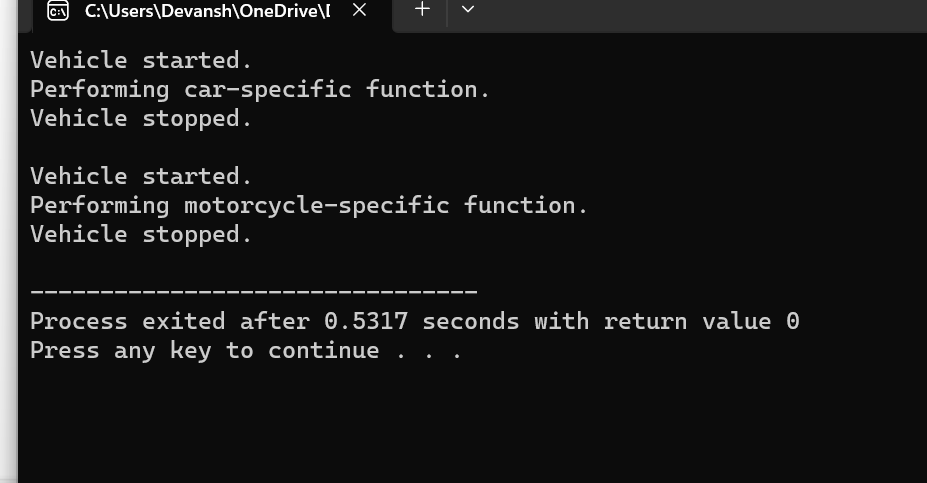
myMotorcycle.start();

myMotorcycle.motorcycleSpecificFunction();

myMotorcycle.stop();

return 0;

}



Q10)

#include <iostream>

#include <cmath>

using namespace std;

class Shape {

protected:

double width;

double height;

public:

void setDimensions(double w, double h) {

width = w;

height = h;

}

void displayDimensions() {

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

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

}

};

class Rectangle : public Shape {

public:

double calculateArea() {

return width \* height;

}

double calculatePerimeter() {

return 2 \* (width + height);

}

};

class Triangle : public Shape {

public:

double calculateArea() {

return 0.5 \* width \* height;

}

double calculatePerimeter() {

double hypotenuse = sqrt((width \* width) + (height \* height));

return width + height + hypotenuse;

}

};

class Circle : public Shape {

public:

double calculateArea() {

return 3.14159 \* pow((width / 2), 2);

}

double calculatePerimeter() {

return 2 \* 3.14159 \* (width / 2);

}

};

int main() {

Rectangle rectangle;

rectangle.setDimensions(5, 8);

rectangle.displayDimensions();

cout << "Rectangle Area: " << rectangle.calculateArea() << endl;

cout << "Rectangle Perimeter: " << rectangle.calculatePerimeter() << endl;

cout << endl;

Triangle triangle;

triangle.setDimensions(4, 6);

triangle.displayDimensions();

cout << "Triangle Area: " << triangle.calculateArea() << endl;

cout << "Triangle Perimeter: " << triangle.calculatePerimeter() << endl;

cout << endl;

Circle circle;

circle.setDimensions(6, 0);

circle.displayDimensions();

cout << "Circle Area: " << circle.calculateArea() << endl;

cout << "Circle Circumference: " << circle.calculatePerimeter() << endl;

return 0;

}

