**Task 1: File Handling**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

class Student

{

private:

string name;

int roll\_number;

float marks;

public:

void input()

{

cout << "Enter name: ";

getline(cin, name);

cout << "Enter roll number: ";

cin >> roll\_number;

cout << "Enter marks: ";

cin >> marks;

cin.ignore();

}

void display() const

{

cout << "Name: " << name << "\nRoll Number: " << roll\_number << "\nMarks: " << marks << "\n";

}

int getRollNumber() const

{

return roll\_number;

}

void updateMarks(float new\_marks)

{

marks = new\_marks;

}

void writeToFile(ofstream& outfile) const

{

outfile << name << "\n" << roll\_number << "\n" << marks << "\n";

}

void readFromFile(ifstream& infile)

{

getline(infile, name);

infile >> roll\_number >> marks;

infile.ignore();

}

};

int main() {

Student students[5];

for (int i = 0; i < 5; ++i)

{

cout << "Enter details for student " << i + 1 << ":\n";

students[i].input();

}

std::ofstream outfile("students.txt");

for (int i = 0; i < 5; ++i)

{

students[i].writeToFile(outfile);

}

outfile.close();

ifstream infile("students.txt");

cout << "\nReading data from file:\n";

for (int i = 0; i < 5; ++i) {

students[i].readFromFile(infile);

students[i].display();

cout << "------------------\n";

}

infile.close();

int target\_roll;

cout << "Enter roll number of student whose marks you want to modify: ";

cin >> target\_roll;

ifstream infile\_mod("students.txt");

for (int i = 0; i < 5; ++i)

{

students[i].readFromFile(infile\_mod);

if (students[i].getRollNumber() == target\_roll)

{

float new\_marks;

cout << "Enter new marks: ";

cin >> new\_marks;

students[i].updateMarks(new\_marks);

}

}

infile\_mod.close();

ofstream outfile\_mod("students.txt");

for (int i = 0; i < 5; ++i)

{

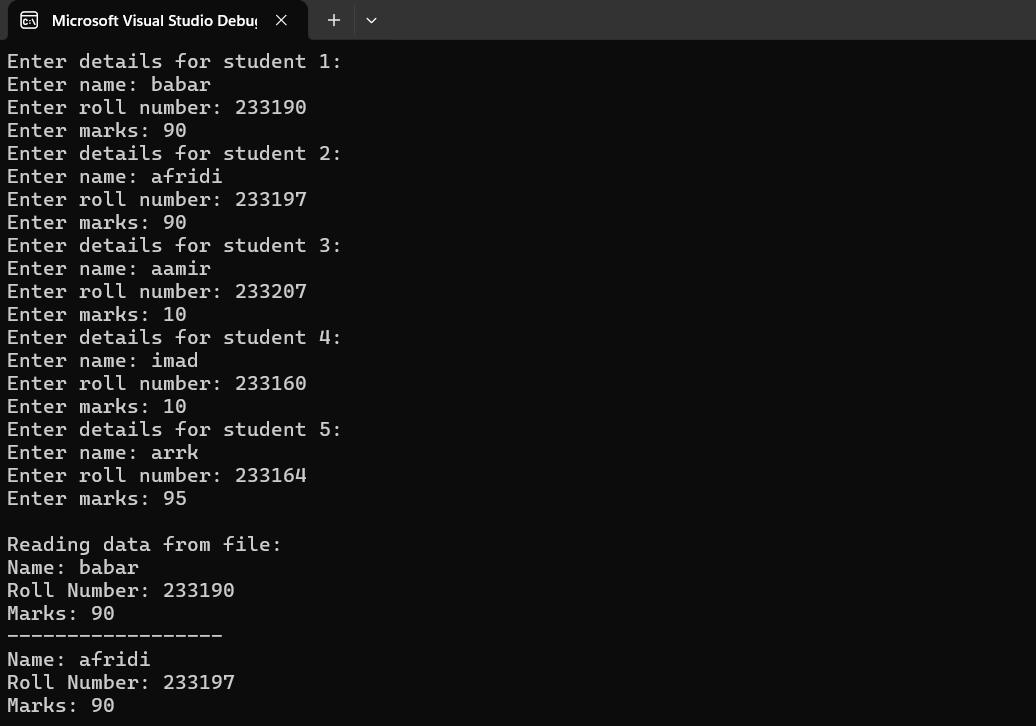
students[i].writeToFile(outfile\_mod);

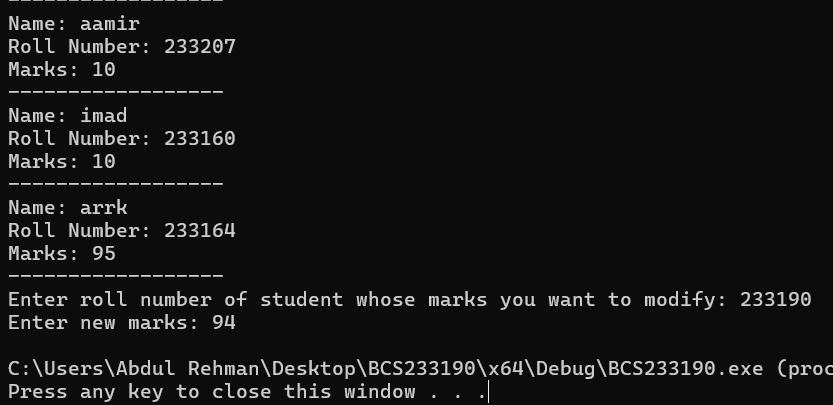
}

outfile\_mod.close();

return 0;

}





**Task 2: Inheritance**

**Part A: Single Inheritance**

#include <iostream>

#include <string>

using namespace std;

class Shape

{

protected:

string color;

public:

Shape(const string& c)

{

color = c;

}

void setColor(const string& c)

{

color = c;

}

string getColor() const

{

return color;

}

};

class Rectangle : public Shape

{

private:

double length;

double breadth;

public:

Rectangle(const string& c, double l, double b) : Shape(c), length(l), breadth(b) {}

double area() const

{

return length \* breadth;

}

double perimeter() const

{

return 2 \* (length + breadth);

}

};

int main()

{

Rectangle r("Blue", 7.0, 9.0);

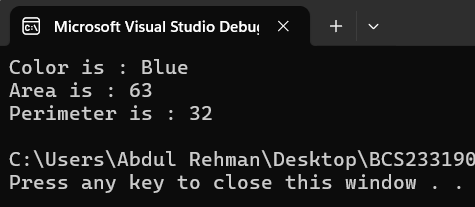
cout << "Color is : " << r.getColor() << endl;

cout << "Area is : " << r.area() << endl;

cout << "Perimeter is : " << r.perimeter() << endl;

return 0;

}



**Part B: Multilevel Inheritance**

#include <iostream>

#include <string>

using namespace std;

class Animal

{

protected:

string name;

public:

Animal(string n)

{

name = n;

}

void display()

{

cout << "Animal name : " << name << endl;

}

};

class Mammal : public Animal

{

protected:

int num\_of\_Legs;

public:

Mammal(string n, int legs) : Animal(n), num\_of\_Legs(legs) {}

void display()

{

Animal::display();

cout << "Number of Legs is " << num\_of\_Legs << endl;

}

};

class Dog : public Mammal

{

public:

Dog(string n, int legs) : Mammal(n, legs) {}

void bark()

{

cout << name << " Dog says Waoh" << endl;

}

};

int main()

{

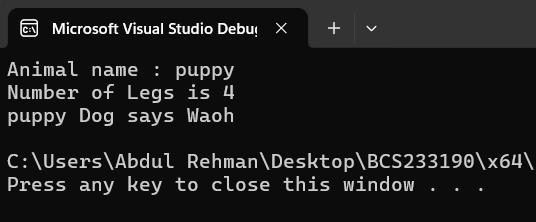
Dog d("puppy", 4);

d.display();

d.bark();

return 0;

}



**Part C: Multiple Inheritance**

#include <iostream>

#include <string>

using namespace std;

class Person

{

public:

string name;

string address;

Person(string n, string a)

{

name = n;

address = a;

}

};

class Employee

{

public:

int emp\_ID;

double sal;

Employee(int id, double s)

{

emp\_ID = id;

sal = s;

}

};

class Teacher : public Person, public Employee

{

public:

string subject;

Teacher(string n, string a, int id, double s, string sub)

: Person(n, a), Employee(id, s), subject(sub) {}

};

int main()

{

Teacher t("Babar", " ep 001", 001, 100000.0, "Batting");

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

cout << "Address: " << t.address << endl;

cout << "Employee ID: " << t.emp\_ID << endl;

cout << "Salary: " << t.sal << endl;

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

return 0;

}

