**Name: Talha Hafiz**

**Roll No: Bse233135**

**Assignment No: 03**

**Codes No 01:**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

class Student {

string name;

int roll\_number;

float marks;

public:

void input() {

cout << "Enter name: ";

cin >> name;

cout << "Enter roll number: ";

cin >> roll\_number;

cout << "Enter marks: ";

cin >> marks;

}

void display() {

cout << "Name: " << name << ", Roll Number: " << roll\_number << ", Marks: " << marks << endl;

}

void writeToFile(fstream& file) {

file << name << " " << roll\_number << " " << marks << endl;

}

void readFromFile(fstream& file) {

file >> name >> roll\_number >> marks;

}

int getRollNumber() {

return roll\_number;

}

void modifyMarks(float new\_marks) {

marks = new\_marks;

}

};

int main() {

Student students[5];

fstream file("students.txt", ios::out);

if (!file) {

cout << "Error: Could not create or open the file for writing." << endl;

return 1;

}

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

cout << "Enter details of student " << i + 1 << ":" << endl;

students[i].input();

students[i].writeToFile(file);

}

file.close();

file.open("students.txt", ios::in | ios::out);

if (!file) {

cout << "Error: Could not open the file for reading and writing." << endl;

return 1;

}

cout << "\nData of students from file:\n";

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

students[i].readFromFile(file);

students[i].display();

}

int roll\_num\_to\_modify;

float new\_marks;

cout << "\nEnter the roll number of the student whose marks you want to modify: ";

cin >> roll\_num\_to\_modify;

cout << "Enter the new marks: ";

cin >> new\_marks;

file.clear();

file.seekg(0);

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

long pos = file.tellg();

students[i].readFromFile(file);

if (students[i].getRollNumber() == roll\_num\_to\_modify) {

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

file.seekp(pos);

students[i].writeToFile(file);

break;

}

}

file.close();

return 0;

}

**Codes No 02:**

#include <iostream>

#include <string>

using namespace std;

class Shape {

private:

string color;

public:

Shape() : color("Unknown") {

cout << "shape constructor called" << endl;

}

Shape(const string& c) : color(c) {

cout << "Shape parameterized constructor called" << endl;

}

void setColor(const string& c) {

color = c;

}

string getColor() const {

return color;

}

};

class Rectangle : public Shape {

private:

double length;

double breadth;

public:

Rectangle() : Shape(), length(0), breadth(0) {

cout << "Rectangle constructor called" << endl;

}

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

cout << "Rectangle parameterized constructor called" << endl;

}

void setLength(double l) {

length = l;

}

void setBreadth(double b) {

breadth = b;

}

double getLength() const {

return length;

}

double getBreadth() const {

return breadth;

}

double calculateArea() const {

return length \* breadth;

}

double calculatePerimeter() const {

return 2 \* (length + breadth);

}

};

int main() {

Rectangle rect("Red", 5.0, 3.0);

cout << "Color of rectangle: " << rect.getColor() << endl;

cout << "Length " << rect.getLength() << endl;

cout << "Breadth " << rect.getBreadth() << endl;

cout << "Area " << rect.calculateArea() << endl;

cout << "Perimeter " << rect.calculatePerimeter() << endl;

return 0;

}

**Codes No 03:**

#include <iostream>

#include <string>

using namespace std;

class Animal {

protected:

string name;

public:

Animal() : name("Unknown") {

cout << "Animal constructor called" << endl;

}

Animal(const string& n) : name(n) {

cout << "Animal parameterized constructor called" << endl;

}

void setName(const string& n) {

name = n;

}

string getName() const {

return name;

}

};

class Mammal : public Animal {

protected:

int legs;

public:

Mammal() : Animal(), legs(0) {

cout << "Mammal constructor called" << endl;

}

Mammal(const string& n, int l) : Animal(n), legs(l) {

cout << "Mammal parameterized constructor called" << endl;

}

void setLegs(int l) {

legs = l;

}

int getLegs() const {

return legs;

}

};

class Dog : public Mammal {

public:

Dog() : Mammal() {

cout << "Dog constructor called" << endl;

}

Dog(const string& n, int l) : Mammal(n, l) {

cout << "Dog parameterized constructor called" << endl;

}

void bark() const {

cout << name << " says: Woof Woof!" << endl;

}

};

int main() {

Dog dog("Buddy", 4);

cout << "Name: " << dog.getName() << endl;

cout << "Legs: " << dog.getLegs() << endl;

dog.bark();

return 0;

}

**Codes No 04:**

#include <iostream>

#include <string>

using namespace std;

class Person {

protected:

string name;

string address;

public:

Person() : name("Unknown"), address("Unknown") {

cout << "Person constructor called" << endl;

}

Person(const string& n, const string& a) : name(n), address(a) {

cout << "Person parameterized constructor called" << endl;

}

void setName(const string& n) {

name = n;

}

string getName() const {

return name;

}

void setAddress(const string& a) {

address = a;

}

string getAddress() const {

return address;

}

};

class Employee {

protected:

string employeeID;

double salary;

public:

Employee() : employeeID("Unknown"), salary(0.0) {

cout << "Employee constructor called" << endl;

}

Employee(const string& id, double sal) : employeeID(id), salary(sal) {

cout << "Employee parameterized constructor called" << endl;

}

void setEmployeeID(const string& id) {

employeeID = id;

}

string getEmployeeID() const {

return employeeID;

}

void setSalary(double sal) {

salary = sal;

}

double getSalary() const {

return salary;

}

};

class Teacher : public Person, public Employee {

private:

string subject;

public:

Teacher() : Person(), Employee(), subject("Unknown") {

cout << "Teacher constructor called" << endl;

}

Teacher(const string& n, const string& a, const string& id, double sal, const string& sub)

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

cout << "Teacher parameterized constructor called" << endl;

}

void setSubject(const string& sub) {

subject = sub;

}

string getSubject() const {

return subject;

}

void displayInfo() const {

cout << "Name: " << getName() << endl;

cout << "Address: " << getAddress() << endl;

cout << "Employee ID: " << getEmployeeID() << endl;

cout << "Salary: " << getSalary() << endl;

cout << "Subject: " << getSubject() << endl;

}

};

int main() {

Teacher teacher("John Doe", "123 Main St", "T001", 50000, "Mathematics");

teacher.displayInfo();

return 0;

}