Mahros Mohamed
Computer Science
223106831

Prof. Mahmoud

Assignment 1

File - Phase_1.cpp

```cpp
/*
    1) Write a program in C++ that creates a class called laptop.
        The data members of the class are brand (string), model (
string), serial (int), colour (string),
            price (float), processor speed (float), RAM (int),
screen size(float).
        Create member function that will set the individual values
.
        Since the RAM can be upgraded therefore create a function
that allows you to upgrade the RAM only.
        In the end, create a function that will display all the
data members.
*/

#include <iostream>
using namespace std;

#define ll long long

class Laptop{
private:
    string _brand, _model, _colour;
    float _price, _speed, _screen;
    int _serial, _ram;
public:
    // Constructor ()
    Laptop(int, string , string, string, float, float, float, int
);

    // Setters
    void ram(int);
    void serial(int);
    void price(float);
    void speed(float);
    void screen(float);
    void brand(string);
    void model(string);
    void colour(string);

    // Getters
    int ram();
    int serial();
    float price();
    float speed();
```

```
    float screen();
    string brand();
    string model();
    string colour();

    // Functionality
    void show(){
        printf("Labtop(brand = %s, colour = %s, price = %.2f,
model = %s, speed = %.2f, ram = %i, serial = %i)\n", _brand.data
(), _colour.data(), _price, _model.data(), _speed, _ram, _serial);
    }

    // Destructor
    ~Laptop();
};

// Header
// Constructor
Laptop::Laptop(int serial, string brand, string model, string
colour, float price, float speed, float screen, int ram): _serial(
serial), _brand(brand), _model(model), _colour(colour), _price(
price), _speed(speed), _screen(screen), _ram(ram){}

// Setters
void Laptop::ram(int ram){
    _ram = ram;
}
void Laptop::serial(int serial){
    _serial = serial;
}
void Laptop::price(float price){
    _price = price;
}
void Laptop::speed(float speed){
    _speed = speed;
}
void Laptop::screen(float screen){
    _screen = screen;
}
void Laptop::brand(string brand){
    _brand = brand;
}
void Laptop::model(string model){
    _model = model;
```

```
}
void Laptop::colour(string colour){
    _colour = colour;
}

// Getters
int Laptop::ram(){
    return _ram;
}
int Laptop::serial(){
    return _serial;
}
float Laptop::price(){
    return _price;
}
float Laptop::speed(){
    return _speed;
}
float Laptop::screen(){
    return _screen;
}
string Laptop::brand(){
    return _brand;
}
string Laptop::model(){
    return _model;
}
string Laptop::colour(){
    return _colour;
}

// Destructor
Laptop::~Laptop(){

}

// main
int main(){
    Laptop Lab1(12, "DELL", "G3", "BLACK", 20000, 7, 980, 16);
    Lab1.show();
    return 0;
}
```

```cpp
/*
    2) Design a class named Rectangle to represent a rectangle.
The class contains:
        • Two private double data fields named side1 and side2
that specify the two sides of the rectangle.
        • Public set and get methods for each of the two data
fields.
        • A public method named getArea() that returns the area of
 a rectangle.
        • A public method named getPerimeter() that returns the
perimeter.

        Write a test program that creates an array of 5 Rectangle
objects.
        Initialize the Rectangle objects by reading the two sides
from the command-line.
        Then, print the two sides, the area and the perimeter of
each Rectangle object.
*/

#include <iostream>
#include <vector>
using namespace std;

#define ll long long

class Course{
private:
    double _length, _width;
public:
    // Constructor ()
    Course(double, double);

    // Setters
    void sideA(double);
    void sideB(double);

    // Getters
    double sideA();
    double sideB();

    // Functionality
    double area();
    double perimeter();
```

```
    void show(){
        printf("Rec(SideA = %.2f, SideB = %.2f, Area = %.2f,
Perimeter = %.2f)\n", _length, _width, area(), perimeter());
    }

    // Destructor
    ~Course();
};

// Header
// Constructor
Course::Course(double sideA, double sideB): _length(sideA), _width
(sideB){}

// Setters
void Course::sideA(double sideA){
    _length = sideA;
}
void Course::sideB(double sideB){
    _width = sideB;
}

// Getters
double Course::sideA(){
    return _length;
}
double Course::sideB(){
    return _width;
}

// Functionality
double Course::area(){
    return _length * _width;
}
double Course::perimeter(){
    return 2 * (_length + _width);
}

// Destructor
Course::~Course(){}

/* This contain all our programm data that will be used in bla,
bla, bla*/
vector<Course> Database;
```

```cpp
void clear(){
    system("cls");
}
void pannel(){
    clear();
    cout << "Choose from here\n";
    cout << "\t1. Create New Rectangle\n";
    cout << "\t2. Show All Rectangles\n";
    cout << "\t8. Clear\n";
    cout << "\t9. Exit\n";
}

void run(){
    pannel();
    // Create array of 5 rectangle, or flex array using vector for exampple.
    bool isRunning = true;
    int input;
    while (isRunning){
        cout << "Input: "; cin >> input;
        switch (input){
            case 1: // new one
                double S1, S2;
                cout << "Enter SideA: "; cin >> S1;
                cout << "Enter SideB: "; cin >> S2;
                Database.push_back(Course(S1, S2));
                pannel();
                break;
            case 2: // SHOW ALL
                for(ll int x = 0; x < Database.size(); x++){
                    cout << "  "; Database.at(x).show();
                }
                break;
            case 8: // clear console
                clear();
                break;
            case 9: // SHOW ALL
                isRunning = false;
                break;
            default:
                cout << "Please! Enter correct value.\n";
                break;
```

```
            }
        }
    }

    // main
    int main(){
        // run
        try{
            run();
        }catch(const exception& ERROR){
            std::cerr << ERROR.what() << '\n';
        }
        return 0;
    }
```

```cpp
/*
    3) Write a class called rectangle.
    Your task is to store the length and width of the rectangle.
    Write a member function called increment that will add 1 to
the value of length and width.
    Also write a function that will compute the area of the
rectangle.
*/

/* As we recently have created a Rectangle class, we can inherit
from it, for reusability
    Rectangle, But: the prof. said "Write a class" not inherit.
*/

#include <iostream>
using namespace std;

#define ll long long

class Course{
private:
    double _length, _width;
public:
    // Constructor ()
    Course(double length, double width) : _length(length), _width(
width){}

    // Functionality
    double area(){
        return _width * _length;
    }
    void increment(){
        _width++; _length++;
    }
};


// main
int main(){
    return 0;
}
```

```cpp
/*
    4) Design a class named Course. The class contains:
        • A private data field named courseName of type String.
        • A private data field named students of type String[].
The size of the array is 100.
        • A private data field named numberOfStudents of type int.

        • Three get methods for data fields courseName, students
[] and numberOfStudents.
        • A method named addStudent(String) that adds a student to
 the Course.
        • A method named dropStudent(String) that drops a student
from the Course.
        • A method named clear() that removes all students from
the Course.

        Write a test class to test your methods.
*/

#include <iostream>
#include <vector>
using namespace std;

#define ll long long

class Course{
private:
    string _name;
    int _studentsCount;
    vector<string> _students;
public:
    // Constructor ()
    Course(string name) : _name(name){}

    // Getters
    string name(){
        return _name;
    }

    vector<string> students(){
        return _students;
    }

    int studentsCount(){
```

```
        return _studentsCount;
    }

    // Setters
    void addStudent(string name){
        _students.push_back(name);
    }
    void dropStudent(string name){
        for(ll int x = 0; x < _students.size(); x++){
            if (_students.at(x) == name){
                _students.erase(_students.begin()+x);
            }
        }
    }

    // Functionality
    void clear(string name){
        _students.clear();
    }
};


// main
int main(){
    // test class
    Course C1("CSE 015");
    C1.addStudent("Mahros Mohamed Mahros");
    C1.addStudent("Ahmed Mohamed Mahros");
    C1.addStudent("Mohamed Mahros");

    C1.dropStudent("Mahros Mohamed Mahros"); // good
    for(string x : C1.students()){
        cout << "Name: " << x << endl;
    }

    return 0;
}
```