



**Department of Computer
Science**

CS 212: Object Oriented Programming

Class: BEE-14D

Fall 2023

**Lab05: Getters and Setters, Separating
implementation from interface**

Name: Irfa Farooq

CMS: 412564

Date: 9th October 2023

Time: 10:00 a.m. - 12:50 p.m. & 2:00 – 4:00 pm

Instructor: Mehreen

Tahir

Lab Engineer: Mehwish Kiran



National University of Sciences and Technology (NUST) School of Electrical Engineering and Computer Science

Task 1: Implement GradeBook example in Visual Studio by creating three files

- a) GradeBook.h
- b) GradeBook.cpp
- c) TestGradeBook

Extend the class implementation by providing data validation for the private data member “courseName”. The length of course name should not exceed 25 characters. In case of an invalid argument the set function should return a code so that the client code can check whether the assignment was successful or not.

Code:

GradeBook.h:

```
#include <string>
using namespace std;

class GradeBook{
public:
    GradeBook(string);
    void setCourseName(string);
    string getCourseName();
    void displayMessage();
private:
    string courseName; // course name for this GradeBook.
};
```

GradeBook.cpp:

```
#include <iostream>
#include "GradeBook.h"
using namespace std;

GradeBook::GradeBook(string name) { courseName = name; }
void GradeBook::setCourseName(string name) {
    courseName = name;
}
string GradeBook::getCourseName() {
    if (courseName.size() >= 26) {
        string Error = "ERROR";
        return Error;
    }
    else {
        return courseName;
    }
}
void GradeBook::displayMessage() {
    string Check;
    if (courseName.size() >= 26) {
    }
    else {
        cout << "Welcome to the grade book for\n" << courseName << "!" <<
endl;
    }
}
```



National University of Sciences and Technology (NUST) School of Electrical Engineering and Computer Science

TestGradeBook.cpp:

```
#include <iostream>
#include "GradeBook.h"
using namespace std;

int main() {
    string check;
    GradeBook gradeBook1("CS101: Introduction to C++ Programming");
    GradeBook gradeBook2("CS102: Data Structures in C++");
    if (gradeBook1.getCourseName() == "ERROR") {
        check = "The assignment was not successful.\n";
    }
    else {
        check = gradeBook1.getCourseName();
    }
    cout << "gradeBook1 created for course: " << check << endl;
    gradeBook2.displayMessage();
}
```

Output Screenshots

The screenshot shows the Microsoft Visual Studio Debug Console window. The output text is as follows:

```
Microsoft Visual Studio Debug Console
gradeBook1 created for course: The assignment was not successful.

C:\Users\Lenovo\OneDrive\Documents\3rd Semester\Object Oriented Programming\OOP Practice\OOP Labs\x64\Debug\OOP Labs.exe
(process 61900) exited with code 0.
Press any key to close this window . . .
```

Task 2: Create a Simple Unit Conversion App that allows converting the following

1. Length (Foot to Meter)
2. Weight (Pounds to Kg.)
3. Temperature (Fahrenheit to Celsius)

Code:

UnitConverter.h:

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

class UnitConversionApp {
public:
    void Feet_Conversion();
    void set_Feet1(float value);
    void set_Feet2(float value);
    float get_Meters();
    void Pounds_Conversion();
    void set_Pounds1(float value);
    void set_Pounds2(float value);
    float get_kgs();
    void Fahrenheit_Conversion();
    void set_Fahrenheit1(float value);
    void set_Fahrenheit2(float value);
    float get_Celcius();
private:
    float input_Data, output_Data;
};
```



UnitConverter.cpp:

```
#include <iostream>
#include "UnitConverter.h"
using namespace std;

void UnitConversionApp::set_Feet1(float feet) {
    input_Data = feet;
}
void UnitConversionApp::set_Feet2(float meter) {
    output_Data = meter;
}
void UnitConversionApp::Feet_Conversion() {
    output_Data = (input_Data / 3.28);
}
float UnitConversionApp::get_Meters() {
    Feet_Conversion();
    return output_Data;
}
void UnitConversionApp::set_Pounds1(float pounds) {
    input_Data = pounds;
}
void UnitConversionApp::set_Pounds2(float kgs) {
    output_Data = kgs;
}
void UnitConversionApp::Pounds_Conversion() {
    output_Data = (input_Data * 0.45359237);
}
float UnitConversionApp::get_kgs() {
    Pounds_Conversion();
    return output_Data;
}
void UnitConversionApp::set_Fahrenheit1(float fehrenheit) {
    input_Data = fehrenheit;
}
void UnitConversionApp::set_Fahrenheit2(float celcius) {
    output_Data = celcius;
}
void UnitConversionApp::Fahrenheit_Conversion() {
    output_Data = (input_Data - 32) * (5 / 9);
}
float UnitConversionApp::get_Celcius() {
    Fahrenheit_Conversion();
    return output_Data;
}
```

TestUnitConverter.cpp:

```
#include <iostream>
#include "UnitConverter.h"
using namespace std;

int main() {
    UnitConversionApp UnitConverter;
    int Switch;
    cout << "-----" << endl;
    cout << "Welcome to Unit Conversion App" << endl;
    cout << "Press 1 to convert from feet to meters: " << endl;
    cout << "Press 2 to convert from pounds to kgs: " << endl;
    cout << "Press 3 to convert from fahrenheit to celcius: " << endl;
```



National University of Sciences and Technology (NUST) School of Electrical Engineering and Computer Science

```
cout << "-----" << endl;
cin >> Switch;
switch (Switch) {
case 1: {
    float feet;
    cout << "Enter Feet = ";
    cin >> feet;
    UnitConverter.set_Feet1(feet);
    cout << "Meters = " << UnitConverter.get_Meters() << " m" << endl;
    break;
}
case 2: {
    float pounds;
    cout << "Enter Pounds = ";
    cin >> pounds;
    UnitConverter.set_Pounds1(pounds);
    cout << "Kilograms = " << UnitConverter.get_kgs() << " kgs" <<
endl;
    break;
}
case 3: {
    float fahrenheit;
    cout << "Enter Fahrenheit = ";
    cin >> fahrenheit;
    UnitConverter.set_Fahrenheit1(fahrenheit);
    cout << "Celcius = " << UnitConverter.get_Celcius() << " C" << endl;
    break;
}
return 0;
}
```

Output Screenshot

```
Microsoft Visual Studio Debug Console
-----
Welcome to Unit Conversion App
Press 1 to convert from feet to meters:
Press 2 to convert from pounds to kgs:
Press 3 to convert from fahrenheit to celcius:
-----
1
Enter Feet = 12
Meters = 3.65854 m
```

```
Microsoft Visual Studio Debug Console
-----
Welcome to Unit Conversion App
Press 1 to convert from feet to meters:
Press 2 to convert from pounds to kgs:
Press 3 to convert from fahrenheit to celcius:
-----
2
Enter Pounds = 12
Kilograms = 5.44311 kgs
```

```
Microsoft Visual Studio Debug Console
-----
Welcome to Unit Conversion App
Press 1 to convert from feet to meters:
Press 2 to convert from pounds to kgs:
Press 3 to convert from fahrenheit to celcius:
-----
3
Enter Fahrenheit = 32
Celcius = 0 C
```