



ASSIGNMENT

COURSE TITLE : Object Oriented Programming
COURSE CODE : CSE 111
ASSIGNMENT NO. : 01
SUBMISSION DATE : N/A

SUBMITTED TO

NAME : N/A
DEPT. OF : N/A

Bangladesh University of Business &
Technology (BUBT)

SUBMITTED BY

NAME : Shadman Shahriar
ID NO. : 20245103408
INTAKE : 53
SECTION : 9
PROGRAM : B.Sc. Engg. in CSE

Object Oriented Programming

Assignment 1

1. A `Box` class

- Implement a class `Box` with attributes `length`, `width`, and `height`.
- Write a function `combineBoxes()` that accepts two `Box` objects, combines their dimensions by adding corresponding sides, and returns the resulting `Box` object.
- Demonstrate the combination of **two boxes** using object passing and returning.

Ans. Here is the C++ program of a class named `Box` that satisfies the conditions above:

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

class Box
{
    int length, width, height;

public:
    void setDimensions(int l, int w, int h)
    {
        length = l;
        width = w;
        height = h;
    }
}
```

```

    int getLength()
    {
        return length;
    }

    int getWidth()
    {
        return width;
    }

    int getHeight()
    {
        return height;
    }
};

Box combineBoxes(Box b1, Box b2)
{
    int length = b1.getLength() + b2.getLength();
    int width = b1.getWidth() + b2.getWidth();
    int height = b1.getHeight() + b2.getHeight();

    Box b3;
    b3.setDimensions(length, width, height);
    return b3;
}

int main()
{
    Box b1, b2;
    b1.setDimensions(36, 12, 48);
    b2.setDimensions(128, 67, 13);

    Box b3 = combineBoxes(b1, b2);
    cout << "Combined Length: " << b3.getLength() << "
units" << endl;
    cout << "Combined Width : " << b3.getWidth() << "

```

```
units" << endl;  
    cout << "Combined Height: " << b3.getHeight() << "  
units" << endl;  
    return 0;  
}
```

Output: The code yields the following output in the terminal:

```
Combined Length: 164 units  
Combined Width : 79 units  
Combined Height: 61 units
```

2. A `Temperature` class

- Design a class `Temperature` with attributes `celsius` and `fahrenheit`.
- Create a constructor that initializes the temperature in Celsius and an **inline function** to convert it to Fahrenheit using an object.

Display the temperature in both **Celsius** and **Fahrenheit** using an object.

Note: There is a slight ambiguity in the last point. Since [class methods are implicitly inlined](#), the inline function will be considered a global function in this answer.

Ans. Here is the C++ program of a class `Temperature` that satisfies the conditions above:

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

class Temperature
{
    float celsius, fahrenheit;

public:
    Temperature(float C) : celsius(C)
    {
    }

    void setTempF(float F)
    {
        fahrenheit = F;
    }

    float getTempC()
    {
```

```

        return celsius;
    }

    float getTempF()
    {
        return fahrenheit;
    }
};

inline float convertToFahrenheit(float C)
{
    return (C * 9 / 5) + 32;
}

int main()
{
    Temperature t1(37);
    float celsius = t1.getTempC();
    float fahrenheit = convertToFahrenheit(celsius);
    t1.setTempF(fahrenheit);

    cout << "Celsius    : " << celsius << "°C" << endl;
    cout << "Fahrenheit: " << t1.getTempF() << "°F" <<
endl;
    return 0;
}

```

Output: The code yields the following output in the terminal:

```

Celsius    : 37°C
Fahrenheit: 98.6°F

```

3. A `Student` class

- There is class `Student` that illustrates some students where each student has a name, an id number, and marks.
- Implement a function to set the students' name, id number, and marks.

Now consider the above scenario and demonstrate a C++ program to set and show the **average marks of 20 students** using array of objects.

Ans. Here is the C++ program of a class `Student` that satisfies the conditions above:

```
#include <iostream>
#define count 20
using namespace std;

class Student
{
    string name;
    int id;
    int marks;

public:
    Student(string s, int i, int m)
    {
        name = s;
        id = i;
        marks = m;
    }

    void setData(string s, int i, int m)
    {
        name = s;
        id = i;
        marks = m;
    }
}
```

```

    }

    int getMarks()
    {
        return marks;
    }
};

int main()
{
    float avg;
    Student students[count] = {
        Student("Afsara", 421, 89),
        Student("Surayea", 422, 89),
        Student("Maria", 402, 91),
        Student("Shumi", 404, 90),
        Student("Tamanna", 406, 97),
        Student("Nusrat", 111, 81),
        Student("Fatema", 107, 80),
        Student("Khadiza", 119, 83),
        Student("Risa", 438, 95),
        Student("Redowan", 338, 94),
        Student("Manzirul", 358, 92),
        Student("Arefin", 347, 97),
        Student("Hakim", 360, 98),
        Student("Rahul", 340, 82),
        Student("Shawon", 357, 82),
        Student("Saif", 361, 84),
        Student("Khairul", 344, 81),
        Student("Mahir", 323, 80),
        Student("John", 331, 87),
        Student("Fikrat", 328, 88)};

    for (int i = 0; i < count; i++)
        avg += (float)students[i].getMarks();
    avg /= count;

    cout << "Average marks of " << count << " students: "

```



```
<< avg << endl;  
    return 0;  
}
```

Output: The code yields the following output in the terminal:

```
Average marks of 20 students: 88
```

Code

You can find all the code snippets [here](#).