CSE-284: Object Oriented Programming
Experiment 1: Introduction to Class and Objects in OOP

Turja Roy
ID: 2108052
Group: G-2

Objectives:

- Introduction with Classes and Objects in C++.
- To create data member and member functions (Methods) in a class.
- To understand the concept of visibility of data member and member functions (Public and Private access).

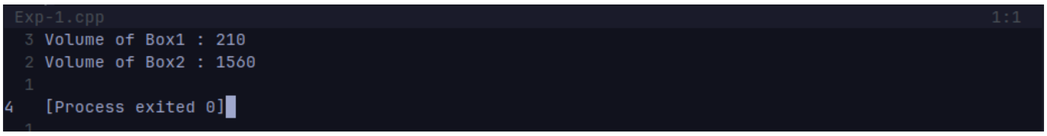
Example 1

Write a C++ program to define a class `Box` and create objects of this class.

Code

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

class Box {
public:
  double length;
  double width;
  double height;
};

int main()
{
  Box Box1;
  Box Box2;
  double volume = 0.0;

  Box1.height = 5.0;
  Box1.length = 6.0;
  Box1.width = 7.0;

  Box2.height = 10.0;
  Box2.length = 12.0;
  Box2.width = 13.0;

  volume = Box1.height * Box1.length * Box1.width;
  cout << "Volume of Box1 : " << volume << endl;

  volume = Box2.height * Box2.length * Box2.width;
  cout << "Volume of Box2 : " << volume << endl;

  return 0;
}
```

Output



```
Volume of Box1 : 210
Volume of Box2 : 1560

[Process exited 0]
```

Figure 1: Output of Exp-1.cpp

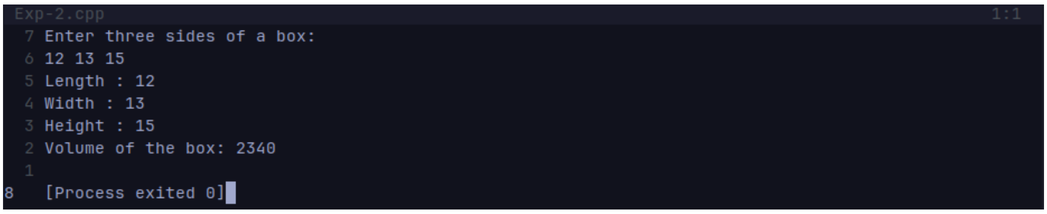
Example 2

Write a C++ program to define a class `Box` with member functions.

Code

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

class BOX {
private:
    double length, width, height;

public:
  void input_value() {
    cout << "Enter three sides of a box: " << endl;
    cin >> length >> width >> height;
  }
  void print_value() {
    cout << "Length : " << length << endl;
    cout << "Width : " << width << endl;
    cout << "Height : " << height << endl;
  }
  double volume() {
    double v = length * width * height;
    return v;
  }
};

int main()
{
  BOX box1, box2;

  box1.input_value();
  box1.print_value();
  double vol = box1.volume();

  cout << "Volume of the box: " << vol << endl;
}
```

Output



```
Enter three sides of a box:
12 13 15
Length : 12
Width : 13
Height : 15
Volume of the box: 2340

[Process exited 0]
```

Figure 2: Output of Exp-2.cpp

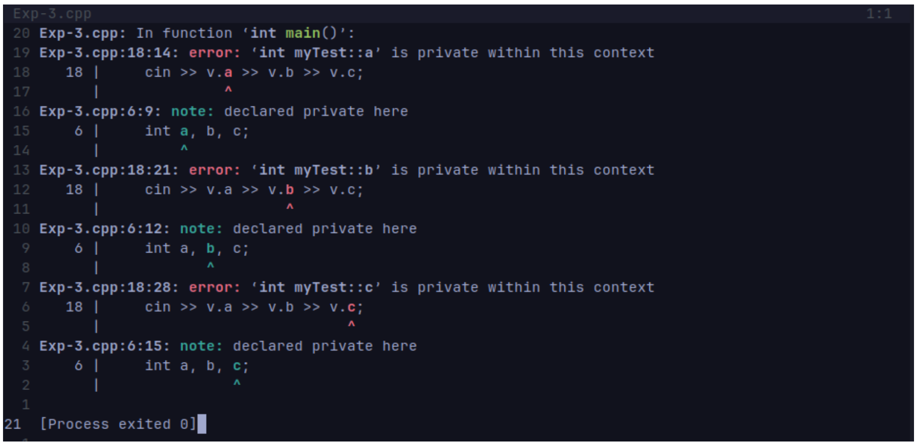
Example 3

Write a C++ program to define a class `Box` with member functions.

Code

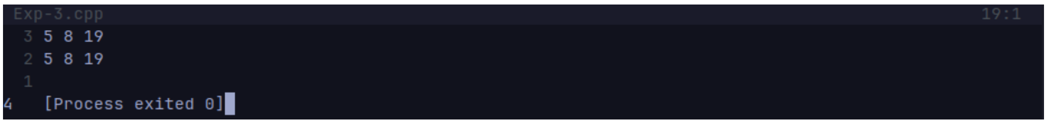
```cpp
#include <iostream>
using namespace std;

class myTest {
private:
    int a, b, c;

public:
    void access_private() {
        cin >> a >> b >> c;
        cout << a << ' ' << b << ' ' << c << endl;
    }
};

int main ()
{
    myTest v;
    // cin >> v.a >> v.b >> v.c;

    v.access_private();

    return 0;
}
```

Output



```
Exp-3.cpp: In function ‘int main()’:
Exp-3.cpp:18:14: error: ‘int myTest::a’ is private within this context
   18 |     cin >> v.a >> v.b >> v.c;
      |              ^
Exp-3.cpp:6:9: note: declared private here
    6 |     int a, b, c;
      |         ^
Exp-3.cpp:18:21: error: ‘int myTest::b’ is private within this context
   18 |     cin >> v.a >> v.b >> v.c;
      |                     ^
Exp-3.cpp:6:12: note: declared private here
    6 |     int a, b, c;
      |            ^
Exp-3.cpp:18:28: error: ‘int myTest::c’ is private within this context
   18 |     cin >> v.a >> v.b >> v.c;
      |                            ^
Exp-3.cpp:6:15: note: declared private here
    6 |     int a, b, c;
      |               ^

[Process exited 0]
```

Figure 3: Exp-3 error log



```
5 8 19
5 8 19

[Process exited 0]
```

Figure 4: Output of Exp-3.cpp After fixing error

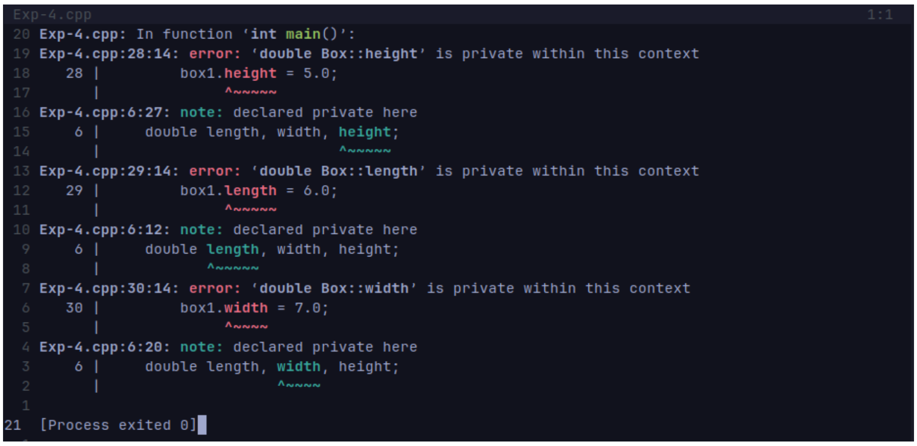
Example 4

Write a C++ program to understand public and private access of class members.

Code

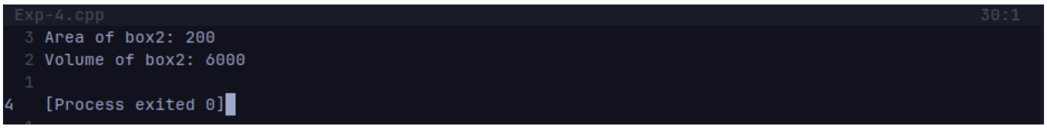
```cpp
#include <iostream>
using namespace std;

class Box {
private:
    double length, width, height;

public:
    void init (double length, double width, double height) {
        this->length = length;
        this->width = width;
        this->height = height;
    }

    double calculateArea () {
        return length * width;
    }
    double calculateVolume () {
        return length * width * height;
    }
};

int main ()
{
    Box box1;
    Box box2;

  // box1.height = 5.0;
  // box1.length = 6.0;
  // box1.width = 7.0;

    box2.init(10, 20, 30);

    cout << "Area of box2: " << box2.calculateArea() << endl;
    cout << "Volume of box2: " << box2.calculateVolume() << endl;

    return 0;
}
```

Output



```
Exp-4.cpp: In function ‘int main()’:
Exp-4.cpp:28:14: error: ‘double Box::height’ is private within this context
   28 |         box1.height = 5.0;
      |              ^~~~~~
Exp-4.cpp:6:27: note: declared private here
    6 |     double length, width, height;
      |                           ^~~~~~
Exp-4.cpp:29:14: error: ‘double Box::length’ is private within this context
   29 |         box1.length = 6.0;
      |              ^~~~~~
Exp-4.cpp:6:12: note: declared private here
    6 |     double length, width, height;
      |            ^~~~~~
Exp-4.cpp:30:14: error: ‘double Box::width’ is private within this context
   30 |         box1.width = 7.0;
      |              ^~~~~
Exp-4.cpp:6:20: note: declared private here
    6 |     double length, width, height;
      |                   ^~~~~

[Process exited 0]
```

Figure 5: Exp-4 error log



```
Area of box2: 200
Volume of box2: 6000

[Process exited 0]
```

Figure 6: Output of Exp-4.cpp After fixing error

Lab Task

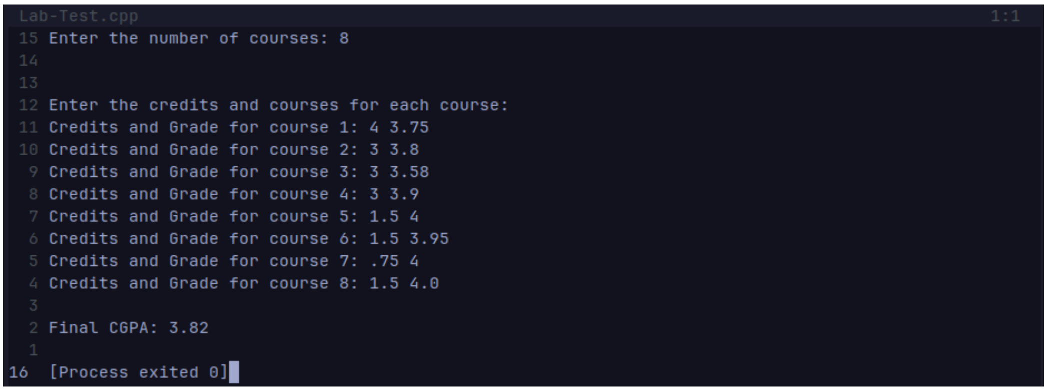
Write a C++ program to calculate the CGPA of a student using class and objects.

Code

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;

class Student {
private:
    int n;
    vector<double> credits, grades;
    double cgpa;

public:
    Student() {
        cgpa = 0;
    }

    void take_inputs () {
        cout << "Enter the number of courses: "; cin >> n;

        credits.resize(n);
        grades.resize(n);

        cout << "\n\nEnter the credits and courses for each course:" << endl;
        for(int i=0 ; i<n ; i++) {
            cout << "Credits and Grade for course " << i+1 << ": ";
            cin >> credits[i] >> grades[i];
        }
    }
    void calculate_CGPA() {
        double total_credits=0, total_grade=0;

```

```
        for(int i=0 ; i<n ; i++)
            total_credits += credits[i], total_grade += credits[i] * grades[i];

        cgpa = total_grade / total_credits;
    }
    double get_cgpa() {
        return cgpa;
    }
};

int main ()
{
    Student st;

    st.take_inputs();
    st.calculate_CGPA();

    printf("\nFinal CGPA: %.2lf\n", st.get_cgpa());

    return 0;
}
```

Output



```
Enter the number of courses: 8


Enter the credits and courses for each course:
Credits and Grade for course 1: 4 3.75
Credits and Grade for course 2: 3 3.8
Credits and Grade for course 3: 3 3.58
Credits and Grade for course 4: 3 3.9
Credits and Grade for course 5: 1.5 4
Credits and Grade for course 6: 1.5 3.95
Credits and Grade for course 7: .75 4
Credits and Grade for course 8: 1.5 4.0

Final CGPA: 3.82

[Process exited 0]
```

Figure 7: Output of CGPA Calculator

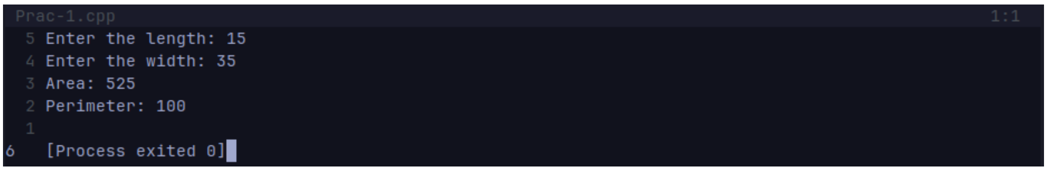
Practice Exercise 1

Write a class having two private variables and one member function which will return the area and parimeter of the rectangle.

Code

```
#include <bits/stdc++.h>
using namespace std;

class Rectangle {
private:
    double length, width;

public:
    void area_perimeter () {
        cout << "Enter the length: "; cin >> length;
        cout << "Enter the width: "; cin >> width;

        cout << "Area: " << length * width << endl;
        cout << "Perimeter: " << 2 * (length + width) << endl;
    }
};

int main ()
{
    Rectangle r;

    r.area_perimeter();

    return 0;
}
```

Output



```
Enter the length: 15
Enter the width: 35
Area: 525
Perimeter: 100

[Process exited 0]
```

Figure 8: Output of Prac-1.cpp

Practice Exercise 2

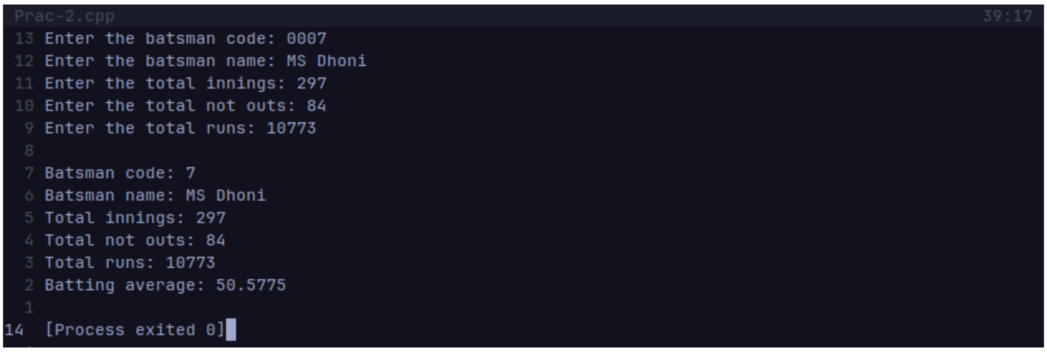
Write a C++ program to define a class `batsman` with the following specifications:

- Private members:
  - `batsman_code`: 4 digits code number
  - `batsman_name` 20 characters (string)
  - `total_innings`, `notout_innings`, `total_runs`: integer type
  - `calcavg()`: Function to compute batavg
- Public members:
  - `readdata()`: Function to accept value from batsman code, batsman name, to- tal innings, notout innings, total runs and invoke the function calcavg().
  - `displaydata()`: Function to display the data members on the screen.

Code

```
#include <bits/stdc++.h>
using namespace std;

class batsman {
private:
    short code;
    string name;
    int tot_innings, not_outs, tot_runs;
    double batavg;

    void calcavg () {
        batavg = (double) tot_runs / (tot_innings - not_outs);
    }

public:
    void readdata () {
        cout << "Enter the batsman code: "; cin >> code;
        cout << "Enter the batsman name: "; getline(cin >> ws, name);
        cout << "Enter the total innings: "; cin >> tot_innings;
        cout << "Enter the total not outs: "; cin >> not_outs;
        cout << "Enter the total runs: "; cin >> tot_runs;
        calcavg();
    }

    void displaydata () {
        cout << "Batsman code: " << code << endl;
        cout << "Batsman name: " << name << endl;
        cout << "Total innings: " << tot_innings << endl;
        cout << "Total not outs: " << not_outs << endl;
        cout << "Total runs: " << tot_runs << endl;
        cout << "Batting average: " << batavg << endl;
    }
};

int main ()
{
    batsman b;

    b.readdata();
    cout << endl;
    b.displaydata();

    return 0;
}
```

Output



```
Enter the batsman code: 0007
Enter the batsman name: MS Dhoni
Enter the total innings: 297
Enter the total not outs: 84
Enter the total runs: 10773

Batsman code: 7
Batsman name: MS Dhoni
Total innings: 297
Total not outs: 84
Total runs: 10773
Batting average: 50.5775

[Process exited 0]
```

Figure 9: Output of Prac-2.cpp

Discussion

- In this lab, basic concepts of class and objects in C++ were discussed.
- The visibility of data members and member functions were discussed.
- When a private member is accessed from outside the class, it generates an error. The errors can be fixed by making the member public, or by using public member functions.