COP4020 Programming Languages C++

Edelis Molina

October 2022

Contents

1 Read Me

All the source code is listed in the document and it is driven by the ‘main.cpp‘ file. Some exercises have a header and implementation files pair. These files have the same name, but the header file has a ‘.h‘ extension and the implementation file has a ‘.cpp‘ extension.

2 main.cpp

```
#include <iostream>
#include "student.h"
#include "batsman.h"
#include "test.h"
#include "flight.h"
#include "book.h"
#include "report.h"
#include "rectangle.h"
#include "complex.h"
#include "distance.h"
#include "time.h"

using namespace std;

int main()
{
    cout << endl
         << "========== Exercise 1 ==========" << endl;
    Student myStudent;
    myStudent.takeData(35, "Lindsay", 3, 4, 5);
    myStudent.showData();

    cout << endl
         << "========== Exercise 2 ==========" << endl;
    Batsman myBatsman;
    myBatsman.readData(1234, "Otani", 6, 4, 3);
    myBatsman.displayData();

    cout << endl
         << "========== Exercise 3 ==========" << endl;
    Test myTest;
    myTest.SCHEDULE();
    myTest.DISPTEST();

    cout << endl
         << "========== Exercise 4 ==========" << endl;
```

```
Flight myFlight;
myFlight.FEEDINFO();
myFlight.SHOWINFO();

cout << endl
     << "========== Exercise 5 ==========" << endl;
Book myBook;
myBook.INPUT(13, "Love in the time of cholera", 12.5);
myBook.PURCHASE();

cout << endl
     << "========== Exercise 6 ==========" << endl;
Report rep;
float mark[5] = {5.0, 4.2, 3.3, 3.1, 4.5};
rep.READINFO(5465, "Edel", mark);
rep.DISPLAYINFO();

cout << endl
     << "========== Exercise 7 ==========" << endl;
Rectangle r1, r2;
r1.setlength(5);
r1.setwidth(2.5);
r2.setlength(5);
r2.setwidth(18.9);
cout << endl;

cout << "First Rectangle: " << endl;
r1.show();
cout << "Area: " << r1.area() << " Perimeter: " << r1.perimeter() << endl;

cout << "Second Rectangle: " << endl;
r2.show();
cout << "Area: " << r2.area() << " Perimeter: " << r2.perimeter() << endl;

cout << endl;
string result = (r1.sameArea(r2) == 1) ? "the same" : "different";
cout << "The area of the rectangles is " << result << endl;

cout << endl;
cout << "Exercise 7 Second Part" << endl;
cout << "First Rectangle: " << endl;
r1.setlength(15);
r1.setwidth(6.3);
r1.show();
cout << "Area: " << r1.area() << " Perimeter: " << r1.perimeter() << endl;
```

```
cout << "Second Rectangle: " << endl;
r2.show();
cout << "Area: " << r2.area() << " Perimeter: " << r2.perimeter() << endl;

cout << endl;
result = (r1.sameArea(r2) == 1) ? "the same" : "different";
cout << "The area of the new rectangles is " << result << endl;

cout << endl
     << "========== Exercise 8 ==========" << endl;
Complex c1, c2, c3;
c1.set(4, 2);
c2.set(5, 1);
c3 = c1.sum(c2);

cout << "Complex number 1:" << endl;
c1.disp();
cout << "Complex number 2:" << endl;
c2.disp();
cout << "Complex number 3 = Complex 1 + Complex 2:" << endl;
c3.disp();

cout << endl
     << "========== Exercise 9 ==========" << endl;
Distance d1, d2, d3;
d1.set(5, 4);
d2.set(9, 20);
d3 = d1.add(d2);

cout << "Distance 1:" << endl;
d1.disp();
cout << "Distance 2:" << endl;
d2.disp();
cout << "Distance 3 = Distance 1 + Distance 2:" << endl;
d3.disp();

cout << endl
     << "========== Exercise 10 ==========" << endl;
Time t1, t2, t3;
t1.settime(10, 50);
t2.settime(15, 40);
t3 = t1.sum(t2);

cout << "Time 1:" << endl;
t1.showtime();
cout << "Time 2:" << endl;
```

```
    t2.showtime();
    cout << "Time 3 = Time 1 + Time 2:" << endl;
    t3.showtime();

    return 0;
}
```

3 Exercise 1

3.1 student.h

```
class Student
{
private:
    int admno;
    char sname[20];
    float eng, math, science;
    float total;
    float ctotal();

public:
    void takeData(int admno, const char sname[], float eng, float math, float science);
    void showData();
};
```

3.2 student.cpp

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

void Student::takeData(int admno, const char sname[], float eng, float math, float science)
{
    this->admno = admno;
    strcpy(this->sname, sname);
    this->eng = eng;
    this->math = math;
    this->science = science;
    this->total = ctotal();
}

void Student::showData()
{
    cout << "Student admno  : " << admno << endl;
    cout << "Student name   : " << sname << endl;
```

```
    cout << "Student eng    : " << eng << endl;
    cout << "Student math   : " << math << endl;
    cout << "Student science: " << science << endl;
    cout << "Student total  : " << total << endl;
}

float Student::ctotal()
{
    return eng + math + science;
}
```

4 Exercise 2

4.1 batsman.h

```
class Batsman
{
private:
    int bcode;
    char bname[20];
    int innings, notout, runs;
    double batavg;
    double calcavg();

public:
    void readData(int bcode, const char bname[], int innings, int notout, int runs);
    void displayData();
};
```

4.2 batsman.cpp

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

void Batsman::readData(int bcode, const char bname[], int innings, int notout, int runs)
{
    this->bcode = bcode;
    strcpy(this->bname, bname);
    this->innings = innings;
    this->notout = notout;
    this->runs = runs;
    this->batavg = calcavg();
}
```

```
void Batsman::displayData()
{
    cout << "Batsman bcode  : " << bcode << endl;
    cout << "Batsman name   : " << bname << endl;
    cout << "Batsman innings: " << innings << endl;
    cout << "Batsman notout : " << notout << endl;
    cout << "Batsman runs   : " << runs << endl;
    cout << "Batsman batavg : " << batavg << endl;
}

double Batsman::calcavg()
{
    return (runs / (double)(innings - notout));
}
```

5 Exercise 3

5.1 test.h

```
#include <string>
class Test
{
private:
    int testCode;
    std::string description;
    int noCandidate;
    int centerReqd;
    double CALCNTR();

public:
    void SCHEDULE();
    void DISPTEST();
};
```

5.2 test.cpp

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

void Test::SCHEDULE()
{
    cout << "Enter Test Code: ";
```

```
    cin >> this->testCode;
    cout << "Enter Test Description: ";
    cin >> this->description;
    cout << "Enter Test No. Candidate:   ";
    cin >> this->noCandidate;

    centerReqd = CALCNTR();
}

void Test::DISPTEST()
{
    cout << "Test Code:             " << testCode << endl;
    cout << "Test Description:      " << description << endl;
    cout << "Test No. Candidate:    " << noCandidate << endl;
    cout << "Test Center Required: " << centerReqd << endl;
}

double Test::CALCNTR()
{
    return (noCandidate / (100 + 1));
}
```

6 Exercise 4

6.1 flight.h

```
#include <string>
class Flight
{
private:
    int flightNo;
    std::string destination;
    float distance, fuel;
    float CALCFUEL();

public:
    void FEEDINFO();
    void SHOWINFO();
};
```

6.2 flight.cpp

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

```
void Flight::FEEDINFO()
{
    cout << "Enter Flight Number: ";
    cin >> this->flightNo;
    cout << "Enter Flight Destination: ";
    cin >> this->destination;
    cout << "Enter Flight Distance:   ";
    cin >> this->distance;

    fuel = CALCFUEL();
}

void Flight::SHOWINFO()
{
    cout << "Flight Number: " << flightNo << endl;
    cout << "Flight Destination: " << destination << endl;
    cout << "Flight Distance: " << distance << endl;
    cout << "Flight Fuel: " << fuel << endl;
}

float Flight::CALCFUEL()
{
    if (distance <= 100)
    {
        return 500;
    }
    else if (distance <= 200)
    {
        return 1100;
    }
    else
    {
        return 2200;
    }
}
```

7 Exercise 5

7.1 book.h

```
#include <string>
class Book
{
private:
```

```
    int bookNo;
    std::string bookTitle;
    float price;
    float TOTAL_COST(int N);

public:
    void INPUT(int bNo, std::string title, float price);
    void PURCHASE();
};
```

7.2 book.cpp

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

void Book::INPUT(int bNo, std::string title, float price)
{
    this->bookNo = bNo;
    this->bookTitle = title;
    this->price = price;
}

void Book::PURCHASE()
{
    int NoCopies;
    float totalCost;

    cout << "Enter the number of copies of " << bookTitle << " to be purchased ";
    cin >> NoCopies;

    totalCost = Book::TOTAL_COST(NoCopies);
    cout << "Total cost of " << NoCopies << " copies of " << bookTitle << " " << totalCost <
}

float Book::TOTAL_COST(int N)
{
    return N * price;
}
```

8 Exercise 6

8.1 report.h

```
class Report
{
private:
    int adno;
    char name[20];
    float marks[5];
    float average;
    float GETAVG();

public:
    void READINFO(int adno, const char name[], float marks[]);
    void DISPLAYINFO();
};
```

8.2 report.cpp

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

void Report::READINFO(int adno, const char name[], float marks[])
{
    this->adno = adno;
    strcpy(this->name, name);
    for (int i = 0; i < 5; i++)
    {
        this->marks[i] = marks[i];
    }
    average = GETAVG();
}

void Report::DISPLAYINFO()
{
    cout << "Admision No.: " << adno << endl;
    cout << "Name: " << name << endl;
    for (int i = 0; i < 5; i++)
    {
        cout << "Mark " << i + 1 << " = " << marks[i + 1] << endl;
    }
    cout << "Average: " << average << endl;
}
```

```
float Report::GETAVG()
{
    int total = 0;
    for (int i = 0; i < 5; i++)
    {
        total += marks[i];
    }
    return total / 5;
}
```

9 Exercise 7

9.1 rectangle.h

```
#include <iostream>
using namespace std;
class Rectangle
{
private:
    float length, width;

public:
    void setlength(float lenght)
    {
        this->length = lenght;
    }

    void setwidth(float width)
    {
        this->width = width;
    }

    float perimeter()
    {
        return 2 * (length + width);
    }

    float area()
    {
        return length * width;
    }

    void show()
    {
        cout << "Length: " << length << endl;
```

```
        cout << "Width: " << width << endl;
    }

    int sameArea(Rectangle r)
    {
        if (area() == r.area())
        {
            return 1;
        }
        return 0;
    }
};
```

10 Exercise 8

10.1 complex.h

```
#include <iostream>
using namespace std;
class Complex
{
private:
    float real;
    float imaginary;

public:
    void set(float r, float img)
    {
        real = r;
        imaginary = img;
    }

    void disp()
    {
        cout << real << " + i" << imaginary << endl;
    }

    Complex sum(Complex c)
    {
        Complex temp;
        temp.real = real + c.real;
        temp.imaginary = imaginary + c.imaginary;

        return temp;
```

```
    }
};
```

11 Exercise 9

11.1 distance.h

```
#include <iostream>
using namespace std;
class Distance
{
private:
    int feet;
    float inches;

public:
    void set(int f, float in)
    {
        feet = f;
        inches = in;
    }

    void disp()
    {
        cout << feet << " ft " << inches << " inches " << endl;
    }

    Distance add(Distance d)
    {
        Distance temp;
        temp.feet = feet + d.feet;
        temp.inches = inches + d.inches;

        // convert inches to feet if greater than 12
        while (temp.inches >= 12.0)
        {
            temp.inches = temp.inches - 12.0;
            ++temp.feet;
        }

        return temp;
    }
};
```

12 Exercise 10

12.1 time.h

```
#include <iostream>
using namespace std;
class Time
{
private:
    int hours, minutes;

public:
    void settime(int h, int min)
    {
        hours = h;
        minutes = min;
    }

    void showtime()
    {
        cout << hours << " hours and " << minutes << " minutes" << endl;
    }

    Time sum(Time t)
    {
        Time temp;
        temp.hours = hours + t.hours;
        temp.minutes = minutes + t.minutes;

        // convert min to hours format if greater than 60
        while (temp.minutes > 60.0)
        {
            temp.minutes = temp.minutes - 60.0;
            temp.hours++;
        }
        // convert hours to 12 hours format if greater than 12
        while (temp.hours > 12.0)
        {
            temp.hours = temp.hours - 12.0;
        }

        return temp;
    }
};
```