San Francisco Bay University

CS360L - Programming in C and C++ Lab
Lab Assignment #2

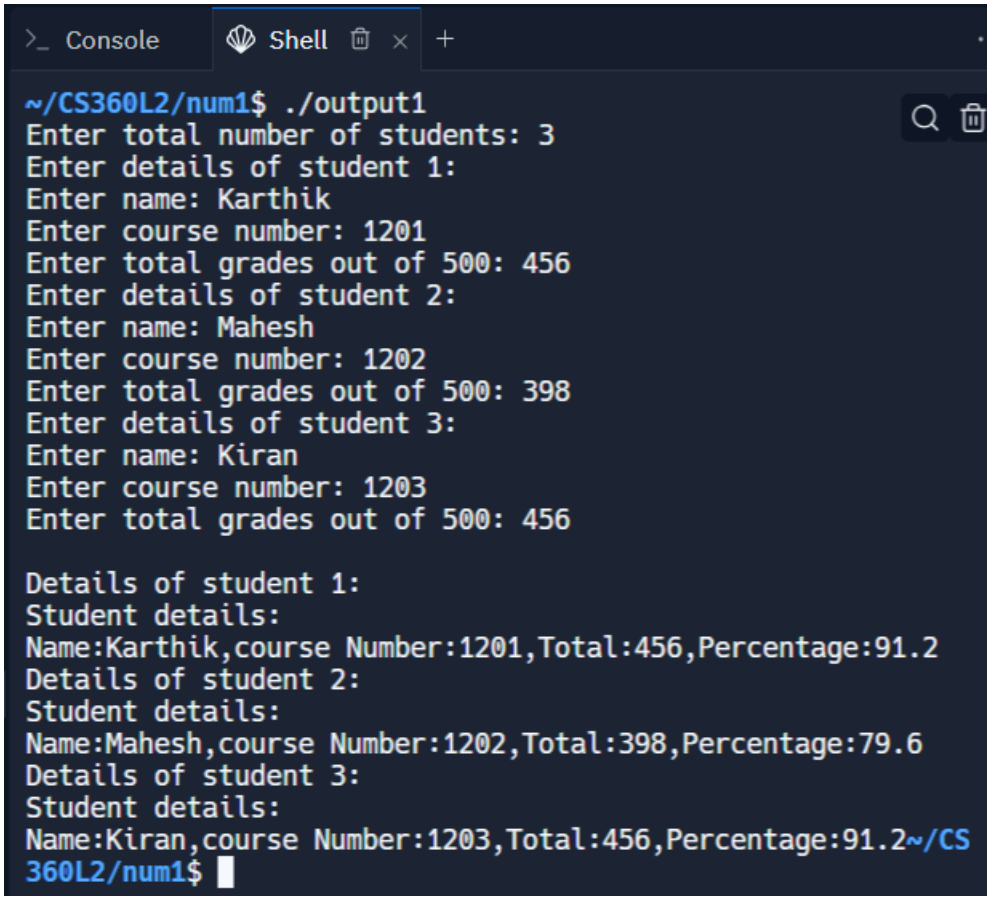
Due day: 2/27/2024

Kritika Regmi
ID:19702

1. Given a *`student`* class with the members and methods as follows, write a C++ test program (a.k.a. main function) to display *names*, *courseNum* and grades of *3* students who have appeared in the examination. Declare the class of *name*, *courseNum*. and *grade*. Create an array of class objects. Read and display the contents of the array.

ANSWER:

OUTPUT:



CODE:

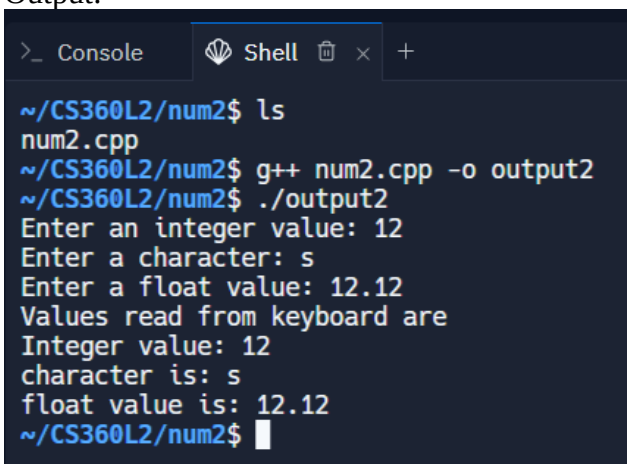
num1 › main.cpp › main

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

#define MAX 10
class student {
private:
    char name[30];
    int courseNum;
    int total;
    float perc;

public:
    void getDetails(void) {
        cout << "Enter name: ";
        cin >> name;
        cout << "Enter course number: ";
        cin >> courseNum;
        cout << "Enter total grades out of 500: ";
        cin >> total;
        perc = (float)total / 500 * 100;
    }

    void putDetails(void) {
        cout << "Student details:\n";
        cout << "Name:" << name << ",course Number:" << courseNum << ",Total:" << total << ",Percentage:" << perc;
    }
};

int main(void) {
    student students[MAX];
    int n;

    cout << "Enter total number of students: ";
    cin >> n;

    for (int i = 0; i < n; ++i) {
        cout << "\n Enter details of student " << i + 1 << ":\n";
        students[i].getDetails();
    }

    for (int i = 0; i < n; ++i) {
        cout << "\n Details of student " << i + 1 << ":\n";
        students[i].putDetails();
    }

    return 0;
}
```

2. Complete two methods, *get_data()* and *print_data()* within the given class *sample* based on the running results shown below.

ANSWER:

Output:



CODE:

main.cpp num2.cpp

num2 > num2.cpp > ...

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

class sample {
private:
    int a;
    char b;
    float c;

public:
    void get_data() {
        cout << "Enter an integer value: ";
        cin >> a;
        cout << "Enter a character: ";
        cin >> b;
        cout << "Enter a float value: ";
        cin >> c;
    }

    void print_data() {
        cout << "Values read from keyboard are\n";
        cout << "Integer value: " << a << endl;
        cout << "character is: " << b << endl;
        cout << "float value is: " << c << endl;
    }
};
```

```
int main(void) {
    sample s;
    s.get_data();
    s.print_data();

    return 0;
}
```

3. Write a class called *Rectangle* that has floating point data members' *length* and *width*. The class has the following member functions: *void setLength(float)* to set the *length* data member; *void setwidth(float)*to set the *width* data member; *float perimeter(void)*to calculate and return the perimeter of the rectangle; *float area(void)*to calculate and return the area of the rectangle; *void show*(*void*)to display the *length* and *width* of the rectangle; *int sameArea(Rectangle)*that has one parameter of type *Rectangle*, and *sameArea* returns *1* if the two *Rectangles* have the same area, otherwiese returns *0* if they don't.

a. Create *Rectangle* class first.

ANS:

num3 > num3.cpp > Rectangle > ...

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

class Rectangle {
private:
  float length;
  float width;

public:
  void setLength(float len) { length = len; }
  void setWidth(float wid) { width = wid; }
  float perimeter() { return 2 * (length + width); }
  float area() { return length * width; }
  void show() {
    cout << "Length: " << length << endl;
    cout << "Width: " << width << endl;
  }
  int sameArea(Rectangle other) { return area() == other.area(); }
};

```

b. Write *main* function to create two *rectangle* objects. Set the *length* and *width* of the first rectangle to *5* and *2.5*, and set the *length* and *width* of the second rectangle to *5* and *18.9*. Display each *rectangle* and its area and perimeter.

ANS:

```
21 int main() {
22     Rectangle rect1, rect2;
23
24     rect1.setLength(5);
25     rect1.setWidth(2.5);
26     cout << "Rectangle 1:" << endl;
27     rect1.show();
28     cout << "Area: " << rect1.area() << endl;
29     cout << "Perimeter: " << rect1.perimeter() << endl;
30
31     rect2.setLength(5);
32     rect2.setWidth(18.9);
33     cout << "\nRectangle 2:" << endl;
34     rect2.show();
35     cout << "Area: " << rect2.area() << endl;
36     cout << "Perimeter: " << rect2.perimeter() << endl;
37 }
```

Console | Shell

```
~/CS360L2/num3$ g++ num3.cpp -o output3
~/CS360L2/num3$ ./output3
Rectangle 1:
Length: 5
Width: 2.5
Area: 12.5
Perimeter: 15

Rectangle 2:
Length: 5
Width: 18.9
Area: 94.5
Perimeter: 47.8

Checking if the two rectangles have the same area...
The two rectangles do not have the same area.
~/CS360L2/num3$
```

c. Check whether the two *Rectangles* have the same area and print a message indicating the result. Set the *length* and *width* of the first *rectangle* to *15* and *6.3*. Display each *Rectangle* and its area and perimeter again. Again, verify whether the two *Rectangles* have the same area and print a message indicating the result.

ANS:

```cpp
int main() {
    Rectangle rect1, rect2;

    rect1.setLength(15);
    rect1.setWidth(6.3);
    cout << "Rectangle 1:" << endl;
    rect1.show();
    cout << "Area: " << rect1.area() << endl;
    cout << "Perimeter: " << rect1.perimeter() << endl;

    rect2.setLength(5);
    rect2.setWidth(18.9);
    cout << "\nRectangle 2:" << endl;
    rect2.show();
    cout << "Area: " << rect2.area() << endl;
    cout << "Perimeter: " << rect2.perimeter() << endl;

    cout << "\nChecking if the two rectangles have the same area..." << endl;
    if (rect1.sameArea(rect2))
        cout << "The two rectangles have the same area." << endl;
    else
        cout << "The two rectangles do not have the same area." << endl;

    return 0;
}
```

```
Console  Shell

~/CS360L2/num3$ g++ num3.cpp -o output3
~/CS360L2/num3$ ./output3
Rectangle 1:
Length: 15
Width: 6.3
Area: 94.5
Perimeter: 42.6

Rectangle 2:
Length: 5
Width: 18.9
Area: 94.5
Perimeter: 47.8

Checking if the two rectangles have the same area...
The two rectangles have the same area.
~/CS360L2/num3$
```

4. Create a class called *MusicIns* to contain three methods *void string(void), void wind(void)* and *void perc(void)*. Each of these methods should initialize a member *string* type *instrument* array to contain the following
    a. *Veena, guitar, sitar, sarod* and *mandolin* under *void string(void)* method
    b. *Flute, clarinet, saxophone, nadaswaram* and *piccolo* under *void wind(void)* method
    c. *Table, mridangam, bongos, drums* and *tambour* under *void perc(void)* method

    It should also have two methods called *void get(void)* and *void show(void)* to display the contents of the arrays initialized. The *void get(void)* methods must display a menu as follows
    a. The values of *instrument* array within *void string(void)* method
    b. The values of *instrument* array within *void wind(void)* method
    c. The values of *instrument* array within *void perc(void)* method

    After that, generate test program *main.cpp* to verify the above class.
    ANS:

num4 > num4.cpp > MusicIns > get

```cpp
#include <iostream>
#include <string>

using namespace std;

class MusicIns {
private:
    string string_instruments[5];
    string wind_instruments[5];
    string perc_instruments[5];

public:
    void setStringInstruments() {
        string_instruments[0] = "Veena";
        string_instruments[1] = "Guitar";
        string_instruments[2] = "Sitar";
        string_instruments[3] = "Sarod";
        string_instruments[4] = "Mandolin";
    }

    void setWindInstruments() {
        wind_instruments[0] = "Flute";
        wind_instruments[1] = "Clarinet";
        wind_instruments[2] = "Saxophone";
        wind_instruments[3] = "Nadaswaram";
        wind_instruments[4] = "Piccolo";
    }
```

```
29    void setPercInstruments() {
30        perc_instruments[0] = "Table";
31        perc_instruments[1] = "Mridangam";
32        perc_instruments[2] = "Bongos";
33        perc_instruments[3] = "Drums";
34        perc_instruments[4] = "Tambour";
35    }
36
37    void get() {
38        char choice;
39        cout << "a. Instrument array within void setStringInstruments() method" << endl;
40        cout << "b. Instrument array within void setWindInstruments() method" << endl;
41        cout << "c. Instrument array within void setPercInstruments() method" << endl;
42        cout << "Enter your choice (a/b/c): ";
43        cin >> choice;
44
45        switch(choice) {
46            case 'a':
47                cout << "String Instruments:" << endl;
48                for (const auto& instrument : string_instruments) {
49                    cout << instrument << endl;
50                }
51                break;
52            case 'b':
53                cout << "Wind Instruments:" << endl;
54                for (const auto& instrument : wind_instruments) {
55                    cout << instrument << endl;
56                }
57                break;
58            case 'c':
59                cout << "Percussion Instruments:" << endl;
60                for (const auto& instrument : perc_instruments) {
61                    cout << instrument << endl;
62                }
63                break;
64            default:
65                cout << "Invalid choice!" << endl;
66        }
67    }
68
69    void show() {
70        cout << "String Instruments:" << endl;
71        for (const auto& instrument : string_instruments) {
72            cout << instrument << endl;
73        }
74
75        cout << "Wind Instruments:" << endl;
76        for (const auto& instrument : wind_instruments) {
77            cout << instrument << endl;
78        }
79
```

```
80          cout << "Percussion Instruments:" << endl;
81          for (const auto& instrument : perc_instruments) {
82              cout << instrument << endl;
83          }
84      }
85  };
86
87  int main() {
88      MusicIns musicInstruments;
89
90      musicInstruments.setStringInstruments();
91      musicInstruments.setWindInstruments();
92      musicInstruments.setPercInstruments();
93      cout << "Displaying all instruments:" << endl;
94      musicInstruments.show();
95      cout << "\nGet information about instruments:" << endl;
96      musicInstruments.get();
97      return 0;
98  }
```

OUTPUT:

```
Console   Shell

~/CS360L2/num4$ g++ num4.cpp -o output4
~/CS360L2/num4$ ./output4
Displaying all instruments:
String Instruments:
Veena
Guitar
Sitar
Sarod
Mandolin
Wind Instruments:
Flute
Clarinet
Saxophone
Nadaswaram
Piccolo
Percussion Instruments:
Table
Mridangam
Bongos
Drums
Tambour

Get information about instruments:
a. Instrument array within void setStringInstruments() method
b. Instrument array within void setWindInstruments() method
c. Instrument array within void setPercInstruments() method
Enter your choice (a/b/c): a
String Instruments:
Veena
Guitar
Sitar
Sarod
Mandolin
~/CS360L2/num4$
```