Question no.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:

The code:

```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); //member function to get student's details
    void putDetails(void); //member function to print student's details
};

void student::getDetails(void) //member function definition, outside of the class
{
    cout << "Enter name: ";
    cin >> name;
    cout << "Enter course number: ";
    cin >> courseNum;
    cout << "Enter total marks out of 500: ";
    cin >> total;
    perc = (float)total / 500 * 100;
}

void student::putDetails(void) //member function definition, outside of the class
```

Q1.cpp Format

```
of the class
{
    cout << "Student details:\n";
    cout << "Name: " << name << ", Course Number: " << courseNum <<
        ", Total: " << total << ", Percentage: " << perc << endl;
}

int main(void) {
    int numStudents;
    cout << "Enter total number of students: ";
    cin >> numStudents;

    student students[MAX]; // Array of class objects

    for (int i = 0; i < numStudents; ++i) {
        cout << "Enter details of student " << i + 1 << ":" << endl;
        students[i].getDetails(); // Get details for each student
    }

    // Display details of all students
    for (int i = 0; i < numStudents; ++i) {
        cout << "Details of student " << i + 1 << ":" << endl;
        students[i].putDetails(); // Display details for each student
    }

    return 0;
}
```

Generate ⌘ I

The output:

```
~/lab2l$ ./Q1.out
Enter total number of students: 5
Enter details of student 1:
Enter name: yunisha
Enter course number: 1230
Enter total marks out of 500: 400
Enter details of student 2:
Enter name: kritika
Enter course number: 1234
Enter total marks out of 500: 350
Enter details of student 3:
Enter name: swek
Enter course number: 1231
Enter total marks out of 500: 50
Enter details of student 4:
Enter name: kalu
Enter course number: 1232
Enter total marks out of 500: 460
Enter details of student 5:
Enter name: nisha
Enter course number: 1234
Enter total marks out of 500: 345
Details of student 1:
Student details:
Name: yunisha, Course Number: 1230, Total: 400, Percentage: 80
Details of student 2:
Student details:
Name: kritika, Course Number: 1234, Total: 350, Percentage: 70
Details of student 3:
Student details:
Name: swek, Course Number: 1231, Total: 50, Percentage: 10
Details of student 4:
Student details:
Name: kalu, Course Number: 1232, Total: 460, Percentage: 92
Details of student 5:
Student details:
Name: nisha, Course Number: 1234, Total: 345, Percentage: 69
~/lab2l$
```

Question no.2

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

⇨ The answer:

The code:

Q2.cpp

```cpp
#include<iostream>
using std::cout;
using std::cin;
using std::endl;

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" << endl;
        cout << "Integer value: " << a << endl;
        cout << "Character is: " << b << endl;
        cout << "Float value is: " << c << endl;
    }
};

int main(void) {
    sample s;
```

```
30      sample s;
31      s.get_data();
32      s.print_data();
33      return 0;
34  }
35
```

The output:

```
~/lab2l$ touch Q2.cpp
~/lab2l$ g++ Q2.cpp -o Q2.out
~/lab2l$ ./Q2.out
Enter an integer value: 12
Enter a character: S
Enter a float value: 12.12
Values read from keyboard are
Integer value: 12
Character is: S
Float value is: 12.12
~/lab2l$
```

Question no.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 thewidth 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 same area returns 1 if the two Rectangles have the same area, otherwise returns 0 if they don't.

a) Create Rectangle class first.
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.
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.

⇨ The answer:
The code:

Q3.cpp > ... Forma

```cpp
#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 << ", Width: " << width << endl;
    }

    int sameArea(Rectangle other) {
```

Q3.cpp > ... Format

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

int main() {
    Rectangle rect1, rect2;

    // Set dimensions for the first rectangle
    rect1.setLength(5);
    rect1.setWidth(2.5);

    // Set dimensions for the second rectangle
    rect2.setLength(5);
    rect2.setWidth(18.9);

    // Display details of the first rectangle
    cout << "Rectangle 1:" << endl;
    rect1.show();
    cout << "Area: " << rect1.area() << endl;
    cout << "Perimeter: " << rect1.perimeter() << endl;

    // Display details of the second rectangle
    cout << "\nRectangle 2:" << endl;
    rect2.show();
    cout << "Area: " << rect2.area() << endl;
    cout << "Perimeter: " << rect2.perimeter() << endl;
```


Q3.cpp
...
Format
58 cout << "Area: " << rect2.area() << endl;
59 cout << "Perimeter: " << rect2.perimeter() << endl;
60
61 // Check if both rectangles have the same area
62 if (rect1.sameArea(rect2))
63 cout << "\nBoth rectangles have the same area." << endl;
64 else
65 cout << "\nThe rectangles have different areas." << endl;
66
67 // Set new dimensions for the first rectangle
68 rect1.setLength(15);
69 rect1.setWidth(6.3);
70
71 // Display details of the first rectangle with new dimensions
72 cout << "\nRectangle 1 with new dimensions:" << endl;
73 rect1.show();
74 cout << "Area: " << rect1.area() << endl;
75 cout << "Perimeter: " << rect1.perimeter() << endl;
76
77 // Check if both rectangles have the same area after updating dimensions
78 if (rect1.sameArea(rect2))
79 cout << "\nNow, both rectangles have the same area." << endl;
80 else
81 cout << "\nNow, the rectangles have different areas." << endl;
82
83 return 0;
84 }
85
Generate ⌘ I
AI
C++
Ln 85, Col 1
Spaces: 2
History

The output:

```
~/lab2l$ g++ Q3.cpp -o Q3.out
~/lab2l$ ./Q3.out
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

The rectangles have different areas.

Rectangle 1 with new dimensions:
Length: 15, Width: 6.3
Area: 94.5
Perimeter: 42.6

Now, both rectangles have the same area.
~/lab2l$
```

Question no.4

Create a class calledMusicIns 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 the instrument array within the void string(void) method.
b) The values of instrument array within void wind(void) method.
c) The values of the instrument array within the void perc(void) method

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

⇨ The answer:

The code:

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

class MusicIns {
private:
    string instruments[5];

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

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

    void initPercussionInstruments() {
        instruments[0] = "Table";
        instruments[1] = "Mridangam";
        instruments[2] = "Bongos";
        instruments[3] = "Drums";
```

Q4.cpp > ...

```
30          instruments[3] = "Drums";
31          instruments[4] = "Tambour";
32      }
33
34      void get() {
35          cout << "Select an option:" << endl;
36          cout << "a. Instruments in string method" << endl;
37          cout << "b. Instruments in wind method" << endl;
38          cout << "c. Instruments in perc method" << endl;
39
40          char choice;
41          cin >> choice;
42
43          switch (choice) {
44              case 'a':
45                  cout << "Instruments in string method:" << endl;
46                  for (int i = 0; i < 5; ++i) {
47                      cout << instruments[i] << endl;
48                  }
49                  break;
50              case 'b':
51                  cout << "Instruments in wind method:" << endl;
52                  for (int i = 0; i < 5; ++i) {
53                      cout << instruments[i] << endl;
54                  }
55                  break;
56              case 'c':
57                  cout << "Instruments in perc method:" << endl;
58                  for (int i = 0; i < 5; ++i) {
59                      cout << instruments[i] << endl;
```


Q4.cpp
cout << "Instruments in perc method:" << endl;
for (int i = 0; i < 5; ++i) {
cout << instruments[i] << endl;
}
break;
default:
cout << "Invalid choice." << endl;
return;
}
}
void show() {
for (int i = 0; i < 5; ++i) {
cout << instruments[i] << endl;
}
}
};
int main() {
MusicIns music;
music.initStringInstruments();
music.get();
return 0;
}
Generate

The output:

```
~/lab2l$ ./Q4.out
Select an option:
a. Instruments in string method
b. Instruments in wind method
c. Instruments in perc method
a
Instruments in string method:
Veena
Guitar
Sitar
Sarod
Mandolin
~/lab2l$
```