Tasmita Tanjim
Student ID 19723

Replit link: https://replit.com/@TASMITA-TANJIMT/CSLAB21

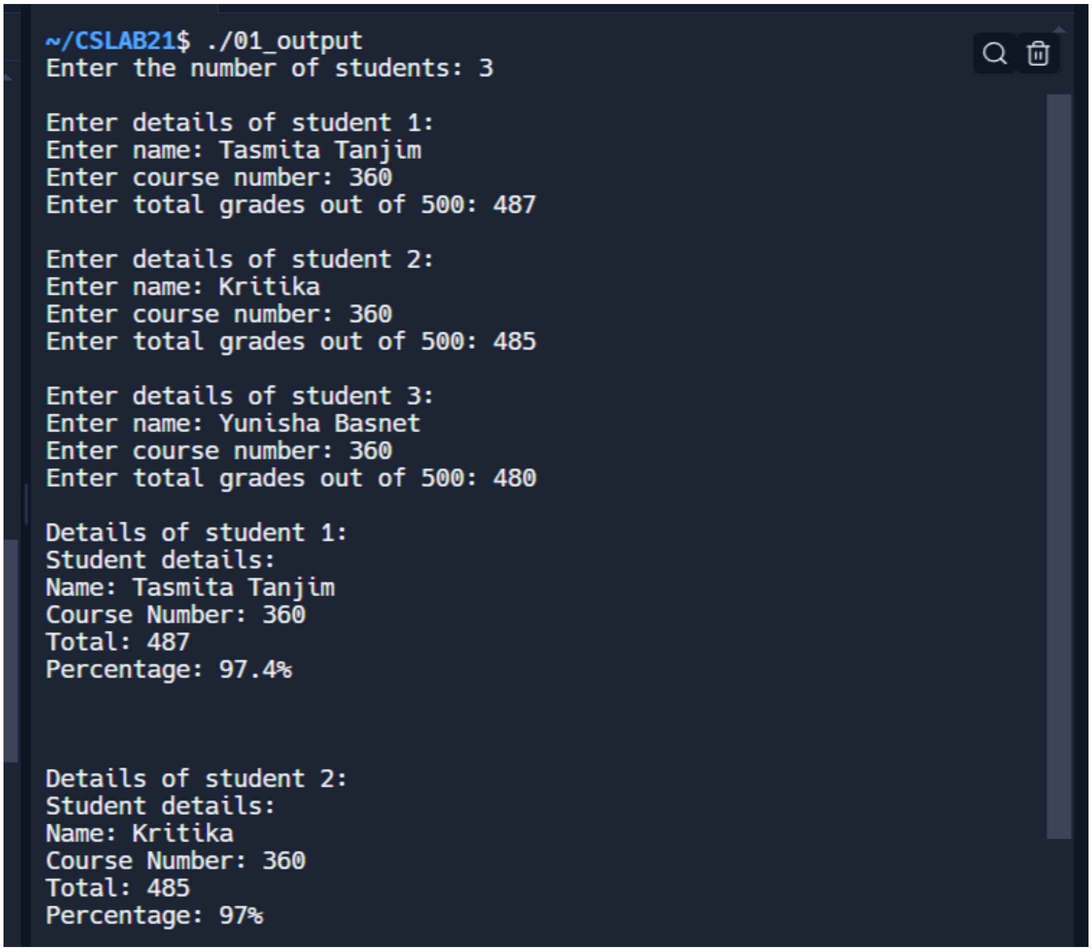
Github link: https://github.com/tasmita0131/cslab2_01_tasmita_19723

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.**

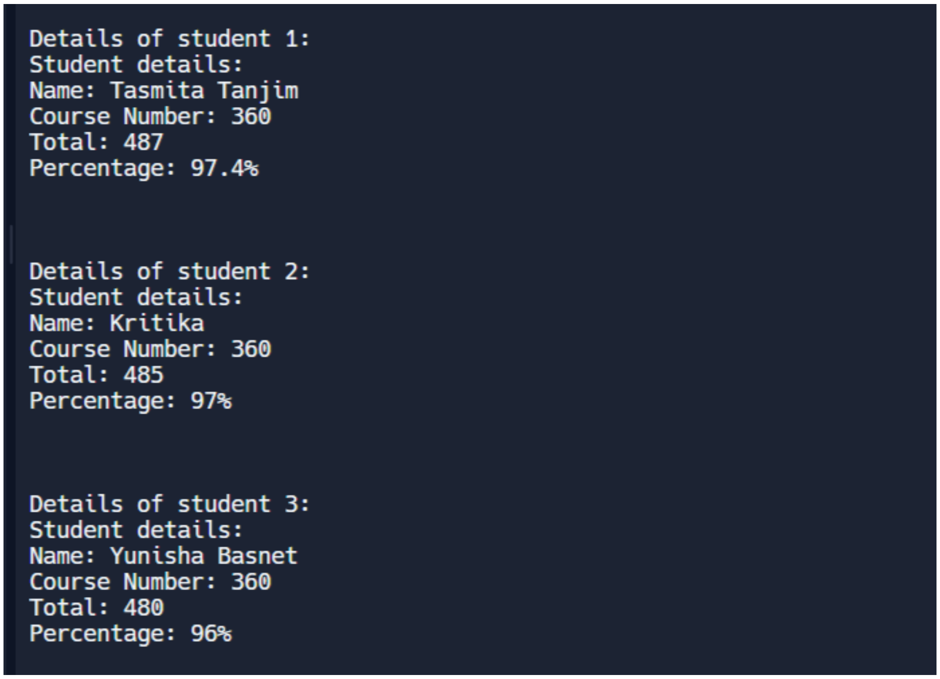
ANSWER:

```
#include <iostream>
#include <string>
#include <limits>

using namespace std;

#define MAX 10

class student {
private:
  string name;
  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) {
  cin.ignore(numeric_limits<streamsize>::max(), '\n'); // Clears the input buffer, crucial before
reading the name
  cout << "Enter name: ";
  getline(cin, name);  // Reads the full name, including spaces

  cout << "Enter course number: ";
  while(!(cin >> courseNum)){ // Loops until a valid integer is entered
    cin.clear(); // Clears the error state of cin
    cin.ignore(numeric_limits<streamsize>::max(), '\n'); // Ignores the rest of the incorrect input
    cout << "Invalid input. Please enter a valid course number: ";
    }

    cout << "Enter total grades out of 500: ";
    while(!(cin >> total)){ // Loops until a valid integer is entered
        cin.clear(); // Clears the error state of cin
        cin.ignore(numeric_limits<streamsize>::max(), '\n'); // Ignores the rest of the incorrect input
        cout << "Invalid input. Please enter total grades out of 500: ";
    }

    perc = static_cast<float>(total) / 500 * 100; // Calculates percentage
}

void student::putDetails(void) {
    cout << "Student details:\n";
    cout << "Name: " << name << "\nCourse Number: " << courseNum << "\nTotal: " << total <<
"\nPercentage: " << perc << "%" << endl;
    cout<<"\n \n \n";
}

int main(void) {
    student std[MAX];  // Array of student objects
    int numberOfStudents;

    cout << "Enter the number of students: ";
    cin >> numberOfStudents;

    if (numberOfStudents > 0 && numberOfStudents <= MAX) {
        for (int i = 0; i < numberOfStudents; i++) {
            cout << "\nEnter details of student " << i + 1 << ":\n";
            std[i].getDetails();
        }

        cout << endl;  // Formatting output

        for (int i = 0; i < numberOfStudents; i++) {
            cout << "Details of student " << i + 1 << ":\n";
            std[i].putDetails();
        }
    } else {
        cout << "The number of students must be between 1 and " << MAX << ".\n";
    }

    return 0;
}
```

OUTPUT:



```
~/CSLAB21$ ./01_output
Enter the number of students: 3

Enter details of student 1:
Enter name: Tasmita Tanjim
Enter course number: 360
Enter total grades out of 500: 487

Enter details of student 2:
Enter name: Kritika
Enter course number: 360
Enter total grades out of 500: 485

Enter details of student 3:
Enter name: Yunisha Basnet
Enter course number: 360
Enter total grades out of 500: 480

Details of student 1:
Student details:
Name: Tasmita Tanjim
Course Number: 360
Total: 487
Percentage: 97.4%



Details of student 2:
Student details:
Name: Kritika
Course Number: 360
Total: 485
Percentage: 97%
```

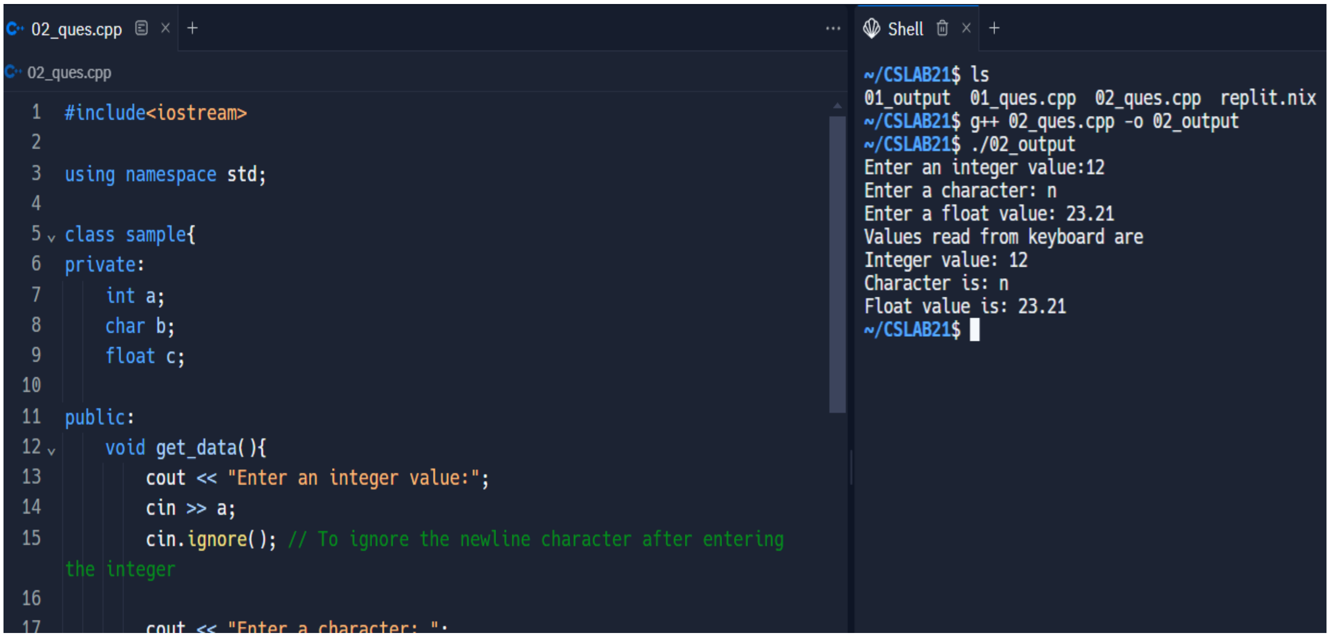


```
Details of student 1:
Student details:
Name: Tasmita Tanjim
Course Number: 360
Total: 487
Percentage: 97.4%



Details of student 2:
Student details:
Name: Kritika
Course Number: 360
Total: 485
Percentage: 97%



Details of student 3:
Student details:
Name: Yunisha Basnet
Course Number: 360
Total: 480
Percentage: 96%
```

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

CODE ANSWER:

```
#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;
      cin.ignore(); // To ignore the newline character after entering the integer

      cout << "Enter a character: ";
      cin >> b;
      cin.ignore(); // To ignore the newline character after entering the character

      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;
   s.get_data();
   s.print_data();

   return 0;
}
```



```
#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;
        cin.ignore( ); // To ignore the newline character after entering the integer

        cout << "Enter a character: ";
```

```
~/CSLAB21$ ls
01_output  01_ques.cpp  02_ques.cpp  replit.nix
~/CSLAB21$ g++ 02_ques.cpp -o 02_output
~/CSLAB21$ ./02_output
Enter an integer value:12
Enter a character: n
Enter a float value: 23.21
Values read from keyboard are
Integer value: 12
Character is: n
Float value is: 23.21
~/CSLAB21$ 
```

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

<u>CODE:</u>

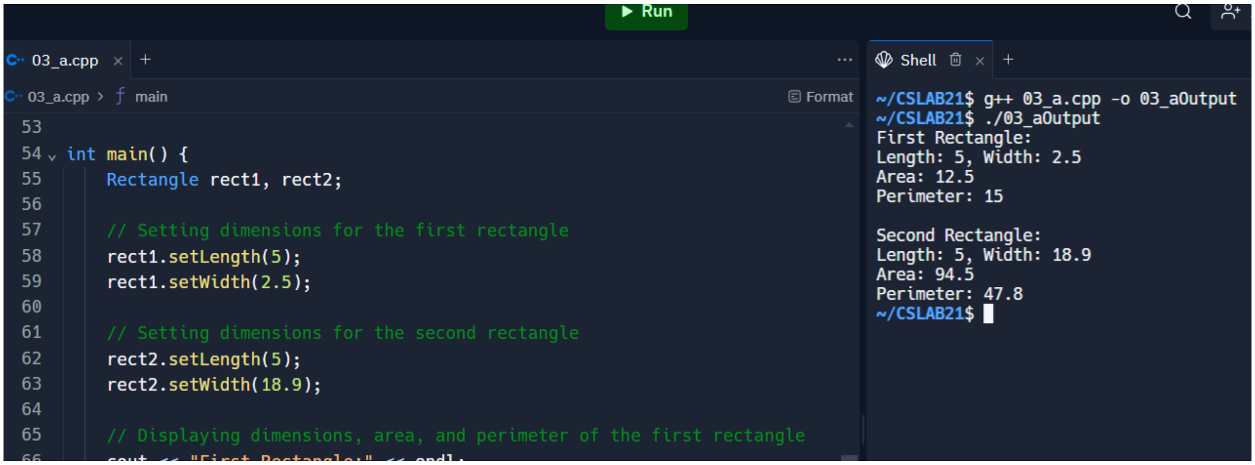
```
#include <iostream>
using namespace std;

class Rectangle {
private:
  float length;
  float width;

public:
  // Default constructor
  Rectangle() : length(0.0f), width(0.0f) {}

  // Function to set the length of the rectangle
  void setLength(float l) {
    if (l > 0) {  // Ensure the length is positive
      length = l;
    } else {
            length = 0;  // Set to default if negative
        }
    }

    // Function to set the width of the rectangle
    void setWidth(float w) {
        if (w > 0) {  // Ensure the width is positive
            width = w;
        } else {
            width = 0;  // Set to default if negative
        }
    }

    // Function to calculate and return the perimeter of the rectangle
    float perimeter() const {
        return 2 * (length + width);
    }

    // Function to calculate and return the area of the rectangle
    float area() const {
        return length * width;
    }

    // Function to display the length and width
    void show() const {
        cout << "Length: " << length << ", Width: " << width << endl;
    }

    // Function to compare the area of this rectangle with another rectangle
    // Returns 1 if areas are the same, otherwise returns 0
    int sameArea(const Rectangle& other) const {
        return (this->area() == other.area()) ? 1 : 0;
    }
};
```

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.**

CODE:

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

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

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

  // Displaying dimensions, area, and perimeter of the first rectangle
  cout << "First Rectangle:" << endl;
  rect1.show();
  cout << "Area: " << rect1.area() << endl;
  cout << "Perimeter: " << rect1.perimeter() << endl << endl;

  // Displaying dimensions, area, and perimeter of the second rectangle
  cout << "Second Rectangle:" << endl;
  rect2.show();
  cout << "Area: " << rect2.area() << endl;
  cout << "Perimeter: " << rect2.perimeter() << endl;

  return 0;
}
```



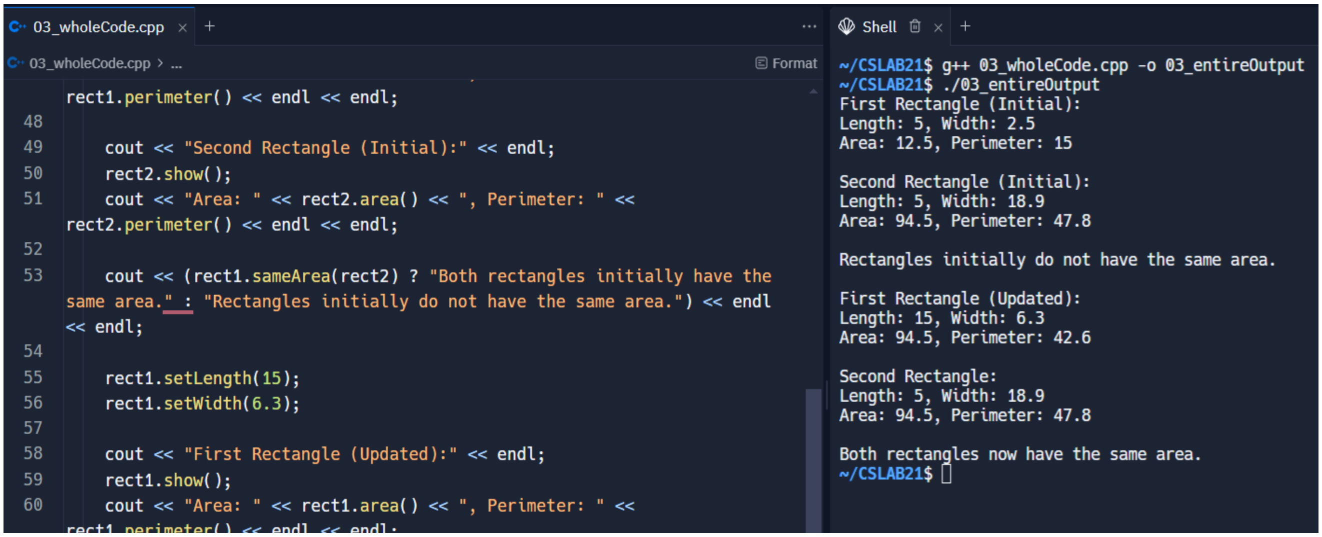
```
int main() {
    Rectangle rect1, rect2;

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

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

    // Displaying dimensions, area, and perimeter of the first rectangle
    cout << "First Rectangle:" << endl;
```

```
~/CSLAB21$ g++ 03_a.cpp -o 03_aOutput
~/CSLAB21$ ./03_aOutput
First Rectangle:
Length: 5, Width: 2.5
Area: 12.5
Perimeter: 15

Second Rectangle:
Length: 5, Width: 18.9
Area: 94.5
Perimeter: 47.8
~/CSLAB21$
```

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**

CODE:

```
int main() {
  Rectangle rect1, rect2;
  rect1.setLength(5);
  rect1.setWidth(2.5);
  rect2.setLength(5);
  rect2.setWidth(18.9);

  cout << "First Rectangle (Initial):" << endl;
  rect1.show();
  cout << "Area: " << rect1.area() << ", Perimeter: " << rect1.perimeter() << endl << endl;

  cout << "Second Rectangle (Initial):" << endl;
  rect2.show();
  cout << "Area: " << rect2.area() << ", Perimeter: " << rect2.perimeter() << endl << endl;

  cout << (rect1.sameArea(rect2) ? "Both rectangles initially have the same area." : "Rectangles
initially do not have the same area.") << endl << endl;

  rect1.setLength(15);
  rect1.setWidth(6.3);

  cout << "First Rectangle (Updated):" << endl;
  rect1.show();
  cout << "Area: " << rect1.area() << ", Perimeter: " << rect1.perimeter() << endl << endl;

  cout << "Second Rectangle:" << endl;
  rect2.show();
  cout << "Area: " << rect2.area() << ", Perimeter: " << rect2.perimeter() << endl << endl;

  cout << (rect1.sameArea(rect2) ? "Both rectangles now have the same area." : "Rectangles now
do not have the same area.") << endl;

  return 0;
}
```



```
rect1.perimeter() << endl << endl;

    cout << "Second Rectangle (Initial):" << endl;
    rect2.show();
    cout << "Area: " << rect2.area() << ", Perimeter: " <<
rect2.perimeter() << endl << endl;

    cout << (rect1.sameArea(rect2) ? "Both rectangles initially have the
same area." : "Rectangles initially do not have the same area.") << endl
<< endl;

    rect1.setLength(15);
    rect1.setWidth(6.3);

    cout << "First Rectangle (Updated):" << endl;
    rect1.show();
    cout << "Area: " << rect1.area() << ", Perimeter: " <<
rect1.perimeter() << endl << endl;
```

```
~/CSLAB21$ g++ 03_wholeCode.cpp -o 03_entireOutput
~/CSLAB21$ ./03_entireOutput
First Rectangle (Initial):
Length: 5, Width: 2.5
Area: 12.5, Perimeter: 15

Second Rectangle (Initial):
Length: 5, Width: 18.9
Area: 94.5, Perimeter: 47.8

Rectangles initially do not have the same area.

First Rectangle (Updated):
Length: 15, Width: 6.3
Area: 94.5, Perimeter: 42.6

Second Rectangle:
Length: 5, Width: 18.9
Area: 94.5, Perimeter: 47.8

Both rectangles now have the same area.
~/CSLAB21$ 
```

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

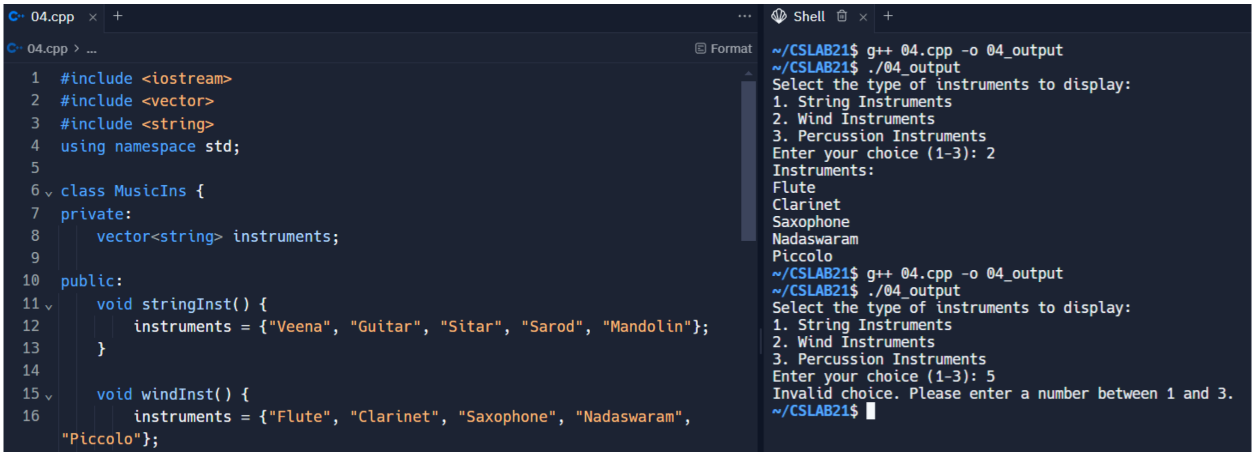
It should also have two methods called void get(void) and void show(void) to display the contents of the arrays initialized. The void get(void) methods must display a menu as follows

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.

CODE:

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

class MusicIns {
private:
  vector<string> instruments;

public:
  void stringInst() {
    instruments = {"Veena", "Guitar", "Sitar", "Sarod", "Mandolin"};
  }

  void windInst() {
    instruments = {"Flute", "Clarinet", "Saxophone", "Nadaswaram", "Piccolo"};
  }

  void percInst() {
    instruments = {"Tabla", "Mridangam", "Bongos", "Drums", "Tambour"};
  }

  void get() {
    cout << "Select the type of instruments to display:" << endl;
    cout << "1. String Instruments" << endl;
    cout << "2. Wind Instruments" << endl;
    cout << "3. Percussion Instruments" << endl;
    cout << "Enter your choice (1-3): ";
  }

  void show(int choice) {
    switch (choice) {
      case 1:
        stringInst();
        break;
      case 2:
        windInst();
        break;
      case 3:
        percInst();
        break;
      default:
        cout << "Invalid choice. Please enter a number between 1 and 3." << endl;
        return;
    }

    cout << "Instruments: " << endl;
    for (const auto& instrument : instruments) {
      cout << instrument << endl;
    }
  }
};

int main() {
  MusicIns musicInstruments;
  int choice;

  musicInstruments.get();
  cin >> choice;
  musicInstruments.show(choice);

  return 0;
}
```



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

class MusicIns {
private:
    vector<string> instruments;

public:
    void stringInst() {
        instruments = {"Veena", "Guitar", "Sitar", "Sarod", "Mandolin"};
    }

    void windInst() {
        instruments = {"Flute", "Clarinet", "Saxophone", "Nadaswaram",
"Piccolo"};
```

```
~/CSLAB21$ g++ 04.cpp -o 04_output
~/CSLAB21$ ./04_output
Select the type of instruments to display:
1. String Instruments
2. Wind Instruments
3. Percussion Instruments
Enter your choice (1-3): 2
Instruments:
Flute
Clarinet
Saxophone
Nadaswaram
Piccolo
~/CSLAB21$ g++ 04.cpp -o 04_output
~/CSLAB21$ ./04_output
Select the type of instruments to display:
1. String Instruments
2. Wind Instruments
3. Percussion Instruments
Enter your choice (1-3): 5
Invalid choice. Please enter a number between 1 and 3.
~/CSLAB21$
```