

Programming in C and C++ Lab (CS360 LAB)

ABU BAKAR SIDDIK

ID: 19897, Assignment- 1

Q1) Let's examine / run the following C++ program regarding string data type and related operators.

Ans:

```
#include <iostream>

using std::cout;
using std::endl;
using std::string;

const char SEMI_COLON = ';';

const string VERB1 = "went up ";
const string VERB2 = "down came ";
const string VERB3 = "washed ";
const string VERB4 = "out came ";
const string VERB5 = "dried up ";

int main(void){
    string firstLine;
    string secondLine;
    string thirdLine;
    string fourthLine;

    firstLine = "The itsy bitsy spider " + VERB1 + "the water spout";
    secondLine = VERB2 + "the rain and " + VERB3 + "the spider out";
    thirdLine = VERB4 + "the sun and " + VERB5 + "all the rain";
    fourthLine = "and the itsy bitsy spider " + VERB1 + "the spout again";
```

```

cout << firstLine << SEMI_COLON << endl;

cout << secondLine << SEMI_COLON << endl;

cout << thirdLine << SEMI_COLON;

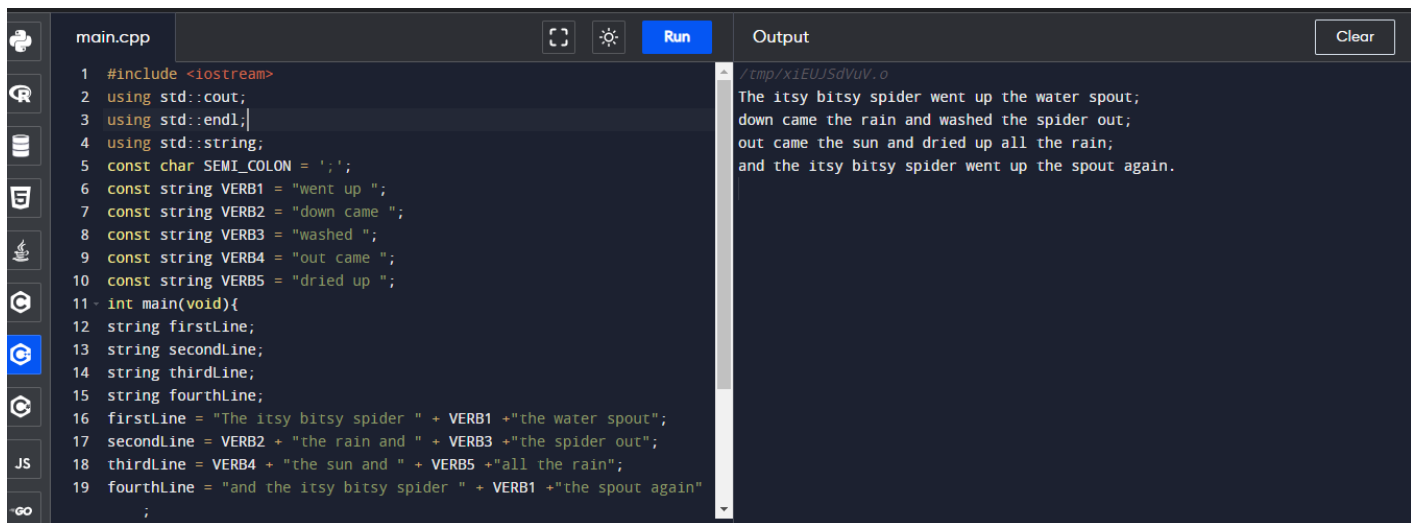
cout << endl;

cout << fourthLine << '.' << endl;

return 0;

}

```



The screenshot shows a C++ IDE with a file named `main.cpp`. The code defines a semicolon as a literal constant and uses it to format the output of a poem. The output window shows the result of running the program.

```

main.cpp
1 #include <iostream>
2 using std::cout;
3 using std::endl;
4 using std::string;
5 const char SEMI_COLON = ';';
6 const string VERB1 = "went up ";
7 const string VERB2 = "down came ";
8 const string VERB3 = "washed ";
9 const string VERB4 = "out came ";
10 const string VERB5 = "dried up ";
11 int main(void){
12     string firstLine;
13     string secondLine;
14     string thirdLine;
15     string fourthLine;
16     firstLine = "The itsy bitsy spider " + VERB1 + "the water spout";
17     secondLine = VERB2 + "the rain and " + VERB3 + "the spider out";
18     thirdLine = VERB4 + "the sun and " + VERB5 + "all the rain";
19     fourthLine = "and the itsy bitsy spider " + VERB1 + "the spout again"
    ;
}

```

Output

```

/tmp/x1EUJSdVuV.o
The itsy bitsy spider went up the water spout;
down came the rain and washed the spider out;
out came the sun and dried up all the rain;
and the itsy bitsy spider went up the spout again.

```

Q2) Focuses on constructing output statements. Program Shell is the outline of a program. Use this shell for Question#1 through #3.

a) Question#1: Write a program to read-in from keyboard and print the following information single spaced on the screen. Use literal constants in the output statements for each of the data items to be written on the screen. Run your program to verify that the output is as specified.

- i. your name (last name, comma, blank, first name)**
- ii. today's date (month:day:year)**

Ans:

```
// Program Shell for Question #1
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

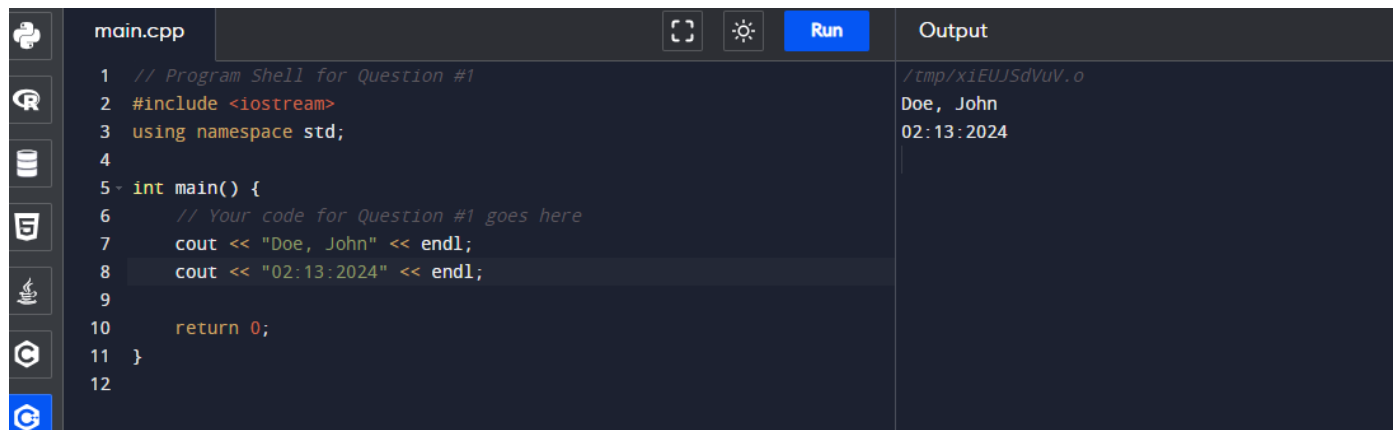
```
    // Your code for Question #1 goes here
```

```
    cout << "Doe, John" << endl;
```

```
    cout << "02:13:2024" << endl;
```

```
    return 0;
```

```
}
```



The screenshot shows a C++ IDE with a file named `main.cpp`. The code in the editor is as follows:

```
1 // Program Shell for Question #1
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6     // Your code for Question #1 goes here
7     cout << "Doe, John" << endl;
8     cout << "02:13:2024" << endl;
9
10    return 0;
11 }
12
```

The IDE has a toolbar with icons for file operations, a 'Run' button, and a settings icon. To the right of the code editor is an 'Output' window showing the execution results:

```
/tmp/xiEUISdVuV.o
Doe, John
02:13:2024
```

b) Question#2: Change your program so that there is a space between the two lines of output.

Ans:

```
// Program Shell for Question #2
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

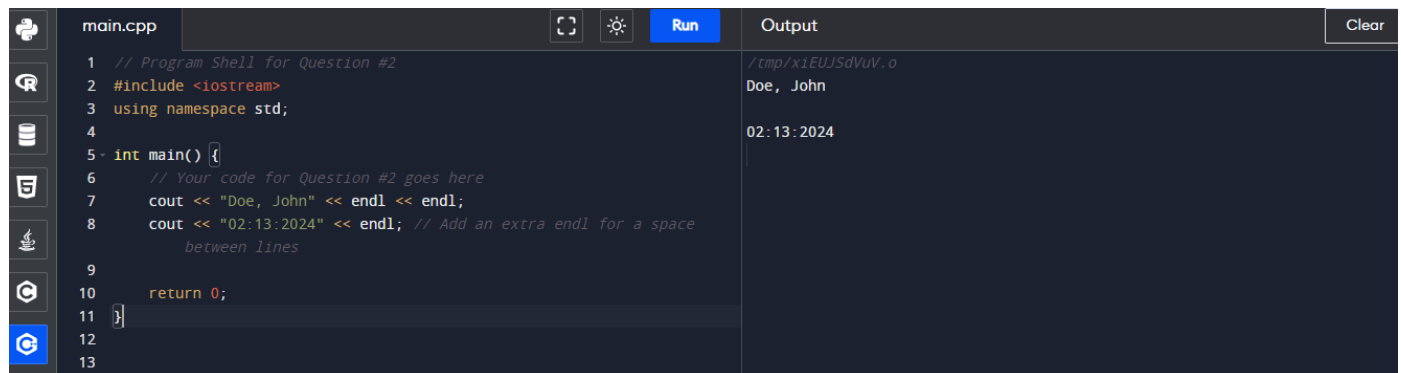
```
    // Your code for Question #2 goes here
```

```
cout << "Doe, John" << endl << endl;
```

```
cout << "02:13:2024" << endl; // Add an extra endl for a space between lines
```

```
return 0;
```

```
}
```



```
main.cpp
1 // Program Shell for Question #2
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6     // Your code for Question #2 goes here
7     cout << "Doe, John" << endl << endl;
8     cout << "02:13:2024" << endl; // Add an extra endl for a space
        between lines
9
10    return 0;
11 }
12
13
```

Output

```
/tmp/x1EUJSdVuV.o
Doe, John

02:13:2024
```

c) Question#3: Change your program so that your first name is printed followed by your last name, with a blank in between the names.

Ans:

```
// Program Shell for Question #3
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

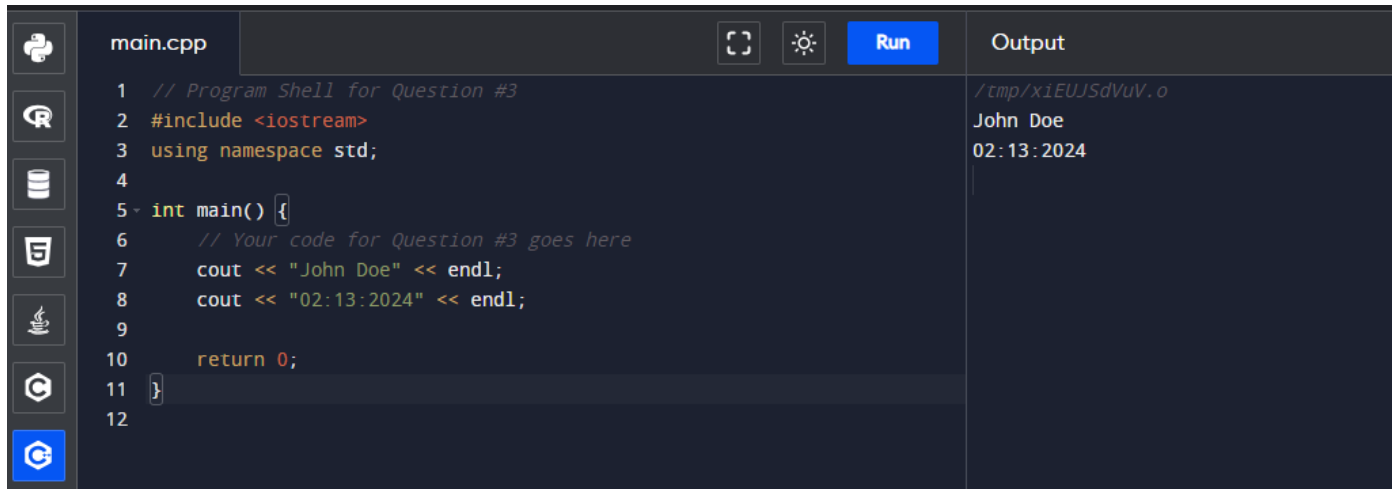
```
    // Your code for Question #3 goes here
```

```
    cout << "John Doe" << endl;
```

```
    cout << "02:13:2024" << endl;
```

```
    return 0;
```

```
}
```



The screenshot shows a C++ IDE with a file named `main.cpp`. The code is as follows:

```
1 // Program Shell for Question #3
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6     // Your code for Question #3 goes here
7     cout << "John Doe" << endl;
8     cout << "02:13:2024" << endl;
9
10    return 0;
11 }
12
```

The IDE has a sidebar on the left with icons for Python, R, SQL, Java, C#, and C++. The top right has a `Run` button. The output window on the right shows the following output:

```
/tmp/x1EUJSdVuV.o
John Doe
02:13:2024
```

Q3) Use the following program shell for Question#1 through #3.

a) Question#1: Write a named string constant made up of your first and last names with a blank in between. Write the statements to print out the result of applying length and size to your named constant object. Compile and run your program.

Ans:

```
// Program Strings applies string functions.
```

```
#include <iostream>
```

```
using std::cout, std::string;
```

```
int main(void) {
```

```
    // Your code for Question #1 goes here
```

```
    const string fullName = "Doe John";
```

```
    cout << "Length: " << fullName.length() << std::endl;
```

```
    cout << "Size: " << fullName.size() << std::endl;
```

```
    return 0;
```

```
}
```



The screenshot shows a C++ IDE with a file named `main.cpp`. The code defines a `main` function that prints the length and size of the string `"Doe John"`. The output window shows the results: `Length: 8` and `Size: 8`.

```
1 // Program Strings applies string functions.
2 #include <iostream>
3 using std::cout, std::string;
4
5 int main(void) {
6     // Your code for Question #1 goes here
7     const string fullName = "Doe John";
8     cout << "Length: " << fullName.length() << std::endl;
9     cout << "Size: " << fullName.size() << std::endl;
10
11     return 0;
12 }
13 |
```

Output: `/tmp/x1EUJSdVuV.o`
`Length: 8`
`Size: 8`

b) Question#2: Add statements to your Question#1 program to print your name formatted as last name first, followed by a comma and your first name. Use function `substr` to accomplish this task. Compile and run your program.

Ans:

```
#include <iostream>
```

```
using std::cout, std::string;
```

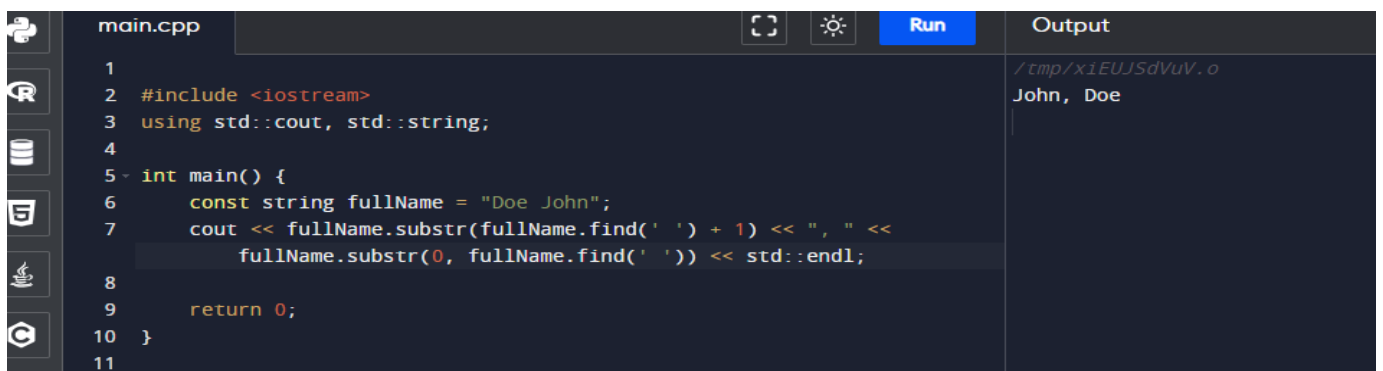
```
int main() {
```

```
    const string fullName = "Doe John";
```

```
    cout << fullName.substr(fullName.find(' ') + 1) << ", " << fullName.substr(0, fullName.find(' '))
    << std::endl;
```

```
    return 0;
```

```
}
```



The screenshot shows the same C++ IDE with the updated code. The program now prints the name in reverse order: `John, Doe`.

```
1
2 #include <iostream>
3 using std::cout, std::string;
4
5 int main() {
6     const string fullName = "Doe John";
7     cout << fullName.substr(fullName.find(' ') + 1) << ", " <<
        fullName.substr(0, fullName.find(' ')) << std::endl;
8
9     return 0;
10 }
11
```

Output: `/tmp/x1EUJSdVuV.o`
`John, Doe`

c) Question#3: Add the statements necessary to print your last name, followed by a comma and your first initial. Compile and run your program.

Ans:

```
#include <iostream>
```

```
using std::cout, std::string;
```

```
int main() {
```

```
    const string fullName = "Doe John";
```

```
    cout << fullName.substr(0, fullName.find(' ')) << ", " << fullName[fullName.find(' ') + 1] << "."  
<< std::endl;
```

```
    return 0;
```

```
}
```



The screenshot shows a C++ IDE with a file named 'main.cpp'. The code is as follows:

```
1 #include <iostream>
2 using std::cout, std::string;
3
4 int main() {
5     const string fullName = "Doe John";
6     cout << fullName.substr(0, fullName.find(' ')) << ", " <<
        fullName[fullName.find(' ') + 1] << "." << std::endl;
7
8     return 0;
9 }
10
```

The IDE has a 'Run' button and an 'Output' panel. The output panel shows the result of running the program:

```
/tmp/xiEUJSdVuV.o
Doe, J.
```

Q4) Use the following program shell for Question#1 through Question#4.

a. Question#1: Write a program to print the following numbers right-justified in a column on the screen. Make the values named constants. 1066 1492 512 1 -23

Ans:

```
#include <iostream>
```

```
#include <iomanip>
```

```
using std::cout, std::fixed, std::showpoint;
```

```
int main() {
```

```
    cout << fixed << showpoint;
```

```
    const int num1 = 1066, num2 = 1492, num3 = 512, num4 = 1, num5 = -23;
```

```
    cout << std::setw(8) << num1 << std::endl;
```

```
    cout << std::setw(8) << num2 << std::endl;
```

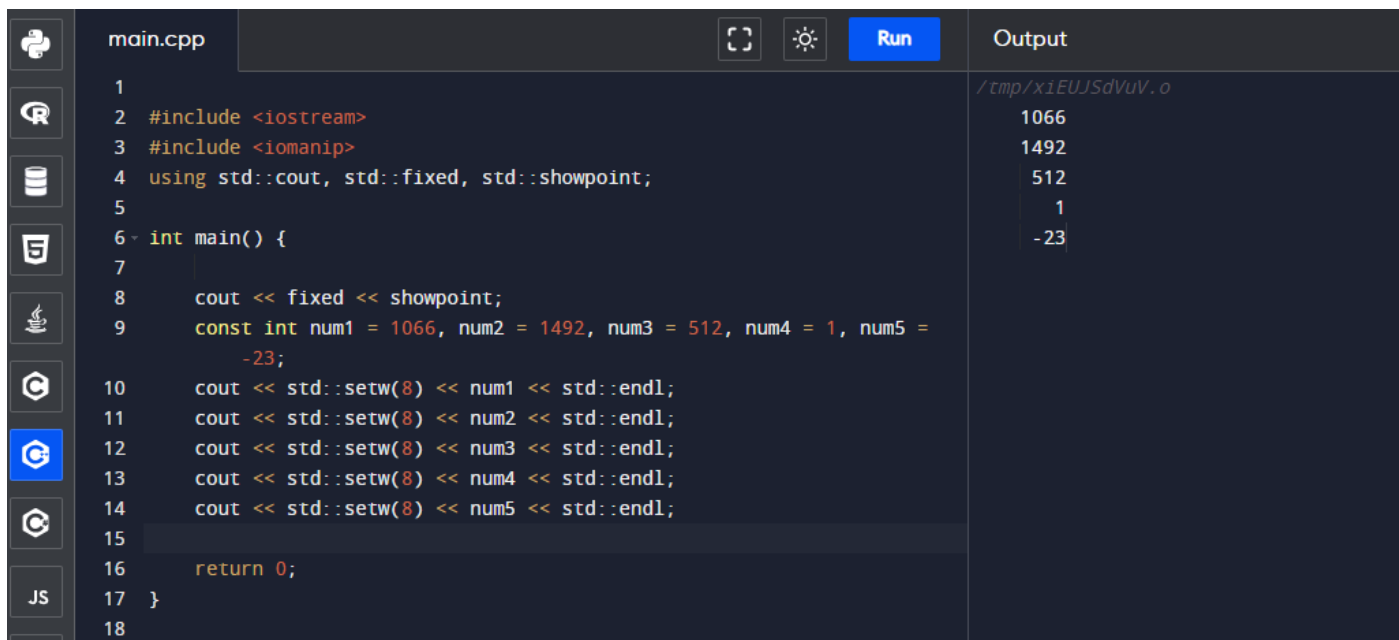
```
    cout << std::setw(8) << num3 << std::endl;
```

```
    cout << std::setw(8) << num4 << std::endl;
```

```
    cout << std::setw(8) << num5 << std::endl;
```

```
    return 0;
```

```
}
```



The screenshot shows a C++ IDE with a file named `main.cpp`. The code is as follows:

```
1
2 #include <iostream>
3 #include <iomanip>
4 using std::cout, std::fixed, std::showpoint;
5
6 int main() {
7
8     cout << fixed << showpoint;
9     const int num1 = 1066, num2 = 1492, num3 = 512, num4 = 1, num5 =
      -23;
10    cout << std::setw(8) << num1 << std::endl;
11    cout << std::setw(8) << num2 << std::endl;
12    cout << std::setw(8) << num3 << std::endl;
13    cout << std::setw(8) << num4 << std::endl;
14    cout << std::setw(8) << num5 << std::endl;
15
16    return 0;
17 }
18
```

The IDE has a sidebar on the left with icons for Python, R, Database, Jupyter, Docker, and C++ (selected). The top bar has icons for a file explorer, settings, and a 'Run' button. The output window on the right shows the following output:

```
/tmp/x1EUJSdVuV.o
1066
1492
512
1
-23
```


b. Question#2: Add two statements to your program. Calculate the floating point result from dividing the sum of the first two values by the sum of the last three values and store it in answer. The second statement should write the contents of answer on the screen to four decimal places. (Do not forget to declare answer.) The answer is _____.

Ans:

```
#include <iostream>

#include <iomanip>

using std::cout, std::fixed, std::showpoint;

int main() {

    cout << fixed << showpoint;

    const int num1 = 1066, num2 = 1492, num3 = 512, num4 = 1, num5 = -23;

    double answer = static_cast<double>(num1 + num2) / (num3 + num4 + num5);

    cout << "The answer is " << std::fixed << std::setprecision(4) << answer << std::endl;

    return 0;

}
```



The screenshot shows a C++ IDE with a dark theme. On the left is a sidebar with icons for Python, R, Java, C++, JavaScript, and a file explorer. The main editor window displays a file named 'main.cpp' with the following code:

```
1
2 #include <iostream>
3 #include <iomanip>
4 using std::cout, std::fixed, std::showpoint;
5
6 int main() {
7
8     cout << fixed << showpoint;
9     const int num1 = 1066, num2 = 1492, num3 = 512, num4 = 1, num5 =
      -23;
10    double answer = static_cast<double>(num1 + num2) / (num3 + num4 +
      num5);
11    cout << "The answer is " << std::fixed << std::setprecision(4) <<
      answer << std::endl;
12
13    return 0;
14 }
15
```

At the top of the editor, there are icons for a full-screen view, a settings gear, and a blue 'Run' button. To the right of the editor is an 'Output' panel. It shows the file path '/tmp/q1QJ0Yhg1U.o' and the output text 'The answer is 5.2204'.

c. Question#3: Write the following numbers right-justified in a column on the screen. Each of the data values should be written in formatted floating-point notation with two decimal places. Use field width specifications rather than listing the numbers in your program with the proper formatting. You may use either literal constants or named constants. 23.62

46.043.4443 100.1 98.98

Ans:

```
#include <iostream>
```

```
#include <iomanip>
```

```
using std::cout, std::fixed, std::showpoint;
```

```
int main() {
```

```
    cout << fixed << showpoint;
```

```
    const double num1 = 23.62, num2 = 46.0, num3 = 43.4443, num4 = 100.1, num5 = 98.98;
```

```
    cout << std::setw(8) << std::setprecision(2) << num1 << std::endl;
```

```
    cout << std::setw(8) << std::setprecision(2) << num2 << std::endl;
```

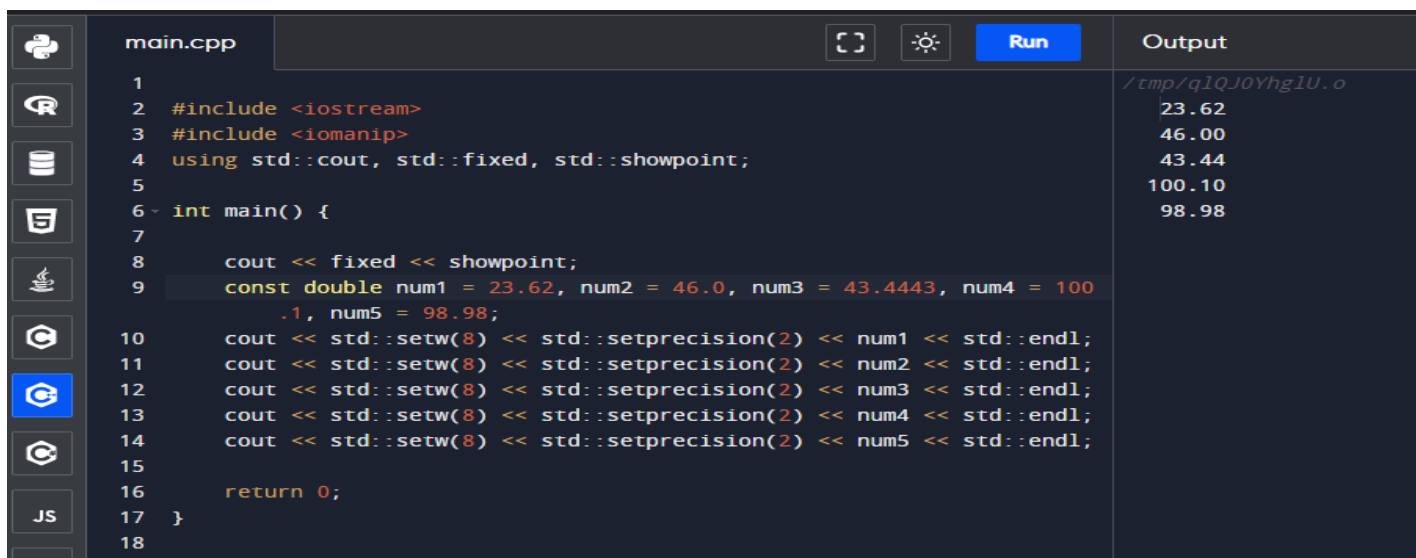
```
    cout << std::setw(8) << std::setprecision(2) << num3 << std::endl;
```

```
    cout << std::setw(8) << std::setprecision(2) << num4 << std::endl;
```

```
    cout << std::setw(8) << std::setprecision(2) << num5 << std::endl;
```

```
    return 0;
```

```
}
```



The screenshot shows a C++ IDE with a file named `main.cpp`. The code is as follows:

```
1
2 #include <iostream>
3 #include <iomanip>
4 using std::cout, std::fixed, std::showpoint;
5
6 int main() {
7     cout << fixed << showpoint;
8     const double num1 = 23.62, num2 = 46.0, num3 = 43.4443, num4 = 100
9       .1, num5 = 98.98;
10    cout << std::setw(8) << std::setprecision(2) << num1 << std::endl;
11    cout << std::setw(8) << std::setprecision(2) << num2 << std::endl;
12    cout << std::setw(8) << std::setprecision(2) << num3 << std::endl;
13    cout << std::setw(8) << std::setprecision(2) << num4 << std::endl;
14    cout << std::setw(8) << std::setprecision(2) << num5 << std::endl;
15
16    return 0;
17 }
18
```

The IDE has a sidebar with icons for Python, R, Java, C++, JavaScript, and others. The C++ icon is selected. The output window on the right shows the following results:

```
/tmp/q1QJ0Yhg1U.o
23.62
46.00
43.44
100.10
98.98
```

d. Question#4: Add two statements to your program for Question#3. The first statement should calculate the sum of the numbers and store the result in variable sum. The second statement should write sum on the screen, properly labeled. The sum of the numbers is _____.

Ans:

```
#include <iostream>

#include <iomanip>

using std::cout, std::fixed, std::showpoint;

int main() {

    cout << fixed << showpoint;

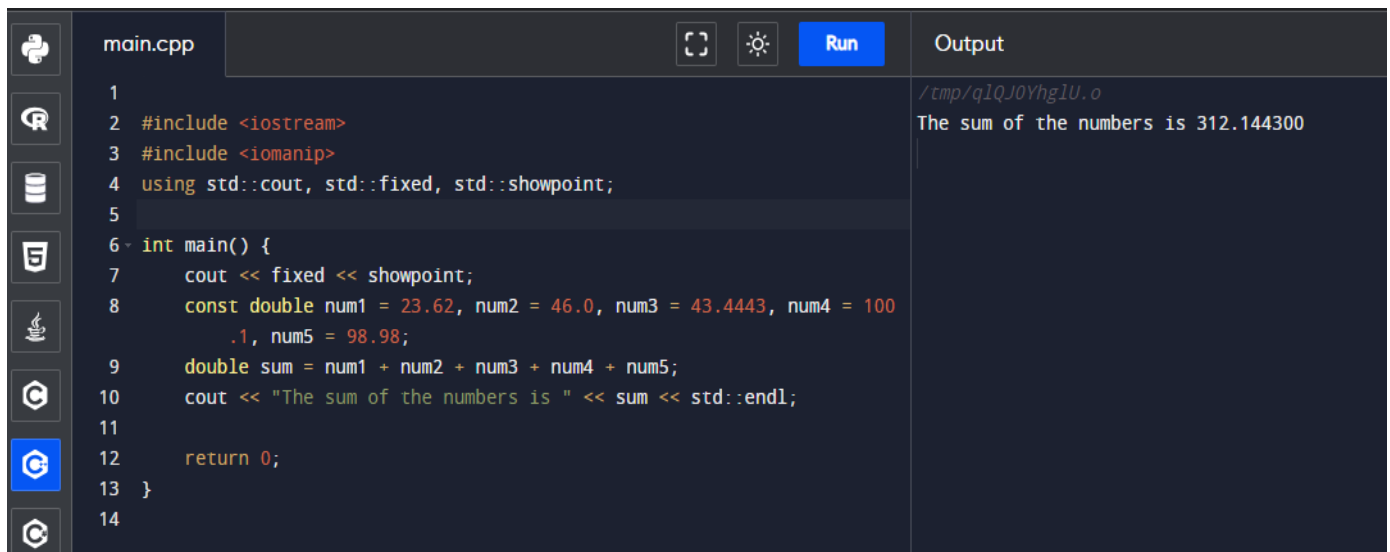
    const double num1 = 23.62, num2 = 46.0, num3 = 43.4443, num4 = 100.1, num5 = 98.98;

    double sum = num1 + num2 + num3 + num4 + num5;

    cout << "The sum of the numbers is " << sum << std::endl;

    return 0;

}
```



The screenshot shows a C++ IDE with a dark theme. On the left is a sidebar with icons for Python, C++, and other languages. The main editor window displays a C++ program named 'main.cpp'. The code is as follows:

```
1
2 #include <iostream>
3 #include <iomanip>
4 using std::cout, std::fixed, std::showpoint;
5
6 int main() {
7     cout << fixed << showpoint;
8     const double num1 = 23.62, num2 = 46.0, num3 = 43.4443, num4 = 100
      .1, num5 = 98.98;
9     double sum = num1 + num2 + num3 + num4 + num5;
10    cout << "The sum of the numbers is " << sum << std::endl;
11
12    return 0;
13 }
14
```

On the right side of the IDE, there is an 'Output' panel. It shows the execution path '/tmp/qlQJOYhg1U.o' and the output of the program: 'The sum of the numbers is 312.144300'. Above the output panel, there are icons for running the code and a 'Run' button.

Q5) Use the following program shell for Question#1 through #3.

a. Question#1: Add the statements necessary to print the following strings centered in fields of 20 characters, all on one line: "Good Morning", "Sarah", and "Sunshine!". Do not use manipulators. Compile and run your program; show your output.

Ans:

// Program Center sends strings to the output stream in specified formats.

```
#include <iostream>
```

```
#include <iomanip>
```

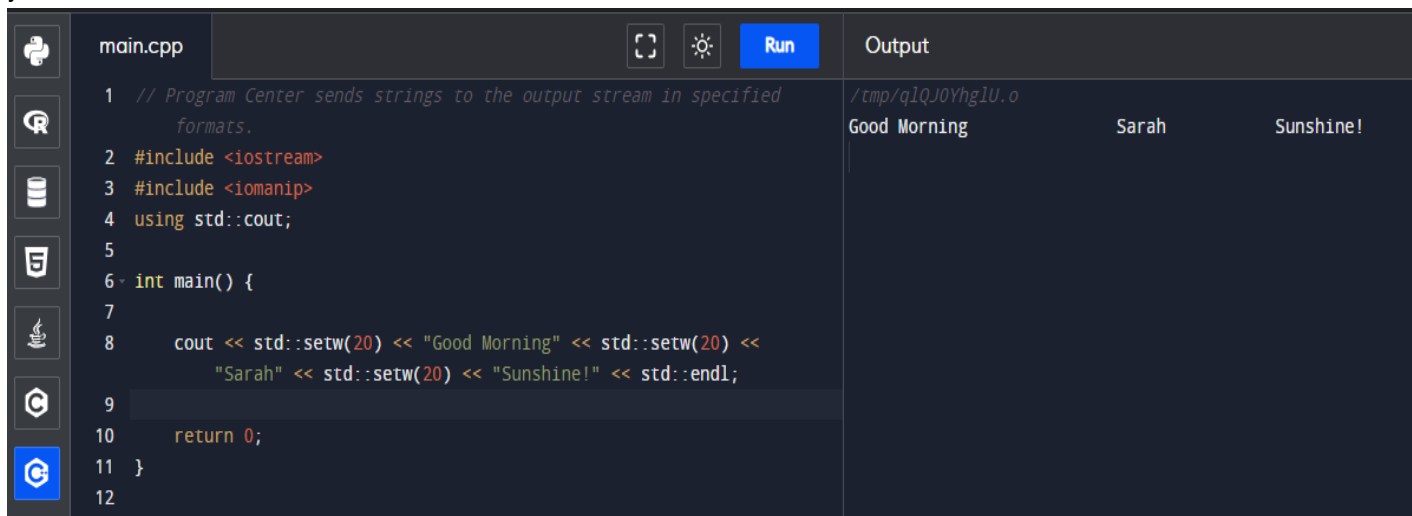
```
using std::cout;
```

```
int main() {
```

```
    cout << std::setw(20) << "Good Morning" << std::setw(20) << "Sarah" << std::setw(20) << "Sunshine!" << std::endl;
```

```
    return 0;
```

```
}
```



The screenshot shows a C++ IDE with a file named `main.cpp`. The code is as follows:

```
1 // Program Center sends strings to the output stream in specified
  formats.
2 #include <iostream>
3 #include <iomanip>
4 using std::cout;
5
6 int main() {
7
8     cout << std::setw(20) << "Good Morning" << std::setw(20) <<
      "Sarah" << std::setw(20) << "Sunshine!" << std::endl;
9
10    return 0;
11 }
12
```

The IDE has a sidebar on the left with icons for Python, C++, and other languages. The top bar has a `Run` button. The output window on the right shows the following output:

```
/tmp/q1QJ0YhglU.o
Good Morning          Sarah          Sunshine!
```

b. Question#2: Repeat Question#1 using manipulators to help center your strings. Compile and run your program. Your output should be the same.

Ans:

// Program Center sends strings to the output stream in specified formats.

```
#include <iostream>
```

```
#include <iomanip>
```

```
using std::cout;
```

```
int main() {
```

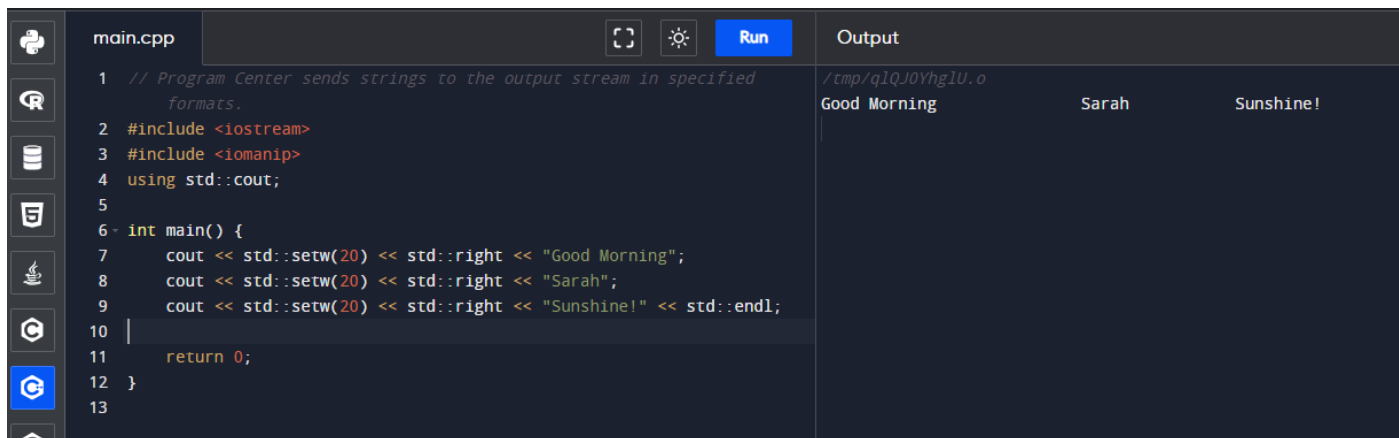
```
    cout << std::setw(20) << std::right << "Good Morning";
```

```
    cout << std::setw(20) << std::right << "Sarah";
```

```
    cout << std::setw(20) << std::right << "Sunshine!" << std::endl;
```

```
    return 0;
```

```
}
```



The screenshot shows a C++ IDE with a file named `main.cpp`. The code is as follows:

```
1 // Program Center sends strings to the output stream in specified
  formats.
2 #include <iostream>
3 #include <iomanip>
4 using std::cout;
5
6 int main() {
7     cout << std::setw(20) << std::right << "Good Morning";
8     cout << std::setw(20) << std::right << "Sarah";
9     cout << std::setw(20) << std::right << "Sunshine!" << std::endl;
10
11     return 0;
12 }
13
```

The output window on the right shows the following output:

```
/tmp/q1QJ0Yhg1U.o
Good Morning          Sarah          Sunshine!
```

c. Question#3: Change the program in Question#2 so that the three strings are printed on three separate lines with a blank line in between each string.

Ans:

// Program Center sends strings to the output stream in specified formats.

```

#include <iostream>

#include <iomanip>

using std::cout;

int main() {

    cout << std::setw(20) << std::left << "Good Morning" << std::endl<< std::endl;

    cout << std::setw(20) << std::left << "Sarah" << std::endl << std::endl;

    cout << std::setw(20) << std::left << "Sunshine!" << std::endl << std::endl;

    return 0;

}

```

main.cpp	Output
1 // Program Center sends strings to the output stream in specified formats.	/tmp/qlQJ0YhglU.o
2 #include <iostream>	Good Morning
3 #include <iomanip>	
4 using std::cout;	Sarah
5	Sunshine!
6 int main() {	
7 cout << std::setw(20) << std::left << "Good Morning" << std::endl	
8 << std::endl;	
9 cout << std::setw(20) << std::left << "Sarah" << std::endl << std	
10 << std::endl;	
11 cout << std::setw(20) << std::left << "Sunshine!" << std::endl <<	
12 std::endl;	
13 return 0;	
14 }	

Github Links:

Q2. [https://github.com/absiddik012/CS360Lab/blob/main/19897_CS360Lab_Assignment-1%2C%20Q-2\(a%2Cb%2Cc\)_Abu%20Bakar_Siddik](https://github.com/absiddik012/CS360Lab/blob/main/19897_CS360Lab_Assignment-1%2C%20Q-2(a%2Cb%2Cc)_Abu%20Bakar_Siddik)

Q3. [https://github.com/absiddik012/CS360Lab/blob/main/19897_CS360Lab_Assignment-1%2C%20Q-3\(a%2Cb%2Cc\)_Abu%20Bakar_Siddik](https://github.com/absiddik012/CS360Lab/blob/main/19897_CS360Lab_Assignment-1%2C%20Q-3(a%2Cb%2Cc)_Abu%20Bakar_Siddik)

Q4. [https://github.com/absiddik012/CS360Lab/blob/main/19897_CS360Lab_Assignment-1%2C%20Q-4\(a%2Cb%2Cc%2Cd\)_Abu%20Bakar_Siddik](https://github.com/absiddik012/CS360Lab/blob/main/19897_CS360Lab_Assignment-1%2C%20Q-4(a%2Cb%2Cc%2Cd)_Abu%20Bakar_Siddik)

Q5. [https://github.com/absiddik012/CS360Lab/blob/main/19897_CS360Lab_Assignment-1%2C%20Q-5\(a%2Cb%2Cc\)_Abu%20Bakar_Siddik](https://github.com/absiddik012/CS360Lab/blob/main/19897_CS360Lab_Assignment-1%2C%20Q-5(a%2Cb%2Cc)_Abu%20Bakar_Siddik)