San Francisco Bay University

CS360L - Programming in C and C++ Lab
Lab Assignment #1

Due day: 2/13/2024

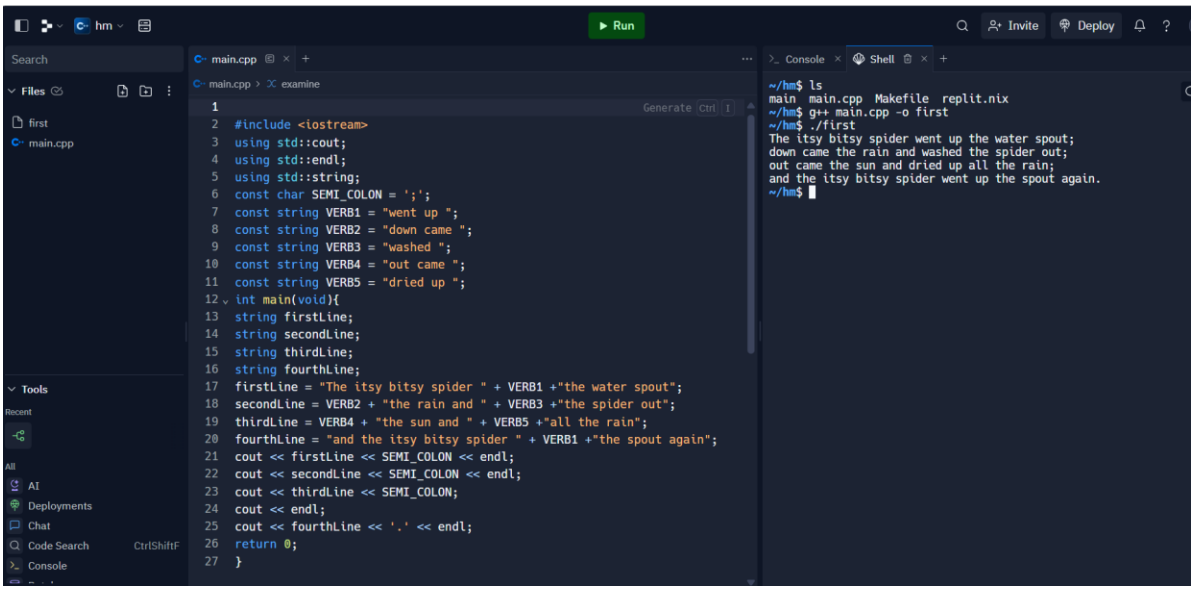
Instruction:

1. **Push the answer sheets/source code to Github**
2. **Please follow the code style rule like programs on handout.**
3. **Overdue lab assignment submission can't be accepted.**
4. **Take academic honesty and integrity seriously (Zero Tolerance of Cheating & Plagiarism)**

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

```
#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;
}
```



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

```
// Program Shell

#include <iostream>

using namespace std;

int main (){

        return 0;
}
```

   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)

```
// Question #1
cout << "Enter your last name: ";
cin >> lastName;
cout << "Enter your first name: ";
cin >> firstName;
cout << "Enter today's date (month:day:year): ";
cin >> date;

cout << "Your name: " << lastName << ", " << firstName << endl;
cout << "Today's date: " << date << endl;
```

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

```
// Question #2
cout << endl;
```

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.

```
// Question #3
cout << "Your name: " << firstName << " " << lastName << endl;
cout << "Today's date: " << date << endl;

return 0;
```

```
Enter your last name: Hamal
Enter your first name: Swekchha
Enter today's date (month:day:year): 04:15:2003
Your name: Hamal, Swekchha
Today's date: 04:15:2003

Your name: Swekchha Hamal
Today's date: 04:15:2003
~/hm$
```

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

```
// Program Strings applies string functions.

#include <iostream>
```

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

int main (void){

        return 0;
}
```

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.

```
int main(void) {
    // Question #1
    const string fullName = "Swekchha Hamal";";
    cout << "Length of fullName: " << fullName.length() << endl;
    cout << "Size of fullName: " << fullName.size() << endl;
```

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.

```
    // Question #2
    string lastNameFirst = fullName.substr(fullName.find(" ") + 1) + ",
" + fullName.substr(0, fullName.find(" "));
    cout << "Last name first: " << lastNameFirst << endl;
```

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.

```
    // Question #3
    string lastName = fullName.substr(fullName.find(" ") + 1);
    string firstNameInitial = fullName.substr(0, 1);
    cout << "Last name, first initial: " << lastName << ", " <<
firstNameInitial << endl;

    return 0;
}
```

```
~/hm$ ls
first  main.cpp  replit.nix  second_result
main   Makefile  second.cpp  third.cpp
~/hm$ g++ third.cpp -o third_result
~/hm$ ./third_result
Length of fullName: 14
Size of fullName: 14
Last name first: Hamal, Swekchha
Last name, first initial: Hamal, S
~/hm$
```

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

```
// Program Numbers sends numbers to the output stream in
// specified formats.

#include <iostream>
#include <iomanip>

using std::cout;

int main (void){

        cout << fixed  << showpoint;
        return 0;
}
```

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

```
int main(void) {
    // Question #1
    cout << "Question #1:" << std::endl;
    const int num1 = 1066;
    const int num2 = 1492;
    const int num3 = 512;
    const int num4 = 1;
    const int num5 = -23;
    cout << std::setw(6) << num1 << std::endl;
    cout << std::setw(6) << num2 << std::endl;
    cout << std::setw(6) << num3 << std::endl;
    cout << std::setw(6) << num4 << std::endl;
    cout << std::setw(6) << num5 << std::endl;
```

```
Question #1:
  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*.)

```
    // Question #2
    cout << "\nQuestion #2:" << std::endl;
    double answer = (num1 + num2) / static_cast<double>(num3 + num4 +
num5);
    cout << "The answer is " << std::fixed << std::setprecision(4) <<
answer << "." << std::endl;
```

```
Question #2:
The answer is 5.2204.
```

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.

```
    // Question #3
    cout << "\nQuestion #3:" << std::endl;
    const double num6 = 23.62;
    const double num7 = 46.0;
    const double num8 = 43.4443;
    const double num9 = 100.1;
    const double num10 = 98.98;
    cout << std::setw(10) << std::fixed << std::setprecision(2) << num6
<< std::endl;
    cout << std::setw(10) << std::fixed << std::setprecision(2) << num7
<< std::endl;
    cout << std::setw(10) << std::fixed << std::setprecision(2) << num8
<< std::endl;
    cout << std::setw(10) << std::fixed << std::setprecision(2) << num9
<< std::endl;
    cout << std::setw(10) << std::fixed << std::setprecision(2) <<
num10 << std::endl;
```

```
Question #3:
     23.62
     46.00
     43.44
    100.10
     98.98
```

23.62 46.0 43.4443 100.1 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.

```
    // Question #4
    cout << "\nQuestion #4:" << std::endl;
    double sum = num6 + num7 + num8 + num9 + num10;
    cout << "The sum of the numbers is " << sum << "." << std::endl;

    return 0;
}
```

```
Question #4:
The sum of the numbers is 312.14.
```

The sum of the numbers is _____312.14_____.

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

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

#include <iostream>
#include <iomanip>

using std::cout;

int main (void){
        return 0;
}
```

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.

```
int main(void) {
    // Question #1
    cout << "Question #1:" << std::endl;
    cout << setw(20) << "Good Morning" << setw(20) << "Sarah" <<
setw(20) << "Sunshine!" << std::endl;
```

```
Question #1:
        Good Morning              Sarah          Sunshine!
```

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

```
    // Question #2
    cout << "\nQuestion #2:" << std::endl;
    cout << std::setw(20) << std::left << "Good Morning" <<
std::setw(20) << "Sarah" << std::setw(20) << "Sunshine!" << std::endl;
```

```
Question #2:
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.

```
    // Question #3
    cout << "\nQuestion #3:" << std::endl;
    cout << std::setw(20) << std::left << "Good Morning" << std::endl;
    cout << std::setw(20) << std::left << "Sarah" << std::endl;
    cout << std::setw(20) << std::left << "Sunshine!" << std::endl;

    return 0;
}
```

```
Question #3:
Good Morning
Sarah
Sunshine!
```