

1. What Would C++ Do?

For each of the code snippets, write down the output of the program. If the code has an error, explain why, and whether it is a compile-time error or a runtime error.

C++ Code	What is the output?
<pre>string s1 = "yesterday"; cout << s1.substr(0,3);</pre>	yes
<pre>string s2 = "revolution"; cout << s2.substr(0);</pre>	revolution
<pre>string s3 = "sgt"; cout << s3 + "peppers";</pre>	sgtpeppers
<pre>cout << "sgt" + "peppers";</pre>	Compile Error: Can't add string literals!
<pre>string s4 = "ForNoOne"; cout << s4.substr(3,2) + s4.substr(0,3);</pre>	NoFor
<pre>string s5 = "Walrus"; s5[1] = "o"; cout << s5;</pre>	Compile Error: Can only assign characters when modifying string, ie: s5[1] = 'o'; // char, not string "o"
<pre>const string s6 = "aow"; s6[0] = 'p'; cout << s6;</pre>	Compile Error: Can't modify a const string!

References

<pre>int a = 42; int &ra = a; a += 3; ra += 1; cout << "a: " << a << " ra: " << ra;</pre>	a: 46 ra: 46
<pre>int a = 1; int &ra = 1; ra += 1; cout << "a: " << a << " ra: " << ra;</pre>	Compile Error: "int &ra = 1;" is invalid, references must point to a previously defined variable.
<pre>int x = 3; int y = x; x = 42; cout << "x: " << x << " y: " << y;</pre>	x: 42 y: 3
<pre>int x = 1; const int& y = x; cout<<"x: " << x << " y: " << y << endl; x = 42; cout << "x: " << x << " y: " << y;</pre>	x: 1 y: 1 x: 42 y: 42
<pre>int x = 1; const int& y = x; y = 2; cout << "x: " << x << " y: " << y;</pre>	Compile Error: Can't modify const value y.
<pre>const int x = 1; int& y = x; y = 42; cout << x << " " << y;</pre>	Compile Error: Can't define "int&y = x;", since x is const!

2. IO Manipulation for Fun and Profit

Louis Reasoner would like to write a program that pseudo-illustrates addition. Given two numbers, each with fewer than 4 digits, the output of the program is:

You entered: 42, 1. The sum is:

```
 00042
+ 00001
-----
 00043
```

Louis writes the following program. Is the program correct? If not, describe what the program would instead output, and try to fix it.

```
#include <iostream>
#include <iomanip>
using namespace std;
int main() {
    cout << "Give me two numbers to add:" << endl;
    int x1, x2;
    cin >> x1 >> x2;
    cout << "You entered: " << x1 << ", " << x2 << ".";
    cout << " The sum is:" << endl;
    cout << setw(5) << setfill('0') << "  " << x1 << endl;
    cout << "+ " << x2 << endl;
    cout << "-----" << endl;
    cout << "  " << x1+x2 << endl;
    return 0;
}
```

[Solution]: The program is incorrect, and would output the following:

```
000 42
+ 1
-----
 43
```

A fixed version is as follows: (.cpp file is in week4b.zip)

```
int main() {
    cout << "Give me two numbers to add:" << endl;
    int x1, x2;
    cin >> x1 >> x2;
    cout << "You entered: " << x1 << ", " << x2 << ".";
    cout << " The sum is:" << endl;
```

```

cout << " " << setw(5) << setfill('0') << x1 << endl;
cout << "+ " << setw(5) << setfill('0') << x2 << endl;
cout << setw(8) << setfill('-') << "" << endl;
cout << " " << setw(5) << setfill('0') << x1+x2 << endl;
return 0;
}

```

3. My Little Barplot

Suppose we would like to visualize some data as a barplot, ie:

```

Title: My Week
Hours of Sleep      | *****
Cups of Coffee     | *****
Existential Crises | *

```

Write a complete C++ program that asks the user for: (a) Title, (b) 3 category title+values. Note that your program should assume that there are exactly 3 categories. For instance, to generate the above barplot, the user would have typed:

```

Enter plot title: My Week
[1/3] Enter category title: Hours of Sleep
      Enter data: 15
[2/3] Enter category title: Cups of Coffee
      Enter data: 20
[3/3] Enter category title: Existential Crises
      Enter data: 1

```

[Solution] (Code is also in week4b.zip)

```

#include <iostream>
#include <iomanip>
#include <string>
#include <cmath>
using namespace std;
int main() {
    /* (1) Ask for plot title */
    cout << "Enter Plot Title: ";
    string plotTitle;
    getline(cin, plotTitle);

    /* Ask for category 1 */
    string catTitle1;
    cout << "[1/3] Enter Category Title: ";
    getline(cin, catTitle1);
    cout << "      Enter Data: ";
    unsigned int catData1;
    cin >> catData1;
    cin.ignore(); // strip that newline!

```

```

/* Ask for category 2 */
string catTitle2;
cout << "[2/3] Enter Category Title: ";
getline(cin, catTitle2);
cout << "      Enter Data: ";
unsigned int catData2;
cin >> catData2;
cin.ignore(); // strip that newline!

/* Ask for category 3 */
string catTitle3;
cout << "[3/3] Enter Category Title: ";
getline(cin, catTitle3);
cout << "      Enter Data: ";
unsigned int catData3;
cin >> catData3;

/** Create barplot **/
cout << "Title: " << plotTitle << endl;
unsigned int catTitleWidth = fmax(fmax(catTitle1.length(), catTitle2.length()),
catTitle3.length());
const unsigned int PADDING = 4; // padding space between separator |
cout << left;
// Category 1
cout << setfill(' ') << setw(catTitleWidth) << catTitle1 << setw(PADDING) << "" <<
"|";
cout << setw(PADDING) << "" << setw(catData1) << setfill('*') << "" << endl;
// Category 2
cout << setfill(' ') << setw(catTitleWidth) << catTitle2 << setw(PADDING) << "" <<
"|";
cout << setw(PADDING) << "" << setw(catData2) << setfill('*') << "" << endl;
// Category 3
cout << setfill(' ') << setw(catTitleWidth) << catTitle3 << setw(PADDING) << "" <<
"|";
cout << setw(PADDING) << "" << setw(catData3) << setfill('*') << "" << endl;
return 0;
}

```

4. Long Division? More like, Wrong Division!

Write a program that illustrates long division. Suppose we want to illustrate $42 / 2$. The output of your program should be:

Let's divide 42 by 2:

00021

00002 | 00042

You may assume that the user will only input integers with at most 5 digits, and that the numbers are evenly divisible. For instance, we disallow dividing 16 by 3 (16 / 3).

Hint: To make the vertical bar "|", use the pipe character on your keyboard: SHIFT+backslash. To make the horizontal line "_", use the underscore character.

[Solution] Here's one way to do it: (.cpp file is in week4b.zip)

```
int main() {
    cout << "Give me two numbers to divide:" << endl;
    /* Ask user for input */
    double x1, x2;
    cin >> x1 >> x2;
    int result = x1 / x2;
    cout << "Let's divide " << x1 << " by " << x2 << ":" << endl;

    /* (1) Output the first line: <SPACE><QUOTIENT> */
    cout << setfill(' ');
    cout << setw(8) << "" << setw(5) << setfill('0') << result << endl;
    /* (2) Output the top '_' of the division rectangle */
    cout << setw(6) << setfill(' ') << " " << setw(7)
        << setfill('_') << "_"<< endl;
    /* (3) Output the numerator and denominator */
    cout << setw(5) << setfill('0') << x2;
    cout << " | ";
    cout << setw(5) << setfill('0') << x1 << endl;

    cin.ignore(); cin.get();
    return 0;
}
```