PIC 10A 1C Week 4b Problems. TA: Eric Kim. [Solutions] *(Updated: 2/9/2016 v2)*

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?**
`string s1 = "yesterday";` `cout << s1.substr(0,3);`	yes
`string s2 = "revolution";` `cout << s2.substr(0);`	revolution
`string s3 = "sgt";` `cout << s3 + "peppers";`	sgt peppers
`cout << "sgt" + "peppers";`	Compile Error: Can't add string literals!

`string s4 = "ForNoOne";`
`cout << s4.substr(3,2) +`
`s4.substr(0,3);`
	NoFor

`string s5 = "Walrus";`
`s5[1] = "o";`
`cout << s5;`
	Compile Error: Can only assign characters when modifying string, ie:
	s5[1] = 'o'; // char, not string "o"

`const string s6 = "aow";` `s6[0] = 'p'; cout << s6;`	Compile Error: Can't modify a const string!

References

`int a = 42;`
`int &ra = a;  a += 3;  ra += 1;`
`cout << "a: " << a << " ra: " << ra;`
	a: 46 ra: 46

`int a = 1;`
`int &ra = 1;  ra += 1;`
`cout << "a: " << a << " ra: " << ra;`
	Compile Error: "int &ra = 1;" is invalid, references must point to a previously defined variable.

`int x = 3;  int y = x;`
`x = 42;`
`cout << "x: " << x << " y: " << y;`
	x: 42 y: 3

`int x = 1;  const int& y = x;`
`cout<<"x: "<< x << " y: "<< y`
`<<endl;`
	x: 1 y: 1
	x: 42 y: 42

`x = 42;` `cout << "x: " << x << " y: " << y;`	

`int x = 1;  const int& y = x;`
`y = 2;`
`cout << "x: " << x << " y: " << y;`
	Compile Error: Can't modify const value y.

`const int x = 1;  int& y = x;`
`y = 42;`
`cout << x << " " << y;`
	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:

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

3. 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:

```
#include <iostream>
#include <iomanip>
using namespace std;
int main() {
  cout << "Give me two numbers to divide:" << endl;
  double x1, x2;
  cin >> x1 >> x2;
  cout << "Let's divide " << x1 << " by " << x2 << ":" << endl;
  cout << setw(8) <<" "<< setw(5) << setfill('0') << x1 / x2 << endl;
  cout << setw(6) << setfill(' ') << " "<< setw(7)
    << setfill('_') <<"_"<< endl;
  cout << setw(5) << setfill('0') << x2;
  cout << " | ";
  cout << setw(5) << setfill('0') << x1 << endl;
  cin.ignore(); cin.get();
  return 0;
}
```