CS 121 – Week 5 Worksheet – Bunch of Programming Stuff

1. Using the below code and output, explain how default parameters work.

PROGRAM	**OUTPUT**

#include <iostream>
#include <string>
using namespace std;
void printStuff(string str = "NO MESSAGE GIVEN")
{
  cout << str << endl;
}
 	
int main()
{
  printStuff("Hello!");
  printStuff();
  printStuff("Supercalifragilisticexpialidocious");
  printStuff("NO MESSAGE GIVEN");
  printStuff();
  printStuff("AND HIS NAME IS JOHN CENA\a");
 	
  return 0;
}
	Hello!
	NO MESSAGE GIVEN
	Supercalifragilisticexpialidocious
	NO MESSAGE GIVEN
	NO MESSAGE GIVEN
	AND HIS NAME IS JOHN CENA

2. Explain the difference between a literal in C++ and a constant in C++.

3. Look at the following function prototypes. Reading from top to bottom, note and explain the ones that are considered syntactically invalid in C++.

```
int addNums(int a, int b);
double addNums(double a = 2.0, double b);
float addNums(float a, float b = 3.0);
short addNums(short a = 3, short b = 6);
long addNums(int z, int c = 3);
```

4. What is the output of the following code segment?

PROGRAM	**OUTPUT**

int x = 3;
int& y = x;
int z = y;
cout << x << " " << y << " " << z << endl;
y = 7;
cout << x << " " << y << " " << z << endl;	

5. Create a function that swaps two strings

6. Note two reasons where by-reference would be considered useful. (**HINT**: Think about how by-value works, the const keyword, and the size of some datatypes).

7. If you wanted to convert 26, a base-10 integer, to its base-2 (binary) representation, you would do something similar to the following:

   26 / 2 = 13 R 0

   13 / 2 = 6 R 1

   6 / 2 = 3 R 0

   3 / 2 = 1 R 1

   1 / 2 = 0 R 1

   Which leads to: $26_{10} = 11010_2$ (you read the bits from bottom to top).

   Using the above algorithm and the following incomplete code, create a function that takes in a base-10 integer and converts it to its binary representation.

```
string convBaseTenToTwo(int num)
{
  string bit_string = "";


  while(             )
  {



    // insert to front of bit_string
    bit_string.insert(0, to_string(            ));
  }


}
```