Instructions: This homework assignment focuses primarily on C++ scope rules and functions. The answers to the following questions can be determined from Chapters 3 through 8 of the lecture notes and Chapters 2 through 8 of the text. Assume any \#include directives, variable declarations, etc, which are needed to make the given code syntactically correct.

After you have analyzed the questions and decided what answers you believe are correct, you may find it useful to write some short programs to test your logic.

The on-line Opscan form for this quiz provided by the Curator system must be used for this assignment. (No other submissions will be accepted!) Check the course web site for the due date for this quiz.

The CS1044 online quizzes are open book, open notes, open Web quizzes. It is an honor code violation to discuss, (in any form: written, verbal or electronic), any portion of these quizzes with any other students, (regardless of whether they are taking the course or not). It is also an honor code violation to have a copy of a quiz, (in any form: written, verbal or electronic), in your possession when not taking a quiz. Failure to adhere to any of these restrictions is a Va Tech Honor Code violation, (www.honorsystem.vt.edu).

1. If an ampersand ('&') is not attached to the data type of a formal parameter, then the corresponding actual parameter can be:
   1) a constant
   2) a variable name
   3) an arbitrary expression
   4) All of these
   5) 1 and 2 only
   6) 2 and 3 only
   7) None of these

2. A function SomeFunc has two formal parameters, alpha and beta, of type int. The data flow for alpha is one-way, into the function. The data flow for beta is two-way, into and out of the function. What is the most appropriate function prototype for SomeFunc?
   1) void SomeFunc( int alpha, int beta );
   2) void SomeFunc( int& alpha, int beta );
   3) void SomeFunc( int alpha, int& beta );
   4) void SomeFunc( int& alpha, int& beta );
   5) 1 and 2 only
   6) 3 and 4 only
   7) None of these

3. For the function definition

   void Func( int& Gamma ) {
       Gamma = 245;
   }

which of the following comments best describes the direction of data flow for Gamma?

   1) one-way, into the function
   2) one-way, out of the function
   3) two-way, into and out of the function
   4) None of these
4. For the function definition

```cpp
void Func( int Gamma ) {
    cout << 3 * Gamma;
}
```

which of the following comments best describes the direction of data flow for Gamma?

1) one-way, into the function
2) one-way, out of the function
3) two-way, into and out of the function
4) None of these

5. For the function definition

```cpp
void Func( int& Gamma ) {
    Gamma = 3 * Gamma;
}
```

which of the following comments describes the direction of data flow for Gamma?

1) one-way, into the function
2) one-way, out of the function
3) two-way, into and out of the function
4) None of these

6. Consider the function definition

```cpp
void Demo( int intVal, double& doubleVal ) {
    intVal = intVal * 2;
    doubleVal = double(intVal) + 3.5;
}
```

What values does the following code fragment print?

```cpp
int myInt = 20;
double myDble = 4.8;
Demo(myInt, myDble);
cout << "myInt = " << myInt
    << " and myDble = " << myDble << endl;
```

1) myInt = 20 and myDble = 43.5
2) myInt = 40 and myDble = 4.8
3) myInt = 20 and myDble = 4.8
4) myInt = 40 and myDble = 43.5
5) None of these
7. Consider the function definition

```cpp
void Demo( int& intVal, double doubleVal ) {
    intVal    = intVal * 2;
    doubleVal = double(intVal) + 3.5;
}
```

What values does the following code fragment print?

```cpp
int myInt   = 20;
float myDble = 4.8;
Demo(myInt, myDble);
cout << "myInt = " << myInt
     << " and myDble = " << myDble << endl;
```

1) myInt = 20 and myDble = 43.5  4) myInt = 40 and myDble = 43.5
2) myInt = 40 and myDble = 4.8  5) None of these
3) myInt = 20 and myDble = 4.8

8. In the following function, the declaration of Beta includes an initialization.

```cpp
void SomeFunc( int Alpha )
{
    int Beta = 25;
    ...
}
```

Which of the following statements about the variable Beta declared above is false?

1) It is initialized once only, the first time the function is called.
2) It is initialized each time the function is called.
3) It cannot be reassigned a different value within the function.
4) 1 and 3 only
5) 2 and 3 only
6) None of these are false

For the next two questions, consider the short program:

```cpp
#include <iostream>                 // Line  1
using namespace std;              // Line  2

int main() {                       // Line  3
    int alpha = 3;                 // Line  4
    int beta  = 20;                 // Line  5
    if (beta > 10)                  // Line  6
    {
        int alpha = 5;             // Line  7
        beta = beta + alpha;        // Line  8
        cout << alpha << ' ' << beta; // Line 12
        return 0;                   // Line 13
    }
    cout << alpha << ' ' << beta;   // Line 11
}                                    // Line 14
```

1) It is initialized once only, the first time the function is called.
2) It is initialized each time the function is called.
3) It cannot be reassigned a different value within the function.
4) 1 and 3 only
5) 2 and 3 only
6) None of these are false
9. What is the scope of the identifier alpha declared in Line 4?

1) Line 4 through Line 14  
2) Lines 4, 5, 6, 12, 13 and 14 only  
3) Lines 4 and 5 only  
4) None of these

10. What is the output of the given program?

1) 3 20  
2) 3 25 3 25  
3) 5 25 5 25  
4) 5 25 3 25  
5) 5 25 3 20  
6) None of these

11. This question demonstrates the hazard of choosing inappropriate parameter-passing mechanisms. Given the function definition:

```c
int Power(int& Base, int& Exponent ) {
    int Product = 1;
    while (Exponent >= 1) {
        Product = Product * Base;
        Exponent--;
    }
    return Product;
}
```

what is the output of the following code?

```c
int N = 2;
int Pow = 3;
int Result = Power(N, Pow);
cout << N << " to the power " << Pow << " is " << Result;
```

1) 2 to the power 3 is 8  
2) 2 to the power 0 is 8  
3) 0 to the power 0 is 0  
4) 2 to the power 3 is 1  
5) None of these

12. Which of the following would be the most appropriate prototype for a function that computes the tax, in cents, on a purchase? Assume that the tax rate will be declared within the function.

1) void Tax(int& Price);  
2) int Tax(int& Price);  
3) void Tax(int Price);  
4) int Tax(int Price);

13. Which of the following things must be specified in a function prototype?

1) name of the function  
2) types of the formal parameters  
3) return type of the function  
4) names of the formal parameters  
5) All of the above  
6) 1 and 2 only  
7) 2 and 3 only  
8) 1 and 3 only  
9) 1, 2 and 3 only  
10) None of these
For the next 2 questions, consider this program, which illustrates one of the hazards of polluting global scope with variable declarations:

```cpp
#include <fstream>
#include <iostream>
#include <string>
using namespace std;

const char SPACE = ' ';
int lCount;

void countSpaces(string Line) {
    int sCount = 0;
    int Idx = 0;
    while ( Idx < Line.length() ) {
        if ( Line[Idx] == SPACE )
            lCount++;
        Idx++;
    }
    cout << "Spaces on line: " << sCount << endl;
}

int main( ) {
    ifstream In("Data.txt");
    string Line;
    lCount = 0;
    getline(In, Line);
    while (In) {
        lCount++;
        countSpaces(Line);
        getline(In, Line);
    }
    cout << "Lines: " << lCount << endl;
    In.close();
    return 0;
}
```

14. If the program is run on the input file shown below, what value will be reported for the number of lines? (The first character in the file is the 'S' and the last is the '.')

```
Software test teams appreciate the use of global variables.
```

1) 0 2) 1 3) 2 4) 3 5) 5 6) 7 8) 9 9) None of these

15. If the declaration of `lCount` were local to `main()`, how would the error in this program have been detected?

1) By the discovery of an incorrect result when the output was examined.
2) By the occurrence of a run-time error.
3) By the occurrence of a compile-time error message.
4) None of these.

On-line submission of a quiz constitutes your Virginia Tech Honor Code Pledge:

Virginia Tech Honor Code:

"On my honor, I have neither given nor received unauthorized aid on this examination."