CS 2605 Online Quiz 2: C++ Functions

Instructions: Submit your answers to these questions to the Curator as Quiz2 by the posted due date and time. No late submissions will be accepted.

For questions 1 through 5, consider the following program:

```cpp
const int LIMIT = 50;                  // Line 1
int AddEm(int x, int y);              // Line 2
int main() {                           // Line 3
  int x = 42,                            // Line 4
  y = 35;                               // Line 5
  int Sum;                              // Line 6
  Sum = AddEm(x, y);                    // Line 7
  return 0;                             // Line 8
} // Line 9

int AddEm(int x, int y) {              // Line 10
  int Total;                            // Line 11
  Total = x + y;                        // Line 12
  if (Total > LIMIT)                    // Line 13
    Total = 0;                          // Line 14
  return (Total);                       // Line 15
} // Line 16
```

1. What is the scope of the identifier Sum which is declared in Line 6?

   1) Line 1 to Line 16  
   2) Line 6 to Line 16  
   3) Line 6  
   4) Line 6 to Line 7  
   5) Line 6 to Line 9  
   6) None of these

2. What is the scope of the identifier x which is declared in Line 4?

   1) Line 1 to Line 16  
   2) Line 4 to Line 16  
   3) Line 4  
   4) Line 4 to Line 7  
   5) Line 4 to Line 9  
   6) None of these

3. What is the scope of the identifier x which is declared in Line 10?

   1) Line 1 to Line 16  
   2) Line 4 to Line 16  
   3) Line 10  
   4) Line 10 to Line 12  
   5) Line 10 to Line 16  
   6) None of these

4. What is the scope of the identifier LIMIT which is declared in Line 1?

   1) Line 1 to Line 16  
   2) Line 1 to Line 3  
   3) Line 1  
   4) Line 10 to Line 13  
   5) Line 10 to Line 16  
   6) None of these

5. Which of the following are true?

   1) LIMIT is local to main()  
   2) Total is local to AddEm()  
   3) Sum is local to main()  
   4) LIMIT is global  
   5) x is global  
   6) All of them are true  
   7) All but 1 are true  
   8) 2 and 3 only  
   9) 2, 3 and 4 only  
   10) None of these
6. Formal parameters are listed in the function ________ and actual parameters are listed in the function ________.

1) call, implementation  3) header, body  5) None of these
2) implementation, call  4) body, header

7. A parameter of a simple type, such as int or double, should be passed by value if that parameter's data flow is:

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

8. Which of the following statements are true when a parameter is passed by value?

1) The actual parameter is never modified by execution of the called function.
2) The formal parameter is never modified by execution of the called function.
3) The actual parameter must be a variable.
4) 1, 2 and 3 are all true.
5) 1, 2, and 3 are all false.
6) Only 1 and 2 are true.
7) Only 1 and 3 are true.
8) Only 2 and 3 are true.
9) None of these

9. Which of the following statements are true when a parameter is passed by reference?

1) The actual parameter can be modified by execution of the called function.
2) The formal parameter can be modified by execution of the called function.
3) The actual parameter cannot be a variable.
4) 1, 2 and 3 are all true.
5) 1, 2, and 3 are all false.
6) Only 1 and 2 are true.
7) Only 1 and 3 are true.
8) Only 2 and 3 are true.
9) None of these

10. Which of the following statements are true when a parameter is passed by constant reference?

1) The actual parameter can be modified by execution of the called function.
2) The formal parameter can be modified by execution of the called function.
3) The actual parameter cannot be a variable.
4) 1, 2 and 3 are all true.
5) 1, 2, and 3 are all false.
6) Only 1 and 2 are true.
7) Only 1 and 3 are true.
8) Only 2 and 3 are true.
9) None of these

11. If an ampersand ('&') is not attached to the data type of a formal parameter, then the corresponding actual parameter can be:

1) a constant  4) All of these  7) None of these
2) a variable name  5) 1 and 2 only
3) an arbitrary expression  6) 2 and 3 only
12. Given the function prototype and declarations:

```cpp
float Fix(int& N, float X);
int someInt = 10;
float someFloat = 4.3;
```

which of the following function calls would be syntactically correct?

1) `Fix(someInt, 6.85);`
2) `someFloat = Fix(24, 6.85);`
3) `someFloat = 0.3 * Fix(someInt, 6.85);`
4) `Fix(someInt + 5, someFloat);`

5) all of the above  
6) 1 and 3 only  
7) 2 and 4 only  
8) None of these

13. For the function definition

```cpp
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

14. 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

15. 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? Disregard formatting.

```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) `myInt = 20` and `myDble = 23.5`  
6) `myInt = 40` and `myDble = 23.5`  
7) None of these
16. For the function definition

```cpp
tvoid 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

17. Consider the function definition

```cpp
tvoid 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
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

18. Given the function definition

```cpp
tvoid SomeFunc( ... ) {
    float Alpha;
    ...
}
```

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

1) The memory allocated to Alpha is deallocated when the function returns.
2) No parameter in the function heading can also be named Alpha.
3) The value of Alpha is unknown at the moment control enters the function.
4) Alpha can be accessed directly from code outside the function.
5) None of these are false.
19. This question demonstrates the hazard of choosing inappropriate parameter-passing mechanisms. Given the function definition

```cpp
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?

```cpp
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

20. This program illustrates some of the hazards of using global variables. What is the output of the following program?

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

void Try(int& a, int b);
int x, y, z;

int main() {
    x = 1;
    y = 2;
    z = 3;
    Try(y, x);
    cout << x << ' ' << y << ' ' << z << endl;
    return 0;
}

void Try(int& a, int b) {
    int x;
    x = a + 2;
    a = a * 3;
    b = x + a;
}
```

1) 10 6 3
2) 10 2 3
3) 1 2 3
4) 1 6 3
5) None of these