

C201 Review Questions #4

T(rue) or F(alse) Talk with other people if want to.

- 1)___ C++ has several storage classes, including auto, static, external and register.
- 2)___ A function that returns void should have a name that conveys action.
- 3)___ An value parameter is used to pass information back and forth from a function.
- 4)___ A function can change the value of an in parameter without fear of side effects.
- 5)___ The actual parameter list needs to have the same number of parameters as formal list.
- 6)___ One should never write functions that are not logically coherent.
- 7)___ Given a reference parameter the corresponding actual parameter may be a constant.
- 8)___ Local static variables are initialized every time a function is called.
- 9)___ A void returning function cannot have the "return;" statement in function body.
- 10)___ Programmers should exercise care when using ">>" to read numeric values.

11) What's wrong with the following main() ?

```
int main()  
{  
    Menu();  
    return 0;  
}
```

12) Which of the following is preferable?

- a) Writing a function that determines the largest proper divisor of an int N and then displays this proper divisor.
- b) Writing a function that returns the largest proper divisor and produces no output.

13) Given the prototype “void FindSumOfDigits (int &Sum, int N);”, determine which of the calls below would be flagged as an error. Assume that S and I have been properly declared as an int.

- ____a) FindSumOfDigits (S);
____b) FindSumOfDigits (S, 1234);
____c) FindSumOfDigits (S + 1, 1234);
____d) FindSumOfDigits (S, S);
____e) FindSumOfDigits (S, I);

14) The function below is supposed to find the sum of the digits of an integer N and it actually does return the correct sum via the parameter Sum. However, the function is not properly designed. Explain.

```
void FindSumOfDigits (int &Sum, int &N, int Copy)  
{  
    Copy = N;  
    Sum = 0;  
    while ( Copy != 0 )  
    {  
        Sum += (Copy % 10);  
        Copy /= 10;  
    }  
}
```