True (A)   False (B)  (2 pts each):

1. A program’s comments should connect the program code to the problem being solved.

2. The value of count is 0; limit is 10. Evaluate:
   \((\text{count} \neq 0) \land \land (\text{limit} < 20)\)

3. Suppose we have these declarations,
   \[
   \text{int } x = -1, \ y = 0, \ z = 1;
   \]
   This Boolean expression is correct and it does what the programmer intends.
   \[x < y < z\]

4. In a do-while loop, the boolean expression is executed before each execution of the loop body.

5. A break statement is used in switches only.

6. There is only one kind of parameter passing in C++, namely call-by-value.

7. A variable declared within a function block is said to be local to the function.

8. Consider two blocks, one within another. C++ prohibits an identifier to be declared as a variable in each of these blocks.

9. Curly braces are required around a single statement contained in a control structure.
10. Creating functions requires you to take a general task and find a more specific category of task that it represents.

11. OOP is an acronym that means Object Oriented Programming.

12. The names \( x \), \( y \), and \( z \) are satisfactory variable names for the lengths of the legs and hypotenuse of a triangle.

13. The range of values for an \texttt{int} variable is from about 0 to \(+2\) billion.

14. A C++ variable declaration includes a name and data type.

15. The standard convention in C++ is to name variables with lower case letters and name constants using all upper case letters.

16. Decomposition requires dividing a problem into smaller problems.

17. C++ uses // for comments.

18. In C++ you can assign an expression of type \texttt{int} to a variable of type \texttt{double} with no problem.

19. Code, within the same block, after a return or a call to the exit function, will be executed.

20. In C++, boolean values are represented only with the int values 0 for \texttt{false} and 1 for \texttt{true}. 
Multiple Choice (3 pts each):

21. A semicolon does not (usually) go after these with the exception of
   a) while(condition)
   b) if(condition)
   c) an expression to make it a statement
   d) int main()
   e) none of the above

22. An r-value is
   a) an expression that can be only placed on the right of any operator such as +, *, / etc.
   b) can never be assigned a value
   c) can have a value fetched from it
   d) is designed for use by a right-handed person.

23. Concerning return statements that functions can have:
   a) Value returning functions can have the statement return;
   b) void functions can have the statement return void;
   c) void functions must have a return; statement, with no argument.
   d) void functions may terminate using a return; statement without an argument, or they may have no return statement at all, terminating by falling off the end of the function block.
   e) Value return functions are allowed to have no return statement at all.

24. In the expression \((j > 0 && (j+1 == 10))\), which operator executes last?
   a) >
   b) &&
   c) +
   d) ==
25. In C++, which of the following is a legal identifier?
   a) Xyz9
   b) $Xyz
   c) x+yz
   d) xy_z
   e) a and d

26. Which is not true about C++ predefined functions
   a) are usually provided in libraries
   b) make C++ harder than necessary.
   c) must #include the proper header file
   d) are usually provided with the C++ compiler

27. What does the & in front of a variable name located outside function parameters represent in C++?
   a) contents of
   b) logical and
   c) address of
   d) inverse of

28. A void function can have
   a) no arguments
   b) as many arguments as the programmer wishes
   c) no more than 3 arguments
   d) exactly one argument
29. When you don’t recall operator precedence, which shouldn’t you do
   a) Look in a table of precedence
   b) Guess
   c) Use parentheses
   d) Experiment with the compiler
   e) None of the above

30. Here is a collection of if and if-else statements with semicolons in various places.
   Assume all variables have been declared and initialized. Which of these is correct
   a) if ( a > b );
      a = b;
      else
      b = a;
   b) if(a > b )
      a = b;
      else;
      b = a;
   c) if(a > b )
      a = b;
      else
      b = a;
   d) if(a > b)
      a = b
      else
      b = a;
   e) b and c
31. Given the function definition, which of the following are correct?

```c
double func(int n, double d)
{
    int j = n;
    double sum = 0;
    while( j >= 0)
    {
        sum += d;
        -j;
    }
    return sum;
}
```

With arguments 7 and 2.0

a) returns 7*2  
b) returns 7+2  
c) returns 7!  
d) There is a syntax error in the program so it won’t run.  
e) It compiles but computes none of these.

32. In a while loop, when a break statement is encountered, an immediate transfer of control is made to

a) the expression part of the while loop  
b) a goto statement  
c) the else clause  
d) the statement beyond the end of the while statement.  
e) none of these
33. What is the output of the following, if it were embedded in an otherwise correct and complete program and run?

```cpp
int x = 10;
while (x > 0)
{
    cout << x << " ";
    x = x + 3;
}
cout << endl;
```

a) 10 13 16 19 . . .
b) The compiler detects that this will be an infinite loop, so it does not compile.
c) This is an infinite loop.
d) 0 3 6 9.
e) Both a and c

34. This question asks about nesting of if, if-else, switch, while, do-while, and for statements:

a) These constructs may not be nested in at all.
b) These constructs may be nested in any way that meets the needs of algorithms the programmer is coding.
c) Only control constructs of a given kind may be nested (while loops within while loops; if-else within if-else etc.)
d) The question does not make sense in C++.

35. The statements int x = 1; int y; y = x++;

a) Assign y the value 2;
b) Change the value of x to 2.
c) Assign y the value 0;
d) Assign y the value 1;
e) Both b and d.
Here is a small program. Which of the statements about this code is not correct?

```cpp
#include <iostream>
const double NUM = 2.9345358;
double num = 3;
double numTimes(int x);
int main( )
{
    using namespace std;
    int value;
    cout << "Enter a value, I’ll multiply it by “ << NUM << endl;
    cin >> value;
    cout << "You entered “ << value << “ NUM times this is “ << numTimes(value) << endl;
    return 0;
}
double numTimes(int x)
{
    double d;
    d = NUM * x;
    return d;
}
```
a) The variable x is a parameter in function numTimes
b) The variable value is an argument in a call to numTimes.
c) The line double numTimes(int x); is a function definition.
d) The line return d; in the function numTimes is necessary.
e) The x in the function declaration is not needed.
37. Consider the following function and code segment.

```c
void One( int first, int & second )
{
    first = 17;
    second = first + 1;
}

int main()
{
    // other code ...
    int j = 4;
    int k = 3;
    One(j, k);
    // other code ..
}
```

After the call to `One(j, k);` what are the values of `j` and `k`?

a) j == 4, k == 3;
b) j == 17, k == 18;
c) j == 4, k == 18;
d) j == 17, k == 3;

38. Which of the following function declarations with default arguments are correct?

a) void g(int length, int width=1, int height);
b) void g(int length = 1, int width, int height);
c) void g(int length, int width = 1, int height = 1);
d) void g(int length = 1, int width = 1, int height);
Given the function, and the `main` function calling it: What is the output of the following code if you omit the ampersand (&) from the first parameter, but not from the second parameter? (You are to assume this code is embedded in a correct function that calls it):

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

void func(int & x, int & y)
{
    int t = x;
    x = y;
    y = t;
}

int main()
{
    int u = 3; v = 4;
    // ...
    cout << u << " " << v << endl;
    func (u, v)
    cout << u << " " << v << endl;
    // ...
}
```

a) 3 4
   3 3
b) 3 4
   4 3
c) 3 4
   3 4
d) 3 4
   4 4
e) none of the above.
40. Which of the following overloading will be invoked by this call?
\[ g(1, 2); \]

a) int g(int count, double value);
b) void g(double value, int count);
c) void g(int value, int count);
d) Neither, the compiler cannot decide which of these to use.

Extra Credit (5 pts each):

Name: ____________________________

1. What is the logical error in this code, and how would you fix it?

```cpp
#include <iostream>
int main() {
    int num;
    std::cout << "Enter a number: ";
    std::cin >> num;
    if (num % 2 == 0) {
        if (num % 3 == 0)
            std::cout << "The number is divisible by 6!\n";
        else
            std::cout << "The number is odd!\n";
    }
}
```

2. What is the output from the following code?

```cpp
#include <iostream>
int main() {
    int num, total = 25;
    for (num = 1; num <= (total / 2); num++) {
        total -= num;
        std::cout << total << "   " << num << std::endl;
    }
}
```