Closed notes, book and neighbor. **If you have any questions ask them.**

**Notes:**
- **Segment of code** – necessary C++ statements to perform the action described – not a complete program
- **Program** – a complete C++ program – what you have been writing in lab.

Write clearly and make sure the case of a letter is clear since C++ is case sensitive. **Unless otherwise noted, assume a single space between all words.**

For this test the two-character sequence \n is to be taken to mean the newline character.

There are no INTENTIONAL syntax errors. Assume that all code in this exam will compile. There may be logic errors in some of the code.

**Multiple Choice (Questions 1 – 13) 26 Points**
Select all correct answers (multiple correct answers are possible)

1) A(n) ____________________ is a function that returns a function value to its caller and is invoked in an expression.
   A) Value-returning function  B) Main Function  C) Subprogram
   D) Void Function  E) None of these

2) How many **function values** does a **void** function have?
   
   A) 0  B) 1  C) As many as necessary  
   D) 2  E) 3  F) None of these

3) **Circle all of the following that are examples of event-controlled loops:**
   
   A) Count-Controlled  B) Sentinel-Controlled  C) Flag-Controlled
   D) Previous-Value  E) End-Of-File Controlled  F) All of these

4) **An individual pass through, or repetition of, the body of a loop is called a(n) _____.**
   
   A) Loop test  B) Iteration  C) Termination condition
   D) Priming read  E) None of the above
5) **Value parameters (passing by value)** are used if a parameters data flow is:

A) one-way, into the function  
B) one-way, out of the function  
C) two-way, into and out of the function  
D) None of these

6) **Reference parameters (passing by reference)** are used if a parameters data flow is:

A) one-way, into the function  
B) one-way, out of the function  
C) two-way, into and out of the function  
D) None of these

7) A ______________ loop is a loop that executes a specified number of times.

A) While  B) Count-Controlled  C) Looping  
D) Event-Controlled  E) None of These

8) A(n) ______________ loop is a loop that terminates when something happens inside the loop body to signal that the loop should be exited.

A) Sequence  B) Selection  C) Event-Controlled  
D) Count-Controlled  E) None of These

9) A(n) ________________ is a variable used in a function call.

A) Function Call  B) Reference  C) Parameter  D) Argument  E) None of These

10) A(n) ________________ parameter is a parameter that receives a copy of the value of the corresponding argument.

A) Function  B) Value  C) Variable  D) Reference  E) None of these

11) A(n) ________________ parameter is a parameter that receives the location (memory address) of the caller’s argument.

A) Function  B) Value  C) Variable  D) Reference  E) None of these

12) ________________ is the precedence that a local identifier in a function has over a global identifier with the same name.

A) Scope  B) Non-Local Identifier  C) Name Precedence  
D) Local Identifier  E) None of these
13) With respect to a given block, a ________________ is any identifier declared outside that block.

A) Scope   B) Non-Local Identifier   C) Name Precedence
D) Local Identifier   E) None of these

14) (2 pts) What are the three **logical** operators for C++ (show the symbols)?

______________________  _______________________  ______________________

15) (6 pts) The **void** function named **GetNums** has two parameters:

- a pass-by-value parameter named **x** of type **float**
- a pass-by-reference parameter named **num** of type **int**.

Write a valid **function prototype** and **function heading** for the function **GetNums**?

**Function Prototype:**

**Function Heading:**

16) (6 pts) Assume that the Boolean variables **X** and **Y** both have the value **true**. What is the Boolean value of the following expressions?

a) ( !(X || !Y) && ( X && (Y || !Y)) ) || X ___________________

b) (Y && (X || !X)) ______________________

c) (Y || !X && X || !Y) && !X && Y _____________________
17) (6 pts) For the operators shown below, CLEARLY indicate if the operator is:

- Relational (use an R), Logical (use an L) or Neither (use an N).

**NOTE:** that there are no spaces between characters even though it may appear that there is a blank in some of the operators

a) >= _______ b) ?? _______ c) & & _______ d) >> _______

d) >! _______ e) ! _______ f) = ! _______ g) != _______

h) =< _______ i) == _______ j) <= _______ k) = _______

18) (12 pts) True/False questions. Select T for true and F for false.

T    F   a) An if statement cannot occur inside of another if statement.

T    F   b) The body of a for loop executes one or more times.

T    F   c) Local identifiers have name precedence over global identifiers.

T    F   d) The use of the statement: return; is not valid in a void function.

T    F   e) A logical expression can consist of a single relational expression?

T    F   f) Static variables in a function maintain their value from function call to function call.

T    F   g) A compile error results when the DataType defining a functions value type is omitted.

T    F   h) The statement for(;;); is a valid C++ statement.

T    F   i) Value parameters receive a copy of the arguments value.

T    F   j) A function call can contain more arguments than the number of parameters in the corresponding function heading.

T    F   k) In sentinel-controlled loops, the sentinel is a value expected as normal input?

T    F   l) An argument corresponding to a reference parameter can be a constant or arbitrary expression?
19) **(8 pts)** In the following code segment, all variables are integers.

```cpp
int maxHeight = 50;
int maxWeight = 30;
if (height < maxHeight)
    if (weight < maxWeight)
        cout << "Message #1\n";
    else
        cout << "Message #2\n";
else
    if (weight < maxWeight)
        cout << "Message #3\n";
    else
        cout << "Message #4\n";
```

If the variables `height` and `weight` have the following values, what is the output of the above segment of code when it is executed?

a) height = 50 , weight = 20

b) height = 30 , weight = 30

20) **(6 pts)** Consider the following **segment of code**

```cpp
int loop = 0;
while (loop < 7)
{
    cout << "Hello" << endl;
    loop = loop + 2;
}
```

Rewrite the above **code segment** as a **for** loop such that the same output is obtained.
21) (8 pts) Finish the segment of code below by using an if-then-else-if statement to print out the following information based on the value of grade:

   “A” if \texttt{grade} has a value of 90,
   “B” if \texttt{grade} has a value of 80,
   “Error” if \texttt{grade} is not 90 or 80.

\textbf{Note:} The output is to be only one value – A, B or Error.

```cpp
int grade;
cout << "Enter in the grade value: ";
cin >> grade;
// if-then-else-if statement follows this comment.
```

22) (8 pts) For the following code segment, write out what is printed to the screen. \textbf{Place a single character in each box, skip a box to indicate a space, and skip a row to indicate a blank line.}

```cpp
int loop_b;
int loop_a = 0;

while ( loop_a < 4 )
{
    for (loop_b = 0; loop_b <= loop_a; loop_b++)
        cout << loop_b;
    loop_a++;
    cout << loop_a << endl;
}
```
23) (6 pts) What is the output for the code segment below:

```cpp
int count = 5;
bool finished = false;
do {
    cout << count << "-";  // note no line termination
    count--;
    finished = count < 0;
} while (!finished);
```

24) (10 pts) Finish the code segment below that reads any character from the standard input stream (cin) until an A is read. The segment is to print out how many characters were entered before the A (do not count the A). A new line (shown below as \n) can be entered and it counts as a single character.

Example of a typical input line: Hello\nWorld\nA
Example of output for the line: 12 characters were entered

```cpp
char ch;  // holds the character read from the input stream
int numChar = 0;  // count of number of characters entered
```
25) (8 pts) What is the output of the following program:  
For the following code segment, write out what is printed to the screen.  
Place a single character in each box, skip a box to indicate a space, and skip a row to indicate a blank line.

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

void Test();

int main()
{
    Test();  // First Call
    Test();  // Second Call
    Test();  // Third Call
    Test();  // Fourth Call
    return 0;
}

void Test()
{
    static int i = 5;
    int j = 5;
    i++; j--;    // increment i and decrement j
    cout << i << "—" << j << endl;
}
```

26) (8 pts) When the program shown below is executed, what is the output to the screen? This problem deals with the scope of a variable in a program, and the order of execution of statements. There will be a total of 4 lines written to the screen from this program. Analysis of this program requires some thought.

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

void function_A(int&);
void function_B(int);
int number = 4;
int main()
{
    int number = 3;
    function_A(number);
    cout << "number in main is: " << number << endl;
    function_B(number);
    return 0;
}
void function_A(int& num)
{
    int number = 2;
    num = num +2;
    cout << "number in function A is: " << number << endl;
}
void function_B(int sum)
{
    cout << "number in function B is: " << number << endl;
    sum = sum + 1;
    cout << "sum in function B is: " << sum << endl;
}
```

The output for this program is as indicated below. In the blank to the left of the lines, place 1,2,3 or 4 to indicate the order the statements are printed (1 for first, 4 for last). In the blank at the end of the line, put in the output value.

_____ number in main is: _____
_____ number in function A is: _____
_____ number in function B is: _____
_____ sum in function B is: _____
27) (10 pts) Finish the program below by adding a value-returning function specified below. Add only a function prototype, function call statement and function definition to the following program.

The name of the float value-returning function is CalcValue. The function has two float parameters. The function is to divide the first parameter by the second parameter and return, as its function value, the floating point result of the division.

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

// Place the function prototype below this line

int main()
{
    float sum = 4563;   // number to be divided
    float number = 20; // number that divides into sum
    float avg;       // holds value returned by the function

    // Place the function call statement below this line

    return 0;
} // Place the function definition below this line
```
28) (10 pts) Finish the segment of code below so that it counts the number of Empty lines in an input file. Hint: An empty line when read is equal to a null string which is represented as "". Use the getline function to read each line of the file. Output to the terminal the number of empty lines contained in the file.

```cpp
int numLines=0;
string line;
ifstream in;
in.open("In.txt");
// put rest of code below this line.
// do not declare any more variables
```
29) (10 pts) **Write a void function** that opens an input file in the following manner:
1) Prompts the user for a file name, reads the name entered and echo prints the file name,
2) Opens the file name provided and associates it with an input file stream
3) If the file did not open:
   a) Print out an appropriate error message
   b) Clear the input file stream (which will reset it)
   c) Repeat 1, 2 and 3 until a file name is successfully entered

**This function requires a single parameter – an input file stream**
The above steps outline what is necessary for a while loop. However, a do-while loop is more compact if written properly.
Bonus #1 (+5 pts)
The following program is executed. The user enters the integer “3” when prompted for a number. What is the output to the screen? Just fill in the blanks indicated in the output line shown

Be careful on this problem. Think about what is being performed with which variables

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

void Summation( int& num, int& result );

int main ()
{
    int number = 0;
    int result = 10;
    cout << "Enter an integer between 1 and 10: ";
    cin >> number;
    Summation(number, result); // first function call
    Summation(number, result); // second function call
    // show the output for this cout statement only
    cout << "Summation for " << number << " is: " << result << endl;
    return 0;
}

void Summation( int& num, int& result )
{
    static int loop = 0;
    for (loop = num; num >= 1; num--)
    {
        result = result + loop;
        num = num - 1;
    }
    return;
}
```

Answer: Summation for ________ is __________

Bonus #2 (+5 pts) What is the output of the following code segment if Input.txt contains the following values: 1 10 4 5 15

```cpp
ifstream inFile;
int value;
inFile.open("Input.txt");
while(inFile) // no priming read, so be careful of end result
{
    inFile >> value;
    if (value < 10)
        continue;
    else
        cout << value << endl;
}
```