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 (where applicable) 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 does not return a function value to its caller and is invoked as a complete, stand-alone statement.
   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) Priming read  
   C) Termination condition
   D) Iteration  
   E) None of the above
5) 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

6) 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

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) Looping   D) Sub-program   E) None of These

9) A(n) ________________ is a variable declared in a function heading.

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++?

_____________________  _______________________  ______________________

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) (10 pts) Match the words with their definitions. Choose the **BEST** definition for each word.

a) Static Variable _____

b) Scope _____

c) Function Prototype _____

d) Lifetime _____

e) Loop _____

f) Event Counter _____

g) Function Call _____

h) Iteration Counter _____

i) Local Variable _____

j) Argument _____

A) Definition is not listed below (may be used more than once).
B) A parameter that receives a copy of the value of the corresponding argument.
C) A variable declared within a block and not accessible outside of that block.
D) The region of a program where it is legal to reference (use) an identifier.

E) The fundamental control structure that allows branches in the flow of control
F) A variable or expression listed in a call to a function.
G) A variable that is incremented when a particular event occurs.
H) The period of time during program execution when an identifier has memory allocated to it.

I) A statement that transfers control to a function.
J) A variable for which memory remains allocated throughout the execution of the entire program.
K) A function declaration without the body of the function.
L) A control structure that causes a statement or group of statements to be executed repeatedly

20) (6 pts) What is the output for the code segment below:

```cpp
int count = 0;
bool finished = false;
while (!finished)
{
    finished = count > 5
    count++;
    cout << count << “-”;
}
```
21) (6 pts) In the following code segment, all variables are integers.

```cpp
maxHeight = 50;
maxWeight = 30;
if (height < maxHeight)
    if (weight < maxWeight)
        cout << “Message #1
”;
    else
        cout << “Message #2
”;
else
    if (weight < maxWeight)
        cout << “Message #3
”;
    else
        cout << “Message #4
”;
```

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 = 50

22) (6 pts) Consider the following segment of code

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

Rewrite the above code segment as a `for` loop such that the same output is obtained.
23) (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 grade has a value of 90,
“B” if grade has a value of 80,
“C” if grade has a value of 70, and
“Error” if grade is not 90, 80 or 70.

Note: The output is to be only one value – A, B, C or Error.

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

24) (8 pts) A program requires a user to enter a positive (greater than zero) integer. Use a while loop to continually prompt the user to enter a positive integer until one is entered. If a negative integer is entered, print out an error message and prompt the user again for a positive integer. Assume that integer values only are entered. Declare any necessary variables.
25) (6 pts) 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();
    Test();
    Test();
    return 0;
}
void Test()
{
    static int i = 5;
    int j = 5;
    i++; j--; // increment i and decrement j
    cout << i << "—" << j << endl;
}
```

26) (6 pts) 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
int loop_b;
int loop_a = 3;

while (loop_a > 0)
{
    for (loop_b = 0; loop_b <= loop_a; loop_b++)
        cout << loop_b;
    loop_a--;
    cout << loop_a << endl;
}
```
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 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 value;  // holds value returned by the function

    // Place the function call statement below this line

    return 0;
}
// Place the function definition below this line
```
28) (8 pts) Rewrite the value-returning function definition below as a void function definition such that the caller of the function still has access to the value returned by the value-returning function.

```c
float Calculate (float num1, float num2)
{
    float result;
    result = (num1 + num2)*5.0;
    return result;
}
```

29) (10 pts) Write a Boolean value-returning function definition called OpenInputFile.

- The function prompts the user for the name of the input file, reads the name of the input file and then tries to open the input file (use `.c_str()`).
- The function returns a value of `true` if the input file was successfully open.
- A value of `false` is returned if the input file was not successfully opened.

- The function requires two parameters. The first parameter represents the name of the input file that is entered for opening, and the name of this file must be available for use in `main()`.
- The second parameter represents the input stream to use.
30) (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.

```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: ____
The following program is executed. **The user enters the integer “6” when prompted for a number.** What is the output to the screen? Just fill in the blanks indicated in the output line shown.

`Bonus Question (+8 pts)`

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 = -1;
    int result = -1;
    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 __________
```