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.**

1) **(2 pts)** A function that does not return a function value is known as what kind of function?

   A) Value returning  B) Reference Parameter  C) Empty
   D) Void  E) Expression less  F) None of these

2) **(2 pts)** 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) **(2 pts)** Circle all of the following that are examples of event-controlled loops:

   A) End-Of-File Controlled  B) Sentinel-Controlled  C) Flag-Controlled
   D) Count-Controlled  E) Previous-Value  F) All of these

4) **(2 pts)** Which operations below are not allowable aggregate operations on *structures*?

   A) Return as a functions return value  B) Assignment  C) Arithmetic
   D) Input/Output (cin, cout statements)  E) Comparison  F) Argument pass by value to a function
   G) Argument pass by reference to a function  H) None of these
5) (2 pts) 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) A and B  
E) B and C  
F) None of these

6) (2 pts) 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) A and B  
E) B and C  
F) None of these

7) (3 pts) What are the three logical operators for C++?

_____________________  _______________________  ________________________

8) (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) = _______
9) (15 pts) True/False questions. Select T for true and F for false.

T  F  a) Reference parameters receive the memory location of an argument.
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 variable declared as a union holds a value for only one member at any given time?
T  F  h) Members of a structure can be different DataTypes.
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 unexpected as normal input?
T  F  l) An argument corresponding to a reference parameter can be a constant or arbitrary expression?
T  F  m) An if statement cannot occur inside of another if statement.
T  F  n) A compile error results when the DataType defining a functions value type is omitted.
T  F  o) The statement for(;;); is a valid C++ statement.
10) (12 pts) Match the words with their definitions. Choose the BEST definition for each word.

<table>
<thead>
<tr>
<th>Word</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static Variable</td>
<td>A) Definition is not listed below (may be used more than once).</td>
</tr>
<tr>
<td>Member Selector</td>
<td>B) The expression used to access components of a struct variable.</td>
</tr>
<tr>
<td>Function Prototype</td>
<td>C) A parameter that receives a copy of the value of the corresponding argument.</td>
</tr>
<tr>
<td>Lifetime</td>
<td>D) A parameter that receives the memory location of the corresponding argument.</td>
</tr>
<tr>
<td>Structure</td>
<td>E) A statement that transfers control to a function.</td>
</tr>
<tr>
<td>Iteration Counter</td>
<td>F) A variable for which memory remains allocated throughout the execution of the entire program.</td>
</tr>
<tr>
<td>Event Counter</td>
<td>G) A structured collection of components, all of the same DataType that is given a single name.</td>
</tr>
<tr>
<td>Function Call</td>
<td>H) A structure in which at least one of the members is itself a structure.</td>
</tr>
<tr>
<td>Hierarchical Structure</td>
<td>I) The expression whose value determines which switch label is selected.</td>
</tr>
<tr>
<td>Local Variable</td>
<td>J) A variable or expression listed in a call to a function.</td>
</tr>
<tr>
<td>Reference Parameter</td>
<td>K) A variable that is incremented when a particular event occurs.</td>
</tr>
<tr>
<td>Scope</td>
<td>L) The period of time during program execution when an identifier has memory allocated to it.</td>
</tr>
<tr>
<td>A) Definition is not listed below (may be used more than once).</td>
<td>B) The expression used to access components of a struct variable.</td>
</tr>
<tr>
<td>C) A parameter that receives a copy of the value of the corresponding argument.</td>
<td>D) A parameter that receives the memory location of the corresponding argument.</td>
</tr>
</tbody>
</table>
11) (8 pts) In the following code segment, all variables are integers.

```cpp
maxHeight = 20;
maxWeight = 50;
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 = 40`

b) `height = 30 , weight = 50`

12) (8 pts) Write the structure declaration for a structure named `LogInfo` containing the following members:

- an integer variable representing the log entry number
- a string variable representing the name of the person making the log entry
- a floating-point variable indicating the cost of the entry
- A `Date` variable indicating the day of the entry where `Date` is a structure already defined
13) (10 pts) Consider the following structure declarations when answering the questions below.

```c
struct Time {
    int hour;
    int minute;
    int second;
};
struct Appointment {
    string name;
    string type;
    string day;
    Time duration;
};
```

a) Write a statement that **declares** the identifier `clock` as a variable of DataType `Time`.

b) Write a `cout` statement that will output the value of the `hour` member of `clock`.

c) Write a statement that **declares** the identifier `date` as a variable of DataType `Appointment`.

d) Write a statement that **assigns** a value of “Dr. Fixit” to the `name` member of `date`.

e) Write a statement to assign a value of 15 to the `minute` member of `duration`, which is a member of `date`.

14) (6 pts) Consider the following **segment of code**

```c
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.
15) (10 pts)
a) What is the output for the code segment below:

```cpp
int count = 10;
bool notFinished = true;
while (notFinished)
{
    if (count > 5)
    {
        notFinished = true;
    }
    else
    {
        notFinished = false;
        count--;
    }
}
// This is the only line that outputs information
cout << "count is: " << count << endl;
```

**count is: __________**

b) *If necessary*, rewrite the code segment in part a (by correcting any possible logic errors) so that the value printed out is 5. You can modify lines that are present only – *do not delete or add any more lines* to the code segment. If no modifications are required, write "No modifications required". A partial copy of the code has been provided to reduce the amount of writing necessary.

```cpp
int count = 10;
bool notFinished = true;
while
{
    if notFinished
    
    else
    notFinished

// This is the only line that outputs information
cout << "count is: " << count << endl;
```
16) (8 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 = 0;

while ( loop_a <= 2 )
{
    for (loop_b = 0; loop_b <= loop_a; loop_b++)
        cout << loop_b;
    loop_a++;
    cout << loop_a << endl;
}
```
17) (8 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.

```
#include <iostream>
using namespace std;
void Test();
int main()
{
    Test();
    Test();
    Test();
    return 0;
}
void Test()
{
    static int i = 0;
    static int j = 5;
    j = 0;
    i++;
    j++;
    cout << i << "-" << j << endl;
}
```
18) (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.
```
19) (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. ➔ No other information (other than what is necessary for the 3 items above) is to be added to the program (i.e. variables) ⇓

The name of the value-returning function is InitStruct. The function has one integer parameter. The function is to initialize each member of a structure variable of data type Date with the value of the integer parameter. The function returns, as its function value, the initialized structure variable.

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

struct Date
{
    int month;
    int day;
    int year;
};
// Place the function prototype below this line

int main()
{
    Date date;
    int value = -1;

    // Place the function call statement below this line
    // information stored in date must be available in main

    return 0;
} // Place the function definition below this line
```
20) (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;
}
```

21) (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.
22) (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_B(number);  
    cout << "number in main is: " << number << endl;  
    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)  
{  
    function_A(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: _____
Bonus Question (+5 pts)

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.

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;
    result = 0;
    for (loop = num; num >= 1; num--) //
    {
        result = result + loop;
        num = num -1;
    }
    return;
}

Answer: Summation for _________ is __________