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) Which of the following can be used as a switch expression? (Circle all correct answers):
   A) floating point variable   B) char variable          C) bool variable
   D) string constant           E) integer variable       F) None of These

4) (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
5) (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

6) (2 pts) Reference parameters (passing by reference) are used if a parameters data flow is:

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

7) (2 pts) Value parameters (passing by value) are used if a parameters data flow is:

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

8) (3 pts) What are the three logical operators for C++?

_____________________  _______________________  ______________________

9) (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) = _______
10) (5 pts) In the following code segment, all variables are integers.

```c
maxHeight = 20;
maxWeight = 50;
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?

height = 50 , weight = 40

11) (4 pts) Write the type declaration for a struct `DataType` named `LogType` 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
12) (10 pts) Match the words with their definitions. Choose the **BEST** definition for each word.

<table>
<thead>
<tr>
<th>a) Static Variable</th>
<th>f) Iteration Counter</th>
</tr>
</thead>
<tbody>
<tr>
<td>b) Member Selector</td>
<td>g) Event Counter</td>
</tr>
<tr>
<td>c) Value Parameter</td>
<td>h) Function Call</td>
</tr>
<tr>
<td>d) Lifetime</td>
<td>i) Hierarchical Structure</td>
</tr>
<tr>
<td>e) Structure</td>
<td>j) Argument</td>
</tr>
</tbody>
</table>

**Definitions:**

A) Definition is not listed below (may be used more than once).
B) The expression used to access components of a struct variable.
C) A parameter that receives a copy of the value of the corresponding argument.
D) A parameter that receives the memory location of the corresponding argument.
E) A statement that transfers control to a function.
F) A variable for which memory remains allocated throughout the execution of the entire program.
G) A structured collection of components, all of the same DataType that is given a single name.
H) A structure in which at least one of the members is itself a structure.
I) The expression whose value determines which switch label is selected.
J) A variable or expression listed in a call to a function.
K) A variable that is incremented when a particular event occurs.
L) The period of time during program execution when an identifier has memory allocated to it.

13) (8 pts) Match the words with their definitions. Choose the **BEST** definition for each word.

<table>
<thead>
<tr>
<th>a) Function Prototype</th>
<th>e) Loop</th>
</tr>
</thead>
<tbody>
<tr>
<td>b) Scope</td>
<td>f) Reference Parameter</td>
</tr>
<tr>
<td>c) Aggregate Operation</td>
<td>g) Switch Expression</td>
</tr>
<tr>
<td>d) Name Precedence</td>
<td>h) Local Variable</td>
</tr>
</tbody>
</table>

**Definitions:**

A) Definition is not listed below (may be used more than once)
B) The precedence a local identifier in a function has over a global identifier with the same name.
C) A component of a structure.
D) A Function declaration without the body of the function.
E) The expression whose value determines which switch label is selected.
F) A control structure that causes a statement or group of statements to be executed repeatedly.
G) A variable declared within a block and not accessible outside of that block.
H) A parameter that receives a copy of the value of the corresponding argument.
I) The region of a program where it is legal to reference (use) an identifier.
J) The expression used to access components of a struct variable.
K) An operation on a data structure as a whole, as opposed to an operation on an individual component of the data structure.
14) (18 pts) True/False questions. Select T for true and F for false.

T  F  a) Value parameters receive the memory location of an argument.
T  F  b) Reference parameters receive a copy of the arguments value.
T  F  c) Global identifiers have name precedence over local identifiers.
T  F  d) The use of the statement: `return;` is valid in a void function.
T  F  e) Members of a structure must be of the same DataType.
T  F  f) Static variables in a function maintain their value from function call to function call.
T  F  g) The body of a `for` loop executes zero or more times.
T  F  h) When a `break` statement is executed, the outermost loop in which it appears is exited.
T  F  i) A `switch` statement MUST have a `default switch label`.
T  F  j) A function call cannot contain more arguments than the number of parameters in the corresponding function heading.
T  F  k) A logical expression can consist of a single Boolean variable?
T  F  l) An argument corresponding to a reference parameter cannot be a constant or arbitrary expression?
T  F  m) In sentinel-controlled loops, the sentinel must be a value expected as normal input?
T  F  n) A compile error results when the DataType defining a functions value type is omitted.
T  F  o) The statement `for(;;);` is an invalid C++ statement.
T  F  p) A `switch` statement can have at most one `default label`.
T  F  q) All possible values for the `switch expression` must be included among the `case labels` for a given `switch` statement.
T  F  r) A variable declared as a union holds a value for only one member at any given time?
15) (10 pts) Consider the following structure declarations when answering the questions below.

```c
struct Date
{
    int day;
    int month;
    int year;
};
struct Person
{
    string name;
    string city;
    int age;
    Date dob; // dob = date of birth
};
```

a) Write a statement that declares the identifier `birthday` as a variable of DataType `Date`.

b) Write a cout statement that will output the value of the `day` member of `birthday`.

c) Write a statement that declares the identifier `Ron` as a variable of DataType `Person`.

d) Write a statement that assigns a value of “Huntsville” to the `city` member of `Ron`.

e) Write a statement to assign a value of 2006 to the `year` member of `dob`, which is a member of `Ron`. 

```c
```
16) (4 pts) Consider the following segment of code

```cpp
int count;
int sum = 0;
for (count = 5; count > 0; count--)
{
    sum += count;
    cout << sum << "-" << count << endl;
}
```

Rewrite the above code segment as a **while** loop such that the same output is obtained.
17) (8 pts)  
a) What is the output for the code segment below:

```c
int count = 0;
bool notFinished = true;
while (notFinished)
{
    count++;
    if (count > 5)
        notFinished = true;
    else
        notFinished = false;
}
// 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 6. 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.

```c
int count = 0;
bool notFinished = true;

while
{
    count++;
    if
        notFinished
    else
        notFinished
}
// This is the only line that outputs information
cout << "count is: " << count << endl;
```
18) (8 pts) Consider the code segment shown when answering the questions below:

```c++
int number;
cout << “Enter an integer between 1 and 6: “;
cin >> number;
switch(number*3)
{
    case 6:  cout << ‘A’;
    case 4:  cout << ‘B’;
    case 2:  cout << ‘C’;
    case 1:  cout << ‘X’;
    case 3:  cout << ‘Y’;
    case 5:  cout << ‘Z’;
    default: cout << “default” << endl;
}
```

a) What is the output if the user enters 2?

b) What is the output if the user enters 1?

19) (4 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.

```c++
int loop_b;
int loop_a = 0;

while ( loop_a <= 2 )
{
    for (loop_b = 3 - loop_a; loop_b >=0; loop_b--)
        cout << loop_b;
    loop_a++;
    cout << loop_a << endl;
}
```
20) (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;
    static int j;
    i = j = 0;
    i++;
    j++;
    cout << i << "—" << j << endl;
}
```

```
+---+---+---+---+---+---+---+---+---+---+---+---+---+
| i |--| j |   |   |   |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+---+---+---+---+
```
21) (8 pts) The int variable grade is used as the switch expression in a switch statement. Write a code segment using a switch statement that prints the information below

“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 switch statement is to output one line only.
22) (10 pts) Finish the program below by adding a void function specified below. Add only a function prototype, function call statement and function definition to the following program. ➔ No other information is to be added (i.e. variables) ➙

The name of the void function is InitStruct.
The function has two parameters. One is of the struct DataType Date, and one is an integer. The function is to initialize each member of the structure with value of the integer parameter. The information stored in the structure must be available in main() after the function call.

```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
```
23) (6 pts) Rewrite the **void function definition** below as a **value returning function definition** such that the caller of the function still has access to the value returned by the void function.
- **Do not use any reference parameters with the value returning function.**
- **Two value parameters** only are allowed for use with the value returning function.
- **You can add a single variable declaration if necessary.**

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

24) (10 pts) Write a **Boolean value-returning function definition** called **OpenFile**.

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

- The function **requires two parameters**. The first parameter represents the name of the 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.
25) (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;
    cout << "number in main is: " << number << endl;
    function_A(number);
    return 0;
}
void function_A(int num)
{
    int number = 2;
    num = num + 2;
    function_B(num);
    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

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

Answer: Summation for ________ is ________