Name _________________________________

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 – 14) 28 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) What is the output of the following code segment if num has a value of 1? Assume all variables are integers.

```
switch(num)
{
    case 1: cout << “a”;
    case 2: cout << “b”;
    case 3: cout << “c”;
        break;
    default: cout << “end”; 
}
```

A) abcend  B) abc  C) ab  D) a

E) b  F) c  G) cend  H) None of these

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

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

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

Function Prototype: ________________________________

Function Heading: ________________________________

16) (4 pts) What is the output of the following code segment?

```
int score = 85;
if (score <= 90)
    score = score - 5;
if (score <= 80)
    score = score - 10;
if (score > 75)
    score = score - 15;
cout << score << endl;
```
17) (6 pts) consider the following code segment where num is a declared int variable

```cpp
if (num <= 10)
    cout << "less than 10" << endl;
if (num > 15)
    cout << "greater than 15\n";
else if (num < 5)
    cout << "num is " << num << endl;
else
    cout << "greater than 10\n";
```

Spaces above are represented as □. Place a single character in each box, skip a box to indicate a space, skip a row to indicate a blank line.

a) What is the output to the screen when num = 10?

b) What is the output to the screen when num = 4?
18) (12 pts) True/False questions. Select T for true and F for false.

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

T F b) The body of a do-while 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 valid in a void function.

T F e) An argument corresponding to a reference parameter can be a constant or arbitrary expression?

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

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

T F h) The default switch label is required in a switch statement.

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

T F j) A break statement is required in a switch statement

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

T F l) A compile error results when the DataType defining a function's value type is omitted.
19) (6 pts) When the following segment of code is executed, the standard input stream contains the numbers 1 2 2 3 6 7 8 9. What is the output when the code executes

```cpp
int sum = 0;
int number;
do{
    cin >> number;
    cout << sum << "-";  // note no line termination
    sum = sum + number;
}while (number < 6);
```

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

```cpp
int loop = 0;
while (loop < 7)
{
    cout << "Hello" << endl;
    loop = loop + 1;
}
```

Rewrite the above code segment as a for loop such that the same output is obtained.
21) (6 pts) There are three functions shown in the code segment below. Assume all variables and function prototypes have been correctly declared before this segment of code.

    status = Average(sum);
    WriteInfo(outFile, Calc(num));

A) Which function(s) is(are) most likely value-returning function(s)?

B) Which function(s) is(are) most likely void function(s)?

22) (8 pts) Consider the following segment of code

    int number;
    cout << "Enter an integer between 0 and 10: ";
    cin >> number;
    switch(number)
    {
    case  2: cout << 'A';
    case  5: cout << 'B';
    case  6: cout << 'C';
    case 12: cout << 'X';
    case 15: cout << 'Y';
    case 16: cout << 'Z';
        break;
    default: cout << "Default" << endl;
    }

a) What is the output if 4 is entered?

b) What is the output if 6 is entered?
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) (12 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) (10 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.

```c++
#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;
}
```

```
5—5
4—4
3—3
2—2
1—1
0—0

```
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.

```
#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;
    function_A(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) (16 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
```
28) (16 pts) **Write a void function definition** 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.
Extra Credit #2) (5 pts) For the following code segment, what is written to the terminal? **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 k;
int j = 0;

while ( j <= 3)
{
    for (k = j+1; k >= 0; k--)
        cout << k;

    cout << "-" << j << endl;
    j++;
}
```

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Bonus #2 (+5 pts) What is the output of the following code segment if Input.txt contains the following values: 10 10 4 15 5 3

```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;
}
```