This worksheet is for your use during and after lecture. It will not be collected or graded, but I think you will find it a useful tool as you learn C++ and study for the exams. Explain all false answers for the "True or False" questions; in general, show enough work and provide enough explanation so that this sheet is a useful pre-exam review. I will be happy to review your answers with you during office-hours, via Email, or instant messaging.

1. C++, like many programming languages, has *repetition statements* or *looping structures*. Explain precisely what is being repeated when these are used.

   Solution: A sequence of program instructions (AKA a "block of code", or "the looping block").

2. Which repetition structure is most likely to have a "loop counter"?

   Solution: A `for`-loop.

3. How many question marks will be printed to the console?

```
1 for( int i(0); i<5; i++ ) {
2     for( int j(i); j>=1; j-- ) {
3         cout << '?' ;
4     }
5 }
6 cout << endl;
```

Solution:

```
??????????
```

4. Using an `if` statement and the `break` keyword, add code to the following program so that it terminates instead of attempting the `sqrt()` of a negative number.

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main()
6  {
7      // Variable to be read from user
8      double x(0);
9      while( true ) {
10
11         cout << "Enter␣a␣number␣for␣SQRT:␣" <<
12                 endl;
13         cin >> x;
14
15         cout << "The␣SQRT␣of␣" << x << "␣is␣" <<
16                 sqrt(x) << endl;
17
18     }
19
20     return 0;
21 }
```

Solution: At line 14 insert

```
if( x < 0 ) break;
```

5. Write a user input loop that waits until a user enters an upper-case vowel (A, E, I, O, or U).

Solution:

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

int main()
{
    char vowel;
    while( true ) {
        // prompts
        cout << "Enter␣an␣upper␣case␣vowel:␣" << flush;
        // input
        cin >> vowel;
        // check for validity
        if( vowel == 'A' ) break;
        if( vowel == 'E' ) break;
        if( vowel == 'I' ) break;
        if( vowel == 'O' ) break;
        if( vowel == 'U' ) break;
        // error message
        cout << "'" << vowel << "'␣is␣not␣an␣upper␣case␣vowel!" << endl;
    }

    cout << "You␣entered␣'" << vowel << "'" << endl;
    return 0;
}
```

6. What are the four control structures we have studied so far (the `if-then` and `if-then-else` *do not* count as two)? Of the four, one does not use the implicit notion of a Boolean expression in its definition. Which is it?

Solution: `sequences`, `if-then`, `for`-loops, and `while`-loops. `sequences` do not require the use of Boolean expressions.

7. Explain the difference between a sentinel loop and a counting loop. Which C++ repetition structure is most easily used for each?

Solution: A counting loops exectutes for a fixed number of times. A sentinel loop executes until some variable assumes a particular value.

8. Explain the difference between a sentinel loop and a conditional loop.

Solution: A sentinel loop predicates on a single, particular value. A conditional loop executes while a set of conditions are met.