Work on the following exercises in the sequence indicated.

Logging On. Log on with the username and password that you were assigned last week. If you experience any difficulty, let the lab instructor know immediately. Insist that your problem be fixed in the beginning of this lab.

Getting the Lab Instructions. Move to your cs201/lab/lab5 directory with the cd command.

$ cd cs201/lab/lab5

Exercise 1: Think of a Simple Program. -- Independent Work --
Think of a simple task for which you are confident that you can write a program in C++. Whatever task you are thinking of, it should be one that involves some input from the user, and some output to the user. Do not write the C++ program, but type up a plain English description of your simple task, and how to solve, it into a file mytask.txt. Use your usual editor, eg., gedit.

Be conscientious and play by the rules: pick a task that is not identical to one that has already been treated in a lab or lecture. In short, keep it simple but novel.
Exercise 2: Write a Simple Program. -- Independent Work --
Implement in C++ the simple program that you thought about in Exercise 1. Recall the pattern of many of the programs that you have already seen in CS201. Model your program after it ...

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

int main()
{
    // variable declarations as needed;

    // user prompting and reading of input

    // processing of user input, as needed for task;

    // output to the user;

    return 0;
}
```

Exercise 3: A Program To Classify Numbers. Your instructor will help you along a bit with this exercise. Write a program that classifies any integer number entered by the user. The program should be able to tell whether the input number is

1. positive or negative
2. odd or even
3. a square (for example, input number 16 is equal to 4x4)

If multiple classifications apply to an input number, then all those that apply should be listed. For example, if the user enters -16, then the program should report that the number is negative, even, and not a square.

The program should allow the user to enter multiple numbers. One at a time, the numbers are read and their classification reported. The program should terminate only after the user has explicitly indicated his/her wish to quit by typing the sentinel -1.

You should find yourself wanting to use if-else and while statements for this program. Also start your program out with
```cpp
#include <iostream>
#include <cmath>
using namespace std;
```

so that you will be able to use the `sqrt(double x)` function in the `<cmath>` library to take the square root of a number. Another useful function for the exercise is the `floor(double x)` function, which for any `double` value `x` returns the largest integer that is smaller or equal to `x` (e.g., `floor(3.75)` returns value `3.0`.

Compile your program. Once it is without error, run it with multiple sets of inputs. Each time, terminate the input by typing `-1`.

**Extra credit.** Realize the deficiency of the program you just wrote: input `-1` will not be classified, but taken as indicator that the user wants to quit. What if the user means for this number to be classified? If we let the program classify input `-1`, then how is the program going to know when to quit? Figure out a way (keep it simple) to handle this dilemma. Then make your changes and/or additions in a copy of your solution to Exercise 3. Comment the lines of code that are designed to resolve the deficiency. Compile and run for several sets of input numbers.