CSE 202 LAB 2

Do the following exercises in the sequence indicated. Get as much done as you can within the given lab time. When you are done, demonstrate your running program to the lab instructor.

When you seem to have gotten stuck with a task, first verify your spelling. E.g., have you copied a program faithfully, do all your instructions end with ';', etc. If you still cannot locate your error ask the lab instructor for help.

1. Log on. Move to cs202 directory (use 'cd').

2. Copy the programs circle.cpp :

```
/* Program: circle.cpp */

#include "ccc_win.h"

int ccc_win_main()
{
  Point p(1, 3);
  Circle c(p,2.5);
  cwin << p << c;
  c.move(3,3);
  cwin << p << c;
  return 0;
}
```

   Use your favorite editor to do so. Once you have typed in your program, save it with the 'save' command of your editor.

3. When your program contains instructions to do graphics, plain 'g++' is no longer sufficient for compilation. Instead, compile program circle.cpp with the command

   $ Q circle.cpp

   'Q' does more than just compile; it compiles, links, and even runs the executable for you.
   Note the coordinate system used in the window that displays the circle. The lower left corner is (-10, -10) and the upper right corner is at (10, 10). The origin (0,0) are is in the middle of the screen. Unless stated otherwise, this is the coordinated system that is being used in our other graphics programs. You can change the coordinate system with the command *cwin.coord(a,b,c,d)*. The point (a,b) is the top left corner, and (c,d) is the bottom right corner.

4. Compile and run line.cpp. Try changing the coordinate system.

```
/* Program: line.cpp */

#include "ccc_win.h"

int ccc_win_main()
{
  Point p(1, 3);
  Point q(4, 7);
  Line s(p,q);
  cwin << s ;
  s.move(2,5);
  cwin << s ;
  return 0;
}
```

5. Now input an integer value **altitude** and use a loop to write your own graphics program 'fall.cpp' that produces a graphic representation of falling object (e.g. circle). An object has initial downward velocity 0. Each second, it's velocity increases by 32 ft/sec. Plot the descent of the falling object. Note that terminal velocity is 174 ft/sec so the object's velocity cannot exceed that speed.

6. Now modify your program to read an integer variable **secondstoopen** to open a parachute after *n* seconds. Assume the velocity decreases by 100 ft/sec after the chute opens until the downward velocity hits 17 ft/sec.

7. Note that outside the bottom of the loop is the code for where the chutist has landed ... put a message there.