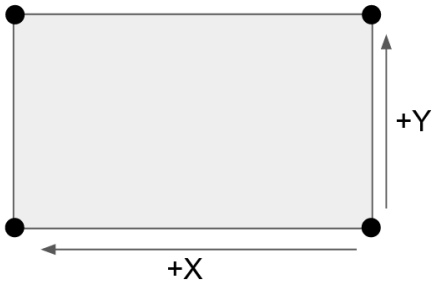
CSE 1384 - C++ Basics
Lab 2

Objectives:

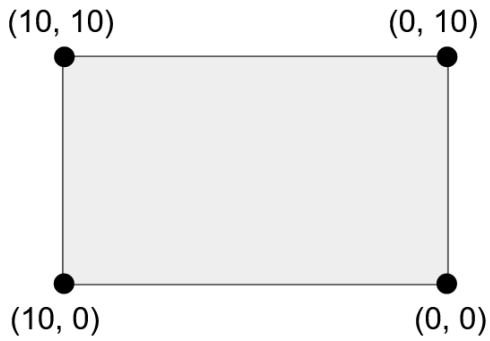
- Practice getting input from the user
- Practice using while loops and nested for loops

Assignment:

Imagine you have a coordinate plane that follows the following rules:



So, the following would be an example of some coordinate points:



Your goal is to take in two sets of coordinate points (as integers!). The upper left corner and the lower right corner. From this, you need to then display a coordinate grid corresponding with the points given. Following the criteria given, this means your lower right corner (the ending points) should always be *lower* in value than your upper left corner (the starting points).

Your program should loop (while loops) to ensure that the ending points are not greater than or equal to the corresponding starting points (x to x, y to y).

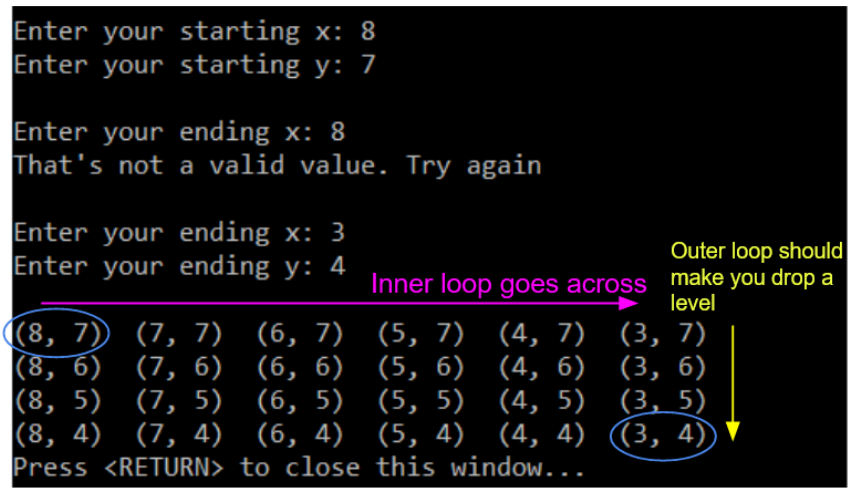
Make sure you're formatting your output properly → (x, y) is what we expect to see with coordinates.

Example Executions:

```
Enter your starting x: 9
Enter your starting y: 9

Enter your ending x: 0
Enter your ending y: 0

(9, 9)  (8, 9)  (7, 9)  (6, 9)  (5, 9)  (4, 9)  (3, 9)  (2, 9)  (1, 9)  (0, 9)
(9, 8)  (8, 8)  (7, 8)  (6, 8)  (5, 8)  (4, 8)  (3, 8)  (2, 8)  (1, 8)  (0, 8)
(9, 7)  (8, 7)  (7, 7)  (6, 7)  (5, 7)  (4, 7)  (3, 7)  (2, 7)  (1, 7)  (0, 7)
(9, 6)  (8, 6)  (7, 6)  (6, 6)  (5, 6)  (4, 6)  (3, 6)  (2, 6)  (1, 6)  (0, 6)
(9, 5)  (8, 5)  (7, 5)  (6, 5)  (5, 5)  (4, 5)  (3, 5)  (2, 5)  (1, 5)  (0, 5)
(9, 4)  (8, 4)  (7, 4)  (6, 4)  (5, 4)  (4, 4)  (3, 4)  (2, 4)  (1, 4)  (0, 4)
(9, 3)  (8, 3)  (7, 3)  (6, 3)  (5, 3)  (4, 3)  (3, 3)  (2, 3)  (1, 3)  (0, 3)
(9, 2)  (8, 2)  (7, 2)  (6, 2)  (5, 2)  (4, 2)  (3, 2)  (2, 2)  (1, 2)  (0, 2)
(9, 1)  (8, 1)  (7, 1)  (6, 1)  (5, 1)  (4, 1)  (3, 1)  (2, 1)  (1, 1)  (0, 1)
(9, 0)  (8, 0)  (7, 0)  (6, 0)  (5, 0)  (4, 0)  (3, 0)  (2, 0)  (1, 0)  (0, 0)
Press <RETURN> to close this window...
```



Hints:

You will *need* nested for-loops for the coordinate plane.

The *outer* loop controls the y, while the *inner* loop controls the x. Keep in mind: we're going down in numbers here. You should be starting your loop at a higher value and decreasing on every iteration of your loop, not increasing. Make sure your comparison coincides! (< VS >)

Also, make sure you're inclusive! (< VS > VS <= VS >=). Notice how your starting and ending points should be reflected in the coordinate plane.

Comment Block:

Your code should contain a comment block at the top containing information on who wrote the code, what the assignment is, when it is due, etc. Here is an example of a good comment block to put:

```
/*
 Name: <your name>                          NetID: <your netID>
 Date: <current date>                       Due Date: <enter in due date>

 Description: <What is the program?>
*/
```

Deliverables:

- C++ code (.cpp file)
- A document (.pdf) with two screenshots showing the program running
    - The two program screenshots should have completely different inputs from each other
    - The two screenshot must be *legible* to count (too small or pixelated text will not be interpreted)
    - Show *all* error messages

Point Breakdown:
(100 points total)
A submission that doesn't contain any code will receive a 0.

- 15pts - IO
    - 5pts - receives input from the user correctly
    - 5pts - receives data as an integer
    - 5pts - coordinates are appropriately formatted
- 25pts - while loops
    - 10pts - loops while ending x is greater than or equal to starting x
    - 10pts - loops while ending y is greater than or equal to starting y
    - 5pts - no infinite loops
- 30pts - for loops
    - 5pts - inclusive to appropriately show both starting and ending points
    - 5pts - begins at a higher number and ends at a lower number
    - 10pts - inner loop appropriately handles the rows and x value
    - 10pts - outer loop appropriately handles adding newlines and the y value
- 20pts - turned in two unique screenshots
- 10pts - programming style *

* Programming style includes good commenting, variable nomenclature, good whitespace, etc.