

CS5001 Assignment 1

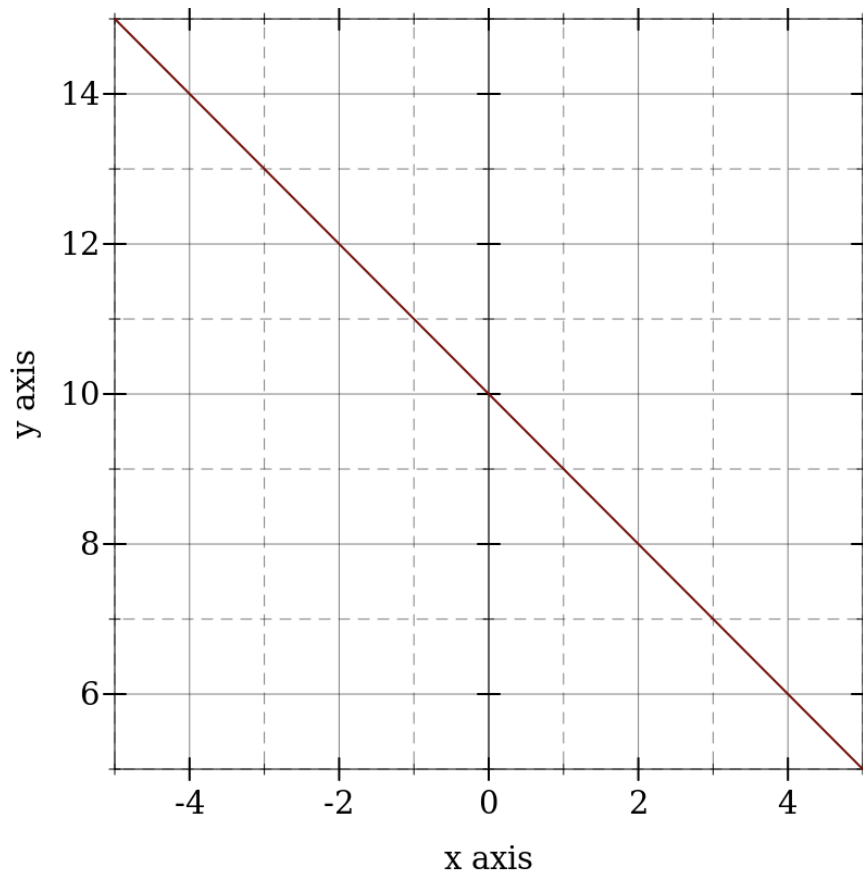
Programming Language: [BSL](#)

Due Date [Wednesday 1/17 at 10:00pm](#)

Purpose To write simple functions, use conditionals, use big-bang.

For this assignment and all future assignments you must upload a .rkt file in the specified language to the Handin server (handins.ccs.neu.edu).

Exercise 1 Take a look at the graph below which is a graph of a function of x . Turn the graph into a table for $x = 0, 1, 2$ and 3 and create a function definition.



Exercise 2 The following table describes how far a person has gone in a race in a certain number of seconds:

<i>t</i>	1	2	3	4	5	6	7	8	9	10
<i>d</i>	3.0	4.5	6.0	7.5	9.0	10.5	12.0	13.5	?	?

Write a function that predicts, based on this data, how far they will have run at time *t*. Write three check-expects: two that test to see the data in the table matches your function's output, and then one more that tests the output when *t* is at least 9.

Exercise 3 Mathematical equations in one variable are claims about an unknown number. For example, the quadratic equation $x^2 + 2x + 1 = 0$ is a claim concerning some unknown number *x*. For *x* = -1, the claim holds: $(-1)^2 + 2(-1) + 1 = 1 - 2 + 1 = 0$. For *x* = 1, the claim doesn't hold because: $(1)^2 + 2(1) + 1 = 4$, not zero. Develop functions that test whether it is true or false that the proposed solution is, in fact, a solution for the following equations:

- a) $8x - 2 = 14$
- b) $10x - 6 = 7x + 9$
- c) $2 - (-12/x) = -4$

You should test your functions with numbers that are solutions and numbers that are not solutions.

Exercise 4 Members of a middle school always walk around in pairs. Each member of a pair will respond with "Yes" or "No" individually when asked an appropriate question. You can only be certain of having received an affirmative answer if at least one replies with "Yes". All other combinations imply "No". Develop the function *determine*, which will accept all pairwise combinations of a pair's answers and produces "Yes" or "No" accordingly.

Exercise 5 Develop the function *sub-chars*. The function consumes a String and a natural number and extracts that many keyboard characters from the string. What happens when the function is applied to a number greater than the length of the String? In that case, *sub-chars* produces the original String.

Exercise 6 Design the function *ticket*. It consumes two natural numbers: one that represents the speed of a car and the other one the speed limit of the road. The result is one of these strings:

- (1) "fine" for a car that goes below the speed limit;
- (2) "danger" for a car that is going at most 5 mph faster than the speed limit; or
- (3) the string "ticket: you drove ____ mph" where the blank is replaced by the car's speed for a car that exceeds 5 mph over the speed limit.

Exercise 7 Design a world program that simulates backspacing on a type writer. The simulation ends when the string is erased.

The [main function](#) consumes the to-be-erased string. For example, if the initial string was "CS5001", then your simulation should produce images of the following strings, one per tick: "CS5001", " CS500", " CS50", " CS5", etc.

Constraint The clock must tick once per second. See [on-tick](#).

Hints

- See [stop-when](#).
- For each clock tick, one letter disappears from the back of the string.