CS 135 Fall 2019
Tutorial 01: Translations, Steppers, Constants, and the Design Recipe

Announcements

• The times and locations of office hours are posted on the “Office & Consulting Hours” page of the course website. Please email us at cs135@uwaterloo.ca to set up an appointment outside of these hours.

• Assignment 1 is due on **Tuesday, September 17, at 9:00 pm**. Submit early and often to MarkUs! Check your basic test results after each submission. We will not be lenient about any late submissions.

• Make sure you complete Assignment 0 before the Assignment 1 deadline, if you haven’t done so already!

• Ensure that your clicker marks posted on the “View Marks” page of the course website are accurate.

Announcements

• MarkUs Basic tests:
  – Are set up for every assignment.
  – Will not thoroughly test your code.
  – Ensure we can run better tests on your code after the due date.
  – The results are automatically emailed to your UWaterloo email. You can also check the results on MarkUs after each submission.
  – Are not related to the tests that you will write in your solutions.
Goals of this tutorial

You should be able to...

• Provide a **direct translation** of mathematical expressions/functions in Racket

• Trace through the steps of evaluating simple Racket functions.

• Understand how and when it is appropriate to use **constants**.

• Write an effective **design recipe** for a simple mathematical expression/function.

---

How to find the help pages

• **Do not** use Google search. It will land you at the wrong language level, which is typically the full Racket help page.

• Open DrRacket: Help menu > Help Desk or Racket Documentation (this opens a browser window) > Teaching > How to Design Programs Languages > Select the appropriate language level (e.g. Beginning Student).

• Note the categorized list of functions on the left side bar.

---

Problem 1: Pythagorean Theorem

![Pythagorean Theorem Diagram](image)

Write a function that computes the hypotenuse of a right angle triangle when given sides A and B.
Problem 1: Pythagorean Theorem

Write a function that computes the hypotenuse of a right angle triangle when given sides A and B. 
Hint: \( F(A, B) = \sqrt{A^2 + B^2} \)

Pythagorean Theorem: Stepper

Do the tutorial 1 stepper on the course course website.

Problem 2: Evaluating Food

For this question, there will be three criteria to evaluate the food:

- How does the food taste?
- What is the cost of the food?
- How much is the food like a sandwich?

Rate each category on a scale of 0-10, 0 being the lowest and 10 the highest.
Problem 2: Evaluating Food

Write a function `evaluate-food` that consumes three parameters: A score from 0-10 based on the taste, cost, sandwich-like, respectively. The function produces the weighted score that the specific food receives, according to the weightings below. For any food, we will add 1.5 as standard score. Include the full design recipe for this function.

- Taste - 25%
- Cost - 30%
- Sandwich-like - 30%

```
(evaluate-food 8 7 1) ⇒ 5.9 ;; Pizza test
(evaluate-food 5 1 10) ⇒ 6.05 ;; Sandwich test
```

```
(define standard 1.5)
(define taste-weight .25)
(define cost-weight .3)
(define sandwich-like-weight .3)
```

Clicker Question: Evaluate Food

Which of these results match the function call `evaluate-food 5 7 10`?

A 7.50
B 5.65
C 7.85
D 7.55
E Error
Clicker Question 2: Stepping
Which of these are the correct next evaluation step to:
(+ 1.5 (* 5 .25) (* 7 cost-weight) (* 10 sandwich-like-weight))

A  (+ 1.5 (* 5 .25) (* 7 .3) (* 10 .3))
B  (+ 1.5 (* 5 .25) (* 7 .3) (* 10 sandwich-like-weight))
C  (+ 2.75 (* 7 cost-weight) (* 10 sandwich-like-weight))
D  (+ 1.5 (* 5 .25) (* 7 cost-weight) (* 10 .3))
E  (+ 1.5 1.25 (* 7 cost-weight) (* 10 sandwich-like-weight))

Bonus Problem 3: Pull Digit
You've received an unreasonably long number in your email inbox.
There is a note at the bottom instructing you to find the nth element.
Define a function that consumes a number and an index (starting from the
tight at index 0, index must be greater than or equal to 0), and produces the
value found at the specified index.

(pull 1234 1) ⇒ 3
(pull 54321 4) ⇒ 5

Bonus Problem 3: Pull Digit - Design Recipe
;; (pull number index)
;; Consumes a number and an index, producing the value found at the specified index.
;; pull: Int Nat → Nat
;; Examples:

(check-expect (pull 1234 1) 3)
(check-expect (pull 54321 3) 4)
(check-expect (pull 12 4) 0)
(check-expect (pull 87 0) 7)