CS 135 Winter 2020

Tutorial 01: Translations, 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, January 21, at 9:00 pm**. Submit early and often to MarkUs. We will not be accept any late submissions. Check your basic test results after each submission!
- 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.
  - Do not thoroughly test your code.
  - Ensure we can run more thorough 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...

- Identify significant errors in a snippet of code and debug them.
- Give a **direct translation** of mathematical expressions and functions in Racket.
- Understand when and how to use **constants**.
- Write the full **design recipe** for simple arithmetic functions.

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.
- If you must use Google, then add the teaching language name to your query, e.g. “Racket beginning student”.

Clicker Question: Errors in Racket

Which of the following is an error-free Racket expression in "Beginning Student"?

A (8 + 6 / 3)

B (* (+ 6 −12 18) −24)

C (/ 5 (− 4 (sqrt 16)))

D (* (+ 5 10) (15))

E (/ (+ 1 3) (− 5 (* 7 9))))

Direct Translations: Tips

When given a mathematical function or expression to translate:

- Do not swap the order of parameters, or change their names.
  - For example, $2x + 2y$ should not be translated as (+ (∗ 2 y) (∗ 2 x)) or (+ (∗ 2 a) (∗ 2 b)).
- A mathematically equivalent expression is not necessarily a direct translation.
  - For example, $\frac{x}{y}$ should be translated as (/ x y), but not (∗ x (expt y −1)).
  - Similarly, $2x + 2y + x$ should be translated as (+ (∗ 2 x) (∗ 2 y) x), but not (+ (∗ 3 x) (∗ 2 y))

Clicker Question: Direct Translation

What is the correct Racket translation of the following mathematical expression?

$$15 + \frac{(6+3)^2}{10} - 18 \cdot 17$$

A (+ (− 15 (sqr (/ (+ 6 3) 10))) (∗ 18 17))

B (− (+ 15 (sqr (/ (+ 6 3) 10))) (∗ 18 17))

C (+ (− 15 (/ (sqr (+ 6 3)) 10)) (∗ 18 17))

D (− (+ 15 (/ (sqr (+ 6 3)) 10)) (∗ 18 17))

E None of the above

Group Problem: Direct Translation

Translate the following mathematical function into Racket:

The area of a regular polygon, given the length of one side, s, and the number of sides, n, can be computed with the following formula:

$$polygon\text{-}area(s, n) = \frac{1}{4} \cdot n \cdot s^2 \cdot \frac{1}{\tan(\frac{\pi}{n})}$$

(Hint: tan is a built-in Racket function that may be useful.)

Ensure that you are giving a **direct translation** of the function above.

Naming parameters

For a correct translation, parameters and constants should be named exactly as they appear in the function you're translating. However, s and n from the previous problem aren't very good names. Which of the following would be the best option for naming s and n differently?

A x and y.

B num-sides and side-length.

C this-parameter-is-the-number-of-sides and the-length-of-each-side.

D Jun-Xue and Yi-Feng

E The current names are the best.

Review: Advantages of constants

- Can give meaningful names to useful values (e.g. interest-rate, passing-grade, and starting-salary).
- Reduces typing and errors when such values need to be changed.
- Makes programs easier to understand.
- Constants can be used in any expression, including the body of function definitions.

An Example

We finally found out who Banksy is - it's you! You are trying to buy paint for your latest street art. You want to save money, so you're only buying red, blue and yellow paint. A can of paint costs 20 dollars each, and you can only store 20 cans of paint at once. Consider the badly written functions below. Why are they bad?

```
(define (yellow-cans-to-buy red-cans blue-cans)
        (- 20 red-cans blue-cans))
...
(define (cost-of-cans red-cans blue-cans)
        (+ (* red-cans 20)
           (* blue-cans 20)
           (* (yellow-cans-to-buy red-cans blue-cans) 20)))
```

What if the price of red paint doubles? Or you can store 30 cans now?

Now if we use constants, then we only need to change 20 to 30 on the first line and 20 to 40 on the second line:

```
(define max-cans 20)
(define red-can-cost 20)
(define blue-can-cost 20)
(define yellow-can-cost 20)
...
(define (yellow-cans-to-buy red-cans blue-cans)
        (- max-cans red-cans blue-cans))
...
(define (cost-of-cans red-cans blue-cans)
        (+ (* red-cans red-can-cost)
           (* blue-cans blue-can-cost)
           (* (yellow-cans-to-buy red-cans blue-cans) yellow-can-cost)))
```

The Design Recipe on Assignments

Although the design recipe is not required for Assignment 1, **all subsequent assignments will require you to write the design recipe.**

A significant portion of your assignment marks will be from the design recipe!

Review: The five design recipe components

Purpose: Describes what the function produces. You should include parameter names in your purpose statement in a meaningfully way.

Contract: Describes what type of arguments the function consumes and what type of value it produces.

Additional contract requirements: If there are important constraints on the parameters that are not fully described in the contract, add an additional **requires** section to "extend" the contract.

Examples: Illustrate the use of the function.

Definition: The Racket definition (header and body) of the function.

Tests: A thorough set of function arguments and expected function values.

Group Problem: evaluate-tutorial-leader

At the end of the term, CS135 students are asked to evaluate their tutorial leaders. For this question, there will be three criteria that students will base their evaluations on:

- How knowledgable their leader is about course content
- How prepared they were for delivering tutorials
- How likely their leader could win in a battle royal against the other tutorial leaders

Group Problem: evaluate-tutorial-leader

Also, each criteria is weighted differently:

- Knowledge - 20%
- Preparedness - 5%
- Battle Royal - 75%

Students can rate each criteria on a scale of 1-10, where 1 is the lowest rating and 10 is the highest rating.

Group Problem: evaluate-tutorial-leader

Write a function evaluate-tutorial-leader, that consumes three parameters: A score from 1-10 based on the knowledge, preparedness, and battle royal criteria, respectively. The function produces the weighted score that to the weightings on the previous slide. Include the full design recipe for this function.

Knowledge: 20%, Preparedness: 5%, Battle: 75%

(evaluate-tutorial-leader 8 7 1) $\Rightarrow$ 2.7

(evaluate-tutorial-leader 5 1 10) $\Rightarrow$ 8.55

Extra Practice: Direct Translation

We will not cover these questions in the tutorial, unless we have some time left over in the end. Solutions will not be posted to these questions - you may discuss these problems with classmates and work together to come up with a solution.

Translate the following function that calculates the terminal velocity, v, of a falling object. Name the Racket function terminal-velocity.

$$v = \sqrt{\frac{2mg}{\rho AC}}$$

where:

- m is the mass of the falling object
- g is the gravitational constant, approximately $9.8 m/s^2$
- ρ (or rho) is the density of the fluid the object is falling through
- A is the projected area of the object
- C the drag coefficient, which is different for every object.

Extra Practice: Design Recipe

We will not cover these questions in the tutorial, unless we have some time left over in the end. Solutions will not be posted to these questions - you may discuss these problems with classmates and work together to come up with a solution.

- Write the design recipe for the two functions in the example on slide 13.
- Using the DrRacket help pages, create purposes and contracts for ceiling, expt, random, boolean? and numerator.

Extra Practice

We will not cover these questions in the tutorial, unless we have some time left over in the end. Solutions will not be posted to these questions - you may discuss these problems with classmates and work together to come up with a solution.

We have 9 equally-weighted assignments in this course that count towards your final grade. Write a function that takes in your assignment marks as parameters and produces your assignment average.