

Class 5: Procedure Practice

Assignments Due

- **Monday, 5 September** (11:01am): Problem Set 1 (both on-line submission and paper)
- **Wednesday, 7 September** (in class): **Quiz 1**: covers Chapters 1-4 of course book, Chapters 1-3 of *The Information*, Classes 1-5 (including questions)

Upcoming Help Schedule (all office hours are now in Rice Hall)

- Sunday, 1:30-6pm, Rice First Floor

If Expression

IfExpression ::= (**if** *Expression*_{Predicate} *Expression*_{Consequent} *Expression*_{Alternate})

To evaluate an if expression:

- Evaluate *Expression*_{Predicate}.
- If it evaluates to a false value, the value of the if expression is the value of *Expression*_{Alternate}; otherwise, the value of the if expression is the value of *Expression*_{Consequent}.

(if false false true)

(if (> 4 3) 4 3)

((lambda (a b) (if (> a b) a b)) 5 6)

Advice on Defining Procedures

1. First, make sure you know what the procedure is intended to do. You should know what the inputs are (and what types of values they are), and what type of value the output is (e.g., a procedure, a number, a list). You should also have some example inputs and outputs in mind.
2. Before worrying about the code, think in English how you will solve the problem.
3. Define the procedure.
4. Test your procedure with the example inputs and outputs. Think about any tricky cases that you should also test.

Procedures Practice

1. Define a procedure, **identity**, that takes one input, and outputs that value.

Example: (identity 37) should evaluate to **37**

2. Define a procedure, **pick-one**, that takes three inputs. The first input is a Boolean value (either **true** or **false**). If the value of the first input is **true**, the output is the value of the second input; if the value of the first input is **false**, the output is the value of the third input.

Examples:

(pick-one (> 3 4) 1 2) should evaluate to **2**.

(pick-one (= 3 3) 1 2) should evaluate to **1**.

3. Define a procedure, **middle**, that takes three numbers as inputs, and outputs outputs the number that is in the middle.

Examples:

(middle 1 2 3) should evaluate to **2**.

(middle 1120 150 200) should evaluate to **200**.

4. (challenging!) Define a procedure, **find-fixedpoint**, that takes as input a function and an initial value, and outputs the fixed point of the function starting from that value. A fixed point of a function f is a value x such that $(f x)$ evaluates to x .

Examples:

> (find-fixedpoint (lambda (n) n) 3)

3

> (find-fixedpoint (lambda (n) (modulo n 7)) 215) ;;; (modulo a b) evaluates to the remainder of a/b

5

> (find-fixedpoint (lambda (n) (modulo n 7)) 1120)

0

> (find-fixedpoint (lambda (x) (cos x)) 0)

0.7390851332151607