CSC 171
Introduction to Computer Programming
Lab 3

Dr. Stephen Bloch
sbloch@adelphi.edu
http://www.adelphi.edu/sbloch/class/171/

Recall example: temperature conversion

; **f->c** takes a number of degrees Fahrenheit and returns the corresponding number of degrees Celsius.

```
(define (f->c Fahrenheit)
  (* 5/9 (- Fahrenheit 32)))
```

"Examples of f->c:"

(f->c 32) "should be 0"

(f->c 212) "should be 100"

(f->c -40) "should be -40"

Testing the f->c function

We can test the **f->c** function in several ways:

- With definition and examples in Definitions window, hit “Run”
- Type examples one at a time into Interactions window

Note that the function works the same way whether it's invoked from Interactions or Definitions.

(BTW, I've put **f->c** on the Web for your convenience: see www.adelphi.edu/sbloch/class/171/examples/f_to_c.scm)

More ways to test f->c

- “Language” menu -> “Add Teachpack” -> htdp -> convert.ss -> click “Run”.

- **(convert-repl f->c)**
  It repeatedly asks you questions; you type # degrees F, it prints # degrees C.

- **(convert-gui f->c)**
  It puts up a GUI with slider bars for Fahrenheit & Celsius

- **(convert-file “input.txt” f->c “output.txt”)**
  It reads a sequence of numbers (degrees F) from file input.txt (which you've previously created with, say, Notepad) and writes the corresponding numbers of degrees C to file output.txt (which you can read with, say, Notepad)

Notes on teachpacks

- Teachpacks provide functions that use advanced language features you don't know about yet...
- ... but you can use them, even in Beginning Student language.
- The function f->c doesn't care where its input comes from, nor where its output goes (keyboard, GUI, disk file, ...)
- The convert functions *use your function f->c*; if it's wrong, they'll produce wrong answers!
- The convert functions *take in a function name* as one of their parameters. We'll learn to do this ourselves in a few weeks.

Syntax Check

- DrScheme feature that checks whether your program is legal Scheme, but doesn't actually run it on test cases
- Colors your code (red = error!)
- Mouse over function name shows where it is defined or used
- Always do this before clicking “Run”

Tracking Mistakes

- Computer programs almost *never* work right on the first try
- Need to *analyze errors*
  - Symptoms (how & when discovered?)
  - Diagnosis (what went wrong?)
  - Treatment (how to fix it?)
  - Prevention (how to not do it again?)

Causes & Symptoms

- Any error is caused at some time, and detected some time later.
- The less time in between, the less time you've wasted.
- The earlier the error is spotted, the easier to diagnose and fix.
- Early cause: bad.
  Early detection: good.

Symptoms/Discovery

- You saw it yourself (congratulations!)
  - You already know what's wrong
- Error message on “Syntax Check”
  - Message tells you exactly what's wrong & highlights the illegal expression
- Error message on “Run”
  - Message tells you how patient died, but disease may have started some time earlier
- Wrong answers on “Run”: look for patterns
  - answer *always* wrong in the same way?
  - answer right on some inputs, wrong on others? Which?
- No answers at all: use Stepper

- Early (good)

- Late (bad)

Diagnosis/Cause

- Wrong contract, wrong choice of function
- Incorrect or insufficient examples
- Wrong function skeleton
- Wrong function body
  - Calling other function w/wrong args
  - Matching parentheses, braces, quotation marks, etc.
  - Misspellings & typoes
  - etc.
- Insufficient testing

- Early (bad)
- Late (good)

Treatment & prevention

- For mismatched parentheses, quotation marks, misspellings, etc. obvious
- If fixing the problem took *thought*, write down how you fixed it
- If there's something *specific* you can watch out for in the future, write down how to prevent it

PSP on-line error forms

- Log in: your initial password is your last name in lower-case (*change it!*)
- Can input new defects, view past defects, view defects recorded by whole class
- Other features (planning, time recording, lines of code) available but not required this semester

Defect Recording Input

Bookkeeping

Homework number	1
If this entry describes a number of similar defects, how many?	1

Symptoms

How did you discover the problem?	Error message on syntax check or compile
Details (e.g. *what* error message did you get?)	Undefined identifier "cubs"

Diagnosis

What caused the problem?	Typos & misspellings
Details about this diagnosis:	I meant to type "cube"

Lab exercise (show me!)

- Write *incorrect* version of **f->c** and test it in various ways
- **difference-of-cubes** from this morning
- **total-minutes** from this morning
- **under-100?** from this morning
- **can-vote?** from this morning
- **can-vote-but-not-drink?** from this morning

End-of-day survey

- http://www.adelphi.edu/sbloch/class/171/daily.html