**CSSE 304 Assignment #18 (5th interpreter assignment) 200 points**

**Piazza Questions and Answers** from previous terms:

**Writing exit-list using given escaper** Are we allowed to implement exit-list using the escaper code on the schedule page if we replace the call/cc with our implementation of call/cc?

**Instructor answer:** Allowed, yes. Advised, no. Once your code is CPS, exit-list can be implemented in a few lines of code, and that does not need escaper or call/cc.

**What needs to be CPS in A18?** According to the A18 project spec:

These should not need to be converted to CPS, unless your eval-exp calls them **and** is called by them:

• parse-exp

• top-level-eval

• syntax-expand

• datatype constructors and continuation constructors

• What about the environment procedures? Depends on how you write them and their interaction with the rest of the interpreter.

 Is this intentionally *and*, or should be *or*?  For example, our eval-exp calls parse-exp, but parse-exp does not call eval-exp. Does parse-exp need to be converted to CPS? Likewise, our eval-exp calls apply-env, but apply-env does not call eval-exp. Does apply-env need to be in CPS form?

**the students' answer:** I believe the "and" is intentional.  Only those procedures that are called by eval-exp and call it.

**the instructors' answer:** The **and** is certainly intentional.  Don't forget that the procs that need to be CPS include apply-proc and apply-prim-proc.  Also, any time you want to call map or apply within a CPS procedure, you need to call cps versions of those.

**followup discussions**

**A student:** You mentioned that we must use the CPS map and apply in our CPS procedures.  So in the our apply-prim-proc, we use apply for most of the primitive procedures.  If we change them all to apply-cps, do we need to provide cps versions of all the primitive procedures to pass to apply-cps?  
Also, can we implement apply-cps using apply, or should we do it another way?

[**Claude Anderson**](https://piazza.com/class/is9cjqgxyh31b?cid=116): My suggestion is to call apply-k on the results of the Scheme procedure call for most of the primprocs.  But for primprocs map, apply, and call/cc you'll need to do something different.

**Another student:** Following up on this question: do apply-env/apply-env-ref need to be in CPS form?

[**Claude Anderson**](https://piazza.com/class/is9cjqgxyh31b?cid=116): For most teams, no.  If these procedures do not call eval-exp or call anything that calls eval-exp, you should be okay without converting them to CPS.

# If apply-env-ref never calls eval-exp, does it need to be in cps?

In Assignment 18, it says "What about the environment procedures? Depends on how you write them and their interaction with the rest of the interpreter." Since our apply-env-ref is only called (indirectly) from eval-exp but never calls eval-exp, does apply-env-ref need to be cps?

**Instructor’s answer:** No.

# Should the fail-proc be in cps?

I noticed in the slide for var-exp the modification is [var-exp (id) (apply-env env id k fail-proc))]

but since fail-proc also will be calling non-primitive procedures, it should be a fail continuation.

However, because it starts with lambda() with no argument, the non-primitive procedures are not evaluated until in apply-env.I'm confused about how to write this fail-proc in cps. Any ideas?

**the instructors' answer,**

If you need the fail procedure to be in CPS, you can always change the interface to apply-k

to be (lambda (k . args) ...).  This allows a continuation to take any number of args, including 0.

# CPS in if statements

If you want to make an if statement into CPS

(if 1 2 3) and statement 1 contains non primitive procedures, how do you handle it?

**the instructors' answer,**

You call the non-primitive procedure first, and put the if in the continuation of that call.