Dove

- * Functions
- * Program is declaration list and expression
- * Declaration is just a function: has name, params, body :expr
  - body can call other functions
  - any function may call any other function
- * Need to know Calling conventions

In C:

can't call
g() here

```
void f() {...}
void g() {...}
```

Callee- and
Caller-saved
registers.

C Calling Conventions

1. Caller must
   Push caller-saved regs
   a. Pushes args
   b. "Call" instr
2. Callee must
   a. Save ebp (push)
   b. Mov ebp, esp
   Push callee-saved regs
   c. Allocate local vars by moving esp
3. Callee runs
4. Callee must
   Pop callee-saved regs
   a. Mov esp, ebp
   b. Restore ebp (pop)
   c. Ret
5. Caller must
   a. Remove params
   Pop caller-saved regs

Callee-saved : ebx, esi, edi

Caller-saved : eax, ecx, edx

Dove Errors

Compile-time checking

New features

- functions
- function calls

Errors

- Call non-existen fn?
- Call wrong # args?
- Declare 2 fns of same name?
- Declare 2 params of same name?
- Unbound variables

1. Check if program has errors
2. If errors, then throw exn
3. Compile normally

```
let rec f ..... =
        ....
and g .... =
    .....
;;
```

```
<expr> ::= .....
           <id>(<exprList>)
```

```
def next(x)
  if isint(x) then x+1 else true
end
```

$\boxed{a + b}$