Newbie

Programming for Dummies (Or highly inexperienced experts)
Who are We?

Clyde Bazile  John Anukem  Sebastien Siclait  Braxton Gunter  Terence Jacobs
Why Newbie?
What can a Newbie do?

➔ **Comment**
   # I’m such a noob
   # enter code below

➔ **Operate**
   # Hi my name is newbie
   set x to 10
   set y to x
   set operators to “+,-,*,/,^,%. ”
   print(operators)

➔ **Infer**
   # Hi my name is newbie

   set x to 10 # type x: int
   set y to “hello” # type y: string
1 define function plus_10 with params (a, b)
2     set c to 9 + 1
3     return a + b + c
4
5 define function main with no params
6     set x to 1
7     set y to 1
8     set out to plus_10(x, y)
Language Features

- Easy syntax makes for simple implementation of algorithms
- Types:
  - Num
  - String
  - Bool
- Control Flow
  - If, Else, While, For
- Lists/Type Conversion/Coercion
```bash
check() {
    error=0
    basename=`echo $1 | sed 's/\.[^/]*$//'`
    reftfile=`echo $1 | sed 's/\.*$//'`
    basedefn=`echo $1 | sed 's/\.[^.]*$//'`
    echo -n "$basename..."
    echo In: Testing $basename $1=$2
    echo generatedfiles="$generatedfiles $basename.ll $basename.s $basename.exe $basename.out" $1 $2
    echo Run "$basename" $1 $2
    echo Compare $basename.out $2 $basename.diff
    echo # Report the status and clean up the generated files
    if [ $error -ne 0 ]; then
        echo "OK"
    else
        echo "# SUCCESS" $1=$2
    fi
    # Kdump never succeeds
    exit $error
}

while getopts kdpsh c; do
    case $c in
        k) # Keep intermediate files
            keep=1
            ;;
        h) # Help
            Usage
            hello
            esac
    esac
    done
}
```

```
define function main with no params
set x to 2 + 3 * 4
print(x)
set x2 to (2 + 3) * 4
print(x2)
set y to 9 - 6 / 3
print(y)
set y2 to (9 - 6) / 3
print(y2)
```
Lessons Learned

- Start early....... very early.
- Don’t underestimate how long something will take, even if you think it’ll be easy.
- Every line of OCaml counts..... EVERY LINE
- If you fail to test, your tests will fail.
Future Work

- Implementing more built-in classes with C.
- More built-in functions to handle walk through of staple algorithms