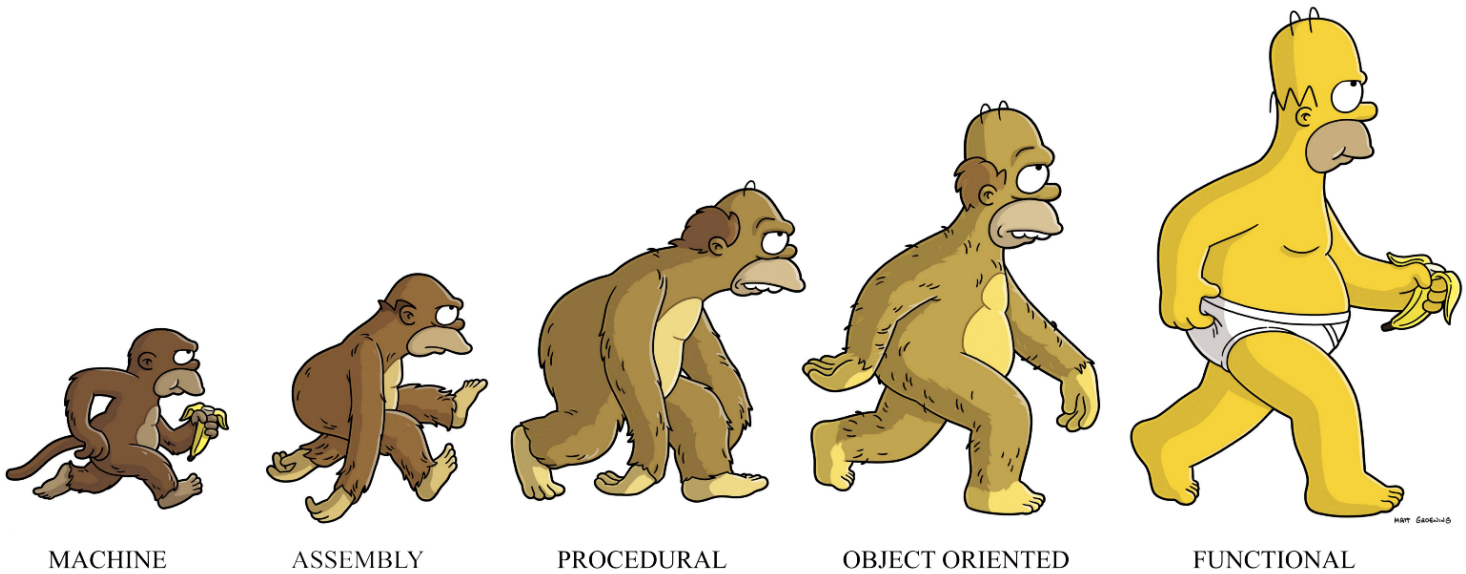
COMP302: Programming Languages and Paradigms

Prof. Brigitte Pientka (Sec 01)
bpientka@cs.mcgill.ca

Francisco Ferreira (Sec 02)
fferre8@cs.mcgill.ca

School of Computer Science
McGill University

Week 6-3, Fall 2017



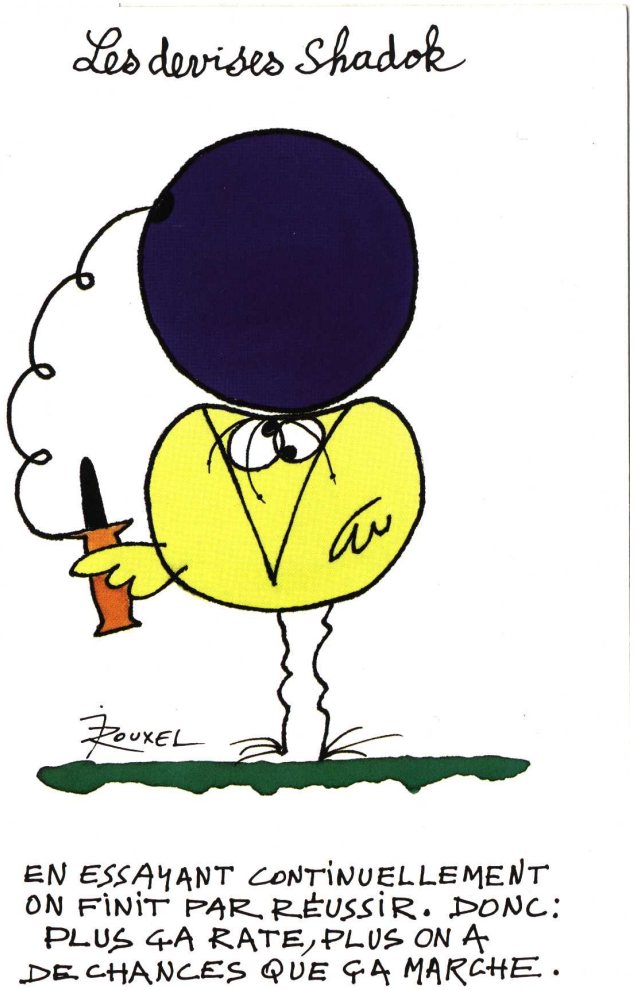
Functional Tidbit: The Midterm – The Aftermath!

Functional Tidbit: The Midterm – The Aftermath!

"Ever tried. Ever failed. No matter.
Try Again. Fail again. Fail better."

Samuel Beckett

Functional Tidbit: The Midterm – The Aftermath!



Functional Tidbit: Imperative vs Functional Programming!

"Clearly, I reject the view that there is one way that is right for everyone and for every problem."

- Bjarne Stroustroup

– Exceptions –

Primary benefits:

- Force you to consider the exceptional case
- Allows you to segregate the special case from other cases in the code (avoids clutter!)
- **Diverting control flow!**

Warm-Up: Type, Values, and Effect

Expression `3 / 0`

Warm-Up: Type, Values, and Effect

Expression `3 / 0`

Type `int`

Warm-Up: Type, Values, and Effect

Expression `3 / 0`

Type `int`

Value ✗

Warm-Up: Type, Values, and Effect

Expression `3 / 0`

Type `int`

Value ✗

Effect raises run-time exception `Division_by_zero`

Warm-Up: Type, Values, and Effect

Expression `3 / 0`

Type `int`

Value ✗

Effect raises run-time exception `Division_by_zero`

Expression

```
1 let head_of_empty_list =
2   let head (x::t) = x in
3     head []
```

Warm-Up: Type, Values, and Effect

Expression `3 / 0`

Type `int`

Value ✗

Effect raises run-time exception `Division_by_zero`

Expression

```
1 let head_of_empty_list =
2   let head (x::t) = x in
3     head []
```

Type `'a`

Warm-Up: Type, Values, and Effect

Expression `3 / 0`

Type `int`

Value ✗

Effect raises run-time exception `Division_by_zero`

Expression

```
1 let head_of_empty_list =
2   let head (x::t) = x in
3     head []
```

Type `'a`

Value ✗

Warm-Up: Type, Values, and Effect

Expression `3 / 0`

Type `int`

Value ✗

Effect raises run-time exception `Division_by_zero`

Expression

```
1 let head_of_empty_list =
2   let head (x::t) = x in
3     head []
```

Type `'a`

Value ✗

Effect raises run-time exception `Match_failure`

User-defined Exceptions

– Demo –