

# COMP302: Programming Languages and Paradigms

Prof. Brigitte Pientka (Sec 01)

`bpientka@cs.mcgill.ca`

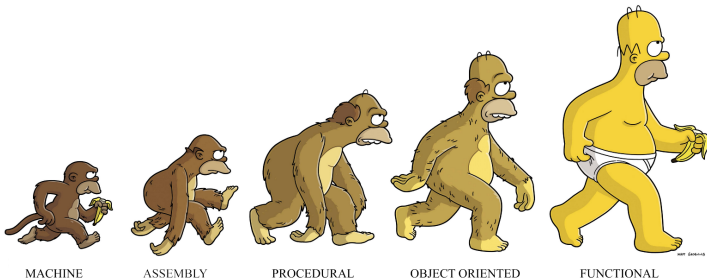
Francisco Ferreira (Sec 02)

`fferre8@cs.mcgill.ca`

School of Computer Science

McGill University

Week 7-1, Fall 2017



# Today: Exceptions (cont.)

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!**

# Backtracking

- General algorithm for finding all (or some) solutions *incrementally* – abandons partial candidates as soon as it determines that it cannot lead to a successful solution.
- Important tool to solve constraint satisfaction problems such as crosswords, puzzles, Sudoku, etc.

# Today's Example: Got Change?



# Today's Example: Got Change?



Implement a function `change`. It takes as input a list of available coins and an amount `amt`. It returns the exact change for the amount (i.e. a list of available coins,  $[c_1; c_2; \dots; c_n]$  such that  $c_1 + c_2 + \dots + c_n = \text{amt}$ ), if possible; otherwise it raises an exception `Change`.