Procedural Abstraction

CMPT 145

Procedural Abstraction

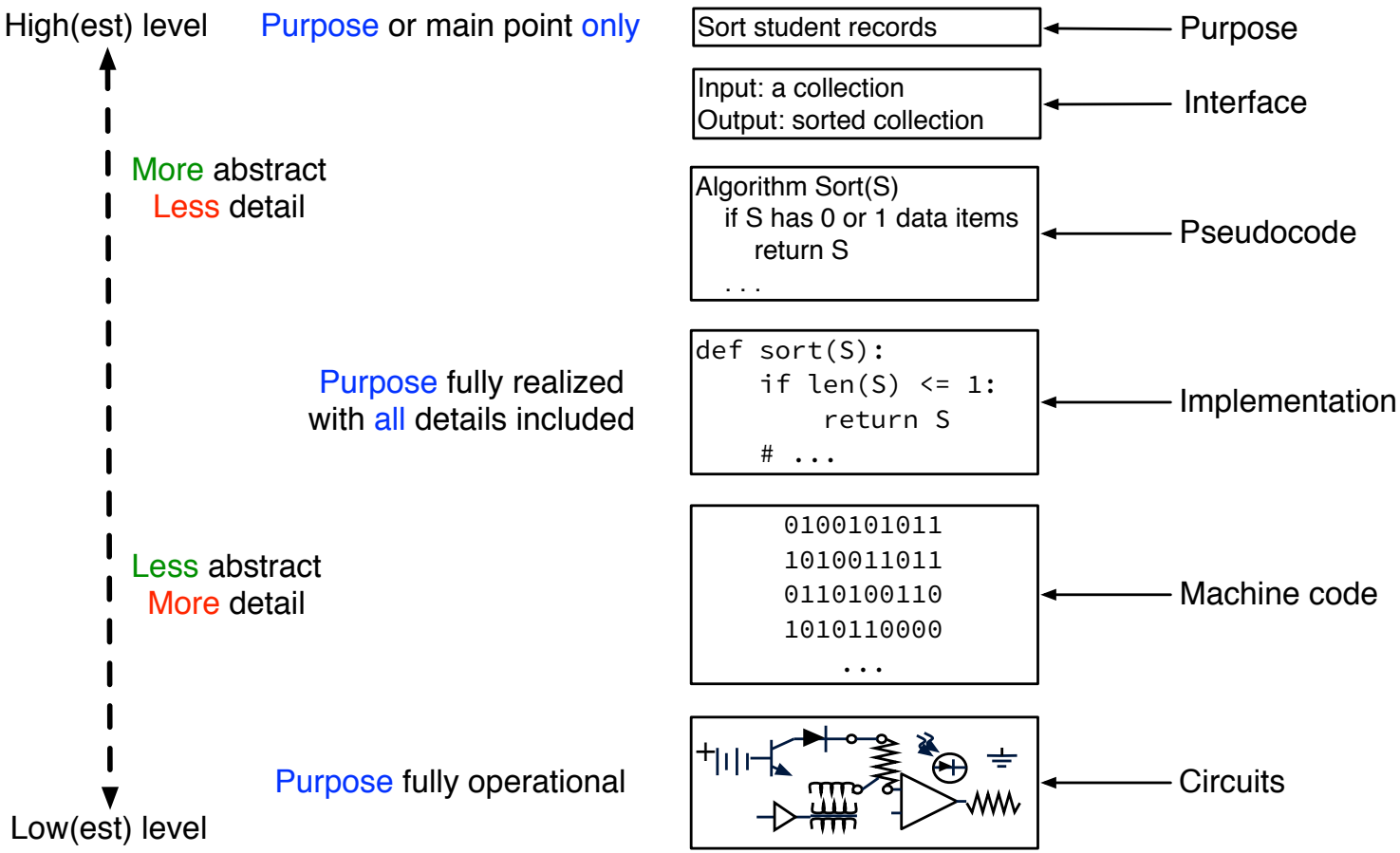
- When we define a function, we are creating a procedural abstraction.
- Allows us to view software from two perspectives:
  - Purpose
    - What is the point of the function? What purpose does the code achieve?
  - Implementation
    - How does the function work? What steps are needed?

Benefits

Procedural abstraction enhances:

- Correctness
- Reusability
- Adaptability

Levels of Procedural Abstraction



Interface

- A function interface defines a function's input-output relationship.
- All interfaces must be documented.
  - Purpose
    - What does the function do?
  - Pre-Condition(s)
    - Parameters and constraints on them, if any
  - Post-Condition(s)
    - Effects outside of function, if any
  - Return
    - Values returned by the function, if any

Demo 1

We will examine some Python code and document its interface using docstrings.

Exercise 1

Describe the interface, as a docstring, for this Python function:

```
def positive_evens( numbers ):
    return [x for x in numbers if x % 2 == 0 and x > 0]
```

Exercise 2

Define an interface, as a docstring, for this Python function:

```
1 def set_currency( customers,country,currency ):
2     for customer in customers:
3         if customer["country"] == country:
4             customer["currency"] = currency
```

Exercise 3

Write an interface (only) for the following abstract purposes:

- Find all the prime factors of a given positive integer $x > 1$
- Remove all the duplicates of a list.
- Microsoft Excel uses letters for column labels. Translate column strings to integers.

Demo 2

Let's refactor the sieve script, and make it a function!

Generalization and Abstraction

- Abstraction hides decisions within a function.
- Generalization exposes decisions to the function caller.
- Combining these two ideas is the essence of procedural abstraction.

Exercise 4

Design an app to play Tic-Tac-Toe in the console.

- Break the task into function interfaces only.
- No need to implement the functions at all.