Why OOP

COMP 102

Victoria University of Wellington

Philosophy of programming

- When creating a computer program there are many different ways to conceptualize programs. They can all reach the same result, but they do it in different ways.

Imperative and procedural paradigm

- “First do this and next do that.”
  - Step by step instructions using basic programming commands
  - Similar to how the computer performs steps at the CPU level
  - Incrementally change the state of the “program”
    - Data
    - Files
    - User inputs
  - Real World Example: Food recipes and furniture assembling descriptions in the real world

Event-driven paradigm

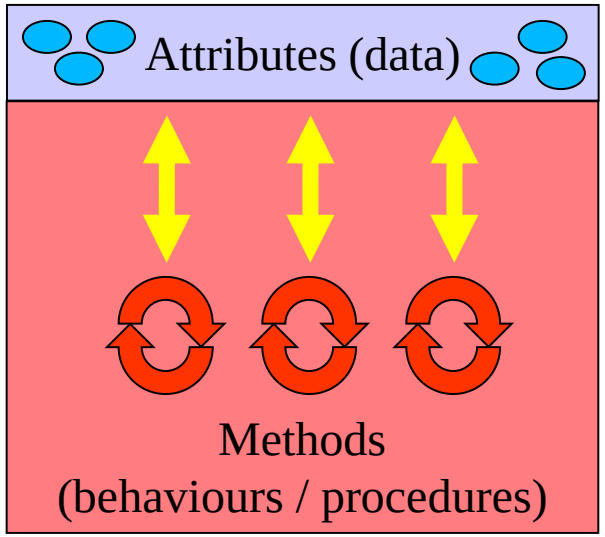
- “Something happens, which triggers a reaction.”
  - Events determine the program flow
    - Events can e.g. be mouse clicks, file reading completions, keyboard press and so on
  - Program consists of
    - Main loop that detects events
    - Main loop calls code to react to event

Object-oriented paradigm

- Why OOP
  - Imperative/procedural/event-driven programming becomes increasingly complex.
    - The code and data, even when being careful, often ends up as spaghetti
  - When designing code for the solution it can help thinking about the problem by creating a “world” of objects
- “Create objects of data and operations. Objects interact to simulate real world behaviour of individual entities.”
  - An object is a self-contained element of a computer program the object is designed to accomplish specific tasks
    - Data and operations on data are encapsulated in objects
    - Information (data) is hidden from other objects and they interact by calling operations (methods) of other objects
  - Objects are defined in classes
    - Classes are recipes of what objects do and how they interact
    - Objects are instances of the class
  - Often incorporate event-driven approach

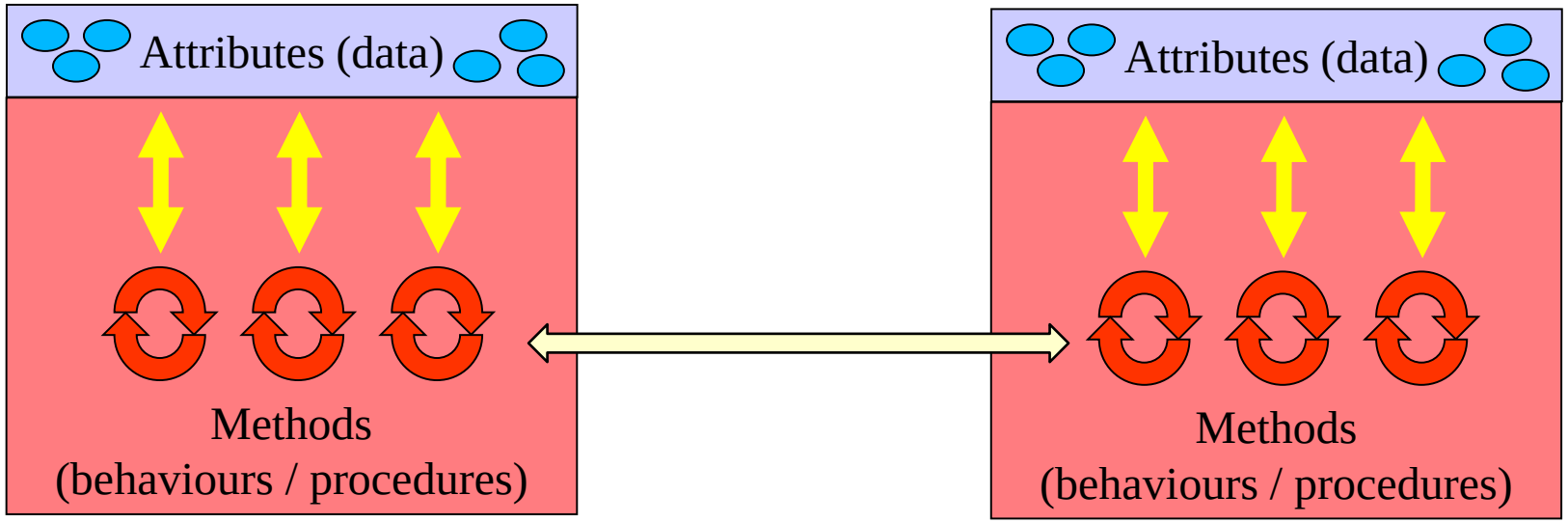
Classes and objects

- A class specifies the **attributes** (also known as **fields**) and **methods** of the objects
  - Methods defines behaviours and make use of attributes to achieve these



Classes and objects

- A class specifies the **attributes** (also known as **fields**) and **methods** of the objects
  - Methods defines behaviours and make use of attributes to achieve these
  - Objects interact through the methods of other objects
    - Attributes are “safe” and only changed through its own methods



Classes and objects

An object is

- A collection of data wrapped up together

plus

- A collection of actions to operate on the collection of data

All specified in a class:

- Fields where data is stored (if it needs stored data)
- Methods describing the actions
- Constructor to make new objects
- Constants

Classes and Objects

- A class is a description of a type of object.
  - includes descriptions of methods you can call on this kind of object

- Some kinds of objects we have used:

String
length, startsWith, substring…

Butterfly
fly, land …

Animal
goLeft, goRight, jump, speak …

Scanner
next, nextInt, hasNext,…

PrintStream
println, print, printf…