Australian National University

Methods

J7

Methods

Parameters

Return values

Methods

Subroutines

- Reusable code to perform specific tasks
- Modularity, Encapsulation

- May take arguments (parameters)
- May return a value

Method Declarations, In Order

1. any **modifiers** (`public`, `private`, `static`, etc.)
2. **generic type parameters**, between angle brackets **<>** (J12)
3. **return type**
4. method **name**
5. **parameters**, in parentheses
6. any **exceptions** the method may throw (J15)
7. the method **body** (code)

Class vs. Instance Methods

Class Methods

- `static` modifier
- may only access `static` fields
- globally accessible

Instance Methods

- no `static` modifier
- may access class *and* instance fields
- associated with objects – model *behavior*

Parameters (method arguments)

Parameters are the mechanism for
passing information to a method or constructor

Primitive Types

- Passed *by value*
  ➔ changes to *parameter*
  are not seen by caller

Reference Types (non-primitive)

- Pass *reference by value*
  ➔ changes to the *reference*
  are not seen by caller
  ➔ changes to *object referred to* **are seen** by caller

The last parameter of a method
may be more than one parameter (varargs), and treated as array

Return Values

Execution returns to the caller of a method when

- execution reaches the end of the method
  - return type needs to be `void`
- execution reaches a return statement
  - return type is `void`: `return;`
  - return type is not `void`: `return` *[expression]*;
    - » type of expression needs to be subtype (details later) of return type