Australian National University

Methods

J7

Methods

Parameters

Return values

Methods

- A subroutine
  - Reusable code to perform a specific task
  - Modularity, encapsulation
- May take arguments (parameters)
- May return a value

Method Declaration

Method declarations will have the following, in order:

- Any **modifiers** (`public`, `private`, etc.)
- **return type**
- method **name**
- **parameters**, in parentheses
- Any **exceptions** the method may throw
- The method **body** (code)

```
class String {…
 public byte[] getBytes(String charsetName)
throws UnsupportedEncodingException {…}
… }
```

Class and Instance methods

A method declared with the `static` modifier is a **class** method (otherwise it is an **instance** method)

- Class methods
  - May operate on class fields only
- Instance methods
  - May operate on class *and* instance fields

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 passed by *value*
  - Changes to the *reference* **are not seen** by caller
  - Changes to *object referred to* **are seen** by caller
- Your last parameter may in fact be more than one parameter (*varargs*), and treated as an array

Returning a Value from a Method

The `return` statement exits the current method

Methods return to caller when:

- all statements in method executed, or
- a `return` statement is reached, or
- the method throws an exception (later)

Methods declared `void` do not return a value.

All other methods must return a value of the declared type

(or a *subclass* of the declared type, described later).