Basic UML Class Diagram Notation

**Class**

- Name
- Attributes (member variables)
- Methods (member functions)
  - `public_method()`
  - `protected_method()`
  - `private_method()`

**Abstract class**

- Name
- `virtual method()`
- `method()`

**Inheritance (is-a) relationship**

- Base
  - Derived1
  - Derived2

**Object**

- `classname: objectname`

**Aggregation and Composition (has-a) relationship**

- Whole
  - Part
  - Whole has Part as a part; lifetimes might be different; Part might be shared with other Wholes.
  - (aggregation)

- Whole
  - Part
  - Whole has Part as a part; lifetime of Part controlled by Whole, Part objects are contained in one Whole object.
  - (composition)

**Association (uses, interacts-with) relationship**

- A
  - A's role
- B
  - B's role
- A
  - B
  - Navigability - can reach B starting from A

**Multiplicity in Aggregation, Composition, or Association**

- `*` - any number
- `0..1` - zero or one
- `1` - exactly 1
- `1..*` - 1 or more
- `n` - exactly `n`
- `n..m` - `n` through `m`

- Follow line from start class to end class, note the multiplicity at the end.
- Say "Each <start> is associated with <multiplicity> <ends>"

- Each A is associated with any number of B's.
- Each B is associated with exactly one A.
Basic UML Sequence Diagram Notation

- **Objects**: One that starts the action at the left.

- **Time Flow**:
  - **Object1**
  - **Object2**

- **Messages**:
  - **Create**: `<<create>>`
  - **Destroy**: `<<destroy>>`
  - **Get Info**: `get_info(spec)`
  - **Do Something**: `do_something()`

- **Information Returned**:
  - **Requested Info**: `requested info`
  - **Search Self**: `search-self`

- **Return**:
  - **Void Return**: `void return is implicit`
  - **Non-Void Return**: `information returned`

- **Function Call**: `message sent`

- **Focus of Control**: `object1`