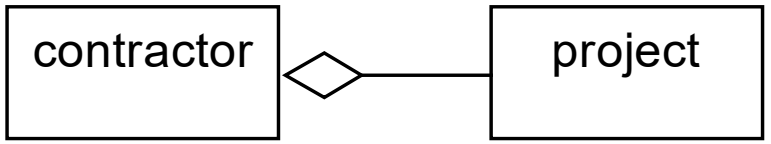
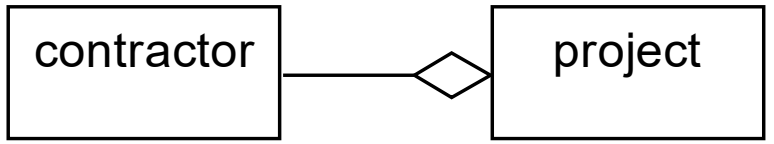
ASSOCIATION

A bidirectional constructive relationship

Delroy A. Brinkerhoff



A contractor has-a project

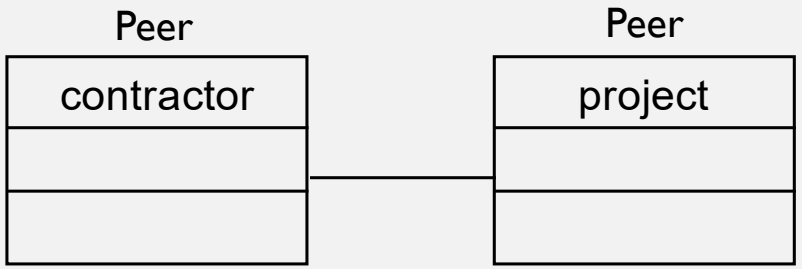


A project has-a contractor

ASSOCIATION: DOUBLE-ENDED AGGREGATION

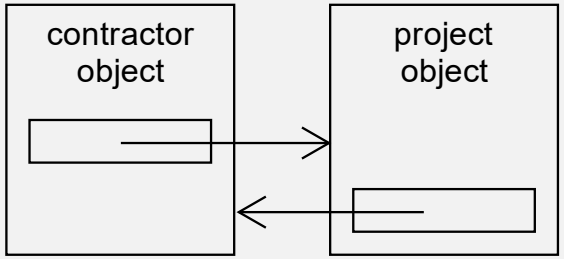
ASSOCIATION: UML SYMBOL AND CLASS ROLES

- Symmetric connector: neither end decorated
- Classes are peers



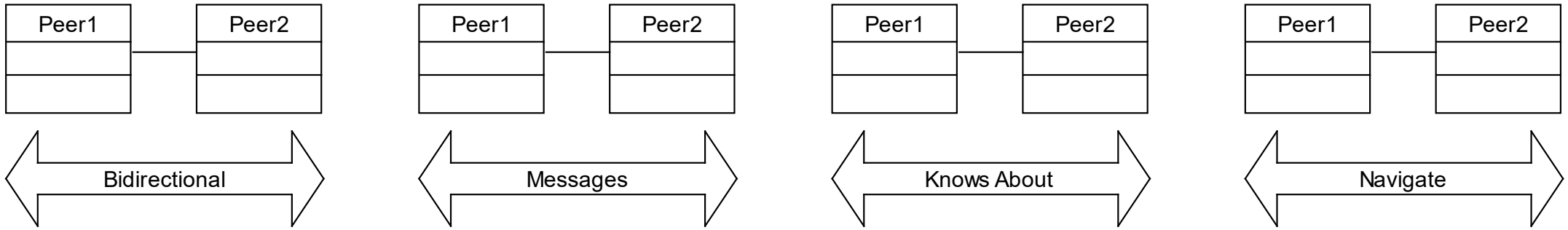
ADDITIONAL ASSOCIATION CHARACTERISTICS

- "Has-A" reading well in both directions
- Implemented with pointer in both classes
- Weak or loose binding
  - Independent lifetimes
  - Objects are shareable

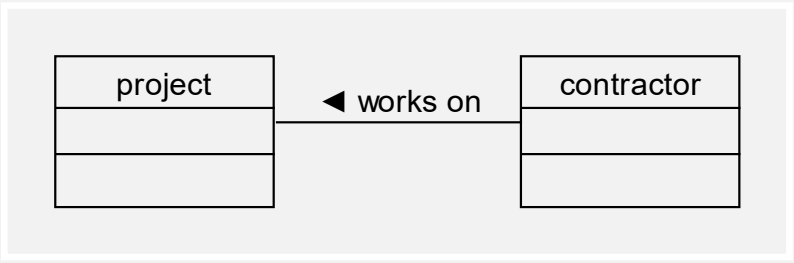


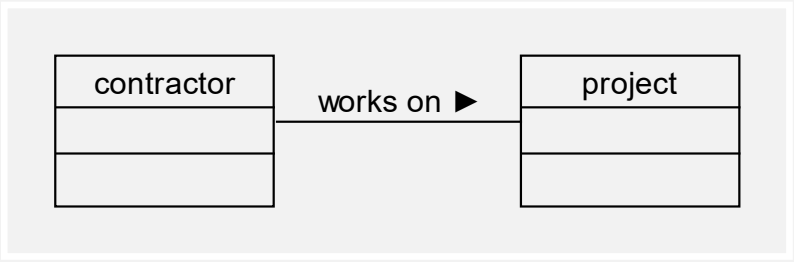

```
class contractor
{
    private:
        project* theProject;
}

class project
{
    private:
        contractor* theContractor;
}
```



DIRECTIONALITY DRIVES ASSOCIATION





READING DIRECTION

- Relationship is still bidirectional
- "Has-A" still reads in both directions
- Reading direction only applies to the relationship label
  - In English, "a contractor works on a project" sounds okay
  - In English, "A project works on a contractor" has a different, strange meaning
  - Indicate the best reading direction with an arrow

IMPLEMENTING ASSOCIATION: FORWARD DECLARATIONS

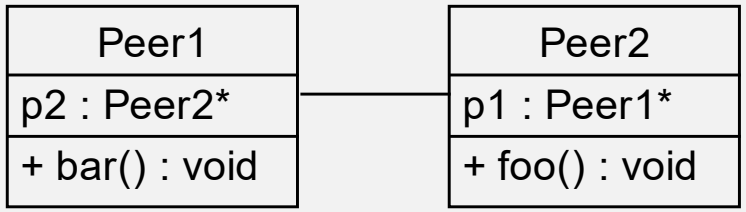
```
class project;

class contractor
{
    private:
        project* theProject;
}
```

```
class contractor;

class project
{
    private:
        contractor* theContractor;
}
```

ASSOCIATION AND INLINE FUNCTIONS




```
class Peer2;

class Peer1
{
    private:
        Peer2* p2;
    public:
        void bar();
};

void Peer1::bar()
{
    p2->foo();
}
```