POLYMORPHISM

POLYMORPHISM

- "having many forms"
- allows class instance to behave like another ancestor class
- **polymorphic reference** - reference variable that can refer to different objects at different points in time

POLYMORPHISM - METHOD BINDING

- Specific method invoked can vary from one invocation to the next
- Depends on what instance the reference currently points to

EXAMPLES

Consider classes:

- `Employee` - abstract
- `StudentWorker` - extends `Employee`
- `Faculty` - extends `Employee`

DUCK TYPING

- Polymorphism matters more in languages with strict typing
- Python has duck typing
    - duck test: "if it walks like a duck and quacks like a duck it's a duck"
    - don't need specific types