Object Oriented Programming – Inheritance vs. Interface vs. Composition

"Code with Passion!"

Choices for implementing changing behavior

- Option #1: Use Inheritance
- Option #2: Use Interface
- Option #3: Use Composition

Option #1: Use Inheritance

- Use it when changing behaviors are aligned only with subclasses
- Use it when changing behaviors depend on the attributes on the subclasses
- Example scenario:
  - Computing monthly pay for all Employee types – hourly employee and salaried employee
  - Both hourly employee and salaried employee are Employee type
  - Hourly employee has “number of hours” attribute while salaried employee has “yearly salary”
  - “Compute monthly pay” behavior needs to be computed differently

Option #2: Use Interface

- Use it when changing behavior need to be imposed to classes that are not related via inheritance
- Example scenario:
  - Computing bonus for salaried employees and board members
  - Salaried employee and board member do not share a parent
  - Yet, “compute bonus” logic can be imposed to both
  - But the logic need to be computed differently between salaries employee and board member

Option #3: Use Composition

- Objects with changing behavior injected to the target class
  - Dependency injection of the changing behavior during runtime
- Use it when runtime selection of behavior is desired
- Use it when reuse of the behavior can be used in other classes
- Example scenario
  - Customer class wants to have payment or shipping preference behavior
  - Customer class has payment and shipping preference dependencies
  - Payment option and Shipping preference behavior injected during runtime

Code with Passion!