

Pattern Exercise

CSCI 4448/5448: Object-Oriented Analysis & Design

Lecture 20

Acknowledgement & Materials Copyright

- I'd like to start by acknowledging Dr. Ken Anderson
- Ken is a Professor and the Chair of the Department of Computer Science
- Ken taught OOAD on several occasions, and has graciously allowed me to use his copyrighted material for this instance of the class
- Although I will modify the materials to update and personalize this class, the original materials this class is based on are all copyrighted © Kenneth M. Anderson; the materials are used with his consent; and this use in no way challenges his copyright

How well do you know your patterns?

- Prepare yourself...
- Clear away all notes and connected machinery (other than Zoom)
- Answer the Task questions WITH NO NOTES or LOOKING THINGS UP ON PHONES/PC/WEB – don't do it!
- Because we're remote for this, it'll be solo work
- You'll need to be able to draw patterns or capture notes on paper (or maybe electronically)
- Score your results: keep track of your total score on all tasks
- Please remember – if you're not here participating in person, or you are, and it doesn't go well, there will be other avenues for extra credit and participation – don't panic 😊

Task 1

- Draw the UML Class Diagram for the Strategy Pattern

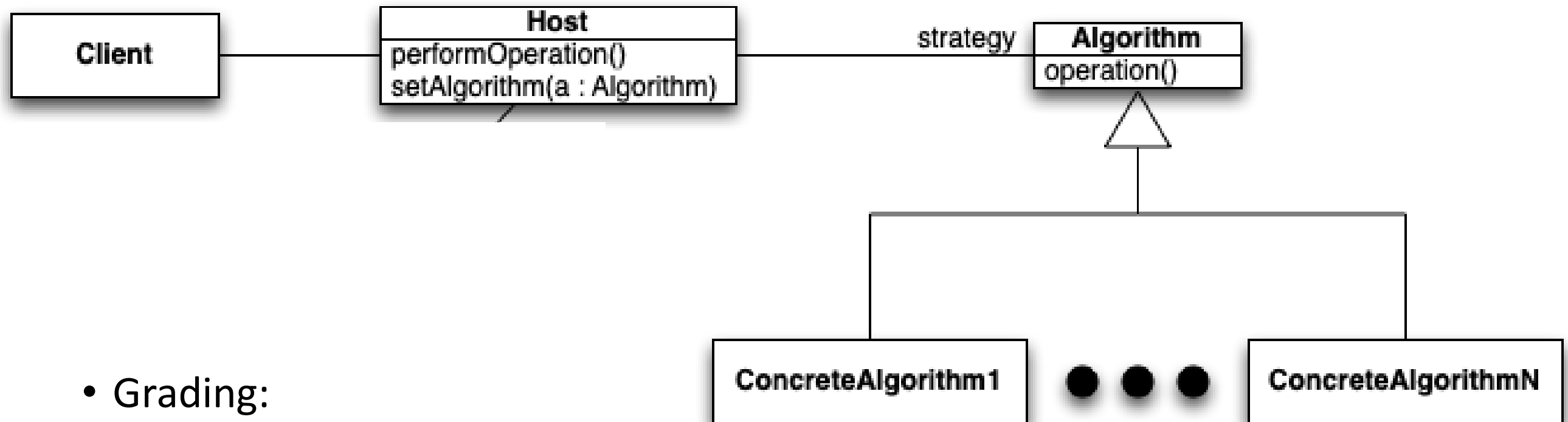
AND

- Complete this OO Principle
- Favor _____ over _____

Grading the tasks

- You'll be grading your own work here...
- Patterns
 - The drawings can be arranged differently – look for parts and connections
 - Exact match or really, really close = Perfect
 - Not bad, missed a couple of things = Close
 - Otherwise = A Duck
- Phrases/Definitions
 - must be right (spelling doesn't count) to get the point

Task 1 – Strategy Answers



- Grading:
- Perfect! = 2 points
- Close! = 1 point
- Looks like a drawing of a duck = 0 points
- Favor **Delegation (or Composition)** over **Inheritance**.
- 1 Point if right, 0 if wrong or somehow rude

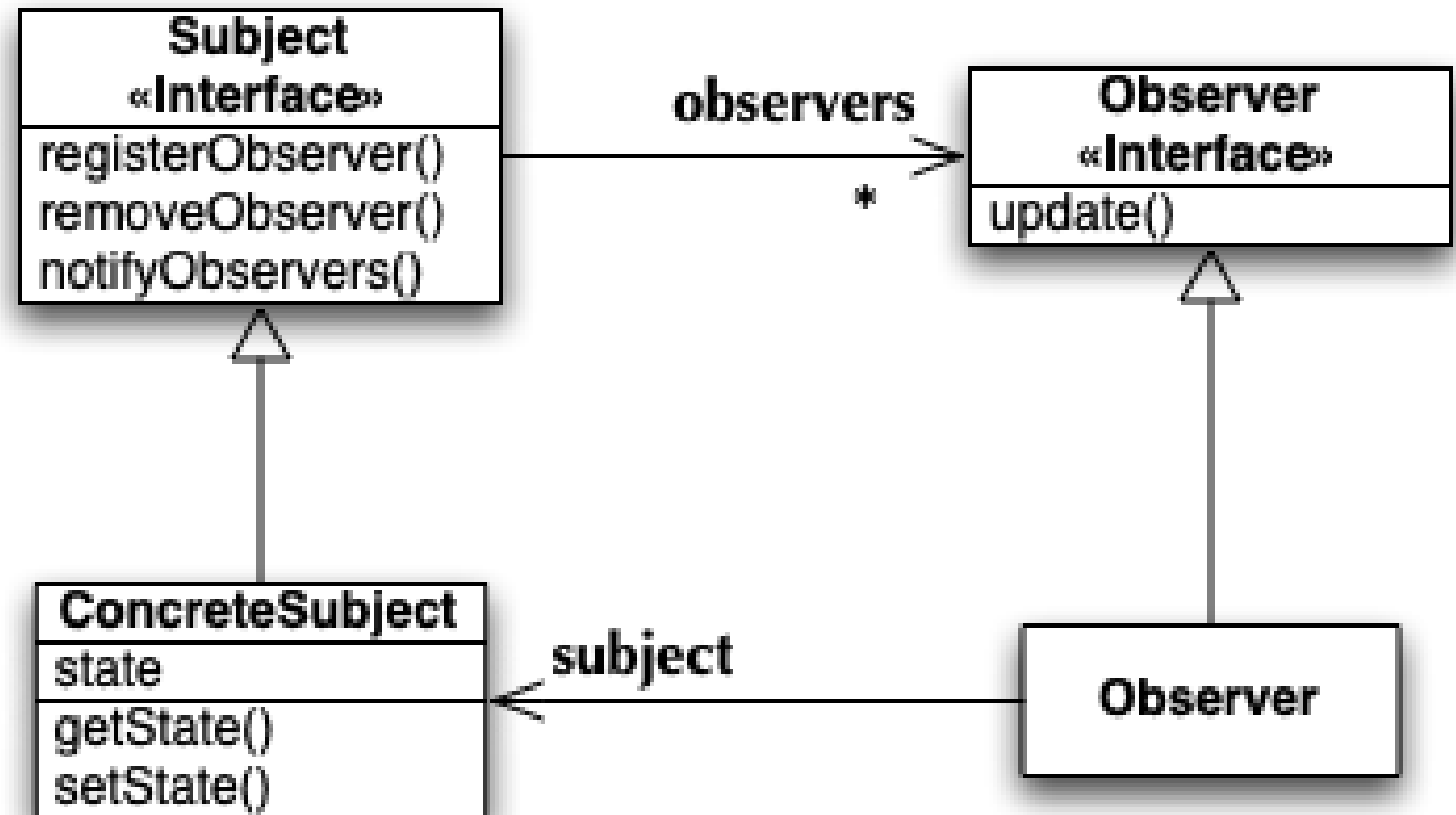
Task 2

- Draw the UML Class Diagram for the Observer Pattern

AND

- Observer implementations support _____ coupling and can use a _____ or _____ approach to get events to subscribers.

Task 2 – Observer Answers



- Grading:
- Perfect! = 2 points
- Close! = 1 point
- Duck issues = 0 points
- Observer implementations support **loose** coupling and can use a **push** or **pull** approach to get events to subscribers.
- 1 Point if right, 0 if otherwise

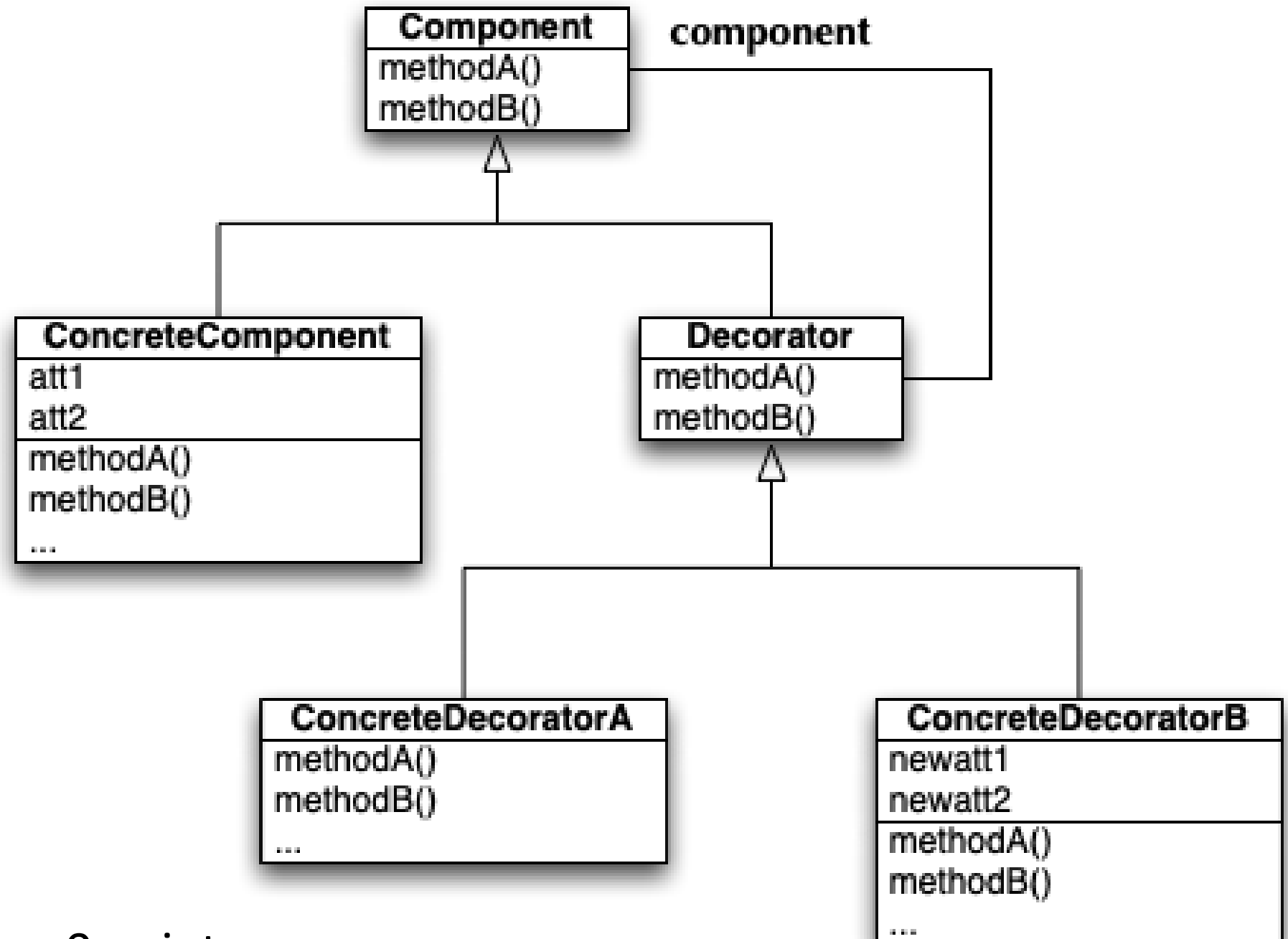
Task 3

- Draw the UML Class Diagram for the Decorator Pattern

AND

- Open Closed Principle: Classes should be open for _____ but closed to _____

Task 3 – Decorator Answers



- Grading:
 - Perfect! = 2 points
 - Close! = 1 point
 - Definitely duck-like = 0 points
-
- Open Closed Principle: Classes should be open for **extension** but closed to **modification**
 - 1 Point if right, 0 if not

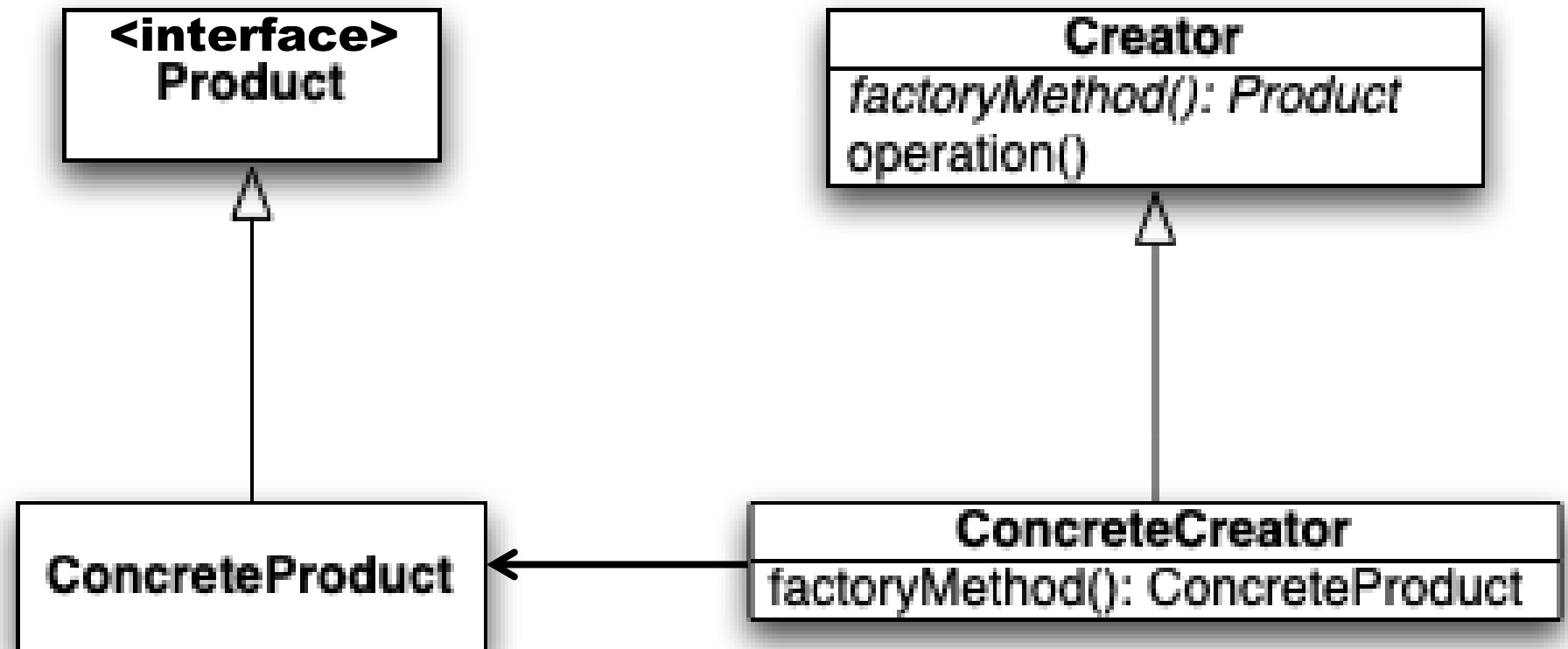
Task 4

- Draw the UML Class Diagram for the Factory Pattern (not simple factory, not abstract factory)

AND

- Dependency Inversion Principle: Depend upon _____. Do not depend upon _____.

Task 4 – Factory Answers



- Grading:
 - Perfect! = 2 points
 - Close! = 1 point
 - A bit too ducky = 0 points
-
- Dependency Inversion Principle: Depend upon abstractions. Do not depend upon concrete classes.
 - 1 Point if right, 0 if otherwise

Possible Points

- Strategy 3
- Observer 3
- Decorator 3
- Factory 3
- 12 points? Who's got it?
- Do we have a tie for first?

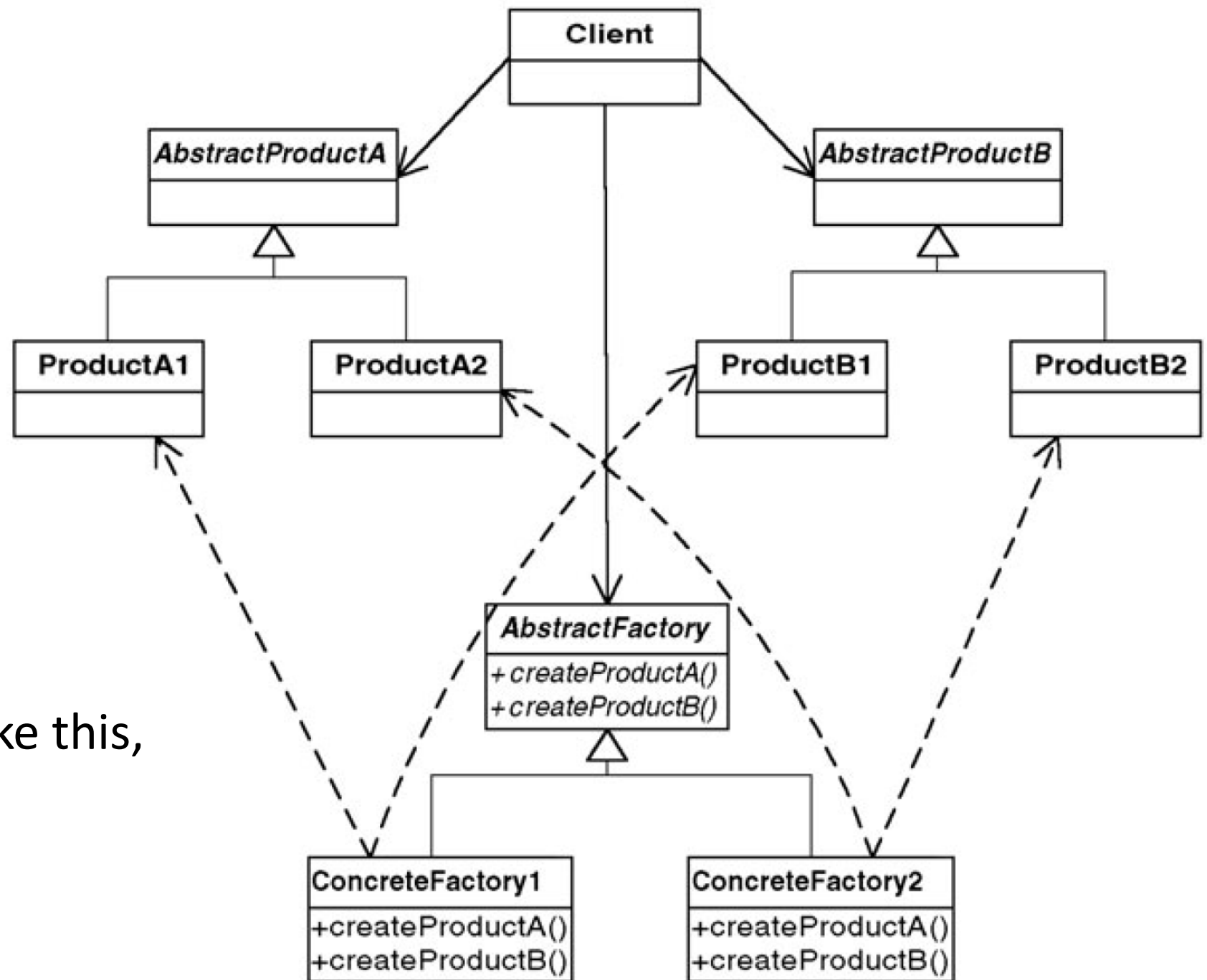
Tiebreaker Task... Abstract Factory

- Draw the UML Class Diagram for the Abstract Factory Pattern

AND

- Factories build objects with _____, abstract factories build families of objects with _____.

Tiebreaker – Abstract Factory Answers



- Grading:
- May not look exactly like this, look for the parts!
- Perfect! = 2 points
- Close! = 1 point
- All I see is duck = 0 points

- Factories build objects with inheritance; abstract factories build families of objects with (object) composition.
- 1 Point if right, 0 if otherwise

Close It Up; Bring It In

- Top Scoring Player(s) – 2 Bonus Points
- Next Level Score(s) – 1 Bonus Points
- Thanks for playing!
- You won't have to reproduce these UML patterns on the online exams, but you may need to recognize and understand them!

Close It Up; Bring It In

- ~~Top Scoring Player(s) – 2 Quiz Points~~
- ~~Next Level Score(s) – 1 Quiz Points~~
- October special: 2 Bonus Points for all Participants! Add your identikey to the Zoom chat for credit!
- Thanks for playing!
- You won't have to reproduce these UML patterns on the online exams, but you may need to recognize and understand them!