

Questions? 😊

# Welcoming Exercise

- Write a Java class that can handle a student's record including:
  - First name
  - Last name
  - Project grade (20%)
  - Midterm exam grade (40%)
  - Final exam grade (40%)
- and also computes and displays the Final Grade.
  - $FG = (PG * 0.20) + (ME * 0.40) + (FE * 0.40)$



**30:00**



Object-Oriented  
Programming

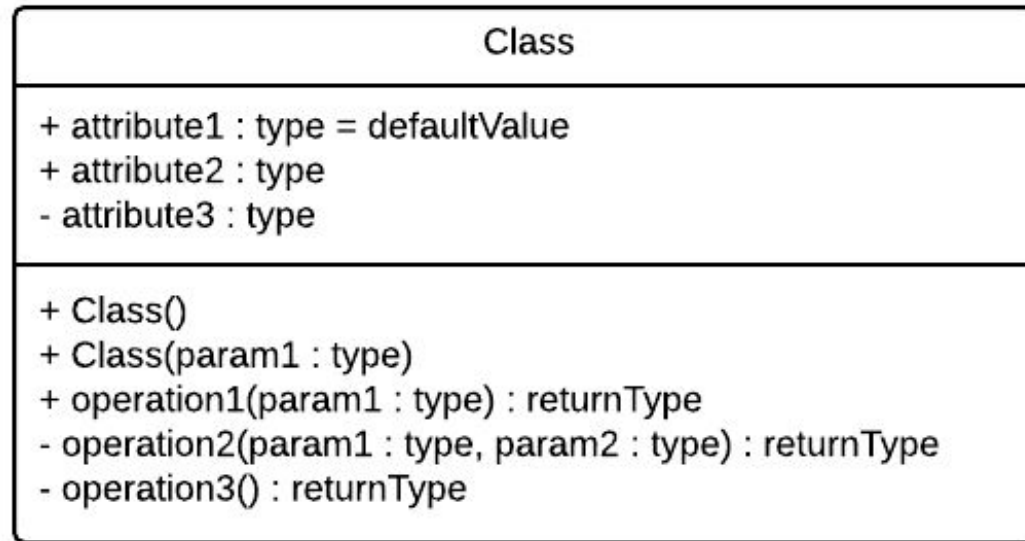
# Basics of Unified Modeling Language (UML)

# Unified Modeling Language

- Able to represent a system / entities
- Aids in object-oriented design process
  - Tool for visual representation
- UML can be **directly translated** to OO languages
- There are many types of UMLs, but we'll mainly focus on the **class diagram**

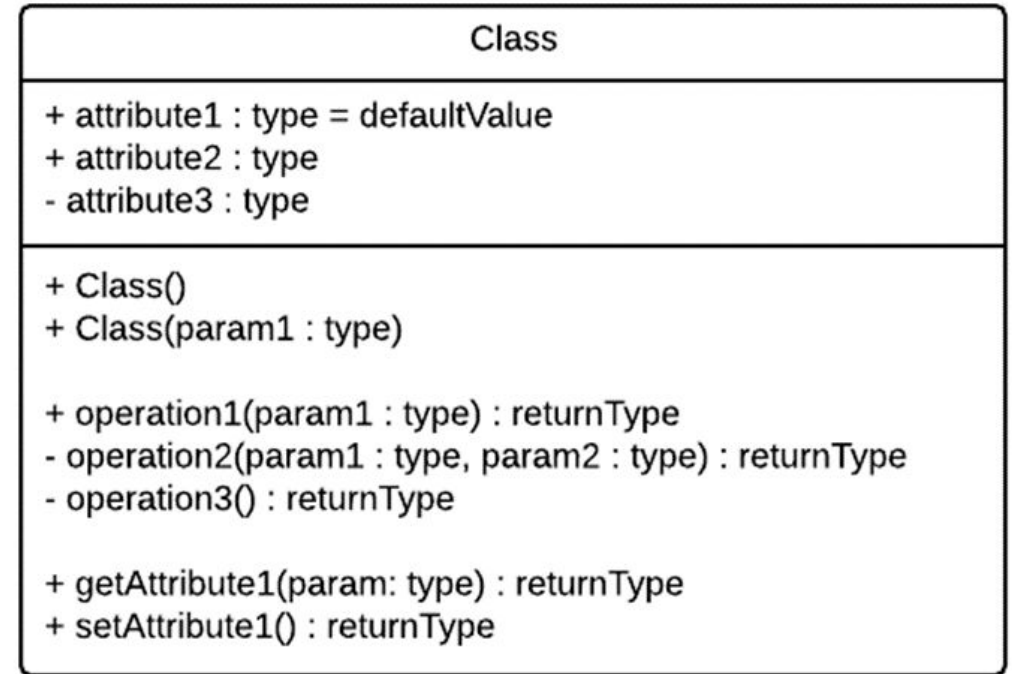
# UML – Class Diagram

- If a class is a blueprint for an object, then we can use UML as the blueprint of the implementation



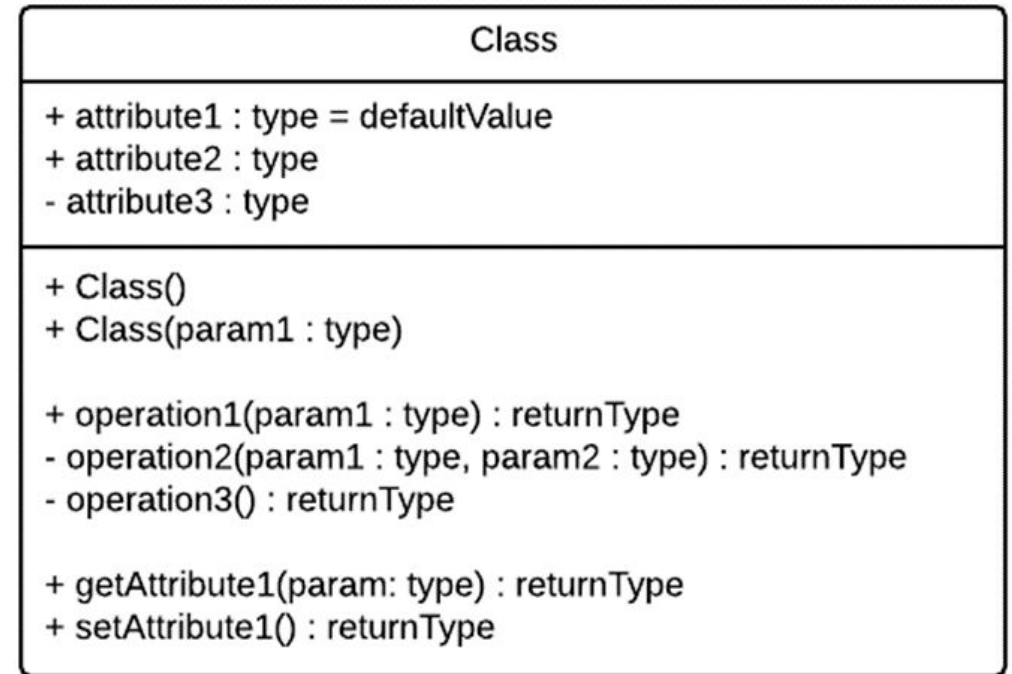
# UML – Class Diagram

- Three parts:
  - Top (class name)
  - Middle (attributes)
  - Bottom (methods)



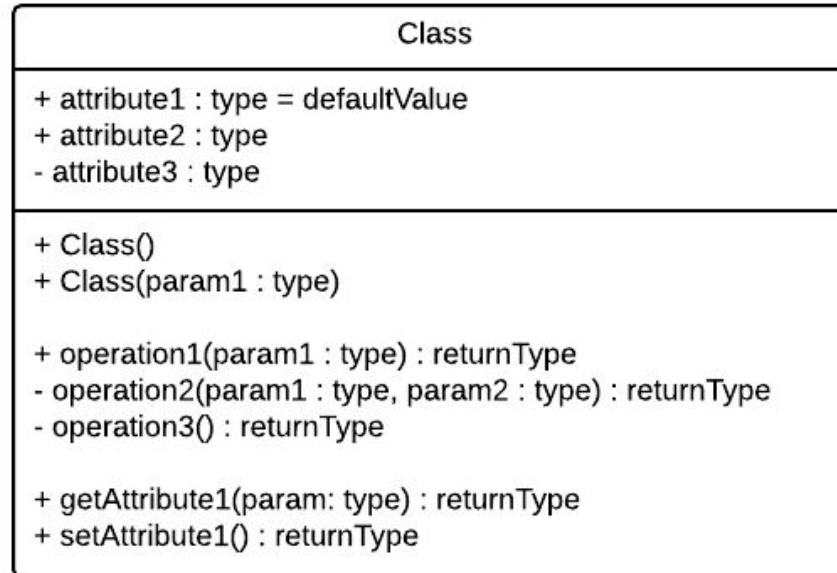
# UML – Class Diagram

- Symbols for access modifiers:
  - Public (+)
  - Private (-)
  - Protected (#)
  - Package-private (~)
  - Derived (/)
  - Static (underlined)



# UML – Class Diagram

- Attributes
  - `<access> <name> : <data type> (= <value>)`

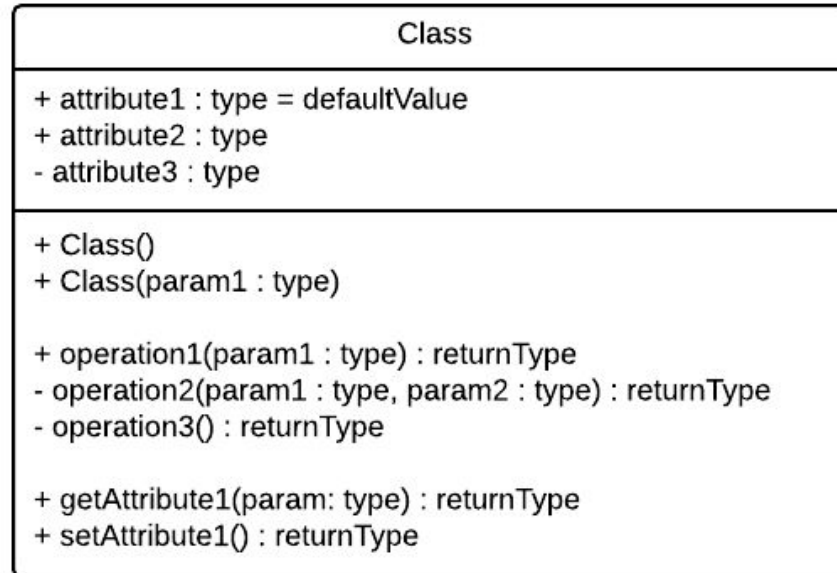




# UML – Class Diagram

- Method

- `<access> <name>(<parameters>) : <returnType>`



# UML – Class Diagram

- Tools for creating these diagrams
  - Powerpoint
    - Painstaking
  - Draw.io
    - Most accessible
  - Lucidchart
    - Best tool, but there are limits to the functionality even if you're on an education plan
- Explore the tools and find one that you'd find comfort in

# Note

- This session only gives you the bare minimum to represent an entity
- We'll talk about other useful class diagram syntax to use when we introduce class relationships

Questions? 😊

# Practice 3 – Class Diagram of a Tweet

- Practice to familiarize yourself with creating diagrams
- Submit before class end
- Serves as your [*mental*] attendance

# Disclaimer

- I may not be able to ALWAYS check your practice exercises and give feedback
- Thus, please raise your questions during the class OR posting a comment under a Discussion thread released after our class

# Assignment

- Practice 4 - Class Diagram of a Shopping Item
- No need to submit but make sure to practice
- This will ready you for the next Graded Exercise

# Next meeting

- (Short) Lecture about ***Overloading***
  - (you may search/read on this in advance)
  - discussion/questions regarding Practice 4
- Graded Exercise 3 (UML and coding)



# Last Remarks

- Your default mindset might be to code first and then implement a class diagram based on your code
- However, try to avoid this and prioritize designing your classes first before implementing them
- A well-designed class diagram gives you the entire structure of a class while abstracting the actual (*detailed*) implementation itself

Keep learning...