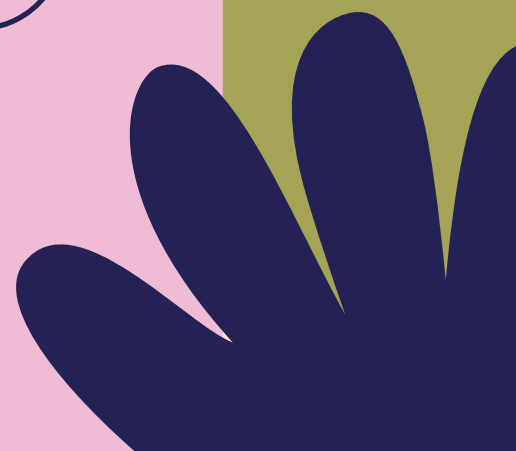# COMP2511

## Tute02

# Agenda

- Code Review
- JavaDoc & Commenting
- Inheritance & Polymorphism
- toString & Equals
- Access Modifiers & Packages
- Assignment i!!!

# Code Review

# Code Review

**super vs this**

- What is the difference between super and this?
- What about super(...) and this(...)?
- What will the main code in `Rectangle print and why?

# Code Review

**Abstract Classes vs Interfaces**

- What is the purpose of using an abstract class in this code?
- What are some downsides of the use of the abstract class here?
- What is the difference between an abstract class and an interface? Why would you use one or the other?

# Code Review

static

- Can you override a static method?
- What is the output of running r2.getArea() in main?
- What is the output of running Shape.getCount() in main?

# JavaDoc & Commenting

- What are the main features of the JavaDoc present in Employee.java?
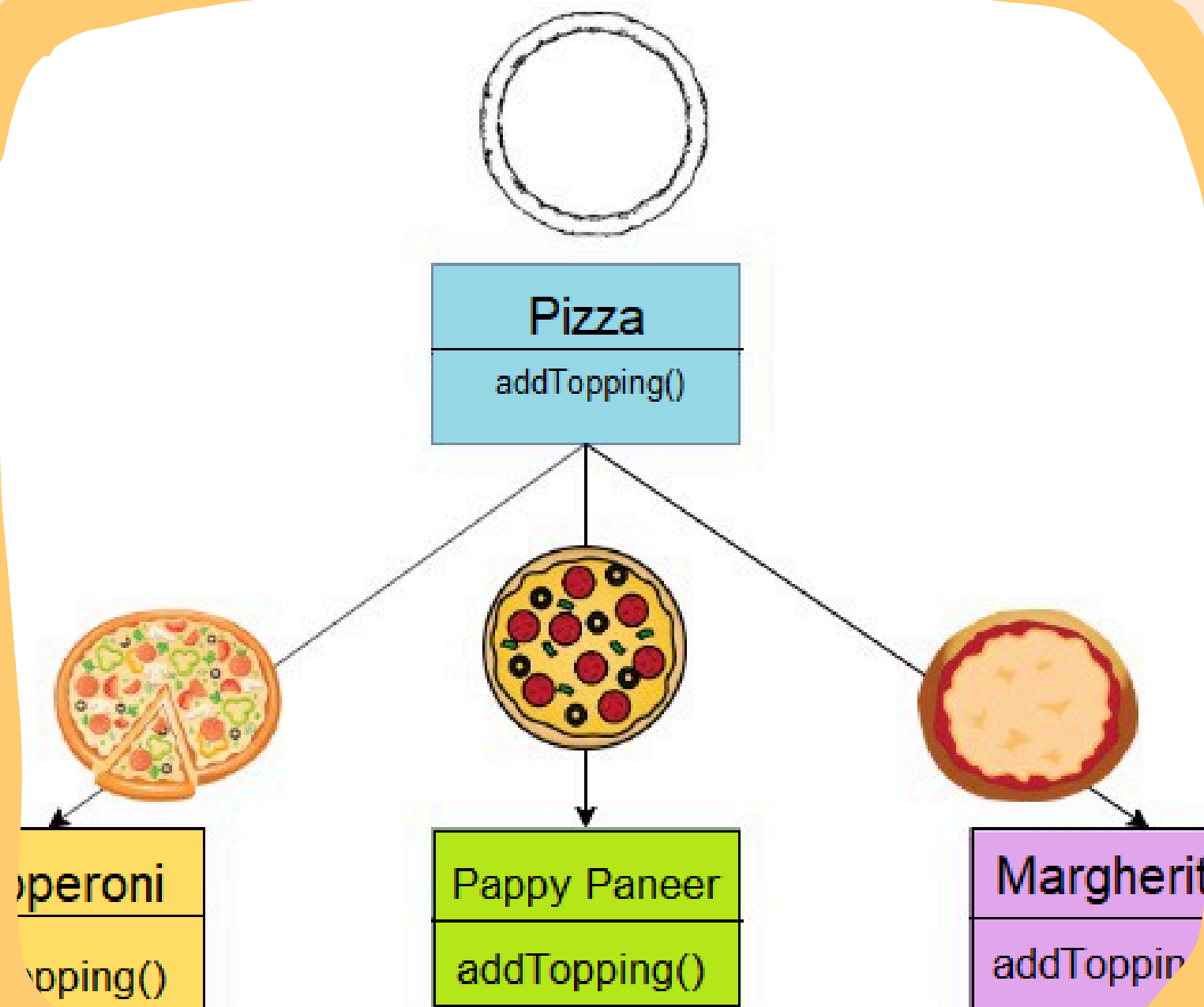- What is meant by the term "self-documenting code"?

# Why Use JavaDoc & Comments?

And when not to?

# Inheritance & Polymorphism



Create a Manager subclass of Employee, that includes hire date.

- What constructors are appropriate?
- Is it appropriate to have a getter for the hire date? What about a setter?
- Why might adding certain getters and setters be bad design?

# toString

Override the toString() method inherited from Object in both classes.

- What should the result of toString() contain?
- Does the method in Manager call the one in Employee?

# The == Operator

**4** What does == do when comparing objects?

**2** Where have we seen something similar in other languages?

**3** How was the underlying data checked for equality?

# The == Operator

**4**

In Java, how can we check 2 objects are equal?

**5**

What does it mean for objects to be equal?

**6**

When are a super type and a sub type equal?

# equals

Override the equals() method inherited from Object in both classes.

- What key things should the equals() method do?
- How should the type of the input object be checked? How should it be compared to the type of the calling object?
- How can the method in Manager utilise code in Employee to avoid repetition?
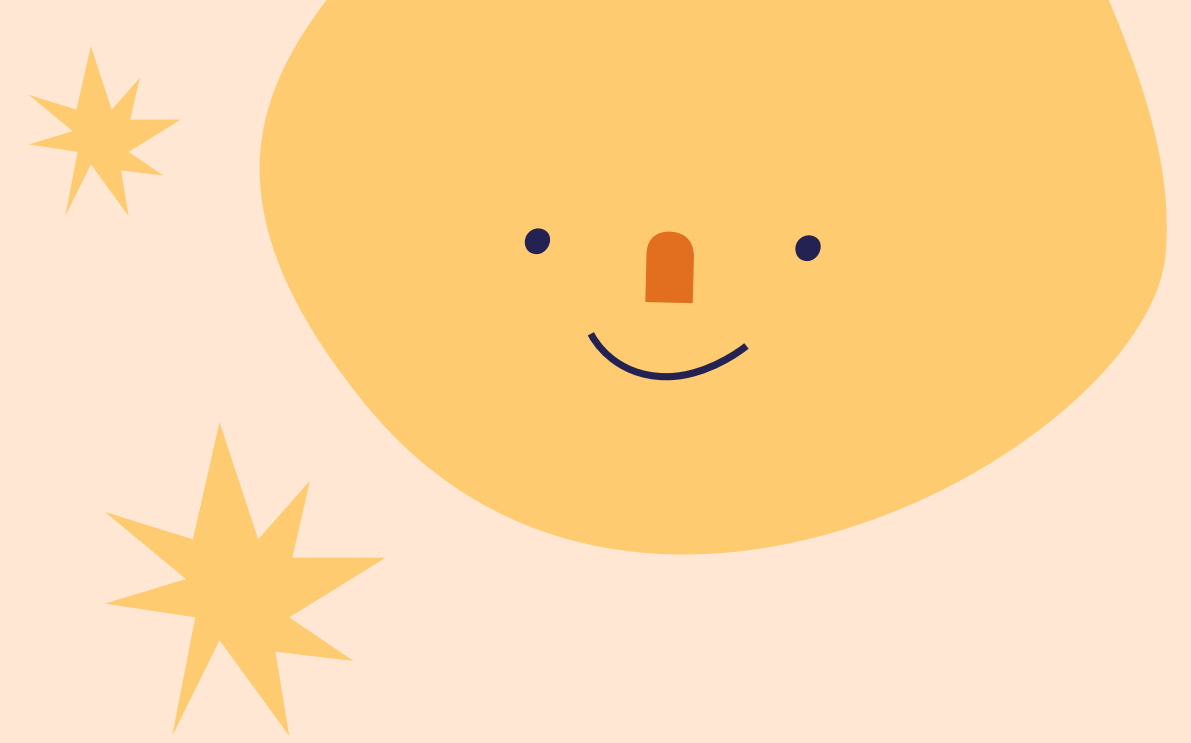
# Access Modifiers & Packages

# Access Modifiers

| Modifier | Class | Package | Subclass | Global |
|---|---|---|---|---|
| Public | ✔ | ✔ | ✔ | ✔ |
| Protected | ✔ | ✔ | ✔ | ✘ |
| Default | ✔ | ✔ | ✘ | ✘ |
| Private | ✔ | ✘ | ✘ | ✘ |

# ASSIGMENT i!!!

- Assignment i IS OUT! - so take a look :D
- We will go through assignment in detail in Wk3 Lecture!
- Start it early!!! - maybe wait until after Wk3 Lecture if you don't understand the assignment well
- REMEMBER THIS IS A DESIGN COURSE!!! PLAN OUT YOUR DESIGN BEFORE YOU START CODING!!!