


COMP 3550


5.2 — DRY, KISS, POLA

Week 5: Design Principles &
Refactoring

DRY — DON'T REPEAT YOURSELF

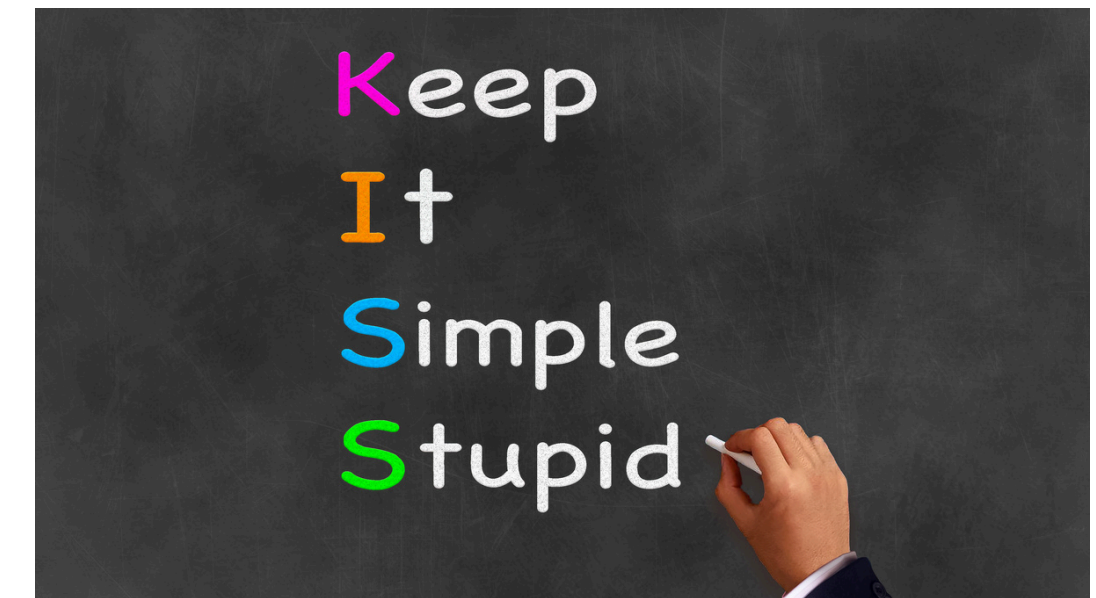
- Copy/Paste code from ANYWHERE is a 
- It should give you pause to ask, “*what am I doing?*”
- Same logic in 2+ places can mean bugs
- Refactor to:
 - methods
 - helpers
 - shared classes

DRY — DON'T REPEAT YOURSELF

- Copy/Paste code from ANYWHERE is a 
- It should give you pause to ask, “*what am I doing?*”
- Same logic in 2+ places can mean bugs
- Refactor to:
 - methods
 - helpers
 - shared classes
- Let's do an example (BadApp52.java)

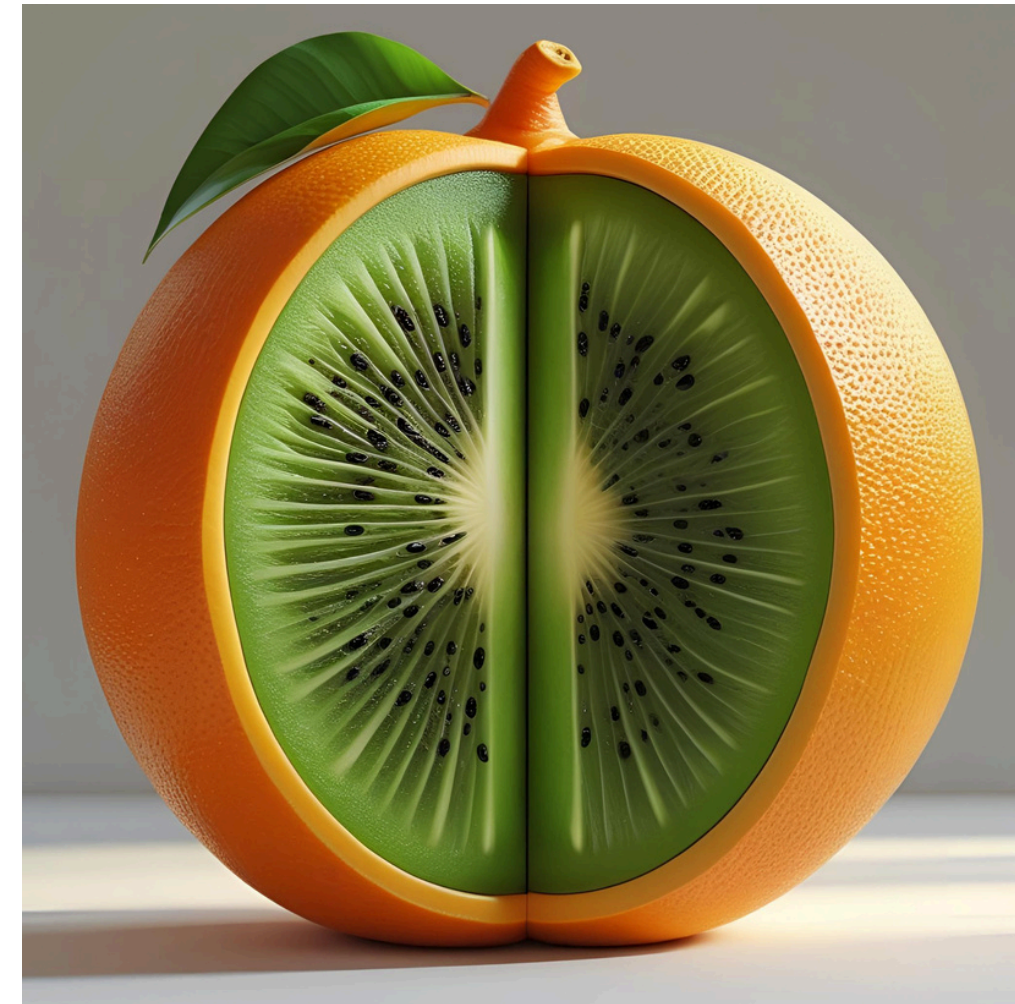
KISS — KEEP IT SIMPLE, STUPID

- Avoid overengineering
- “Don’t build a rocketship when you need a bike”
- Let’s look at a complete program (BankApp52.java):
 - Let’s say a business person wants to write a program which Greet’s the user when they sign in. That’s all it needs to do.



POLA — PRINCIPLE OF LEAST ASTONISHMENT

- Code should do what users/devs expect
- Add vs Save
- Let's look at a subtle example existing in Java example of this (PoLA52.java)



BAD EXAMPLE REFACTOR

- Let's look at a larger example of code (OrderProcessor52.java)

PAUSE & PROJECT REFLECT

- Go look through the code you have worked on this week and find one thing in your project that's more complex than needed.
- Can you simplify it?
- Make a git issue defining the problem and solution.