



University of
South Australia

INFS 2044

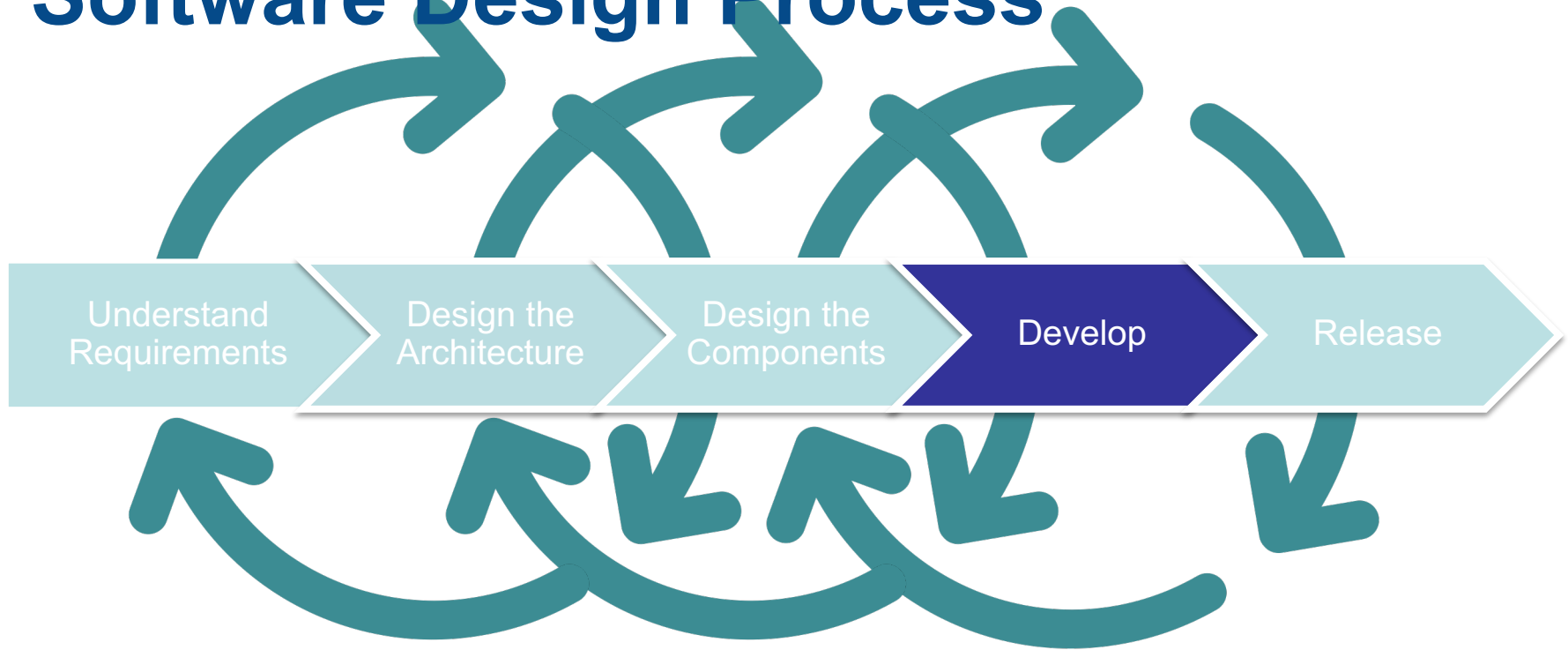
Practical 3

Preparation

- Read the required readings
- Watch the Week 12 Lecture



Software Design Process



Learning Objectives

- Select intention-revealing names for variables, methods, and classes, and write informative comments (CO4)
- Focus on code quality, style, and documentation



Task 1. Review Names

- Review the names chosen in the implementation design shown on the subsequent slide. Download and examine the corresponding code in `Prac3_initial_code.py`
 - Do all classes, methods, variables, etc have intention-revealing names?
 - Is it clear from the method name what the method does?
 - Are names used consistently?
 - If not, try to come up with better names.

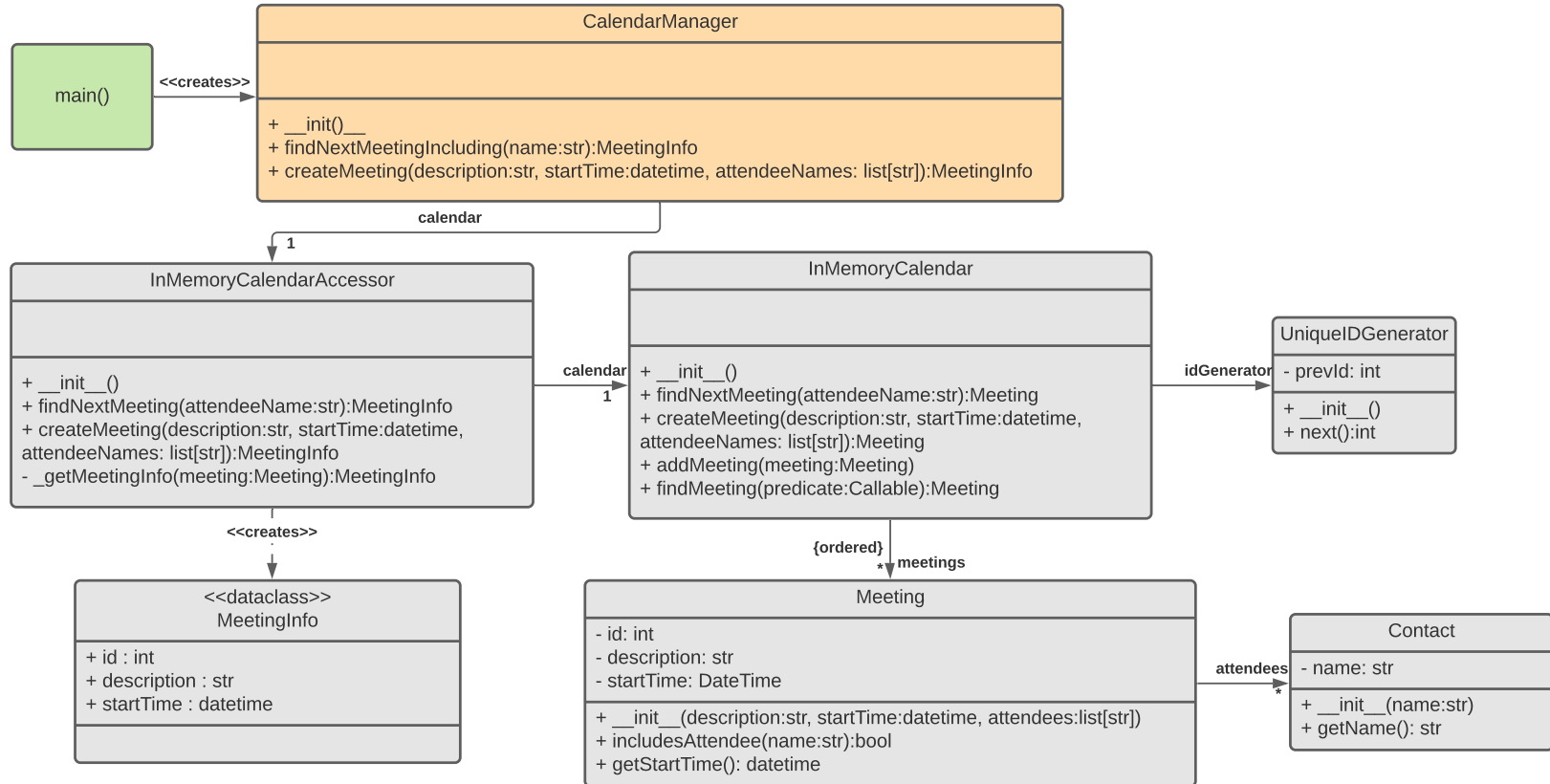


Principles of Good Names

- Intention-revealing
- Ambiguity avoiding
- No “false clues”
- Pronounceable
- Searchable
- Phrasing
 - Classes: noun phrases
 - Methods: verb phrases
- Specific
- Consistent



Practical 1 Implementation Design



Refactor the Code

- Download the code `Prac3_initial_code.py` code (if not already done)
- Rename the methods as shown on the previous slides
- Also take the opportunity to ensure that all tests have intention-revealing names
 - It should be clear from the name of the test what is being tested



Task 2. Commenting

- Which parts of the design and its corresponding source code of the Calendar system require a comment?
- Why?
- Is there something the programmer should know that is not immediately evident from reading each class/method in isolation?



Principles of Good Comments

- **Describe things that are not obvious from the code**

Other acceptable comments:

- Legal comments, copyright, etc
- Explanation of intent
- Warning of consequences
- Amplification
- TODOs
- Self Documentation



Task 3. Removing Comments

- Are there any comments in the given code that are undesirable?
- Why?



Unacceptable Comments

- Redundant (delete)
- Misleading (change or delete)
- Change logs, journals (use version control for this)
- Position and section markers (break up files)
- Closing brace markers (use shorter functions)
- Commented out code (delete, use version control)
- Too much information (relegate to a separate file)



You Should Know

- Detect and resolve poor naming
- Write explanatory comments
- Organise source code into separate files or packages



Activities this Week

- Complete Quiz 8
- Continue working on Assignment 2





**University of
South Australia**