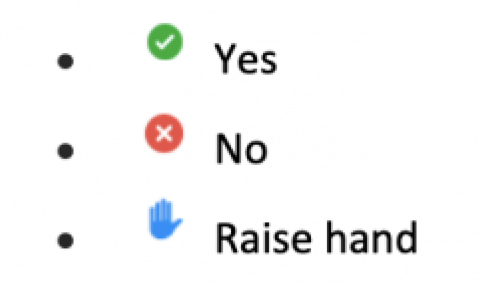
48024 Applications Programming

Dr Angela Huo

Zoom Manner

- Chat: Message the lecturer if something is wrong
- Icon: Slow down or faster?
  - When you have a question, click “Raise hand”, I will unmute you and you can speak
  - Vote for a poll, click “Yes” or “No”
- Padlet: https://padlet.com/angelahuo/appsprog
  - I will answer the posted questions at the end of the lecture.



- ✓ Yes
- ✗ No
- ✋ Raise hand

This subject adopts “flipped learning” teaching strategy. You need to complete the “Pre-class” activities to prepare for the interactive quiz and application questions in lectures, which help you prepare for the LMS exam.

Contents

Prep: Open Question Board(https://padlet.com/angelahuo/appsprog)

1. Announcement
2. Lab Review
3. Week 5 Reflection
4. Lab5 Preview
5. Question Board

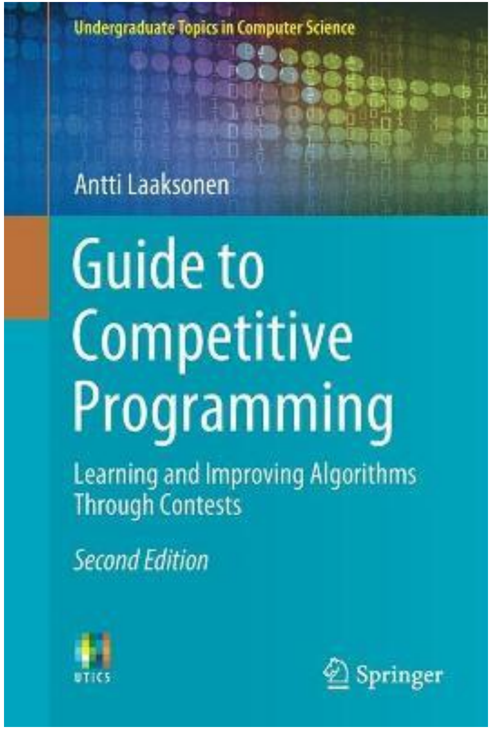
Post your question to https://padlet.com/angelahuo/appsprog

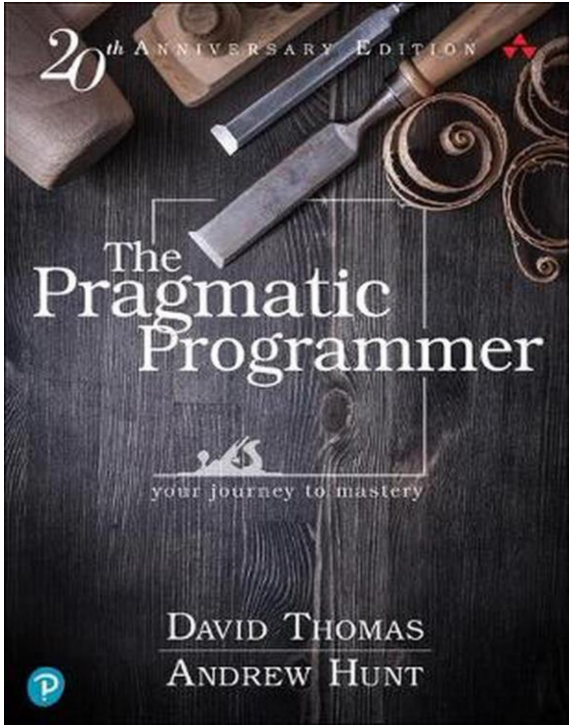
Announcement

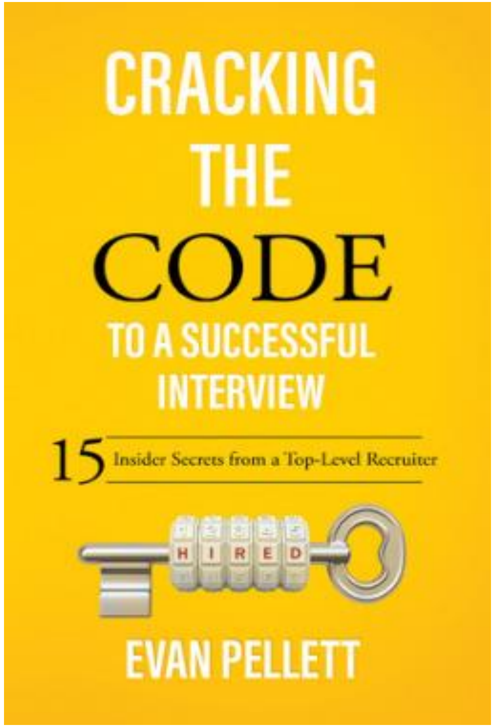
- South Pacific Programming Contests
  - South Pacific ICPC: 15th October
  - ANZAC Contests: 26th March
- code jam by Google: 1st April 2022, 02:00(UTC) 27 hrs
- Refresher module 3 and module 4 are helpful for Lab 5.
- Tutor code solution will be posted on Canvas. Assessment code please consult your tutor.
- Assignment 1 will be released at the end of this week.
  - Study 6 is the foundation for assignment 1. You must complete Study 6 before access assignment 1.

Post your question to https://padlet.com/angelahuo/appsprog

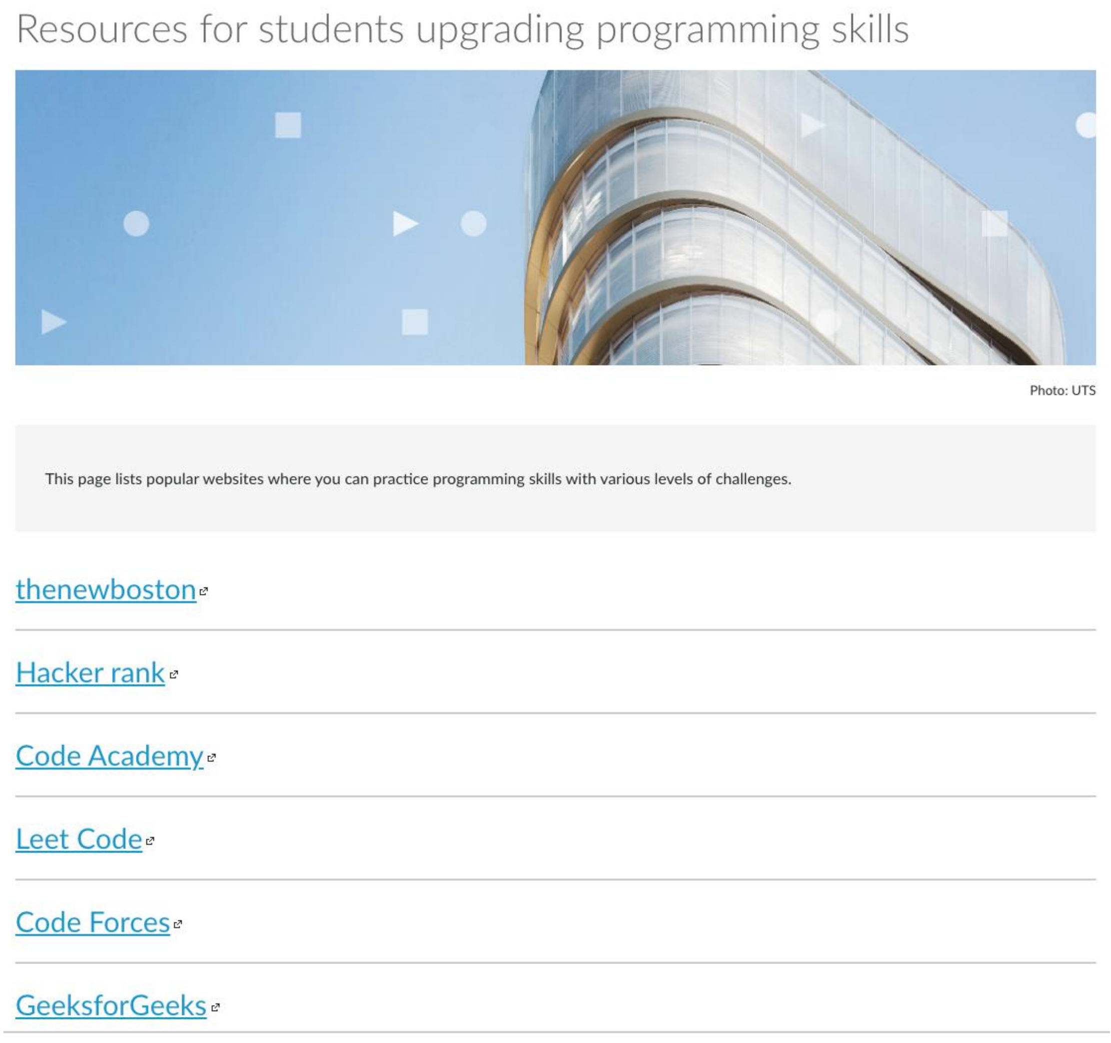
Upgrade Books


Undergraduate Topics in Computer Science
Antti Laaksonen
Guide to Competitive Programming
Learning and Improving Algorithms Through Contests
Second Edition
UTICS
Springer




CRACKING THE CODE TO A SUCCESSFUL INTERVIEW
15 Insider Secrets from a Top-Level Recruiter
HIRED
EVAN PELLETT

Online Resource



Resources for students upgrading programming skills

Photo: UTS

This page lists popular websites where you can practice programming skills with various levels of challenges.

- thenewboston
- Hacker rank
- Code Academy
- Leet Code
- Code Forces
- GeeksforGeeks

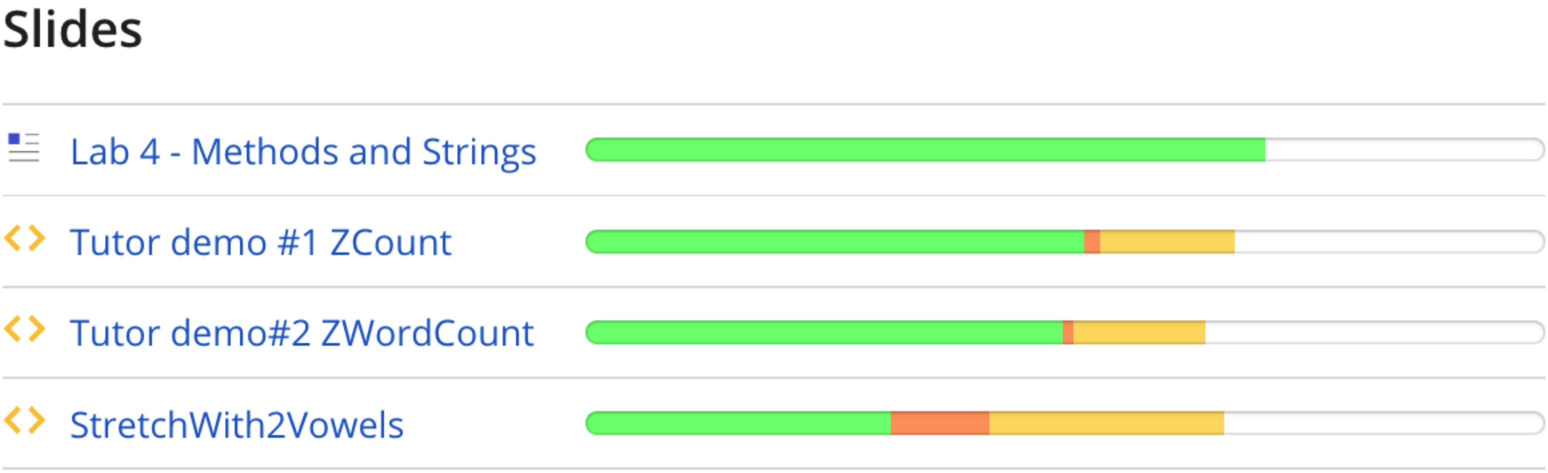
Lab 3 Result:

- 20 students’ submissions are detected as “High similarity” by the system.
  It means 100% codes are identical.
  If you receive a feedback with grade “0”, you are on the blacklist.

- **NOTE: You should write every line of code yourself except for code copied from the lecture notes, lecture code or lab code.**

- What if it happens again?
  You will be marked ”0” and reported for misconduct investigation!

Lab 4 Review

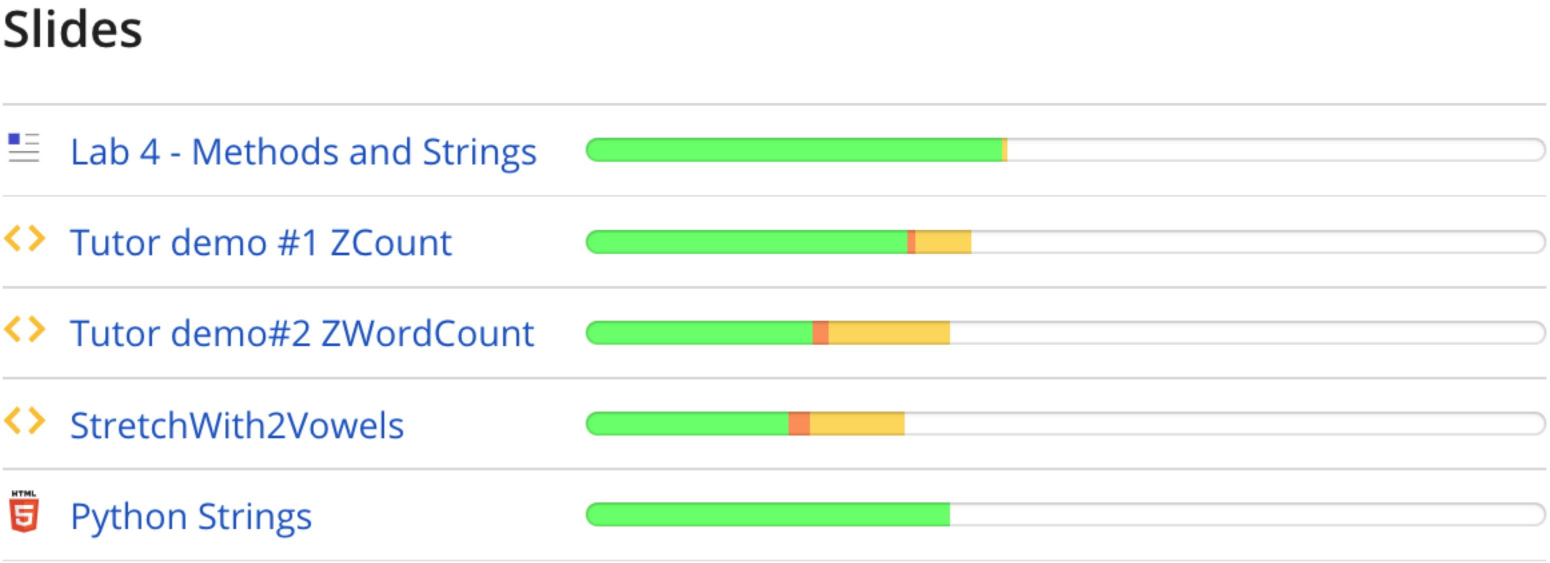


Slides

- Lab 4 - Methods and Strings
- Tutor demo #1 ZCount
- Tutor demo#2 ZWordCount
- StretchWith2Vowels

Post your question to https://padlet.com/angelahuo/appsprog

Lab 4 Python Review



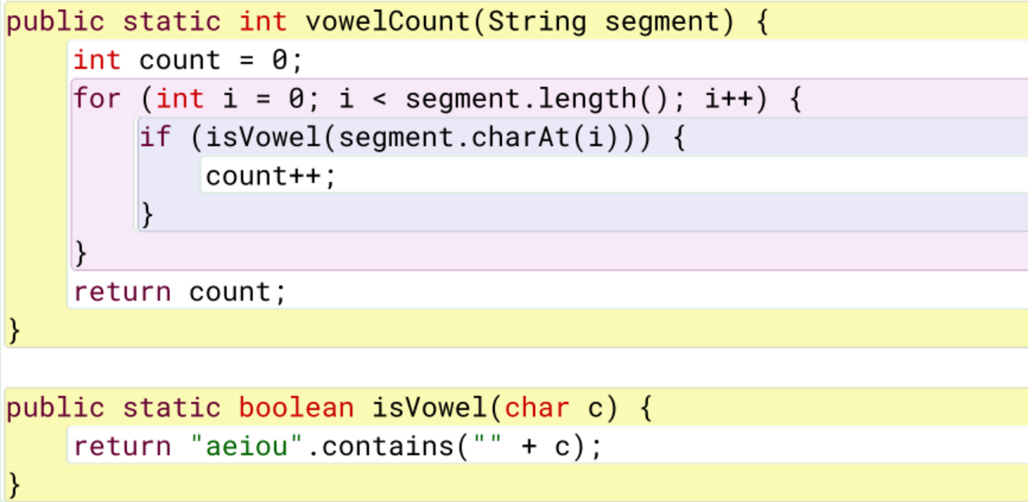
Slides

- Lab 4 - Methods and Strings
- Tutor demo #1 ZCount
- Tutor demo#2 ZWordCount
- StretchWith2Vowels
- Python Strings

Post your question to https://padlet.com/angelahuo/appsprog

Lab 4 Review

- Methods
  - ○ Design: input(Formal Parameter), process, output
  - ○ Application: parameter(Real Parameter)
- Life Span
  - ○ Close brace



```
public static int vowelCount(String segment) {
    int count = 0;
    for (int i = 0; i < segment.length(); i++) {
        if (isVowel(segment.charAt(i))) {
            count++;
        }
    }
    return count;
}

public static boolean isVowel(char c) {
    return "aeiou".contains("" + c);
}
```

P

Lab4 Review

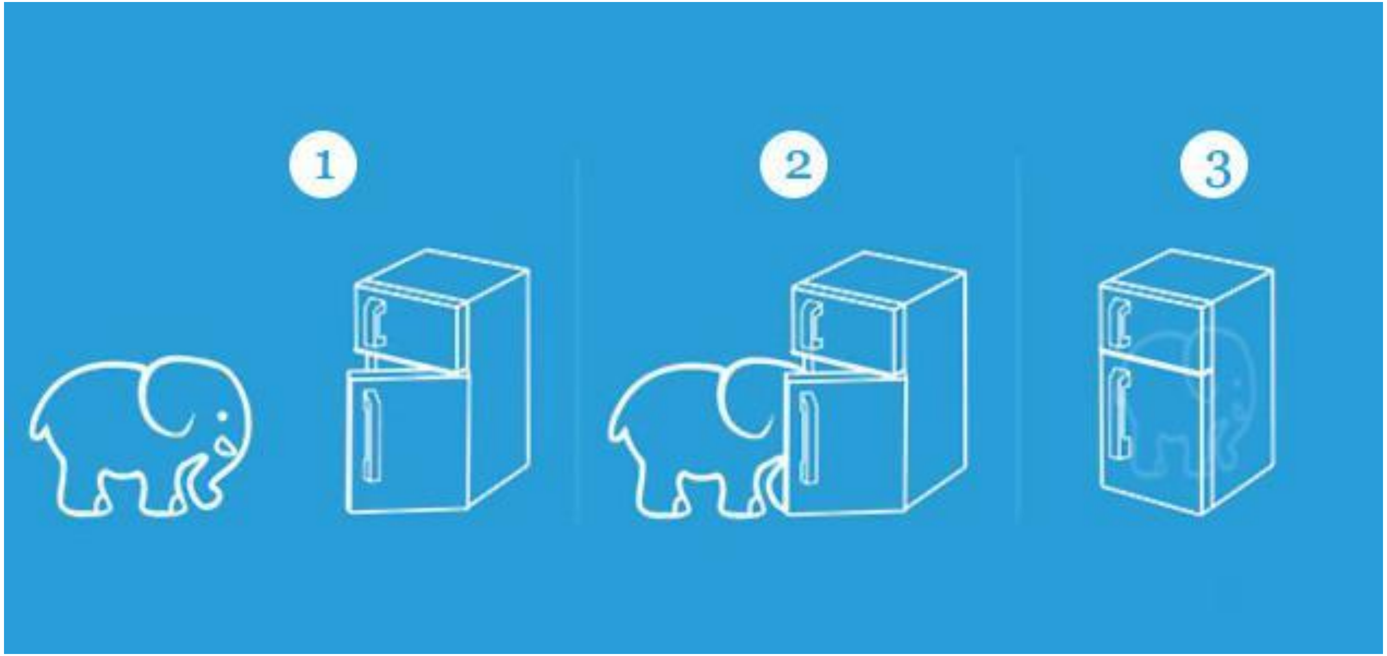
- Procedural Programming
  - Incremental Goals
  - Top-Down & Bottom-up

Lab5 Preview

- Object Oriented Programming
  - Class and Objects
  - Instance vs Static
  - Access modifiers
  - Constructor
  - Methods
  - Patterns

Post your question to https://padlet.com/angelahuo/appsprog

From Procedure to Object Oriented



1. Input

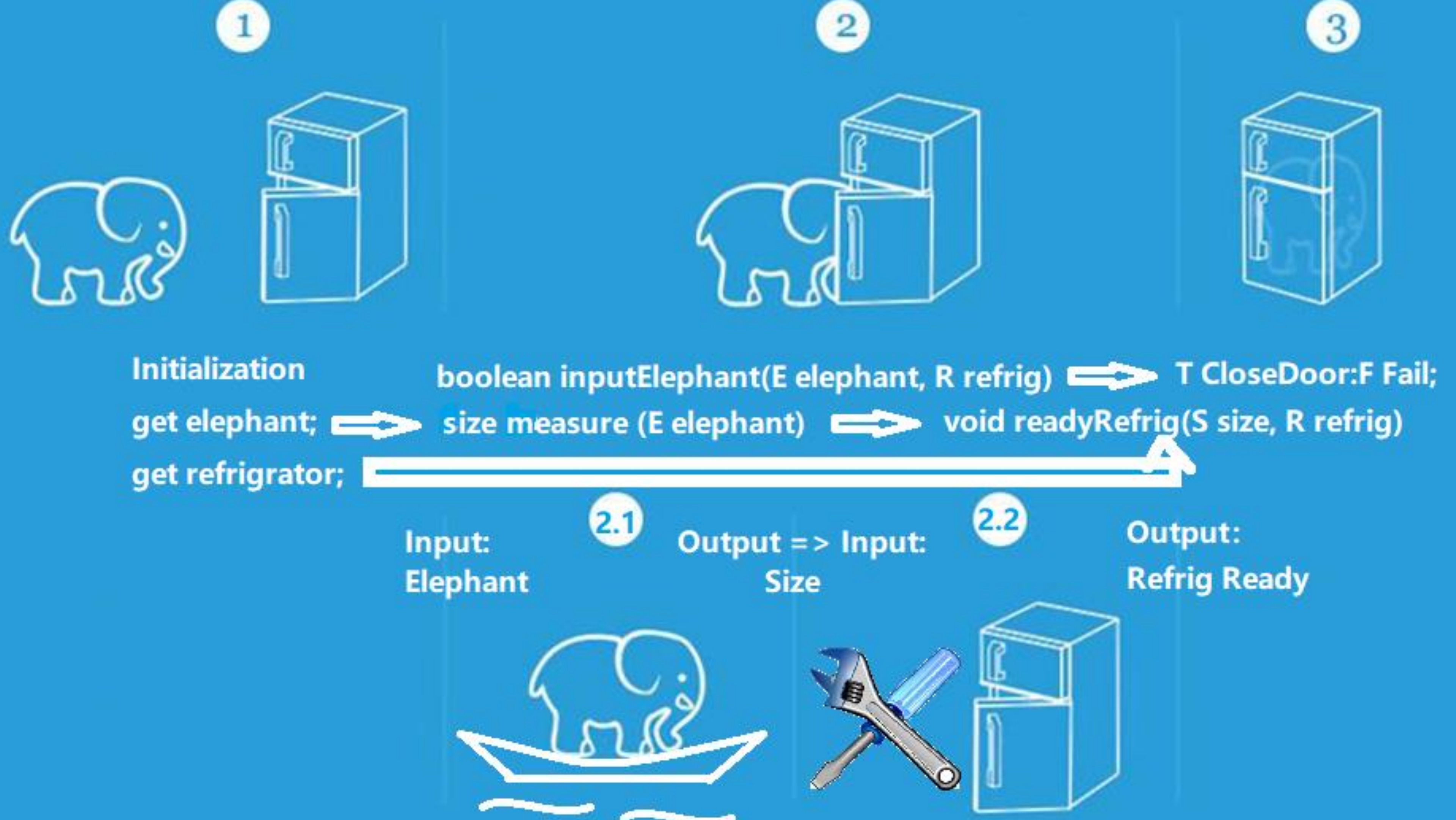
2. Process

3. Output

1. Elephant

2. Refrig

Post your question to https://padlet.com/angelahuo/appsprog



1

2

3

Initialization

get elephant; ⟹ size measure (E elephant) ⟹ void readyRefrig(S size, R refrig)

get refrigrator;

boolean inputElephant(E elephant, R refrig) ⟹ T CloseDoor:F Fail;

2.1

Input:
Elephant

Output => Input:
Size

2.2

Output:
Refrig Ready

Design 1

Elephant
Elephant(){}

int size
String zoo

int measure()

Refrigerator

int size
String brand

Refrigerator(size)
void open()
Bool putEle(Obj)
void close(suc)

Analysis 2

void open()

Int measure()

void buildRefige(size)

Bool putEle(Ele)

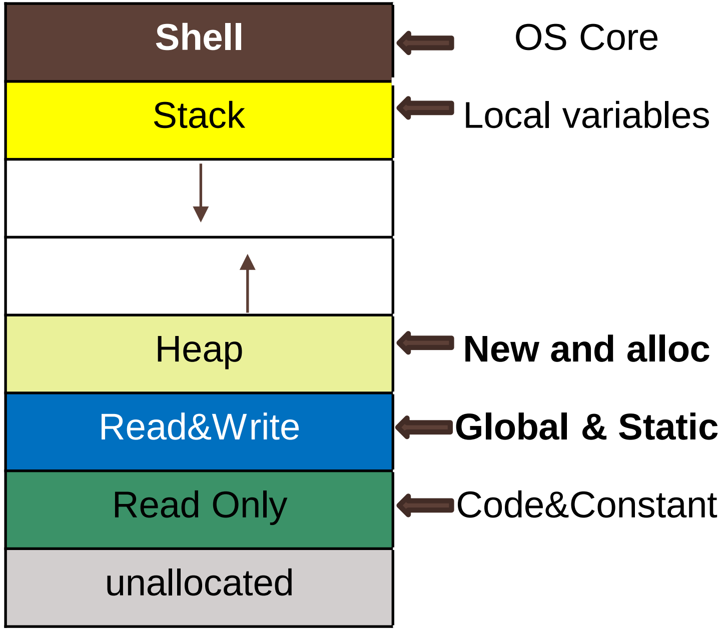
void close(suc)

Utilize 3

Question: Where to put the main method?

```
Public static void main (String[ ] args){
  Elephant dumbo=new Elephant();
  Refrigerator refrige=new
Refrigerator(dumbo.mesure());
   refrige. open();
   Bool suc=refrige. putEle(dumbo);
   refrige. close(suc);
}
```

Instance vs Static



Shell ← OS Core

Stack ← Local variables

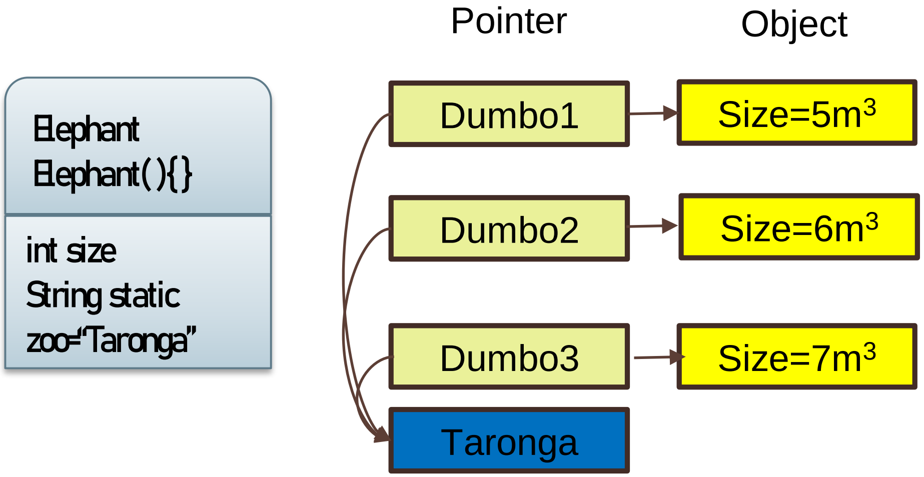
Heap ← **New and alloc**

Read&Write ← **Global & Static**

Read Only ← Code&Constant

unallocated

- Static Class, only 1 copy
- Instance Objects, multiply copies
- Final: can't change after initialized



Pointer

Object

Elephant
Elephant(){}

int size
String static zoo=“Taronga”

Dumbo1 → Size=5m^3

Dumbo2 → Size=6m^3

Dumbo3 → Size=7m^3

Taronga

Post your question to https://padlet.com/angelahuo/appsprog

Access Control

- Public
- Protected: Self, subclass, package
- Private: inner
- Default: Self, Package

Post your question to https://padlet.com/angelahuo/appsprog

Constructor

- Same name after class
- No return type
- No “void”
- Run with “new”
- Can be overloaded

- Python: field and constructor

- Field initialize
    - Literals
    - Read pattern
    - Parameters

- Python:

```
class Account:
    def __init__(self, accountType):
        self.accountType = accountType
        self.balance = self.readBalance()
```

Other Methods

- Overloading—Differentiate by parameters
- @Override—**toString()**

- This—current object (Python: self)
- Super—parent object

- Setter
- Getter

- Main method: creates and uses the first object

Design Rules

- Fields are always private!!
- Methods are private only if no other classes need to use them
- If a method uses a field, it is defined in the same class
- Avoid using getters and setters for single field
- Main method is the only static method
- Define it before use it   vs. Instantiate it then use it

Lab 5

- From this week, the scale of projects that students work on gets bigger, so we do analysis together as a class and get them onto the coding sooner.

- For the Goals Matrix
  - list the goals down the left first,
  - then distribute the methods across the classes.

  ----- This chart basically captures the content of a class diagram in a more concise way, and allows you to see how one goal can distribute across several classes more easily.

Pattern book

- 3 constructor patterns
- Formatted : Format to 2 decimal places
- Menu pattern

- Unfamiliar ones
    - initialising a field with a new object, toString.

Coding--skeleton code

- Code the classes and fields.

  They must be private otherwise your exercise will be marked “0”.

- Code the constructors.

  Decide whether a field is initialised from a default value, a read pattern or a parameter.

- Code the menu.

  Clarify the methods for each goal.

- Code the methods towards each goal.

  Code the complete goal so that it cuts across the classes. If you do this, you will be able to receive marks right from the start as you knock off each goal.

Post your question to https://padlet.com/angelahuo/appsprog

Coding-Menu

- Please do the menu goal first.

  Of course each menu action points to an empty procedure at this stage so each action does nothing. But run it and show them that the menu works, even though the actions don't work yet. Then knock off each action.

- After you submit the constructors, you should look at what ED is trying to test next, and that is how you should decide which goal you should implement first.
- Just implement things in the order that ED tests them. So, menu first!

Coding-Skeleton

- Two Rules:

  (1) one method per goal,

  (2) good names per method.

This makes code more reusable and more readable. For example, while writing the client code for withdraw:

```
if (account.has(amount))
        account.withdraw(amount);
```

The has() function is both reusable and reads almost like English.

Timing

- 30 min Intro + Demo
- 10 min analysis
- Remaining time -- Coding

See you next week!

Post your question to https://padlet.com/angelahuo/appsprog

Contact

- Subject Coordinator and Lecturer: Angela Huo
- Email: huan.huo@uts.edu.au
- Contact information on Canvas