Lecture 0
Course Overview

TIC1001 Introduction to Computing and Programming I

13 Aug 2020

Welcome to NUS

WHY

WHAT

HOW

But introductions first…

Wai Kay

Wai Kay

Timeline About Friends 10 Mutual

About

Overview

Work and Education

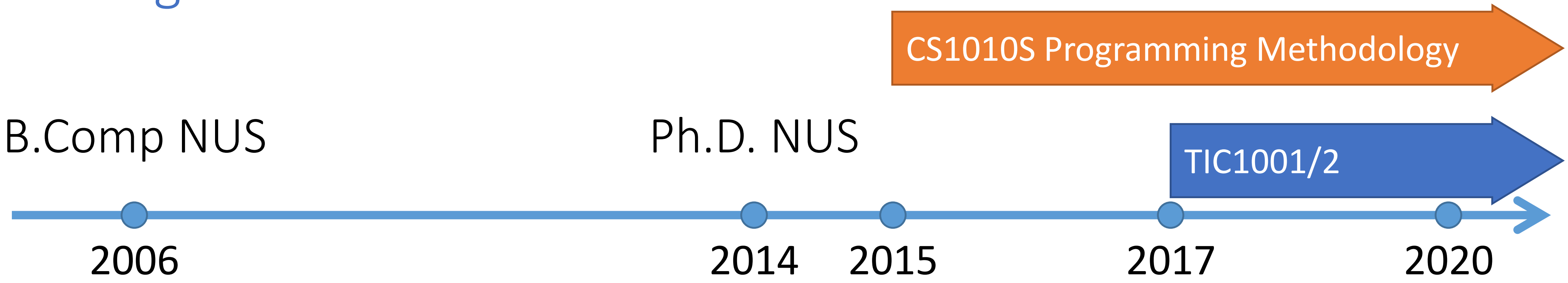
Works at National University of Singapore

waikay@nus.edu.sg

April 2

Dr Leong Wai Kay

Background



CS1010S Programming Methodology

TIC1001/2

B.Comp NUS

Ph.D. NUS

2006

2014

2015

2017

2020

Contact

– Office: COM2 02-11

– Email: waikay@comp.nus.edu.sg

Why take TIC1001?

To obtain a foundation in programming methodology

- because BTech Computing

Unfortunately,

- it is easier said than done

Teaching Assistants/Tutors

Siddharth Bhatia

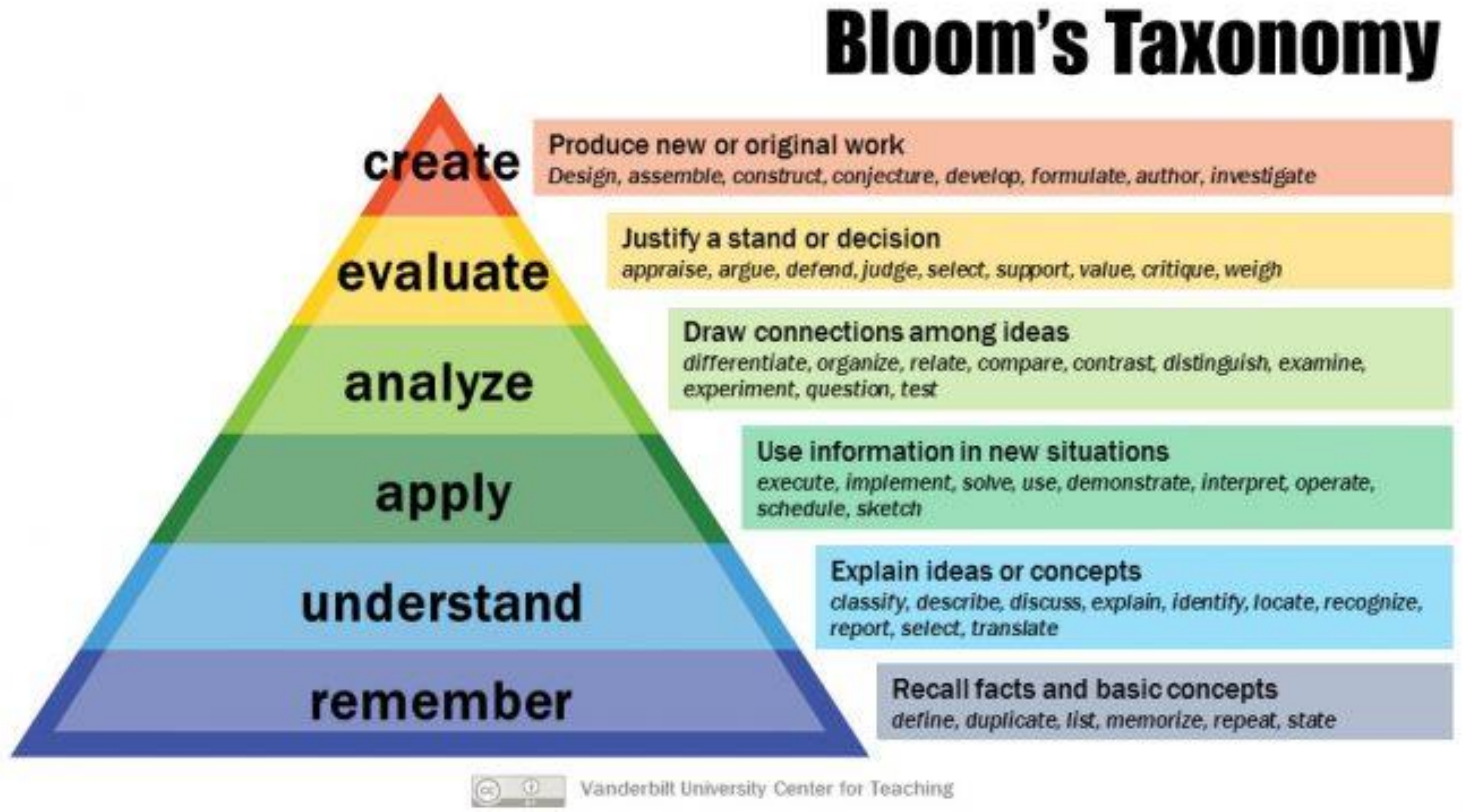
Saif Uddin Mahmud

Teo Wei Jie, Shaun

Michael Dinata

Ang Cheng Jun

Why is programming hard?



Bloom's Taxonomy

create — Produce new or original work
Design, assemble, construct, conjecture, develop, formulate, author, investigate

evaluate — Justify a stand or decision
appraise, argue, defend, judge, select, support, value, critique, weigh

analyze — Draw connections among ideas
differentiate, organize, relate, compare, contrast, distinguish, examine, experiment, question, test

apply — Use information in new situations
execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch

understand — Explain ideas or concepts
classify, describe, discuss, explain, identify, locate, recognize, report, select, translate

remember — Recall facts and basic concepts
define, duplicate, list, memorize, repeat, state

Vanderbilt University Center for Teaching

Why is programming hard?

Computers only do what you say, not what you mean.

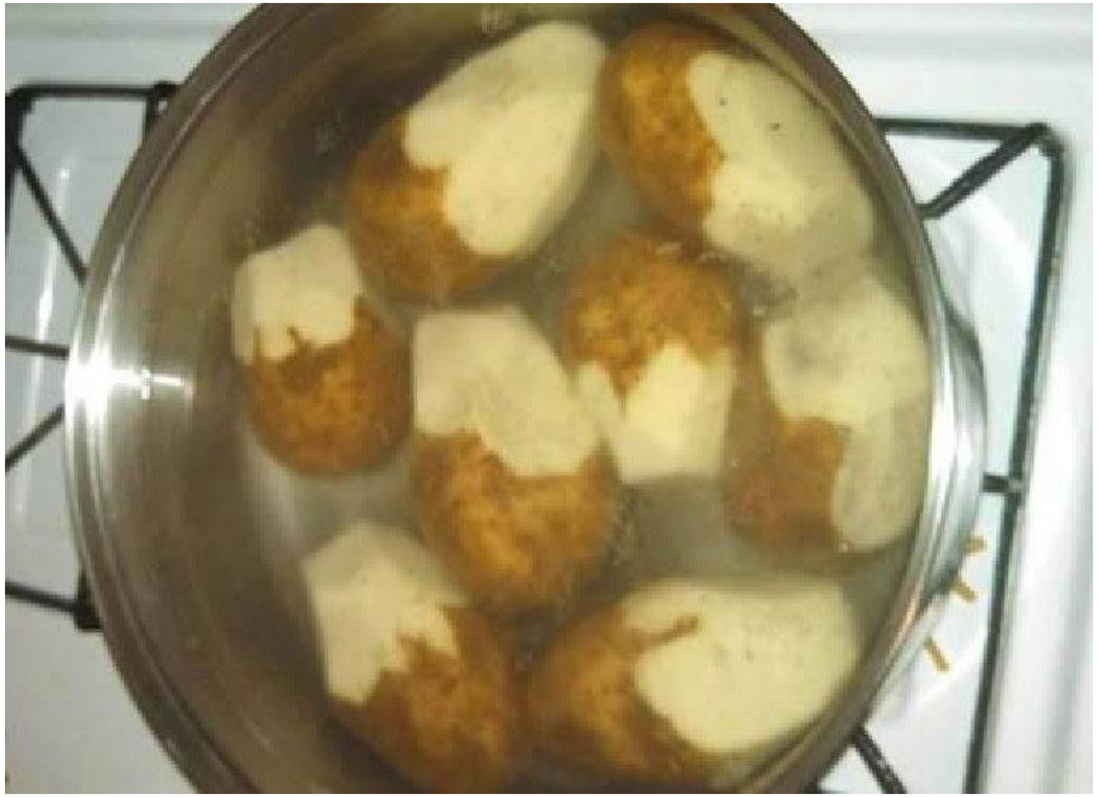
Natural language is not precise

- Not trivial to code-switch

Why is programming hard?

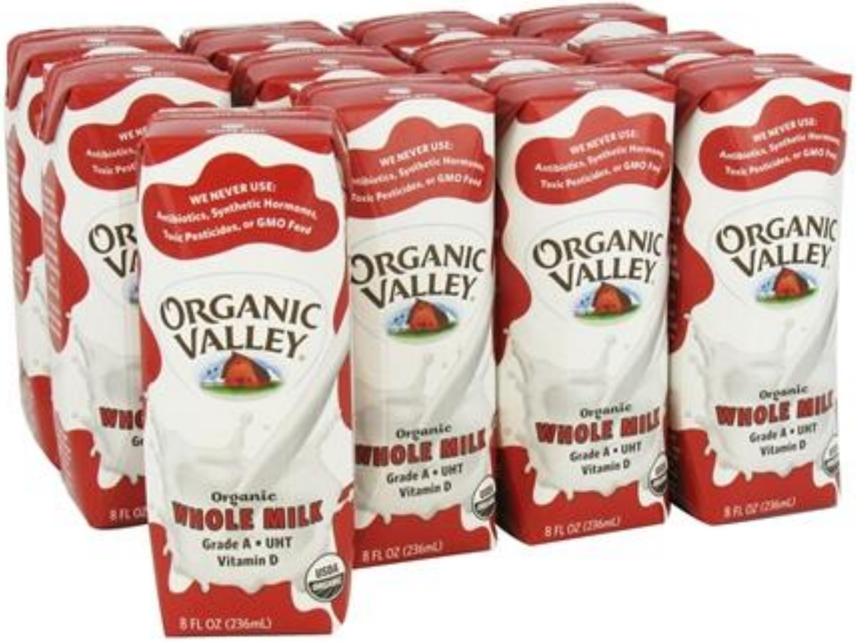
A man sees his wife busy in the kitchen and offers to help

She says, “take this bag of potatoes, peel half of them, and put them in a pot to boil”



Why is programming hard?

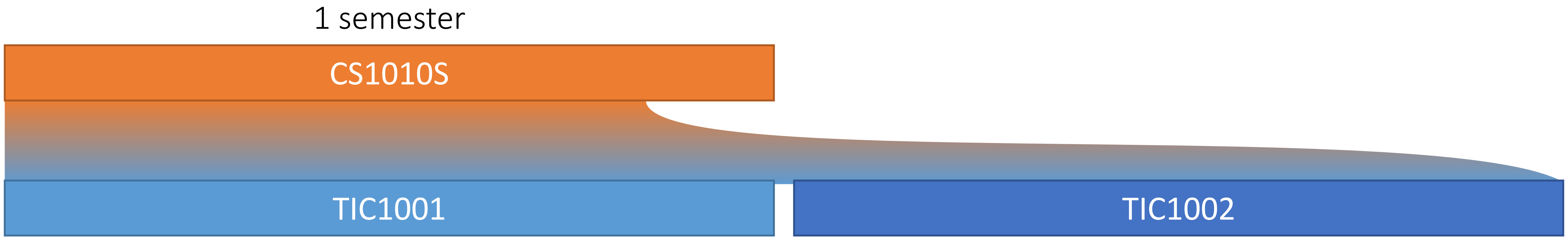
My wife said, “Please go to the store and buy a carton of milk.
If they have eggs, get 12.”



They had eggs

So how?

Split across two semesters



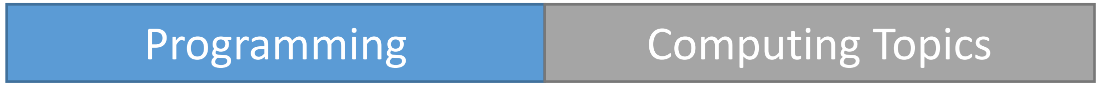
1 semester

CS1010S

TIC1001

TIC1002

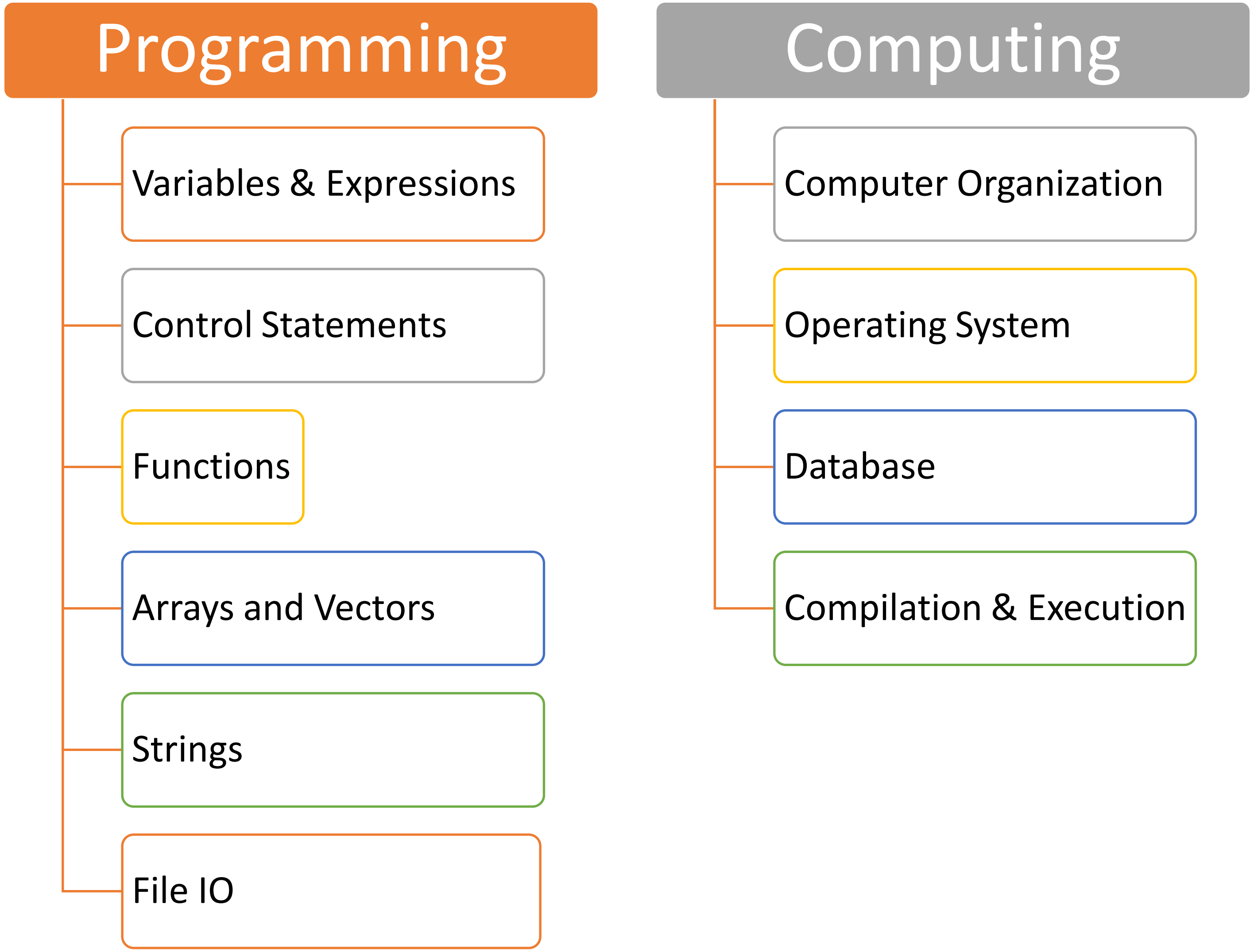
Half content in first semester



Programming

Computing Topics

The Two Threads



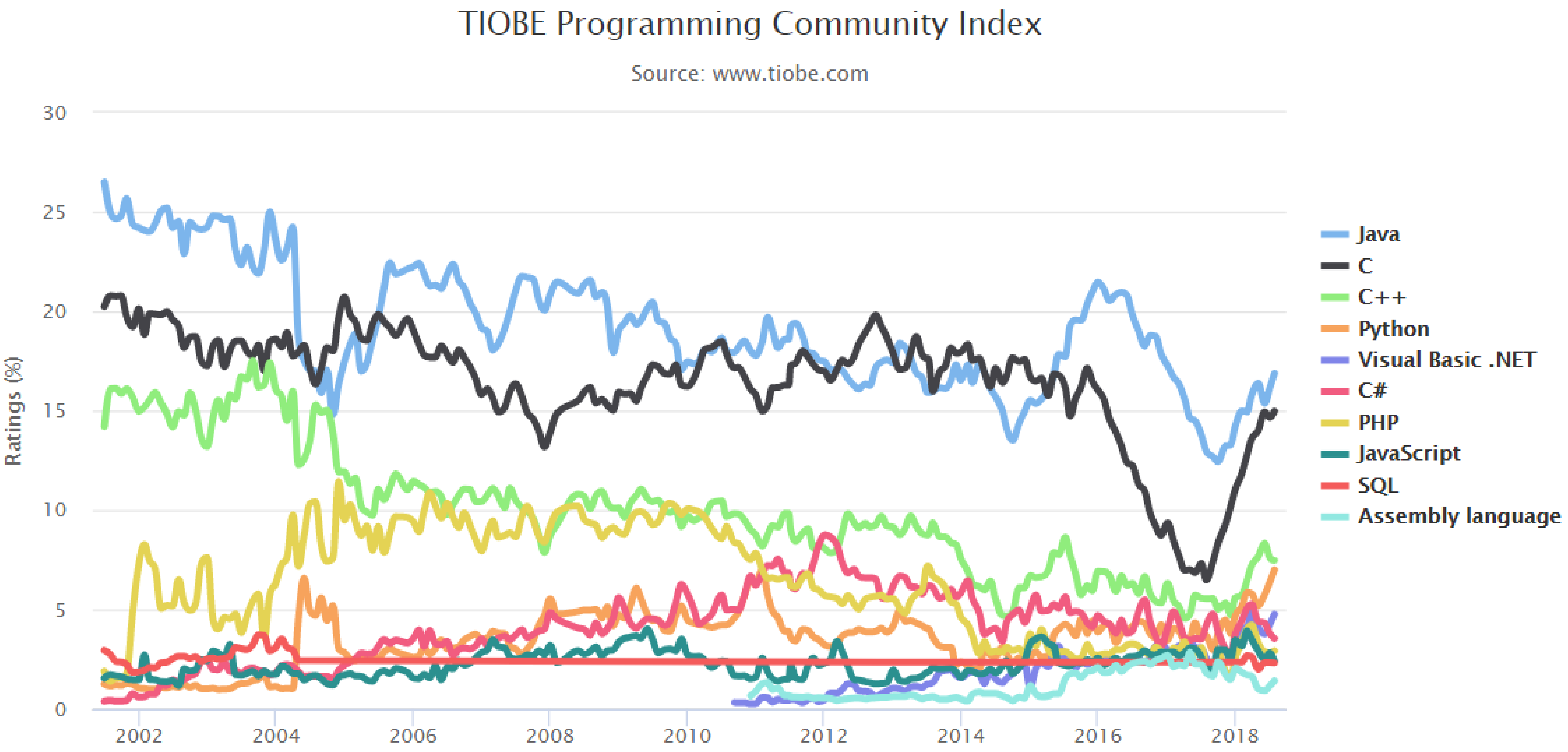
Programming

- Variables & Expressions
- Control Statements
- Functions
- Arrays and Vectors
- Strings
- File IO

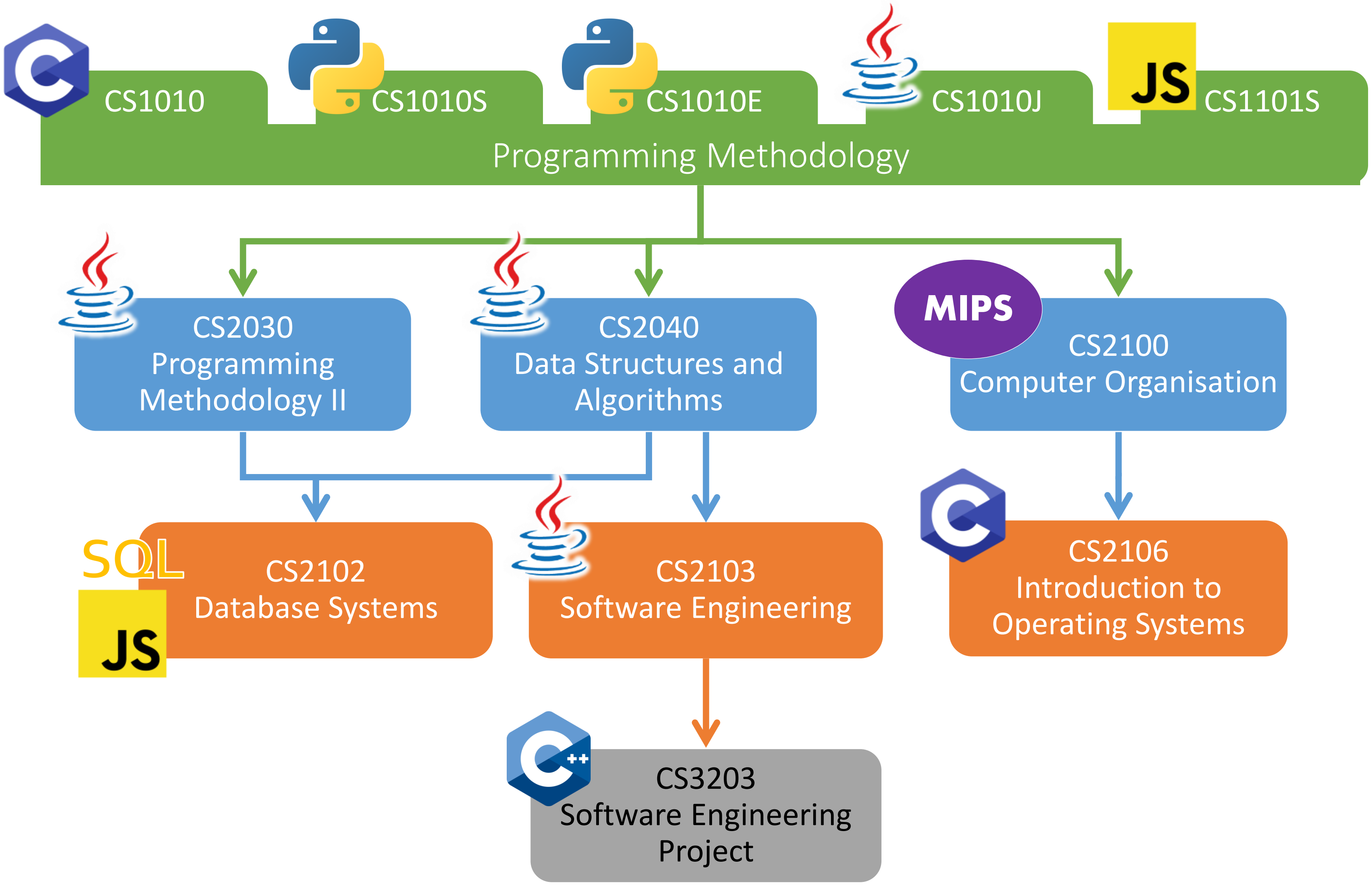
Computing

- Computer Organization
- Operating System
- Database
- Compilation & Execution

Why teach C/C++


TIOBE Programming Community Index
Source: www.tiobe.com
Ratings (%)
30
25
20
15
10
5
0
2002
2004
2006
2008
2010
2012
2014
2016
2018
Java
C
C++
Python
Visual Basic .NET
C#
PHP
JavaScript
SQL
Assembly language

Programming language used in modules



CS1010 CS1010S CS1010E CS1010J CS1101S

Programming Methodology

CS2030 Programming Methodology II

CS2040 Data Structures and Algorithms

CS2100 Computer Organisation

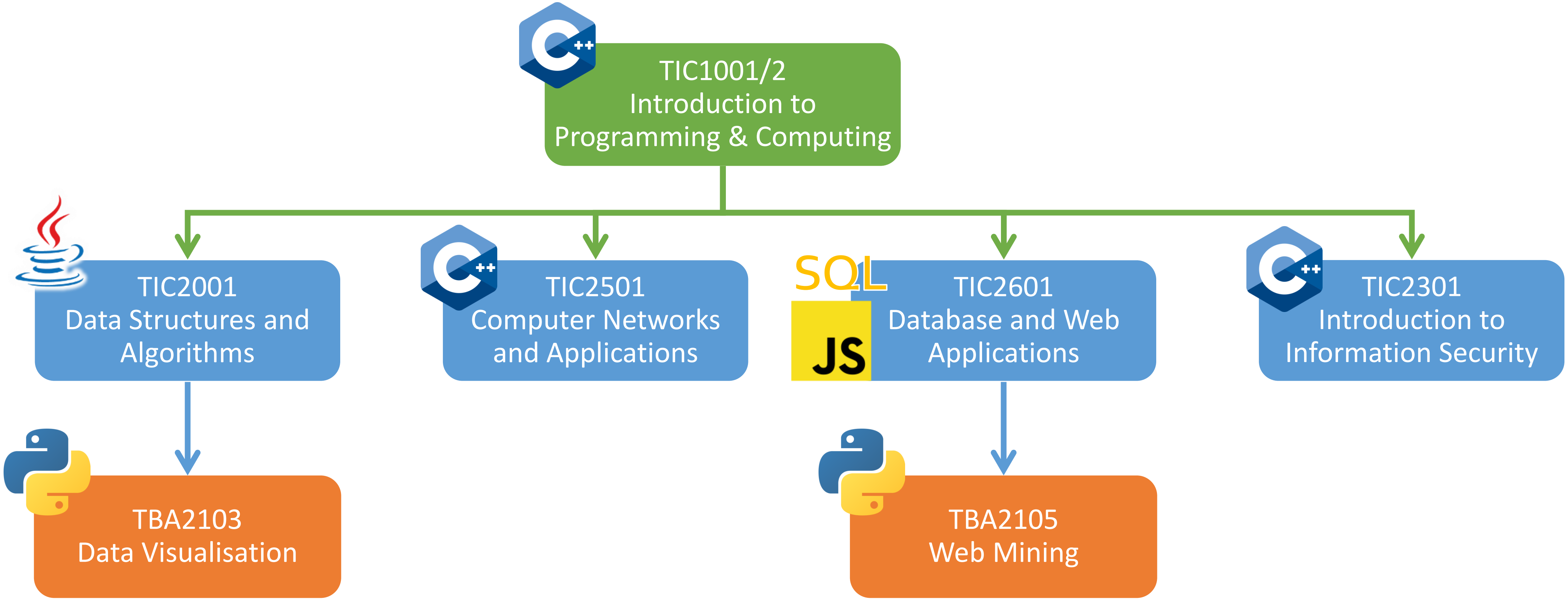
CS2102 Database Systems

CS2103 Software Engineering

CS2106 Introduction to Operating Systems

CS3203 Software Engineering Project

Programming language used in modules



TIC1001/2
Introduction to
Programming & Computing

TIC2001
Data Structures and
Algorithms

TIC2501
Computer Networks
and Applications

TIC2601
Database and Web
Applications

TIC2301
Introduction to
Information Security

TBA2103
Data Visualisation

TBA2105
Web Mining

Why teach C/C++ and not Python?

Analogy: Suppose you are learning to drive

- Which car to use?

Why teach C/C++ and not Python?

Analogy: Suppose you want to learn to cook

Why teach C/C++

Expose Low Level Internals

High Performance

Cultivates Good Habit

Why teach Computing Topics?

Kickstart

Context

Future Modules

Course Objectives

Expose students to **computing principles**

- Abstraction and Composition

Introduce key **computing concepts**:

- Computer organization
- Operating systems
- Data management

Introduce basic **programming methodologies** and **problem solving techniques**

- Using simple structured programming language

Specific Learning Outcomes

After taking this module, students should be able to:

- Appreciate the use of abstractions and composition to deal with complexities in computing systems
- Able to seek out and use existing tools to explore how computer systems and applications works
- Explain commonly used computing terms and the relations between them
- Understand how data and programs are represented and executed within a computer
- Develop simple programs using branches, loops, standard input/output, and functions

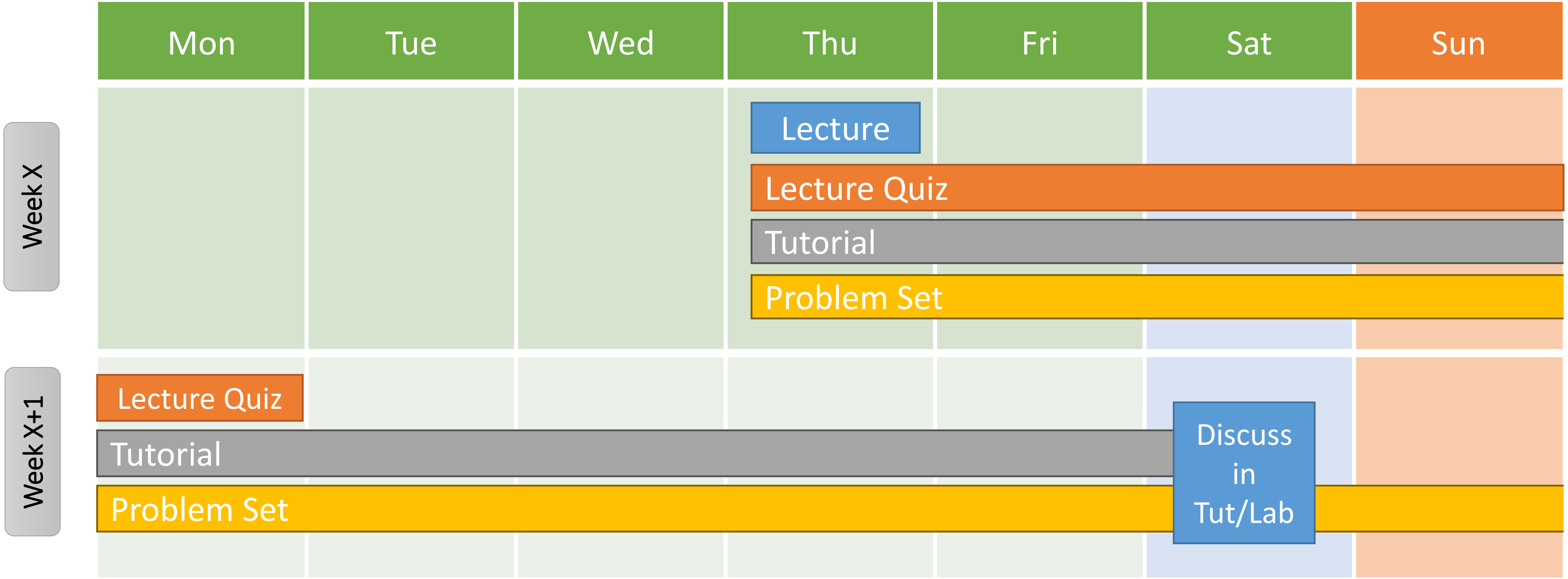
Calendar

Week	Mon	Tue	Wed	Thu	Fri	Sat	Sun
1				Lecture (P)		Technical Support	
2				Lecture (P)		Lab	
3				Lecture (C)		Tutorial 1 + Lab	
4				Lecture (P)		Tutorial 2 + Lab	
5				Lecture (C)		Tutorial3 + Lab	
6				Lecture (P)		Practical Exam 1	
R							

Calendar

Week	Mon	Tue	Wed	Thu	Fri	Sat	Sun
7				Midterm Test		Tutorial 4 + Lab	
8				Lecture (P)		Tutorial 5 + Lab	
9				Lecture (C)		Tutorial 6 + Lab	
10				Lecture (P)		Tutorial 7 + Lab	
11				Lecture (C)		Tutorial 8 + Lab	
12				Lecture (P)		Tutorial 9 + Lab	
13				Practical Exam 2			
R							
E							
E		Final Exam					

Typical Weekly Schedule



Mon

Tue

Wed

Thu

Fri

Sat

Sun

Week X

Lecture

Lecture Quiz

Tutorial

Problem Set

Week X+1

Lecture Quiz

Tutorial

Problem Set

Discuss in Tut/Lab

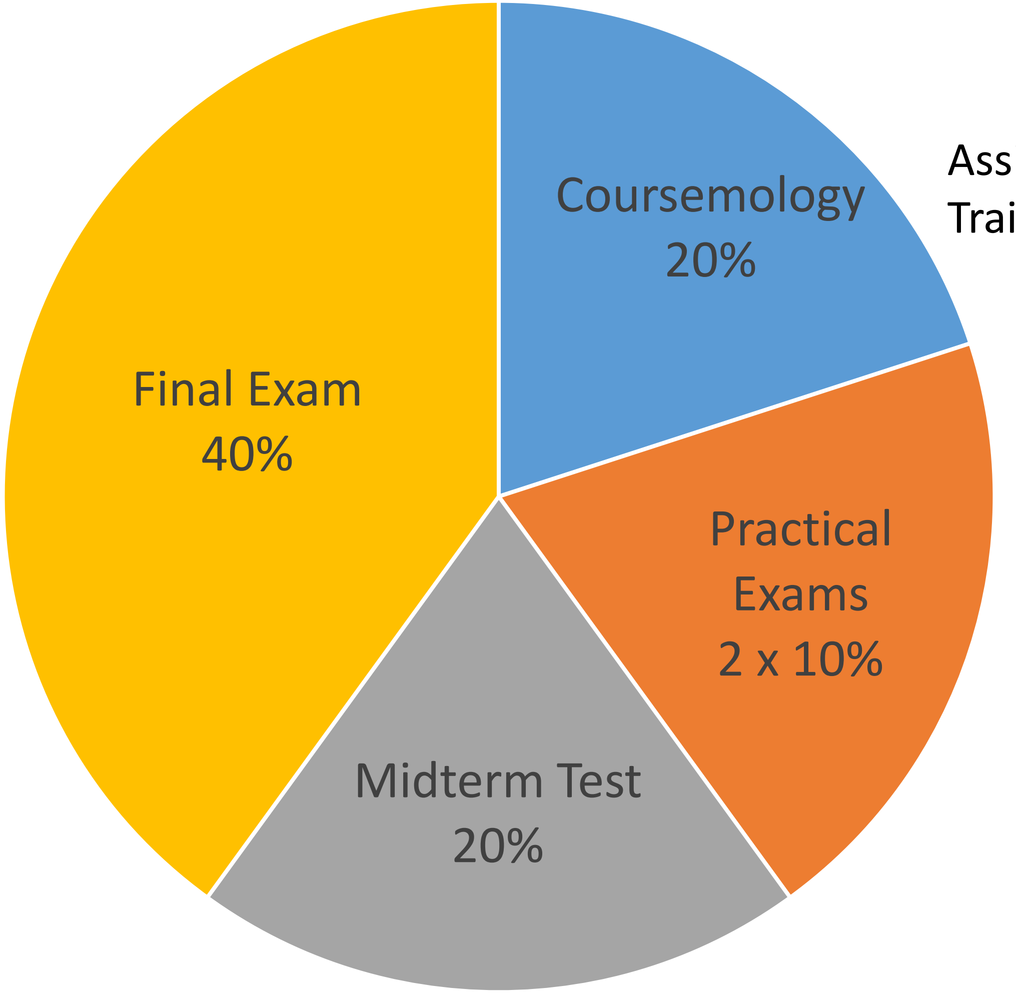
Tutorial/Lab Timing

Saturday Morning/Afternoon

Tutorial	Lab
1 hour	2 hours
9.30 – 10.30am	10.30 – 12.30pm
(2 – 3pm)	(3 – 5pm)

- Length of tutorial / lab is dynamic (i.e. adjustable depending on the topics)
- Total length is fixed though (at most 3 hours)

Assessment Weightage



Assignments, Tutorials, Trainings, etc.

Week 6 & **Week 13**
Saturday AM

Week 7 During Lecture

Tue 3rd Dec
Evening

Resources

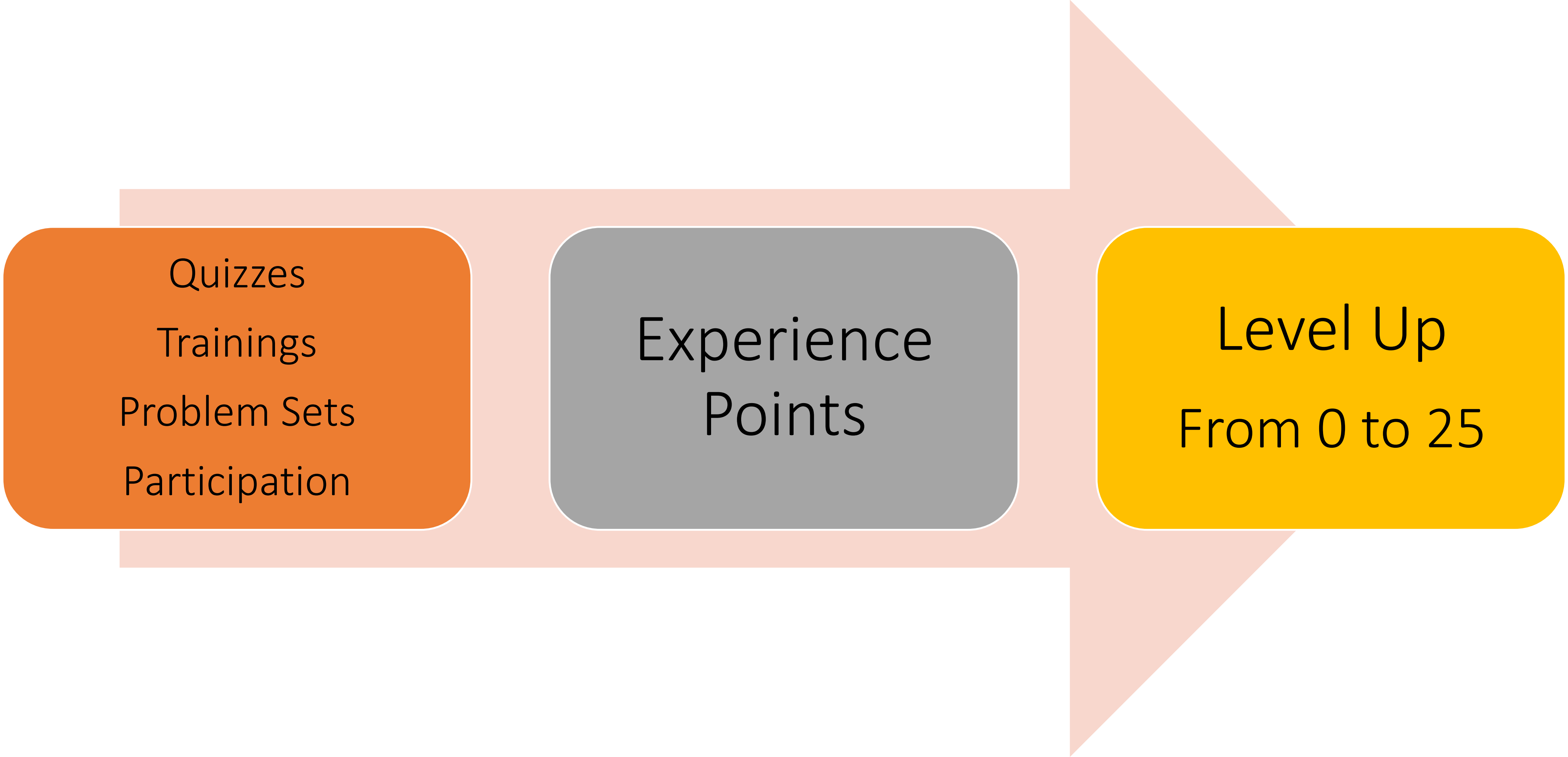
LumiNUS

- Gradebook
- Webcast (Panopto)

Coursemology

- Announcements
- Quizzes
- Trainings
- Problem Sets
- Forum participation

Coursemology



Quizzes

Trainings

Problem Sets

Participation

Experience Points

Level Up

From 0 to 25

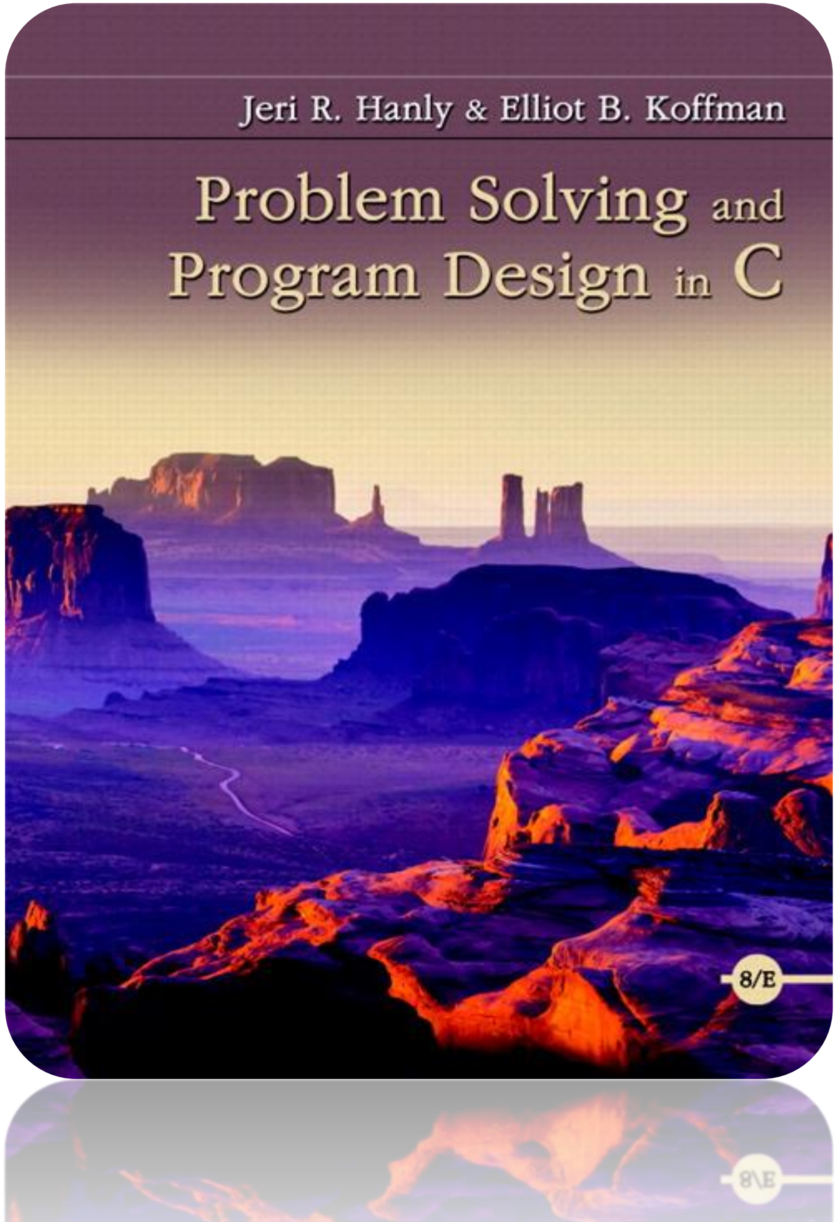
References

No textbook needed

- Save money

If you die die need a book

- Problem Solving and Program Design in C



Jeri R. Hanly & Elliot B. Koffman
Problem Solving and Program Design in C
8/E