

Welcome to CSSE1001/7030

Introduction to Software Engineering

Lecture 0

Dr. Paul Vrbik

February 20, 2023

Course Coordinator

Dr. Paul Vrbik

e-mail uqpvrbik@uq.edu.au

office 78-204C

hours Friday 12P–1P.

Coordinating TA

Ms. Ashleigh Richardson

e-mail a.richardson@uq.edu.au

About your instructor (Education)

PhD *Computer Science*. University of Western Ontario.

Computational Algebraic Geometry

MSc *Pure Mathematics*. Simon Fraser University.

Computer Algebra

BSc *Pure Mathematics*. McMaster University.

Algebra + Computer Science

About your instructor (Previous Teaching)

2020– *Lecturer (Teaching Stream)* of Computer Science at the
University of Queensland

2018–2020 *Assistant Professor (Teaching Stream)* of Mathematics
and Computer Science at the *University of Toronto, Mississauga*

2014–2016 *Associate Lecturer* of Mathematics at the *University of
Newcastle, Australia*

About the course

- ▶ This is an *introductory* course for people who have never programmed in the *imperative* and *object oriented* paradigm.
- ▶ *1,226* students, *20* teaching assistants, *1* instructor, and *1* coordinator.
- ▶ *Weekly* tutorials and practicals. All optional. *Starts week 2*.
- ▶ *11 minor formative assessments* called *ShiFoo*. All optional.
- ▶ *3 major assessments* submitted to *GradeScope*.
- ▶ *1 final examination* (multiple choice, fill in the blank, and one full solution).

Marking Scheme

Assignment 1	15%
Assignment 2	15%
Assignment 3	15%
Final Exam ^a	55%

^a> 45% required to pass course.

Grade

$M \geq 85\%$ and $E \geq 80\%$ and $A3 \geq 80\%$	7
$M \geq 75\%$ and $E \geq 70\%$	6
$M \geq 65\%$ and $E \geq 60\%$	5
$M \geq 50\%$ and $E \geq 45\%$	4
$M \geq 45\%$ and $E \geq 40\%$	3
$M \geq 20\%$	2
otherwise	1

Final grades and marks may be *scaled upwards*, but
not downwards.

What a grade means

Demonstrated evidence of ...

Grade 3 *Marginal Fail* developing achievement of course learning outcomes.

Grade 4 *Pass* functional achievement of course learning outcomes:

Grade 5 *Credit* proficient achievement of course learning outcomes.

Grade 6 *Distinction* advanced achievement of course learning outcomes.

Grade 7 *High Distinction* mastery of course learning outcomes.

Previous Results

	7	6	5	4	3	≥ 3
Last Term	12%	16%	23%	30%	4%	85%

23-S1	Week Starting	Hour 1	Hour 2	Tutorial	Tutorial Coverage	ShiFoo	Assignments	Date
1	Feb 20	- Intro to SWE - Variables, Types, and Arithmetic						
2	Feb 27	- Functions (Sequence) - If Statements (Selection)	- Expressions and Control Structures	Print, Input, Arithmetic, If-Statements, Casting, Types, Randomness	- Intro Programming	A1 (Assigned)	03/03/2023	
3	Mar 6	- While Loops (Iteration) - Strings - For Loops (Iteration)	- Functional Decomposition	Functions, Turtle,	- Control Structures			
4	Mar 13	- Tuples (0) and Lists (37) - String Methods (10) - Dictionaries (26)	- Data Structures	Strings, Lists, String Methods,	- Functions			
5	Mar 20	- File IO (40) - Testing (55)	- Debugging and Dictionaries	Tuples, Dictionaries	- List, Strings, Tuples	A1 (Due)	24/03/2023	
6	Mar 27	- Scope (30) - Introduction to OO (60)	- File IO and processing	Reading from files	- Dict, Lists and Strings	A2 (Assigned)	31/03/2023	
7	Apr 3	- Exceptions (30) - Composition, Inheritance, and Polymorphism (65)	- Class Design	Classes, Methods, Lists	- Classes, Objects, and Exception Handling			
Break	Apr 10							
8	Apr 17	- GUI: Basic Widgets - MVC: Connect Four	- Inheritance	Class Inheritance,	- Inheritance			
9	Apr 24	CANCELLED ANZAC DAY	CANCELLED ANZAC			A2 (Due)	28/04/2023	
10	May 1	- GUI: Animation, Text Widgets, Canvas Widgets - Calculator Example	- GUI 1	tk.Label, tk.Entry,	- GUI programming 1	A3 (Assigned)	05/05/2023	
11	May 8	- GUI: File Menus, File Dialog, and Drawing - GUIs: Model View Controller	- GUI 2	tk.Canvas, tk.Label, tk.Button,	- GUI programming 2			
12	May 15	- Recursion - Recursion	- Recursion	Recursion, import os	- Recursion			
13	May 22	- FP: Lambda, List comp, Iterators - FP Operators	- Functional Programming	Lambda and list comprehension	- Functional Programming	A3 (Due)	26/05/2023	
14	May 29	Rev Week						

Course Resources

What's all this then, Amen!

— Monty Python

Introduction (Section 1 of 17)

What is Software Engineering?

Software Engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software (*IEEE std 610.12-1990*). A software engineer must have a good understanding of tools and techniques for requirements gathering, specification, design, implementation, testing and maintenance.

These days, software systems are often very large and many contain key safety or mission-critical components. The complexity of modern software systems require the application of good software engineering principles. Indeed, many agencies mandate the use of particular tools and techniques to achieve very high quality and reliable software.

Course Resources (BlackBoard)

The screenshot shows the BlackBoard course resources interface for the course [CSSE1001/7030] Introduction to Software Engineering (St Lucia & external). Semester 2, 2021. The main navigation bar includes links for Request Support, Paul Vrbik, Welcome, Courses (highlighted), Organisations, Staff, Help, and Guides. The left sidebar contains a navigation tree with sections like Announcements, Course Profile (ECP), Course Staff, Help Procedures, Weekly Study Plan, Assessment, My Grades, Library Links, Learning Resources, Final Exam, Ed Discussion Board, Tutorials, Practicals, and Programming Hints. The main content area is titled "Weekly Study Plan". It features a "Jump To:" dropdown set to "Week 1". The "Orientation" section lists "Need to Do" tasks: Read the Course Profile (ECP), Read the Help Procedures, and Familiarise Yourself with Blackboard. The "Week 1" section lists "Need to Know" topics: Topic 1A - Introduction to Software Engineering and Programming, Topic 1B - Variables and Types, and an "Assessment" task: MyPyTutor - Introductory Programming. The "Week 2" section lists "Need to Know" topics: Topic 2A - Control Structures and Topic 2B - Functional Decomposition, and an "Assessment" task: Tutorial 1 - Expressions and Control Structures. The footer shows dates for each week: 19th Jul - 25th Jul for Week 1, 26th Jul - 1st Aug for Week 2, and 2nd Aug - 8th Aug for Week 3.

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Welcome Courses Organisations Staff Help Guides

[CSSE1001/7030] Introduction to Software Engineering (St Lucia & external). Semester 2, 2021 CSSE1001S_7160_61545

Weekly Study Plan

Edit Mode is: ON

Weekly Study Plan

Build Content Assessments Tools

Jump To: Week 1

Settings ?

Orientation

Need to Do

- Read the Course Profile (ECP)
- Read the Help Procedures
- Familiarise Yourself with Blackboard

Week 1

Need to Know

- Topic 1A - Introduction to Software Engineering and Programming
- Topic 1B - Variables and Types

Assessment

MyPyTutor - Introductory Programming

Week 2

Need to Know

- Topic 2A - Control Structures
- Topic 2B - Functional Decomposition

Need to Do

Tutorial 1 - Expressions and Control Structures

Assessment

MyPyTutor - Introductory Programming

Assignment 0

19th Jul - 25th Jul

26th Jul - 1st Aug

2nd Aug - 8th Aug

Course Resources (EdStem Discussion Board)

ed CSSE1001/7030 – Discussion

New Thread

COURSES +

COURSES	
CSSE1001/7030	MyPyTutor
SCIE1000	Assessments - MyPyTutor

CATEGORIES

- General
- Lectures
- Tutorials
- Assessments
- Administration
- Social

Filter ▾

This Week

MyPyTutor	✓
Assessments - MyPyTutor	Anonymous 21h

Wk 1

MyPyTutor	✓
Assessments - MyPyTutor	Anonymous 1d

Idle help

General	Anonymous 1d
---------	--------------

Do we have the recording for all the tutorials

Tutorials	Yulong Liu 2d
-----------	---------------

Zoom link for today's tutorial at 6:00pm?

Labs	Anonymous 3d
------	--------------

Delayed and clashable

Lectures	Paul Wade 3d
----------	--------------

Last Week

Welcome!	General Paul Vrbik STAFF 10d
----------	------------------------------

Search

Filter ▾

Welcome! #1

 Paul Vrbik STAFF 10 days ago in General

PIN STAR WATCHING VIEWS 441

Hi everyone,

20 We're using Ed Discussion for class Q&A.

This is the best place to ask questions about the course, whether curricular or administrative. You will get faster answers here from staff and peers than through email.

Here are some tips:

- Search before you post
- Heart questions and answers you find useful
- Answer questions you feel confident answering
- Share interesting course related content with staff and peers

For more information on Ed Discussion, you can refer to the [Quick Start Guide](#).

All the best this semester!

Paul Vrbik

Comment Edit Delete ...

Add comment

Individual tech help csse1001@helpdesk.eait.uq.edu.au

Course Resources (ShiFoo)

The screenshot shows the ShiFoo course resources interface. On the left, there's a sidebar with navigation links: Summary, Repositories, Results, and Extensions. Under Extensions, several categories are listed with their completion status and due dates:

- intro: 0/1
- input: 0/1
- input_number: 0/1
- Control Structures: 0 / 6 (Due Date: 12/8/2022 16:00)
- Functions: 0 / 3 (Due Date: 19/8/2022 16:00)
- List, Strings, Tuples: 0 / 9 (Due Date: 26/8/2022 16:00)
- Dict Lists and String: 0 / 7 (Due Date: 2/9/2022 16:00)
- Classes, Objects and Exception: 0 / 5 (Due Date: 9/9/2022 16:00)

The main content area displays a task titled "[CSSE1001/7030] Introduction to Software Engineering (St Lucia & external). Semester 2, 2022". A code editor window titled "attempt.py" contains the placeholder code: `1. ### WRITE YOUR CODE HERE ###`. To the right of the code editor is a "Description" panel with the following sections:

- Introduction: Hello, World!**

Each problem we assign you in the ShiFoo system will have a 'Background', 'Task', 'Precondition', and 'Example' section.

This problem is meant to introduce you to the ShiFoo system and only requires you write a single line of code.
- Background**

We usually discuss the function or topic being introduced in this section.

To output strings to the screen use the `print` command.

```
>>> print("Hello, World!")
Hello, World!
```

Note `>>>` is just notation for distinguishing input from output, as if we are working in the REPL. It is useful for illustrating usage but do not include `>>>` in your working.

To complete this task simply type `print("Hello, World")` in the adjacent

At the bottom of the main content area, there's a "Results" section with the message: "Test output will appear here when you run the code".

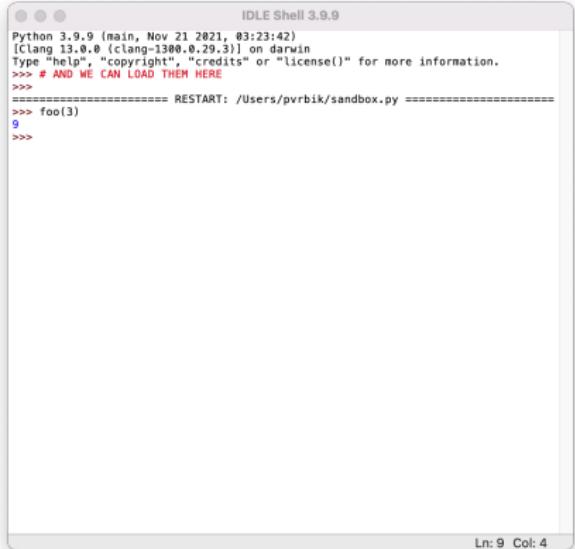
Course Resources (Idle)



A screenshot of a Mac OS X application window titled "sandbox.py - /Users/pvrbik/sandbox.py (3.9.9)". The window contains the following Python code:

```
# We can write definitions here
def foo(x: int) -> int:
    return x**2
```

The status bar at the bottom right shows "Ln: 4 Col: 0".



A screenshot of a Mac OS X application window titled "IDLE Shell 3.9.9". The window contains the following text from a Python shell:

```
Python 3.9.9 (main, Nov 21 2021, 03:23:42)
[Clang 13.0.0 (clang-1300.0.29.5) on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> # AND WE CAN LOAD THEM HERE
>>>
=====
===== RESTART: /Users/pvrbik/sandbox.py =====
>>> foo(3)
9
>>>
```

The status bar at the bottom right shows "Ln: 9 Col: 4".

This is the *Integrated Development Environment* (IDE) that is *officially* supported by our TAs.

Course Resources (Open Help Center)



Open Help Center (a.k.a. "ITLC" a.k.a. "Code Clinic") in 78-217.

Hours Monday to Friday from 10A-5P.

Late Policy

Late work will receive a 10% penalty for seven days unless a medical certificate is provided.

Longest possible extension is *seven days* with *no exceptions*.

We must *receive* your work before the deadline. This is *not* the same as when you *transmit* your work.

There is an (undisclosed) buffer. That is to say, we do not count work as late if it is *merely a few minutes late*.

Academic Integrity

Don't cheat.

There are *harsh* penalties.

Always *cite code* that you take from someone else or from the internet in a comment. This insulates you from academic misconduct but *not* from getting a zero.

We have pretty *sophisticated tools for detecting cheating*.

You *are encouraged* to talk about solutions to problems. However: *do not share code*.

Secondary V Post-Secondary: Setting Expectations

Secondary	Post Secondary
20 weeks. 5 hours pw.	12 weeks. 2 hours pw.
<i>100 hours</i> instruction.	<i>24 hours</i> instruction.
30 classmates	1000 classmates
<i>Teachers:</i> teach you	<i>Instructors:</i> guide you

Secondary V Post-Secondary: Setting Expectations

Lectures

We cannot teach someone to program in 24 hours. The purpose of a *lecture* is to give you *first exposure* to a topic. We expect you to do most of your learning by *completing assigned tasks*.

You should *not* have an expectation that you will understand everything “taught” to you in during lecture. You need to revise after (and indeed you are *supposed* to revise *before*) lectures to absorb the material.

Those who *engage* with the material and *practice* will pass this course.

Setting Expectations

Humanity

Your lecturer and TAs *are humans* with a myriad of other responsibilities that go far beyond this course.

Please understand that the course staff will sometimes make errors, write terse responses, and are usually sleep deprived.

There is an *incredible* amount of work that happens behind the scenes. I am trying my *absolute hardest* to provide you the best educational experience. The *TAs are giving it there all* as well.

Questions?