Lesson 1: Week 1 - Overview

Created: 2024-05-27T01:25:36.374378+10:00

Week 1 - Introduction

Week 1 Tuesday To Do List

Week 1 Tuesday To Do ListAdmin/TipsCheck Lectures and Tutorials Timetable since some locations have changedPython 3 Cheat Sheet - allowed in final exam (as a soft copy)You can generate a PDF version of "Week 1 - Python Programming Fundamentals" by clicking on "..." on the top right corner then "Preview Lesson" then "Download PDF"ContentGo through Week 1 - Introduction PPT slidesStart "Week 1 - Python Programming Fundamentals"

Week 1 Thursday To Do List

Week 1 Thursday To Do ListAdmin/TipsEd Python EditorCTRL + / to comment/uncomment a selected codeCTRL +] to add one indentation for a selected codeCTRL + [to remove one indentation for a selected codeReset to Scaffold if accidently deleting lesson/practice/quiz/assignment files. However, save your work first (if any).ContentContinue "Week 1 - Python Programming Fundamentals" with "Importing modules" slideIf statements including match new Python statementWhile statementsShow the flowchart - problem solving from requirements to Python programShow how to run Turing machine simulatorHandling exceptions (if time permits)Working with files (if time permits)Useful LinksWinSCP is an open source free SFTP client, FTP client, WebDAV client, S3 client and SCP client and file manager for Windows. Its main function is file transfer between a local and a remote computer. Download it using the link below:https://winscp.net/eng/download.phpNumbers in Pythonhttps://realpython.com/python-numbers/

More Examples

School of Computer Science and Engineering (CSE)

COMP9021 Principles of Programming

2024 Term 2

Week 1

Dr Rachid Hamadi (Lecturer in Charge)

UNSW SYDNEY

Outline

- Course Introduction
- Moodle
- Ed Platform
- CSE Machines
- Academic Honesty and Plagiarism
- Week 1 - Python Programming Fundamentals
- Week 1 - From Problem Description to Python Program Example
- Week 1 - Notes 1 Turing Machines (**optional**)
- Q & A

Course Introduction

Course Introduction

- Moodle Course Website
  https://moodle.telt.unsw.edu.au/course/view.php?id=84354

- Lectures:
  Tuesday 18:00 - 20:00 in Ainsworth G03 (K-J17-G03)
  Thursday 18:00 - 20:00 in Keith Burrows Theatre (K-J14-G5)

- Tutorial:
  1 hour per week (Weeks 2-5, 7-10) either **F2F**
  or **Online** using **Blackboard Collaborate**

Course Introduction (cont'd)

	Monday	Tuesday	Wednesday			Thursday	Friday	
9am - 10am			W09A Abhay Quad G040					
10am - 11am							F10A Aayush Law 275	
11am - 12pm							F11A Hao Mat 226	F11B Ziting Block G15
12pm - 1pm			W12A Ziming OMB151					
1pm - 2pm			W13A Yiping Online	W13B Hao OMB151				
2pm - 3pm								
3pm - 4pm	M15A Hao Sqhouse115					H15A Hao Block G6		
4pm - 5pm	M16A Ziming Sqhouse115							
5pm - 6pm	M17A Abhay Sqhouse115		W17A Abhay Online	W17B Ziming Online	W17C Ziting Sqhouse214	H17A Yiping Law 301		
6pm - 7pm	M18A Aayush Ainswth101	9021 Lecture Ainsworth G03				9021 Lecture Burrows Th		
7pm - 8pm								
8pm - 9pm		T20A Dominic Online						

Course Introduction (cont'd)

Section	Class	Times (Weeks, Location)	Location	Tutor	Email
M15A	4089	Mon 15-16 (w2,4-5,7-10)	Squarehouse 115 (K-E4-115)	Hao	hao.wang19@student.unsw.edu.au
M16A	4090	Mon 16-17 (w2,4-5,7-10)	Squarehouse 115 (K-E4-115)	Ziming	ziming.gong@student.unsw.edu.au
M17A	4091	Mon 17-18 (w2,4-5,7-10)	Squarehouse 115 (K-E4-115)	Abhay	abhay.gupta1@unsw.edu.au
M18A	4092	Mon 18-19 (w2,4-5,7-10)	Ainsworth 101 (K-J17-101)	Aayush	aayush.gupta@unsw.edu.au
T20A	4093	Tue 20-21 (w2-5,7-10)	Online on Collaborate	Dominic	dominic.h.wong@unsw.edu.au
W09A	4094	Wed 09-10 (w2-5,7-10)	Quadrangle G040 (K-E15-G040)	Abhay	abhay.gupta1@unsw.edu.au
W12A	4095	Wed 12-13 (w2-5,7-10)	Old Main Building 151 (K-K15-151)	Ziming	ziming.gong@student.unsw.edu.au
W13A	4096	Wed 13-14 (w2-5,7-10)	Online on Collaborate	Yiping	yiping.tang@student.unsw.edu.au
W13B	4097	Wed 13-14 (w2-5,7-10)	Old Main Building 151 (K-K15-151)	Hao	hao.wang19@student.unsw.edu.au
W17A	4098	Wed 17-18 (w2-5,7-10)	Online on Collaborate	Abhay	abhay.gupta1@student.unsw.edu.au
W17B	4099	Wed 17-18 (w2-5,7-10)	Online on Collaborate	Ziming	ziming.gong@student.unsw.edu.au
W17C	4100	Wed 17-18 (w2-5,7-10)	Old Main Building 151 (K-K15-151)	Ziting	ziting.li@student.unsw.edu.au
H15A	4087	Thu 15-16 (w2-5,7-10)	Blockhouse G6 (K-G6-G6)	Hao	hao.wang19@student.unsw.edu.au
H17A	4088	Thu 17-18 (w2-5,7-10)	Law Building 301 (K-F8-301)	Yiping	yiping.tang@student.unsw.edu.au
F10A	4084	Fri 10-11 (w2-5,7-10)	Law Building 275 (K-F8-275)	Aayush	aayush.gupta@unsw.edu.au
F11A	4085	Fri 11-12 (w2-5,7-10)	Mathews 226 (K-F23-226)	Hao	hao.wang19@student.unsw.edu.au
F11B	4086	Fri 11-12 (w2-5,7-10)	Blockhouse G15 (K-G6-G15)	Ziting	ziting.li@student.unsw.edu.au

Course Summary

- This course provides students with solid **conceptual knowledge** and **practical skills** of both **generic programming techniques** and **Python programming**
- The features of the language are covered to a significant depth, and there is a strong emphasis on problem solving
- There is a lot of contents to study in limited time and the learning curve is not gentle

Course Resources

- No textbook and the provided material is self-contained
- Ed resources
- **Jupyter notebook sheets**, together with **static PDF files** produced from those, will be provided as **notes**
- Some of the notes are complemented with **automatically produced videos**

Course Resources (cont'd)

- Jupyter notebook sheets offer many advantages over the more traditional lecture notes

- **Cells** that make up a Jupyter notebook sheet can be edited, cells can be added or deleted, cells that contain code can be executed, allowing students to guess what the output will be and check that the guess is correct

- This will let students play a more **active** role when they learn from existing code

Assumed Knowledge

This course **does not assume** any prior knowledge of **programming** in general, or of **Python** programming in particular, as its content is self-contained for students with the expected mathematical background

Learning outcomes

1. Design, implement and test programs written in a language with procedural, object-oriented, and functional constructs
2. Apply Python language, including advanced syntax and programming techniques
3. Analyse what happens behind the scene when operating on Python data types, with an understanding of efficiency and memory use
4. Make use of fundamental data structures and algorithms
5. Design programs to solve small to medium scale problems
6. Create clear, reliable, well-structured, well-tested, well-documented programs
7. Apply appropriate tools for editing, testing and debugging

Course Schedule

The following table outlines a **provisional** schedule for this course. The contents of the lectures are described **roughly** and are subject to **adjustments**:

Week	Topics covered	Activities	Assessments
1	Introduction to operators, strings, lists, tuples, dictionaries, control structures, reading from files, printing, functions.		
2	Functions from the random module. Exceptions. Base systems, modulo operations. Unicode character set. Sorting, lambda expressions.	Practice Exercises	Quiz 1 released
3	Approximation in computations. String formatting. Lists and sets, with a view on time complexity, plotting, timing. Slices, lists with a view on space complexity.	Practice Exercises	Assignment 1 released Quiz 2 released Quiz 1 due Thursday 9pm
4	Operations on files and directories, system operations. Default arguments. Bitwise operations. The collections and matplotlib modules.	Practice Exercises	Quiz 3 released Quiz 2 due Thursday 9pm
5	Special modules. Generator functions. 2-dimensional lists, numpy arrays and operations. Regular expressions.	Practice Exercises	Quiz 4 released Quiz 3 due Thursday 9pm
6	Flex Week: No Classes.		
7	More special modules. Recursion. Memoisation. From recursive implementations to iterative implementations.	Practice Exercises	Assignment 1 due Monday 10am Assignment 2 released Quiz 5 released Quiz 4 due Thursday 9pm
8	Classes, objects. Object-oriented programming. Special methods.	Practice Exercises	Quiz 6 released Quiz 5 due Thursday 9pm
9	Dynamic programming. Inheritance. Decorators.	Practice Exercises	Quiz 6 due Thursday 9pm
10	Searching. Sorting.	Practice Exercises	
11			Assignment 2 due Monday 10am

Assessment Tasks Overview

Item	Topics	Due	Marks
Six Coding Quizzes	Various and worth 4 marks each	Thursday Weeks 3, 4, 5, 7, 8, and 9 @ 9pm	24%
Assignment 1	Procedural programming	Monday Week 7 @ 10am	13%
Assignment 2	Object oriented programming	Monday Week 11 @ 10am	13%
Final exam	Everything covered in the course	Exam period	50%

Moodle

Moodle

Used for:

- **Lecture Recordings** on Echo360 available shortly after the lectures end
- **Online Tutorials** using Blackboard Collaborate
- Check your **marks** for A1, A2, and Quizzes
- **myExperience** survey at the end of the term

Ed Platform

Ed Platform

Used for:

- Everything else
- Quizzes submission
- Assignments submission

CSE Machines

CSE Machines

Used for:

- Practicing (IDLE, Spyder, VS Code, Jupyter Notebook)
- Final Exam
- https://vlabgateway.cse.unsw.edu.au/

Academic Honesty and Plagiarism

Academic Honesty and Plagiarism

- Group submissions are not allowed
- Your programs must be entirely your own work
- Plagiarism detection software will be used to compare all submissions pairwise (including submissions for similar assignments in previous terms, if applicable) and serious penalties will be applied
- Do not copy code from others
- Do not use a publicly accessible repository to save your work or allow anyone to see your code

Academic Honesty and Plagiarism (cont'd)

- Refer to the following online resources to help you understand what plagiarism is and how it is dealt with at UNSW:
  - Plagiarism and Academic Integrity
  - Academic integrity reminder | ChatGPT
  - UNSW Plagiarism Management Procedure

- You should also read the following page which describes your responsibilities and conduct:
  - Student Responsibilities and Conduct

Academic Honesty and Plagiarism (cont'd)

- Note that the **permission level** of **generative AI** in assessments for COMP9021 is "**No assistance**"
- It is prohibited to use any **software** or **service** to search for or generate information or answers
- If its use is detected, it will be regarded as serious academic misconduct and subject to the standard penalties, which may include **00 FL** (Fail), **suspension**, and **exclusion**

Week 1 - Python Programming Fundamentals (See Ed Lessons)

Week 1 - From Problem Description to Python Program Example (See Ed Lessons)

Week 1 - Notes 1 Turing Machines (See Ed Lessons)

Q & A