ACE Network Subject Information Guide

CSC2410 Computational Thinking with Python

Semester 2, 2022

Administration and contact details

Host department	School of Sciences
Host institution	University of Southern Queensland
Name of lecturer	Wencheng Yang
Phone number	
Email address	wencheng.yang@usq.edu.au
Homepage	https://staffprofile.usq.edu.au/profile/wencheng-yang
Name of honours coordinator	Joanna Turner
Phone number	+61 7 4631 2096
Email address	Joanna.Turner@usq.edu.au
Name of masters coordinator	Xiaohui Tao
Phone number	+61 7 4631 1576
Email address	Xiaohui.Tao@usq.edu.au

Subject details

Handbook entry URL	**https://www.usq.edu.au/course/specification/2022/CSC2410-S2-2022-WEB-TWMBA.html**
Subject homepage URL	
Honours student hand-out URL	
Start date:	11 July 2022
End date:	17 October 2022
Contact hours per week:	4 hours
Census date:	5/8/22
Lecture day(s) and time(s):	Wednesday 4-6pm and Thursday 12-2pm AEST
Description of electronic access arrangements for students (for example, WebCT)	USQ Connect (Moodle)

Subject content

1. Subject content description

This course covers fundamental computational problem solving concepts, tools and methodologies. Computational thinking is a core skill across many cross disciplinary fields. And CSC2410 is a core course for USQ's Bachelor of Science students majoring in Mathematics and Statistics, or Physical Sciences, or Astronomy and Space Sciences as well as Bachelor of IT students completing the Data Analytics major.
The topics in this course are intended to introduce students not merely to the coding of computer programs, but algorithmic thinking, data management, the methodology of computer programming, and the principles of good program design including modularity, encapsulation and abstraction. The Python language is used because of its extensive application libraries and its effectiveness and popularity as a modern programming language.

While we are aware that most honours programs focus on 3rd year courses, most 3rd year programming courses will assume prior knowledge of at least one programming language and generally focus on more advanced programming topics. This 2nd year programming course provides training in key computational problem solving concepts without requiring prior programming knowledge.

2. Week-by-week topic overview

Week 1:	Introduction to modelling and simulation of physical systems. Intro to python programming. Software setup
Week 2:	Time series data and plotting of bike share system model. Python fundamentals: variables, functions, conditional statements and loops.
Week 3:	Iterative modelling and system metrics. Python fundamentals: function parameters, classes and objects.
Week 4:	Incremental development. Sweeping parameters. Python fundamentals: function return values, loops and arrays.
Week 5:	Extract data from web page with Pandas library. Model and simulate constant population growth.
Week 6:	Proportional growth model.
Week 7:	Quadratic growth and equilibrium. Python fundamentals: common problems with functions.
Week 8:	Comparing predictions.
Week 9:	Case Studies: Queueing theory, Predicting salmon populations, Tree growth. Python fundamentals: Pandas DataFrame and Series objects.
Week 10:	Epidemiology - modelling an epidemic. Evaluate the effectiveness of possible interventions.
Week 11:	Optimisation – metrics to quantify effect of a disease and possible interventions. Determine optimal interventions within fixed budget.
Week 12:	Improving the epidemic model - sweeping two parameters to explore relationship between them, using data to estimate parameters.

3. Assumed prerequisite knowledge and capabilities

Familiarity with beginner level foundational computing concepts such as variables, looping constructs and conditional statements. Ability to solve problems in the context of programming by designing, implementing, debugging and testing a solution to a prescribed problem, verifying that the solutions meets expected criteria.

4. Learning outcomes and objectives

On successful completion of this course students should be able to:

- effectively conduct program designs including modularity, encapsulation and abstraction;
- differentiate between available data types and demonstrate their appropriate problem application;
- apply available libraries to solve problems;
- identify and apply the appropriate technical processes and problem-solving skills to successfully process a variety of data sets.

AQF specific Program Learning Outcomes and Learning Outcome Descriptors (if available):

AQF Program Learning Outcomes addressed in this subject	Associated AQF Learning Outcome Descriptors for this subject
Insert Program Learning Outcome here	Choose from list below
Insert Program Learning Outcome here	Choose from list below
Insert Program Learning Outcome here	Choose from list below
Insert Program Learning Outcome here	Choose from list below
Insert Program Learning Outcome here	Choose from list below
Insert Program Learning Outcome here	Choose from list below
Insert Program Learning Outcome here	Choose from list below

Learning Outcome Descriptors at AQF Level 8

Knowledge

K1: coherent and advanced knowledge of the underlying principles and concepts in one or more disciplines

K2: knowledge of research principles and methods

Skills

S1: cognitive skills to review, analyse, consolidate and synthesise knowledge to identify and provide solutions to complex problem with intellectual independence

S2: cognitive and technical skills to demonstrate a broad understanding of a body of knowledge and theoretical concepts with advanced understanding in some areas

S3: cognitive skills to exercise critical thinking and judgement in developing new understanding

S4: technical skills to design and use in a research project

S5: communication skills to present clear and coherent exposition of knowledge and ideas to a variety of audiences

Application of Knowledge and Skills

A1: with initiative and judgement in professional practice and/or scholarship

A2: to adapt knowledge and skills in diverse contexts

A3: with responsibility and accountability for own learning and practice and in collaboration with others within broad parameters

5. Learning resources

Texts/Lecture notebooks:
The course readings and lecture notebooks are available on the USQ course StudyDesk.
Lectures notebooks use the Jupyter notebook format.
Software:
The course lectures, exercises and assignments use Python and Jupyter notebooks. Students are advised to install the open-source Anaconda Distribution which provides the necessary tools and libraries for the course.

6. Assessment

Exam/assignment/classwork breakdown					
Exam	50 %	**Assignment**	50 %	**Class work**	0%
Assignment due dates		Assignment1 (20%)26 Aug 2021	Assignment2(30%) 07 Oct 2021	Click here to enter a date.	Click here to enter a date.
Approximate exam date				25 Oct- 7 Nov	

Institution honours program details

Weight of subject in total honours assessment at host department	Click here to enter text.
Thesis/subject split at host department	Click here to enter text.
Honours grade ranges at host department	
H1	Enter range %
H2a	Enter range %
H2b	Enter range %
H3	Enter range %

Institution masters program details

Weight of subject in total masters assessment at host department	Click here to enter text.
Thesis/subject split at host department	Click here to enter text.
Masters grade ranges at host department	
H1	Enter range %
H2a	Enter range %
H2b	Enter range %
H3	Enter range %