

Subject Outline

Subject Title	Programming II
Subject Code	CP1404
Credit Points	3
Study Period	SP51
Attendance Mode	Internal
Campus	Singapore
Prerequisite/s	CP1401 / CP5369
Subject Coordinator/Division /College	Lindsay Ward / DTES / CSE
Subject Convenor	Randy Zhu
Subject Lecturer	Cynthia Chan

At James Cook University, we acknowledge the Australian Aboriginal and Torres Strait Islander peoples of this nation. We acknowledge the Traditional Owners of the lands on which our campuses and study centres are located and where we conduct our business. We pay our respects to ancestors and Elders, past, present and future. JCU is committed to honouring Australian Aboriginal and Torres Strait Islander peoples' unique cultural and spiritual relationships to the land, waters and seas and their rich contribution to JCU and society.

© Copyright 2023

This publication is covered by copyright regulations. Apart from any fair dealing for the purpose of private study, research, criticism, or review as permitted under the Copyright Act, no part may be reproduced by any process or placed in computer memory without written permission.

Contents

1	Subject details.....	3
1.1	Student participation requirements	3
1.2	Teaching Staff contact details	3
1.3	Subject description.....	4
1.4	Subject learning outcomes and course learning outcomes.....	4
1.5	Student feedback on subject and teaching.....	4
1.6	Subject resources and special requirements	4
2	Assessment details.....	5
2.1	Key dates.....	5
2.2	Requirements for successful completion of this subject	5
2.3	AccessAbility Services and Support.....	6
2.4	Assessment items.....	6
3	Submission and return of assessment	10
3.1	Submission of assessment	10
3.2	Late submissions	11
3.3	Special Consideration (including deferrals and extensions)	11
3.4	Academic Integrity	11
3.5	Return of assessment.....	11
3.6	Review of assessment	12
4	Learning and teaching in this subject.....	13
4.1	Subject calendar	13
4.2	Learning and teaching activities/expectations	14

This Subject Outline has been prepared by Lindsay Ward for the College of Science and Engineering, Division of Tropical Environments and Societies, James Cook University. Updated 06 March 2023.

The information provided in this subject outline is correct as at the time of completion and may change in response to changing University resources. Any changes will be approved by the College Dean or representative and will be communicated to students by the LearnJCU subject site.

1 Subject details

1.1 Student participation requirements

The JCU [Learning, Teaching and Assessment Procedures](#) (2.1.2d) indicates a typical student workload for a **three (3) credit point subject** requires a **130 hour work load** of study related activities, including attendance, assessment and self-directed study over the duration of the subject with equivalency across all modes of delivery.

Note that attendance at specified classes will be a mandatory requirement for satisfactory completion of some subjects ([Learning, Teaching and Assessment Procedures](#), 3.1.8e) and that additional hours may be required per week for those students in need of **English language, numeracy** or **other learning support**.

Key subject activities	
Seminar (2 hours per week)	Refer to class timing via the Timetable service from StudentFirst - https://secure.jcu.edu.sg/StudentFirst/
Tutorial (1 hour per week)	Refer to class timing via the Timetable service from StudentFirst - https://secure.jcu.edu.sg/StudentFirst/

For information regarding class registration, visit the [Class Registration Schedule](#).

Learning and teaching activities may be recorded for this subject. Personal Information in the form of images and audio may be collected by JCU during the recording. This Personal Information may appear as part of the recording which is accessible to students and staff in this subject on LearnJCU.

1.2 Teaching Staff contact details

Teaching team	Staff member	Room	Phone	Email	Consultation times*
Subject Coordinator	Lindsay Ward	-	-	lindsay.ward@jcu.edu.au	
Subject Convenor	Dr Randy Zhu	C3-06	6709 3726	Randy.zhu@jcu.edu.au	
Lecturer	Cynthia Chan	-	-	Cynthia.chan@jcu.edu.au	
Learning Advisors	JCU Singapore Learning Centre	C1-12B	-	learningcentre-singapore@jcu.edu.au	By email
Librarian	JCUS Library	C1-08	-	library-singapore@jcu.edu.au	By email

*Other consultation times by appointment only.

1.3 Subject description

In this subject, students will learn fundamental principles and practices of computer programming. Students will learn and apply coding skills in Python as well as software development techniques such as version control, testing and debugging. Students will develop procedural and object-oriented programs with console and graphical user interfaces.

1.4 Subject learning outcomes and course learning outcomes

On successful completion of this subject, you will be able to:

- develop and utilise best-practice coding techniques to develop solutions;
- select and apply appropriate and efficient data structures;
- manage software projects using version control.

These outcomes will contribute to your overall achievement of [course learning outcomes](#).

The course learning outcomes that this subject contributes to are:

- Demonstrate essential knowledge for a career in technology related professions and practice
- Convey information clearly and fluently, in high quality written form appropriate for their audience
- Apply technical skills, including numeracy, necessary for professional practice

1.5 Student feedback on subject and teaching

Students are at the heart of JCU and as part of our commitment to improving the quality of our subjects and teaching, we regularly seek feedback on the JCU student experience.

YourJCU Surveys are available to all students through [LearnJCU](#). You will receive an email invitation when the survey opens.

In response to previous student feedback and other data, the following enhancements to this subject have been made:

- The order has been adjusted to place object-oriented programming (classes) before Kivy to give people more time to focus on classes.
- The helpful programming patterns guide has been extended and enhanced.
- The topic of recursion has been moved from Programming II to Programming III.

1.6 Subject resources and special requirements

All subject readings and resources, including journal articles, book chapters, websites, videos, print and eTextbooks, are available to view online from your *Readings list* via your LearnJCU subject site.

Textbook:

- **The Practice of Computing Using Python** 3rd Edition by Punch & Enbody. Pearson, 2017. ISBN-13: 978-0134379760 (or the 2nd edition is fine)

If you already have the following textbook, then it is a suitable substitute (you do not need to get a new/different one):

- Starting Out with Python by Tony Gaddis. Global edition, 4th edition, Pearson Education <https://www.pearson.com.au/9781292225753>

Software:

- All the software used in this subject is free or freely available for JCU IT students. Instructions for getting setup on your own computer can be found at: <https://github.com/CP1404/Starter/wiki/Software-Setup> Please complete this setup as soon as you can.

2 Assessment details

2.1 Key dates

Key dates	Date
Census date and Last date to withdraw without financial penalty	See 2023 JCUS Study Period and Census Dates
Last date to withdraw without academic penalty	See 2023 JCUS Study Period and Census Dates
Assessment item 1: Assignments - Software development/creation, 40% (20% each)	A1 Due 30/04/23 A2 Due 11/06/23
Assessment item 2: Practical assessment/practical skills demonstration, 30%	Weekly
Assessment item 3: Examination (centrally administered), 30%	Due during examination period JCUS exam timetable

2.2 Requirements for successful completion of this subject

In order to pass this subject, you must:

- Achieve an overall percentage of 50% or more
- Achieve at least 33% in the final exam

Final results for this subject will be graded as described in the [Student Results Policy](#).

Graded = HD, D, C, P, F etc

Supplementary examinations/supplementary assessments are available for this subject, in accordance with the [Learning, Teaching and Assessment Procedure](#) (3.2.9) and the [Student Results Policy](#) (2.1)

2.2.1 How do I track my progress in this subject?

You will be given feedback as appropriate each week during practicals, and through the formal assessment results. You should also consider that your weekly preparation and practice will provide a good guide to how well you are progressing with the subject. You are welcome to ask teaching staff for further feedback as appropriate. You will also learn and do "code reviews" where you evaluate other students' code.

2.3 AccessAbility Services and Support

Reasonable adjustments may be made to assist you to manage additional circumstances impacting on your studies provided these do not change the academic integrity of a degree. Reasonable adjustments do not alter the need to be able to demonstrate the [inherent requirements](#) of the course.

If you believe you will experience challenges completing your degree or course because of a disability, health condition or other reason, you should fill up the [Accessibility Intake form](#) (click Forms for Prospective Students to locate this) and submit to the relevant [Student Ambassador](#) at JCUS who will be in touch with you. . For course and subject handbook refer to [Course and Subject Handbook](#).

2.4 Assessment items

ASSESSMENT ITEM 1:	
Aligned subject learning outcomes	<ul style="list-style-type: none">• develop and utilise best-practice coding techniques to develop solutions• select and apply appropriate and efficient data structures• manage software projects using version control
Aligned professional standards/ competencies	<ul style="list-style-type: none">• ACS CBoK - Modelling, abstraction, design; Programming; Systems development• SFIA - Programming/Software Development; Testing
Group or individual	Individual
Weighting and due date	40% (20% each), A1 Due 30/04/23 ; A2 Due 11/06/23

ASSESSMENT ITEM 1: DESCRIPTION

You will be given two programming assignments in this subject to help you develop new skills as well as giving you more experience with clean coding practices. Each assignment will involve planning and implementing a small-to-medium-sized Python program. Assessment will be based on the functionality of the solution as well as the quality of your code and documentation. It is important that you don't just try and get your programs to work, but that you systematically develop high-quality programs, starting with the planning process and then writing good clean code. You will use version control with Git and GitHub for each assignment.

Assignment 1 will help you build skills using techniques like decision and repetition structures, file input/output, exception handling, lists, tuples, functions and string formatting.

Assignment 2 will build on the assignment 1 program with similar functionality but more advanced code constructs such as dictionaries, classes and a Graphical User Interface (GUI) using Kivy. Classes are a fundamental tool for developing object-oriented programs and are one of the key learning goals of the subject. This second assignment will also include a reflection focused on your development process to help you become a better software developer.

ASSESSMENT ITEM 1: CRITERIA SHEET (OR RUBRIC)

- SLO 1 - develop and utilise best-practice coding techniques to develop solutions
- SLO 2 - select and apply appropriate and efficient data structures
- SLO 3 - manage software projects using version control

Assignment 1

Criteria	Exemplary (9, 10)	Good (7, 8)	Satisfactory (5, 6)	Limited (2, 3, 4)	Very Limited (0, 1)
Correctness SLO1 20%	Program works correctly for all functionality required.	Exhibits aspects of exemplary (left) and satisfactory (right)	Program mostly works correctly for most functionality, but there is/are some required aspects missing or that have problems.	Exhibits aspects of satisfactory (left) and very limited (right)	Program works incorrectly for all functionality required.
Error checking SLO1 10%	Invalid inputs are handled well using exceptions and control logic as instructed, for all user inputs.		Invalid inputs are mostly handled correctly as instructed, but there is/are some problem(s), e.g., exceptions not well used.		Error checking is not done or is very poorly attempted.
Similarity to sample output SLO1 10%	All outputs match sample output perfectly, or only one minor difference, e.g., wording, spacing.		Multiple differences (e.g., typos, spacing, formatting) in program output compared to sample output.		No reasonable attempt made to match sample output. Very many differences.
Identifier naming SLO1 10%	All function, variable and constant names are appropriate, meaningful and consistent.		Multiple function, variable or constant names are not appropriate, meaningful or consistent.		Many function, variable or constant names are not appropriate, meaningful or consistent.
Use of code constructs SLO1, 2 15%	Appropriate and efficient code use, including good logical choices for data structures and loops, good use of constants, etc.		Mostly appropriate code use but with definite problems, e.g., unnecessary code, poor choice of data structures or loops, no use of constants.		Many significant problems with code use.
Use of functions SLO1 10%	Functions and parameters are appropriately used, functions are well reused to avoid code duplication.		Reasonable but not good use of functions, e.g., poor parameter choices, function choices don't follow SRP well, too much low-level detail in main.		No functions used or functions used very poorly. Any use of global variables.
Formatting SLO1 5%	All formatting is appropriate, including correct indentation, horizontal spacing and consistent vertical line spacing. PyCharm shows no formatting warnings.		Multiple problems with formatting reduces readability of code. PyCharm shows formatting warnings.		Readability is poor due to formatting problems. PyCharm shows many formatting warnings.
Commenting SLO1 10%	Helpful block/inline comments and meaningful docstrings for all functions, top docstring contains all program details (name, date, basic description, GitHub URL).		Comments contain some noise (too many/unhelpful comments) or some missing program details in top docstring or some inappropriate or missing block/inline comments.		Commenting is very poor either through having too many comments (noise) or too few comments.
Use of version control SLO3 10%	Git/GitHub used effectively and the repository contains a good number of		Aspects of the use of version control are poor, e.g., not many commits, meaningless		Git/GitHub not used at all.

	commits with good messages that demonstrate incremental code development.		messages that don't represent valuable incremental development.		
--	---	--	---	--	--

Assignment 2

Criteria	Exemplary (9, 10)	Good (7, 8)	Satisfactory (5, 6)	Limited (2, 3, 4)	Very Limited (0, 1)
Project reflection SLO1 15%	The project reflection is complete and describes development and learning well, shows careful thought, highlights insights made during code development.	Exhibits aspects of exemplary (left) and satisfactory (right)	Project reflection contains some good content but is insufficient in coverage, depth or insight.	Exhibits aspects of satisfactory (left) and very limited (right)	Many aspects of the project reflection are missing or could be improved.
Use of version control SLO3 10%	Git/GitHub has been used effectively and the repository contains a good number of commits with good messages that demonstrate incremental code development starting with classes and testing then console before GUI .		Git/GitHub used but several aspects of the use of version control are poor, e.g., not enough commits, or meaningless messages that don't represent valuable incremental development in an appropriate order.		Git/GitHub not used.
Console program SLO1, 2 8%	Class is used correctly in console program.		Class is used in console program but not correctly.		Class is not used in console program.
Error handling SLO1 8%	Errors are handled correctly and robustly as required.		Some errors are handled but not all, or errors are not handled properly.		No reasonable error handling.
Correctness SLO1 16%	GUI layout is correct and program works correctly for all functionality required.		Aspects of the GUI layout are incomplete or poorly done or there are significant problems with functionality required.		GUI layout is very poor or not done. Program works incorrectly for all functionality required.
Identifier naming SLO1 10%	All function, variable and constant names are appropriate, meaningful and consistent.		Several function, variable or constant names are not appropriate, meaningful or consistent.		Many function, variable or constant names are not appropriate, meaningful or consistent.
Use of code constructs SLO1, 2 12%	Appropriate and efficient code use, including no unnecessary duplication, good logical choices for control and storage, good use of constants, no global variables, good use of functions in main app, etc.		Several problems, e.g., unnecessary duplication, poor control, no use of constants, improper use of global variables, poor use of functions in main app.		Many problems with code use.
Use of classes and methods SLO1, 2 10%	Classes and methods are used correctly as required. Method inputs and outputs are well designed.		Some aspects of classes and methods are not well used, e.g., methods not used where they should be, problems with method/parameter design, incorrect use of objects.		Classes and methods used very poorly or not used at all.
Commenting SLO1 6%	Code contains helpful # block comments, all classes and methods have meaningful docstrings and main module docstring contains all details (name, date, basic description, GitHub URL).		Comments are reasonable, but some classes and methods have no docstrings, and/or there is some noise (too many comments), and/or missing details in main module docstrings.		Commenting is very poor or not done.
Formatting SLO1 5%	All formatting is appropriate, including indentation, horizontal spacing and vertical line spacing. PyCharm shows no formatting warnings.		Problems with formatting reduces readability of code. PyCharm shows multiple formatting warnings.		Readability is poor due to formatting problems. PyCharm shows many formatting warnings.

ASSESSMENT ITEM 2: [INSERT ASSESSMENT METHOD HERE]

Aligned subject learning outcomes	<ul style="list-style-type: none">• develop and utilise best-practice coding techniques to develop solutions• select and apply appropriate and efficient data structures• manage software projects using version control
Aligned professional standards/ competencies	<ul style="list-style-type: none">• ACS CBoK - Modelling, abstraction, design; Programming; Systems development• SFIA - Programming/Software Development; Testing
Group or individual	Individual
Weighting and due date	30% (Weekly)

ASSESSMENT ITEM 2: DESCRIPTION

There will be 10 assessed practicals, due weekly as per the subject calendar. If you find yourself not completing the work most weeks, then please start the work earlier to give yourself more time to get it finished.

You also have **online tests** via LearnJCU, which are marked as part of the practical assessments. They are designed to help you assess your own learning and get feedback on how you are going with the subject. They must be completed by their due dates. Usually each test is made available at the start of the week *after* the related lecture, after which you have one week to complete it. This schedule is designed to use the psychological [spacing effect](#) to help you learn the material.

ASSESSMENT ITEM 2: CRITERIA SHEET (OR RUBRIC)

The 10 numbered **practicals** in the subject calendar below will be marked. Practical assessment will be as follows:

- 4: Excellent response to all questions
- 3: Good response to most or all questions, but some omissions or problems
- 2: Reasonable response to some questions, but some significant omissions or problems
- 1: Reasonable response to at least one question
- 0: Minimal or no attempt

Note: This subject is all about learning best practices and patterns for solving problems using programming (in Python) and it is important that you learn these important foundations according to the teaching philosophy of the subject. For this reason (to help you learn properly!), you will not achieve full marks for working/complete practicals that are not done in the way that we prescribe. Here are some reasons that you might lose marks for your practicals (and assignments):

- Your work looks too similar to any provided solutions
- Your work includes concepts and code not taught in the subject
- You miss or skip part of a question (you must attempt all requirements for all questions)
- Your work does not follow the core concepts and patterns (e.g. using global variables, while True loops or similar techniques), even if they work!

Because the **online tests** are mostly for keeping you on track, the actual test scores are not as important as completing them. Online tests are marked as follows:

1: test result of more than zero

0: not attempted or a test result of zero or not completed by the due date

The total for this assessment will be out of 50: your total for all 10 practicals (out of 40) + your total for all tests (out of 10). You can determine the percentage weighting of each item from this.

ASSESSMENT ITEM 3: EXAMINATION (CENTRALLY ADMINISTERED)

Aligned subject learning outcomes	<ul style="list-style-type: none">• develop and utilise best-practice coding techniques to develop solutions• select and apply appropriate and efficient data structures
Aligned professional standards/ competencies	<ul style="list-style-type: none">• ACS CBoK - Modelling, abstraction, design; Programming; Systems development• SFIA - Programming/Software Development; Testing
Group or individual	Individual
Weighting	30%
Date	Examination period
Duration	2.5 hours

ASSESSMENT ITEM 3: DESCRIPTION

You will complete a final examination during the University's exam period. The examination is the final opportunity to show your understanding of important concepts as well as demonstrate specific programming skills. It is important to note that while you may do most of your programming during the semester on a computer with the help of a development environment, the exam will test your abilities on paper, without the computer's help. Some students question whether a written exam is realistic or appropriate for a programming subject. You will need to build your understanding and experience more deeply to be able to read and write code on paper rather than with a computer. This is a valuable skill and something often covered in IT job interviews. You will be given opportunities in class to practise writing code with pen and paper so make sure you take advantage of this.

The exam will use both multiple-choice questions and short-answer written questions to test your knowledge and your ability to apply that knowledge. Questions will include reading and writing code. The exam will cover material from the entire subject (unless otherwise stated in class).

ASSESSMENT ITEM 3: CRITERIA SHEET (OR RUBRIC)

Marks are given for correct answers, and will not be deducted for incorrect answers (i.e. you will not be penalised for an incorrect answer, you will simply not receive the marks for that question). Answers to short-answer questions can earn part marks for partially correct answers.

3 Submission and return of assessment

3.1 Submission of assessment

All assessment in this subject must be submitted via the LearnJCU subject site according to any provided instructions.

3.2 Late submissions

The [Learning, Teaching and Assessment Procedure](#) (3.1.8d) outlines a uniform formula of penalties imposed for submission of an assessment item after the due date. This formula is 5% of the total possible marks for the assessment item per day including part-days, weekends, and public holidays. If submitted after 20 days, the assessment item thus would be awarded 0 marks (i.e. $5\% \times 20 = 100\%$ of total possible marks in penalties). For assessment items weighted 0%, and submitted after 10 days a DNS grade is awarded.

3.3 Special Consideration (including deferrals and extensions)

You are encouraged to access equity measures if you are affected by extenuating circumstances while undertaking the subject. JCU's [Learning, Teaching and Assessment Procedure](#) 3.1 requires that you must make yourself available for assessments and examinations at the scheduled times and extensions or deferrals for an assessment item due to previously scheduled commitments such as weddings or holidays, will not be granted.

All Special Consideration requests can be applied for through the Special Consideration application form. The form is linked to the [Special Consideration Procedure](#) and also available on the [Student Forms](#) webpage.

3.4 Academic Integrity

As outlined in the Coursework Academic Integrity [Policy](#) and [Procedure](#), you are required to complete the Coursework Academic Integrity Modules available in your LearnJCU site. Penalties for non-completion may be applied.

All non-examination items of assessment are required to be submitted with the Assessment Declaration available through LearnJCU. The Assessment Declaration contains statements relating to academic integrity under the [Coursework Academic Integrity Policy and Procedures](#). All instances of [academic misconduct](#) are treated very seriously by the University and students may be severely penalised for committing any form of academic misconduct.

For more information regarding academic integrity, see <https://www.jcu.edu.sg/current-students/student-support-services/learning-support/academic-integrity-at-james-cook-university-singapore>

3.5 Return of assessment

The requirements for an assessment's return date, time and manner will be determined by the Subject Coordinator in line with the JCU [Learning, Teaching and Assessment Procedures](#). Feedback will be given, and you can discuss your assessment with the marker as per clause 3.5 of the [Learning, Teaching and Assessment Procedures](#). You will be informed of your grade for every component of assessment in the subject under clause 3.5.1 and 3.5.2 of the [Learning Teaching and Assessment Procedures](#).

3.6 Review of assessment

Assessment items and final grades will be reviewed through moderation processes ([Learning, Teaching and Assessment Procedures](#), 3.6). It is important to be aware that assessment results “must always undergo final ratification for each study period. No single grade or mark represents a final result in a subject” ([Learning, Teaching and Assessment Procedures](#), 3.7.4.).

Assessment in this subject may involve the use of Respondus with camera surveillance or webcams. Respondus can record an assessment attempt, and that recording will be used for the investigation of cheating or any other conduct which may contravene JCU Policies and Procedures. Footage will only be accessed by persons authorised by the University to do so and may be shared with internal or external investigators. The footage constitutes Personal Information and will be stored and accessed in accordance with JCU’s [Information Privacy Policy](#).

Students can seek a review of individual assessment pieces through the process identified in clause 3.8 of the [Learning, Teaching and Assessment Procedures](#).

Students can seek a review of the final subject result through the process contained in the [Review and Appeal of a Final Subject Result Procedure](#).

4 Learning and teaching in this subject

4.1 Subject calendar

Please note, the sequence of some topics may change due to staff availability, resourcing, or due to unforeseen circumstances. Please monitor announcements made via LearnJCU.

Week/Module		Lecture	Practical	Readings/Preparation	Relationship to assessment
1	13/03	IPO, Decision, Repetition	IPO, Decision, Repetition	2, 3, 4	Assignments, Practicals, Exam
2	20/03	Functions, Version Control (Git & GitHub)	Functions, Version Control (Git & GitHub)	5	Assignments, Practicals, Exam
3	27/03	Files and Exceptions, Testing and Debugging	Files and Exceptions, Testing and Debugging	6	Assignments, Practicals, Exam
4	03/04	Lists and Tuples, More Strings	Lists and Tuples, More Strings	7, 8	Assignments, Practicals, Exam
5	10/04	Dictionaries, Code Reviews	Dictionaries, Code Reviews	9	Assignments, Practicals, Exam
LR	LECTURE RECESS				
6	24/04	Classes and OOP	Classes and OOP	10	Assignments, Practicals, Exam
7	02/05	Classes and OOP	Classes and OOP	10	Assignments, Practicals, Exam
8	8/05	GUI Programming & Kivy	GUI Programming & Kivy	13	Assignments, Practicals, Exam
9	15/05	Inheritance	Inheritance	11	Practicals, Exam
10	22/05	Database Programming, Using APIs	Database Programming, Using APIs	14	Practicals, Exam
SV	STUDY VACATION				

4.2 Learning and teaching activities/expectations

This subject is designed to develop your programming skills step-by-step using a variety of teaching and learning approaches. For most people, the best way to learn to program is to program! The more time you spend actually planning and writing programs, the better. The lectures teach specific programming skills and patterns (often with recommended ways to do common tasks) in an interactive manner. Here is one student's comment about the value of lectures (applies to external video lectures as well as internal lectures):

- *"The in class examples were the best part of this class. It is one thing to read about it in the book, but working out the problems with <the lecturer> was very fun and helped me learn the various programming techniques much quicker."*

Both live and video recording lectures often have a "Do This Now" component, where you get given a task to complete to help practise what you've learned. These are valuable not just for revision but for you to assess how you are going and what you need to catch up on. Students say:

- *"Do this now activities were helpful in understanding the content"*

It is important for you to understand that the subject focuses not just on getting a program to work, but on best practices and "clean code", including things like good identifier naming, formatting and commenting for both code readability and efficiency. Students say:

- *"Good job focusing on improving code readability and code development process, setting the right foundation for future programmers. I've learned a lot!"*

You will then use and develop the concepts and skills in the practical sessions. As an idea for external students, internal practical sessions are scheduled for 2 hours, but many students find it takes longer to complete them, that's fine - just start them before coming to the practical session, as the following student comment shows:

- *"The practicals were very long, it was hard to finish in the allocated time. Making it a necessity to start the practical before entering. Although this isn't a huge problem, it means that it is hard to manage the time efficiently."*

Aim for a minimum of 2 hours per week practising your skills in addition to the scheduled class times. It's like learning to play guitar or basketball - you can't expect to get really good if you just go to your lessons and do nothing in between - you also need to spend time practising on your own. Keep working at each skill until you really understand it and can use it comfortably. For example, when you learn about using lists to store sequential data, write as many different (but probably similar) programs as you can that use lists. We have a great collection of practice projects to help you with this at:

<https://github.com/CP1404/Starter/wiki/Practice-Programming-Projects>

Assessment, therefore, is not about learning new things or researching outside the subject to figure things out but rather using what you have learned in lectures and practised in practicals. Students say:

- *"This has to be the best constructed subject I have taken. Extremely straight forward, never felt as if the assessment was designed to trick/fool students. Simply designed to benefit those taking the subject."*
- *The subject seems to be very well organized that incorporates everything I need to know about the assignments."*

When you get stuck or need to do something you're unsure of, it is very tempting to just "Google it", but you will often find inconsistent, misleading and sometimes unhelpful material online and so we suggest you start first with the material in the subject (including lecture notes, practicals, and the programming patterns summary at <https://github.com/CP1404/Starter/wiki/Programming-Patterns>). We are very confident that you will find exactly what you need to know in the subject resources because that's how we've designed the subject - it's all here.

100%

 of what you need to know to do well in the subject is taught in the subject!

The assignments are built based on the concepts and skills taught in lectures and practicals, so anything you need to know for the assignment, you can find here. If you find yourself struggling with an assignment, go back to the lectures and practicals that relate to your area of concern and work on that area until you can comfortably apply it in your assignment.

Here are some actual comments from CP1404 students, written as part of their reflection on the second assignment:

- *Instead searching the internet for solutions i found it a lot easier to run programs from the kivy demos to find examples of the answers i was looking for.*
- *I would dedicate more time to following lectures and practicals which I have missed a lot of this semester and I am feeling the effects now.*
- *there were times I had problems (which I wasn't sure how to solve). After analysing the problem, finding appropriate resources (e.g. lecture notes), and a bit of iterative development, I was able to overcome these problems*
- *The Divide and conquer technique we had been taught in class I found worked better than I had expected. Being able to break up each of the classes into smaller parts made it easier for testing and finding out where the errors were. The fact that this technique was emphasised so much in class definitely helped in the assignment. If I had just tested the entire class all at once an error that may have been present in one part of the class may not have been detected straight away resulting in the program having a bug further down the line.*
- *Next time I'm required to completed a project like this I will read the task sheet thoroughly to ensure a proper understanding is gained of the task and the aspects involved as not realising how testing or classes worked properly this caused many issues. Testing will all also be implemented as it is handy to know that the functions work correctly before implementation into the proper program.*
- *The task sheet was filled with many useful information such as the lecture notes on testing. The task sheet was very useful and lead me to two other resources, those being the pracs and the lecture materials. The lecture materials contained everything we needed for this assignment and the pracs gave us examples to allow us to fully understand the process behind the code.*

You are expected to be an **active participant in the learning process** and are encouraged to participate in the weekly activities – lectures, practicals, and readings (prior to the lectures).

Learning and teaching activities may be recorded for this subject. Personal Information in the form of images and audio may be collected by JCU during the recording. This Personal Information may appear as part of the recording which is accessible to students and staff in this subject on Learn JCU

EMAS IT Peer Tutoring Program

EMAS means “gold” in the Malay language, and EMAS IT peer tutors provide support to students who want to achieve “gold” in their subject.

EMAS IT peer tutors assist students who have questions about the content and skills taught in the subject and are not sure who to ask. The peer tutors are students who have achieved good grades in the subject, and have been recommended by their lecturers. They are trained by the Learning Advisors in peer tutoring strategies to guide students to learn more effectively.