Unit Outline

ISYS2001 Introduction to Business Programming

Semester 1, 2024

Unit study package code:	ISYS2001
Mode of study:	Internal
Tuition pattern summary:	Note: For any specific variations to this tuition pattern and for precise information refer to the Learning Activities section. Lecture: 1 x 1 Hours Weekly Computer Laboratory: 1 x 2 Hours Weekly This unit does not have a fieldwork component.
Credit Value:	25.0
Pre-requisite units:	Nil
Co-requisite units:	Nil
Anti-requisite units:	COMP1005 (v.0) Fundamentals of Programming or any previous version
Result type:	Grade/Mark
Approved incidental fees:	Information about approved incidental fees can be obtained from our website. Visit https://www.curtin.edu.au/students/essentials/fees/understanding-your-fees/ for details.
Unit coordinator:	**Title:** Dr **Name:** Michael Borck **Phone:** +61 8 9266 3976 **Email:** Michael.Borck@curtin.edu.au **Location:** Building: 402 - Room: 720
Teaching Staff:	
Administrative contact:	**Name:** Contact Administration **Phone:** Please email **Email:** FBL-TSUnitAdmin@curtin.edu.au
Learning Management System:	Blackboard (lms.curtin.edu.au)

Acknowledgement of Country

We respectfully acknowledge the Indigenous Elders, custodians, their descendants and kin of this land past and present. The Centre for Aboriginal Studies aspires to contribute to positive social change for Indigenous Australians through higher education and research.

Coronavirus (COVID-19) Update

Curtin University is committed to supporting all our students and staff whether they are on campus, working remotely or overseas. Your health, safety and wellbeing are our priority and the continuing COVID-19 pandemic may require changes to the unit schedule, learning activities, delivery modes and assessment to provide flexible and safe options to our community. Curtin will endeavour to keep changes and disruptions to a minimum at all times. For current advice and further information visit https://www.curtin.edu.au/novel-coronavirus/.

Syllabus

This unit is designed to provide students with a solid foundation in basic programming and computational concepts, regardless of their previous coding experience. Through a variety of interactive learning activities, students will gain the knowledge and skills necessary to confidently apply programming tools and techniques to solve problems in a range of fields, such as business, science, and engineering. By the end of the unit, students will be able to develop and implement effective programs using appropriate programming languages and apply best practices in program design and development.

Introduction

In this course, we will be exploring the fundamentals of Python, one of the most popular and versatile programming languages in use today, and how it can be applied to solve real-world problems in the business environment.

Our focus will be on the use of Jupyter Notebooks, an interactive programming environment that enables us to write, run, and share code in a collaborative and user-friendly way. We will also be using GitHub, a web-based platform for version control and collaborative development, to share and manage our code and projects.

In addition to learning the language itself, we will also focus on the practical application of Python in the business context. We will explore how Python can be used to automate repetitive tasks, streamline workflows, and create customized solutions to unique business challenges. We will also cover best practices for working with data, including cleaning and preprocessing, and demonstrate how to use Python to perform statistical analysis and create visualizations that help to communicate insights.

By the end of the course, you will have a strong foundation in Python programming and be able to apply your skills to a range of business tasks. You will have experience using Jupyter Notebooks to write and run code, and you will understand how to use GitHub to share and manage your projects. You will also have a portfolio of projects that demonstrate your understanding of Python and your ability to apply it in a business context.

This course is an excellent introduction to the world of Python programming for business. With its focus on practical applications and real-world examples, you will gain the skills and experience you need to be successful in this exciting and dynamic field.

Unit Learning Outcomes

All graduates of Curtin University achieve a set of six Graduate Capabilities during their course of study. These inform an employer that, through your studies, you have acquired discipline knowledge and a range of other skills and capabilities which employers would value in a professional setting. Each unit in your course addresses the Graduate Capabilities through a clearly identified set of learning outcomes. They form a vital part in the process referred to as assurance of learning. The learning outcomes notify you of what you are expected to know, understand or be able to do in order to be successful in this unit. Each assessment for this unit is carefully designed to test your knowledge of one or more of the unit learning outcomes. On successfully completing all of the assessments you will have achieved all of these learning outcomes.

Your course has been designed so that on graduating you will have achieved all of Curtin's Graduate Capabilities through the assurance of learning processes in each unit.

	On successful completion of this unit students can:	Graduate Capabilities addressed
1	Configure and navigate an IDE to develop, debug, and execute programs efficiently. Use IDE features to optimize program performance and functionality	
2	Recognize, describe, and effectively use different data types and algorithmic constructs to design and develop innovative and responsible programs that are both effective and efficient	
3	Plan and execute effective testing strategies for programs using standard techniques and notations. Refine and optimize programs using testing results and collaborate with team members to ensure comprehensive testing	
4	Develop problem-solving strategies and apply programming methodologies to create efficient algorithms and programs that solve real-world problems and approach programming challenges from an innovative and responsible perspective, taking into account the social, ethical, and global implications of your programs	

Curtin's Graduate Capabilities

	Apply discipline knowledge, principles and concepts		Innovative, creative and entrepreneurial		Effective communicators with digital competency
	Globally engaged and responsive		Culturally competent to engage respectfully with local First Peoples and other diverse cultures		Industry connected and career capable

Find out more about Curtin's Graduate Capabilities at the Learning Innovation and Teaching Excellence Centre (LITEC) website: litec.curtin.edu.au

Learning Activities

In this unit, students will have several learning experiences that will help them build a strong foundation in Python programming for business using notebooks and GitHub. These experiences include:

1. Lectures: The unit will feature lectures that introduce students to the basics of Python programming, including data types, control structures, functions, and object-oriented programming. Lectures will also cover the use of popular Python libraries, such as NumPy, Pandas, and Matplotlib.
2. Hands-on programming: To reinforce the concepts covered in the lectures, students will participate in hands-on programming exercises using Jupyter Notebooks. These exercises will allow students to write and run code, and to practice applying their skills to real-world business problems.
3. Collaborative projects: Students will work in small groups on collaborative projects that apply Python programming to business tasks. These projects will be shared and managed using GitHub, allowing students to practice using version control and collaborative development tools.
4. Guest speakers: The unit may feature guest speakers from industry, who will share their experiences and insights on using Python in a business setting. These talks will provide students with a deeper understanding of the practical applications of Python in the real world.
5. Assessments: To measure their understanding of the material, students will complete a range of assessments, including programming assignments, quizzes, and exams. These assessments will test both their knowledge of the language itself, as well as their ability to apply it to solve real-world business problems.

Overall, the unit will provide students with a comprehensive learning experience that will equip them with the skills and knowledge they need to apply Python programming to a range of business tasks. By combining lectures, hands-on programming exercises, collaborative projects, guest speakers, and assessments, students will gain a well-rounded understanding of the language and its practical applications.

Learning Resources

Essential texts

The required textbook(s) for this unit are:

- Al Sweigart. 2015. Automate the Boring Stuff with Python: Practical Programming for Total Beginners (1st. ed.). No Starch Press, USA.
  (ISBN/ISSN: ISBN-10: 1-59327-992-2 ISBN-13: 978-1-59327-992-9)

Online resources

- Al Sweigart. 2015. Automate the Boring Stuff with Python: Practical Programming for Total Beginners (1st. ed.). No Starch Press, USA.
  (https://automatetheboringstuff.com/)
  (ISBN/ISSN: ISBN-10: 1-59327-992-2 ISBN-13: 978-1-59327-992-9)

Assessment

Assessment policy exemptions

- There are no exemptions to the assessment policy

Assessment schedule

	Task	Value %	Date Due	Unit Learning Outcome(s) Assessed	Late Assessments Accepted?*	Assessment Extensions Considered?*
1	Reflective Journal	20%	**Week:** Weeks 1 to 6, and 8 **Day:** Every Sunday **Time:** 23:59	1,2,3	Yes	Yes
2	Programming Assignment	30%	**Week:** 13 **Day:** 26 May 2023 **Time:** 23:59	1,3,4	Yes	Yes
3	Extended Learning Portfolio	50%	**Week:** Examinations **Day:** TBA **Time:** TBA	1,2,3	Yes	Yes

*Please refer to the Late Assessment and the Assessment Extension sections below for specific details and conditions.

Detailed information on assessment tasks

1. **Weekly Journal.** In this assesment students will keep a reflective journal to document their learning experiences and reflect on their progress throughout the course. The reflective journal will serve as a personal record of the student's growth and development, as well as a tool for self-assessment and goal-setting. Students may use the journal to reflect on the lectures, programming exercises, collaborative projects, and assessments, and to identify areas of strength and areas for improvement. The reflective journal will be a valuable resource for students as they continue to develop their skills in Python programming and apply them to solve real-world business problems.
2. **Programming Assignment.** For this assesment students will be tasked with creating a Python program that automates a business task. The program should be designed to save time and increase efficiency, and

should demonstrate the practical application of Python programming in a business setting. For example, automate a task related to data analysis, web scraping, or report generation, among others. The program should be well-documented and easy to use, and should showcase the student's skills in using Jupyter Notebooks and Python libraries, as well as their ability to apply best practices in programming and data management.

3. **Extended Learning Portfolio.** An extended learning portfolio consists of a collection of programming projects and reflective writing that demonstrate the student's mastery of Python programming for business. The portfolio projects that showcase the student's ability to apply Python to real-world business problems. In addition to the programming work, the portfolio should also feature reflective writing that documents the student's learning journey, highlights key insights and areas of growth, and identifies future goals for developing their skills. The extended learning portfolio will provide a comprehensive record of the student's achievements and growth throughout the unit, and serve as a valuable resource for future learning and career development.

Pass requirements

To pass this unit, students must achieve overall a minimum grade 50% and have submitted ALL assesments. In addition to meeting the minimum grade requirements, students are also expected to attend all lectures and participate in all hands-on programming exercises and collaborative projects. Students must also demonstrate an understanding of the practical application of Python programming in a business context, and be able to apply their skills to solve real-world problems. Finally, students must submit a reflective journal and an extended learning portfolio that document their learning experiences and progress throughout the unit. By meeting these requirements, students will be well-equipped to apply Python programming to a range of business tasks and pursue further study in this exciting and dynamic field

Assessment Moderation

Fair assessment through moderation

Moderation describes a quality assurance process to ensure that assessments are appropriate to the learning outcomes, and that students work is evaluated consistently by assessors. Minimum standards for the moderation of assessments are described in the Assessment and Student Progression Manual, available from policies.curtin.edu.au/findapolicy/

Pre-marking moderation

To ensure quality assessment practices and experiences in this unit, pre-marking moderation strategies could include the following:

1. Clear assessment criteria: Providing clear and specific assessment criteria for each assessment task will ensure that students understand what is expected of them and can meet the learning outcomes of the unit.
2. Standardization of grading: Ensuring consistency in grading through standardization of grading rubrics and pre-marking meetings between assessors can help to minimize variations in grading practices and ensure fair and consistent assessment outcomes.
3. Blind marking: Blind marking, in which the assessor does not know the identity of the student, can help to eliminate any potential biases and ensure that students are assessed solely on the quality of their work.
4. Sampling of student work: Sampling of student work across multiple assessors can help to ensure that the quality of assessment is maintained and that grading is consistent across multiple assessors.
5. Feedback to students: Providing clear and constructive feedback to students on their work, including areas for improvement and opportunities for further development, can help to promote learning and ensure that students are meeting the required standards.

Implementing these pre-marking moderation strategies, the unit can ensure that assessment practices and experiences are of high quality, consistent, and aligned with the learning outcomes of the unit.

Intra-marking / Post-marking moderation

assessment practices and experiences in this unit. These strategies include:

1. Double marking: Double marking, in which a second assessor marks the same piece of work independently, can help to ensure consistency and fairness in grading.
2. Moderation of grades: Moderation of grades, in which multiple assessors review the grades assigned to a particular assessment task, can help to ensure that grades are consistent and fair across all students.
3. Feedback to assessors: Providing feedback to assessors on their grading practices, including areas for improvement and opportunities for further development, can help to ensure that grading is consistent, fair, and aligned with the assessment criteria.
4. Review of assessment tasks: Reviewing the assessment tasks and rubrics after grading is complete can help to identify any areas for improvement in future assessments and ensure that assessment tasks are appropriately aligned with the learning outcomes of the unit.
5. Appeals process: Providing an appeals process for students to challenge their grades and seek further clarification on the grading process can help to ensure that grades are fair, consistent, and aligned with the learning outcomes of the unit.

Implementing these intra-marking and post-marking moderation strategies, the unit can ensure that assessment practices and experiences are of high quality, consistent, and aligned with the learning outcomes of the unit.

Late assessment

Where the submission of a late assessment is permitted, late penalties will be consistently applied in this unit.

Where a late assessment **is** permitted for an assessment item or the entirety of the unit (refer to the Assessment Schedule table in this Unit Outline) and the student does not have an approved assessment extension:

1. For assessment items submitted within the first 24 hours after the due date/time, students will be penalised by a deduction of 5% of the total marks allocated for the assessment task;
2. For each additional 24 hour period commenced an additional penalty of 10% of the total marks allocated for the assessment item will be deducted; and
3. Assessment items submitted more than 168 hours late (7 calendar days) will receive a mark of zero.

Where late assessment **is NOT** permitted for an assessment item or the entirety of the unit (refer to the Assessment Schedule table in this Unit Outline) and the student does not have an approved assessment extension:

1. All assessment items submitted after the due date/time will receive a mark of zero.

Assessment extension

Where an application for an assessment extension **is** permitted for an assessment item(s) within this unit (refer to the Assessment Schedule table in this Unit Outline):

1. A student who is unable to complete an assessment item by/on the due date/time as a result of exceptional circumstances beyond the student's control, may apply for an assessment extension on the Assessment Extension Application Form as prescribed by the Academic Registrar. The form is available on the Forms page at https://students.curtin.edu.au/essentials/forms-documents/forms/ and also within the student's OASIS (My Studies tab – Quick Forms) account.
2. The student will be expected to submit their application for an Assessment Extension with supporting documentation via the online form.
3. Timely submission of this information supports the assessment process. For applications that are declined, delayed submission may have significant ramifications on the possible marks awarded.
4. An application may be accepted up to five working days after the due date/time of the assessment item where the student is able to provide a verifiable explanation as to why they were not able to submit the application prior to the assessment due date/time

Where an application for an assessment extension **is NOT** permitted for an assessment item(s) within this unit (refer to the Assessment Schedule table in this Unit Outline):

1. All assessment items submitted after the due date/time will be subject to late penalties or receive a mark of zero depending on the unit permitting late assessment submissions.

Deferred assessments

If your results show that you have been granted a deferred assessment you should immediately check OASIS for details.

Deferred examinations/tests will be held from 15/07/2024 to 24/07/2024 . Notification to students will be made after the Board of Examiners' meeting via the Official Communications Channel (OCC) in OASIS.

Further assessment

Further assessments, if granted by the Board of Examiners, will be held between 15/07/2024 and 24/07/2024 . Notification to students will be made after the Board of Examiners meeting via the Official Communications Channel in OASIS.

It is the responsibility of the student to be available to complete the requirements of a further assessment. If your results show that you have been granted a further assessment you should immediately check OASIS for details.

Reasonable adjustments for students with disabilities/health circumstances likely to impact on studies

A Curtin Access Plan (CAP) is a document that outlines the type and level of support required by a student with a disability or health condition to have equitable access to their studies at Curtin. Carers for people with disability may also be eligible for support. This support can include alternative exam or test arrangements, study materials in accessible formats, access to Curtin's facilities and services or other support as discussed with an advisor from AccessAbility Services.

Documentation is required from your treating Health Professional to confirm your health circumstances or carer responsibilities.

If you think you may be eligible for a CAP, please contact AccessAbility Services. If you already have a CAP please provide it to the Unit Coordinator in week 1 of each study period.

Referencing style

The referencing style for this unit is Chicago 17th Author-Date.

More information can be found on this style from the Library web site: https://libguides.library.curtin.edu.au/uniskills/referencing/chicago17.

Privacy

As part of a learning or assessment activity, or class participation, your image or voice may be recorded or transmitted by equipment and systems operated by Curtin University. Transmission may be to other venues on campus or to others both in Australia and overseas.

Your image or voice may also be recorded by students on personal equipment for individual or group study or assessment purposes. Such recordings may not be reproduced or uploaded to a publicly accessible web environment. If you wish to make such recordings for study purposes as a courtesy you should always seek the permission of those who are impacted by the recording.

Recording of classes or course materials may not be exchanged or distributed for commercial purposes, for compensation, or for any other purpose other than personal study for the enrolled students in the unit. Breach of this may subject a student to disciplinary action under Statute No 10 – Student Disciplinary Statute.

If you wish to discuss this please talk to your Unit Coordinator.

Copyright

Academic Integrity (including plagiarism and cheating)

Academic Integrity

Curtin's Student Charter, Academic Integrity Program (AIP), and core Values guide expectations regarding student behaviour and responsibilities. Information on these topics can be found on the Academic Integrity Website.

Academic Integrity Warnings

An Academic Integrity Warning may be issued to a student in limited circumstances and only where misconduct is not involved.

Academic Misconduct

Staff members are required to report poor academic practice and suspected misconduct. Academic Misconduct means conduct by a student that is dishonest or unfair in connection with any academic work. This includes all types of plagiarism, cheating, collusion, falsification or fabrication of content, and behaviours like falsifying medical certificates for extension. Contract cheating, the use of file sharing, translation services/apps, paraphrasing tools (text-spinners), article generators, and assignment help websites also may be considered academic misconduct.

Check your assessment instructions carefully before using any generative artificial intelligence (Gen-AI) software (e.g. Chat GPT, Midjourney, GitHub Copilot, etc.). You are not permitted to use Gen-AI software in any assessment task unless written permission is explicitly granted by the Unit Coordinator (e.g. within Blackboard or the assignment specifications). If the use of Gen-AI software has been approved, you must document its use, apply appropriate acknowledgement and attribution rules, and include a statement as to the nature and extent of the use when submitting the assessment. Unapproved, inappropriate, or undisclosed use may be dishonest or unfair behaviour, and thus considered misconduct. For further information on the use of Gen-AI software see the Academic Integrity Website.

The longer term personal, social, and financial consequences of misconduct can be severe, so please ask your tutors or unit coordinator if you need clarification or are unsure what to do. If your work is the subject of an inquiry, you will be given an opportunity to respond and appropriate support will be provided. Academic work under inquiry will not be graded until the process has concluded. Penalties for misconduct may include a warning, a reduced or nil grade, a requirement to repeat the assessment, an annulled grade (ANN) or termination from the course. For more information refer to Statute No.10 Student Discipline and Academic Misconduct Rules.

Information and Communications Technology (ICT) Expectations

Curtin students are expected to have reliable internet access in order to connect to OASIS email and learning systems such as Blackboard and Library Services.

You may also require a computer or mobile device for preparing and submitting your work.

For general ICT assistance, in the first instance please contact OASIS Student Support: oasisapps.curtin.edu.au/help/general/support.cfm

For specific assistance with any of the items listed below, please visit UniSkills and the IT tools and guides webpage.

- Using Blackboard, the I Drive and Back-Up files
- Introduction to PowerPoint, Word and Excel

Additional information

Further Assessment Student Eligibility

The Board of Examiners grants further Assessments.

Accreditation Bodies

This unit develops topics from the Australian Computer Society (ACS) ICT Profession Body of Knowledge: Systems Acquisition. You can find more information about these topics and the ICT Professional Body of Knowledge at: https://www.acs.org.au/__data/assets/pdf_file/0007/7792/The-ICT-Profession-Body-of- Knowledge.pdf

This unit supports AACSB International, Assurance of Learning through course-embedded assessments. In particular, it supports the broad course learning goals of discipline knowledge (dk), critical thinking (ct), written communication (wc), oral communication (oc), and ethical reasoning (er). Tou can fine more information at: http://www.aacsb.edu/accreditation/standards/2013-business/learning- and teaching/standard8.aspx

Enrolment

It is your responsibility to ensure that your enrolment is correct - you can check your enrolment through the eStudent option on OASIS, where you can also print an Enrolment Advice.

Student Rights and Responsibilities

It is the responsibility of every student to be aware of all relevant legislation, policies and procedures relating to their rights and responsibilities as a student. These include:

- the Student Charter
- Values and Signature Behaviours
- the University's policy and statements on plagiarism and academic integrity
- copyright principles and responsibilities
- the University's policies on appropriate use of software and computer facilities

Information on all of the above is available through the University's "Student Rights and Responsibilities" website at: students.curtin.edu.au/rights.

Note: In Australia and other jurisdictions, students are required to complete a screening check prior to undertaking any activities that include children (e.g. surveying children at a school as part of a project). If this applies to you, start by contacting your unit coordinator for advice.

Student Equity

There are a number of factors that might disadvantage some students from participating in their studies or assessments to the best of their ability, under standard conditions. These factors may include a disability or medical condition (e.g. mental illness, chronic illness, physical or sensory disability, learning disability), significant caring responsibilities, pregnancy, religious practices, living in a remote location,or another reason. If you believe you may be unfairly disadvantaged on these or other grounds please contact the appropriate service below. It is important to note that the staff of the University may not be able to meet your needs if they are not informed of your individual circumstances, so please get in touch with the appropriate service if you require assistance.

To discuss your needs in relation to:

- Disability or medical conditions, contact AccessAbility Services: https://students.curtin.edu.au/personal-support/disability/
- Elite athletes, contact Elite Athlete Coordinator: https://stadium.curtin.edu.au/sport/academy/elite-athlete-program/
- All other grounds, contact the Student Wellbeing Advisory Service: https://students.curtin.edu.au/personal-support/counselling-guidance/wellbeing/

Recent Unit Changes & Response to Student Feedback

Students are encouraged to provide feedback through student surveys (such as Insight (Curtin's new unit and teaching survey developed in collaboration with students and staff) and the annual Student Experience Survey) and interactions with teaching staff.

Listed below are some recent changes to the unit as a result of student feedback.

Student feedback is a crucial tool for improving the quality and effectiveness of the unit. Based on student feedback, changes made include adjustments to the content, assessment tasks, and delivery methods. In addition to student feedback, the use of ChatGPT, an artificial intelligence language model, will be used to provide valuable insights into student learning and engagement. By analyzing student interactions with ChatGPT, instructors can gain a deeper understanding of students' strengths and weaknesses, and adjust the content and delivery of the unit accordingly. Through these ongoing improvements, the unit can provide a high-quality and effective learning experience for all students.

Program calendar

Week	Begin Date			
Orientation	19 February	Orientation Week		
1.	26 February	Computational Thinking	Chapter 1	Weekly Journal
2.	4 March	Data Types and Objects	Chapter 4 & 5	Weekly Journal
3.	11 March	Building Blocks	Chapter 3	Weekly Journal
4.	18 March	Input Output	Chapter 8	Weekly Journal
5.	25 March	Modularity	Chapter 10	Weekly Journal
7.	1 April	Tuition Free Week		
6.	8 April	Managing Code		Weekly Journal
8.	15 April	Web Applications	Chapter 12	Completed Journal
9.	22 April	Databases		
10.	29 April	User Interfaces		
11.	6 May	Object Orientation		
12.	13 May	Algorithms		
13.	19 May	Catch-Up/Review		Programming Assignment
14.	27 May	Study Week		
15.	3 June	Examinations		
16.	10 June	Examinations		