



Faculty of Science, Engineering and Built Environment

SIT232 Object-Oriented Development

Deakin University Unit Guide

Trimester 2, 2020

CONTENTS

WELCOME	2
WHO IS THE UNIT TEAM?	2
Unit chair: leads the teaching team and is responsible for overall delivery of this unit	2
Unit chair details	2
Other members of the team and how to contact them	3
Administrative queries	3
ABOUT THIS UNIT	3
Unit development in response to student feedback	3
Your course and Deakin's Graduate Learning Outcomes	4
Your Unit Learning Outcomes	4
ASSESSING YOUR ACHIEVEMENT OF THE UNIT LEARNING OUTCOMES	5
Hurdle requirements	5
Summative Assessment during teaching period	6
- Summative assessment task 1	7
Your learning experiences in this Unit - and your expected commitment	9
Scheduled learning activities - campus	9
Scheduled learning activities - cloud	9
Note	10
UNIT LEARNING RESOURCES	10
Essential learning resources	11
KEY DATES FOR THIS TRIMESTER	11
UNIT WEEKLY ACTIVITIES	11

WELCOME

Welcome to the unit **SIT232 Object-Oriented Development**.

In this unit, we will undertake a study of the object-oriented approach to software development. This unit forms part of the Bachelor of Information Technology as a compulsory unit for students studying computer science or software development, and an elective unit for all other streams.

This unit is roughly divided into three areas of study. First, we review the fundamentals of the C# programming language including program construction, data types, control structures, and mechanisms for modularity. Students go through how to construct computer programs following the procedural design principles. Second, we shift our focus to object-oriented development where students study analysis, design, and programming principles, explore class definition, the relationships between classes and objects, inheritance and polymorphism, and undertake a study of UML and object modelling. Third and finally, we examine features common to object-oriented programming languages including overloading, exceptions, generics, and use of programming language libraries.

Learning resources for this unit will be mainly delivered via the unit site (accessed in DeakinSync). This material contains weekly learning activities including video and article steps to help guide your learning. Accompanying this are associated assessment tasks in [OnTrack](#), your progress on these tasks will form the basis of your portfolio which will be assessed at the end of the unit.

If you have any queries or feedback regarding the unit and its content, please feel free to post a message in the discussion forums or email the unit staff. Note that queries of a personal nature should be communicated directly with unit staff and must not be posted in the public discussion forums. Finally, we would like to wish you the very best for your study of SIT232 and hope you enjoy the trimester.

This Unit Guide provides you with the key information about this Unit. For the best chance of success, you should read it very carefully and refer to it frequently throughout the trimester. Your Unit site (accessed in **DeakinSync**) also provides information about your **rights and responsibilities**. We will assume you have read this before the Unit commences, and we expect you to refer to it throughout the trimester.

Due to the coronavirus (COVID-19) situation, you may be learning in a way that is new to you. We appreciate your flexibility and dedication to learning. For a range of helpful services and resources, please go to study support <https://www.deakin.edu.au/students/studying/study-support>.

WHO IS THE UNIT TEAM?

Unit chair: leads the teaching team and is responsible for overall delivery of this unit

Nayyar Zaidi

Unit chair details

Name	Dr Nayyar Zaidi
Campus:	Melbourne Burwood Campus 221 Burwood Highway BURWOOD VIC 3125
Email:	nayyar.zaidi@deakin.edu.au
Phone:	+61 3 924 45963

Other members of the team and how to contact them**Melbourne Burwood Campus and Cloud Leader:**

contact the campus leader for assistance at your campus

Name: Dr Sergey Polyakovskiy (Unit Chair)

Email: sergey.polyakovskiy@deakin.edu.au

Phone: +61 3 924 68813

Geelong Waurin Ponds Campus Leader:

contact the campus leader for assistance at your campus

Name: Dr Mengmeng Ge

Email: mengmeng.ge@deakin.edu.au

Phone: +61 3 522 73078

Administrative queries

- Contact your Unit Chair or Campus Leader
- Drop in or contact [Student Central](#) to speak with a Student Adviser

For additional support information, please see the Rights and Responsibilities section under 'Content' in your unit site.

ABOUT THIS UNIT

SIT232 introduces students to object-oriented programming as the fundamental paradigm of modern programming languages and software development. Upon completion of the unit, the students will be ready to develop real-world software following the software design and structure best practices.

Unit development in response to student feedback

Every trimester, we ask students to tell us, through eVALUate, what helped and hindered their learning in each Unit. You are strongly encouraged to provide constructive feedback for this Unit when eVALUate opens (you will be emailed a link).

In previous versions of this unit, students have told us that these aspects of the Unit have helped them to achieve the learning outcomes:

- Prompt responses to questions on discussion boards, which gave a lot of helpful advice and made the unit material and assessment easier for students
- The lectures and practicals had clear goals and were good to support learning and applied knowledge rather than just testing it.
- The programming assignments were well-thought and appeared to be representative of “real-world” issues.
- Release of weekly practical solutions greatly helped when working on larger assignments.
- The content of the unit was very detailed and went well in depth of programming concepts.
- The workbook acted as an informative learning tool and provided a wonderful breakdown of C# and much of the theory that students used within the unit.
- The extra learning material (e.g. instructional videos and external resources) was really helpful.

They have also made suggestions for improvement, and so this is what we have done:

- This offer of the unit follows the portfolio-based model, and therefore its structure has been redeveloped.
- Firstly, there is now no exam and the assessment is based on a suite of projects/assignments of different difficulty. Students are allowed to set a goal that they wish to achieve in terms of their grades and select specific tasks to fulfil the requirements of the selected grade.
- Secondly, the material previously given within this unit has been slightly reduced in order to pay more attention on important concepts and implementation issues in object-oriented programming. This change reduced the introductory part inherited from the SIT102 Introduction to Programming unit.
- Finally, this unit has got more diverse assessment. Most of these changes have been made to meet the feedback collected from the students in the previous offers.

If you have any concerns about the Unit during the trimester, please contact the unit teaching team - preferably early in the trimester - so we can discuss your concerns, and make adjustments, if appropriate.

Your course and Deakin's Graduate Learning Outcomes

GLO1 Discipline knowledge and capabilities:	appropriate to the level of study related to a discipline or profession
GLO2 Communication:	using oral, written and interpersonal communication to inform, motivate and effect change
GLO3 Digital literacy:	using technologies to find, use and disseminate information
GLO4 Critical thinking:	evaluating information using critical and analytical thinking and judgment
GLO5 Problem solving:	creating solutions to authentic (real world and ill-defined) problems
GLO6 Self-management:	working and learning independently, and taking responsibility for personal actions
GLO7 Teamwork:	working and learning with others from different disciplines and backgrounds
GLO8 Global citizenship:	engaging ethically and productively in the professional context and with diverse communities and cultures in a global context

Each Deakin course has **course learning outcomes** which explain what the Deakin Learning Outcomes mean in your discipline. Learning in each unit builds towards the course learning outcomes.

Your Unit Learning Outcomes

Each Unit in your course is a building block towards these Graduate Learning Outcomes - not all Units develop and assess every Graduate Learning Outcome (GLO).

	These are the Learning Outcomes (ULO) for this Unit At the completion of this unit successful students can:	Deakin Graduate Learning Outcomes
ULO1	Evaluate simple program code for correct use of coding conventions, and use code tracing and debugging techniques to identify and correct issues.	GLO1: Discipline-specific knowledge and capabilities GLO4: Critical thinking
ULO2	Apply and explain the principles of object-oriented programming including abstraction, encapsulation inheritance and polymorphism.	GLO1: Discipline-specific knowledge and capabilities GLO4: Critical thinking
ULO3	Implement and test small object-oriented programs that conform to planned system structures and requirements.	GLO1: Discipline-specific knowledge and capabilities GLO5: Problem solving
ULO4	Design, communicate, and evaluate solution structures using appropriate diagrams and textual descriptions.	GLO2: Communication

	These are the Learning Outcomes (ULO) for this Unit At the completion of this unit successful students can:	Deakin Graduate Learning Outcomes
ULO5	Justify meeting specified outcomes through providing relevant evidence and critiquing the quality of that evidence against given criteria.	GLO4: Critical thinking GLO6: Self-management

These Unit Learning Outcomes are applicable for all teaching periods throughout the year

ASSESSING YOUR ACHIEVEMENT OF THE UNIT LEARNING OUTCOMES

Hurdle requirements

To be eligible to obtain a pass in this unit, students must meet certain milestones as part of the portfolio.

Brief summary of hurdle requirement/s	Rationale
<p>1. Unit Milestones</p> <p>Students are required to submit tasks by their indicated due dates (S) and collaborate with their teaching staff to resolve any issues identified, and discuss their understanding of associated concepts by the task deadline (D). You will be at risk of failing the unit if start missing the deadlines and your progress becomes slower than the timelines presented below.</p> <p>The following indicative timelines will apply (shown visually below). Task descriptions published in OnTrack system will provide a more up-to-date timelines during the trimester. Here,</p> <ul style="list-style-type: none"> - S indicates the submission week (For example, Week 1 Pass Tasks must be submitted before the end of week 2). - D indicates the discussion deadline (For example, Week 5 Pass Tasks must be signed off as complete by the end of week 9). 	<p>This unit requires a pass in the interview component to ensure that each student can demonstrate their learning independently.</p>

All tasks must be completed and discussed with teaching staff before the end of week 11.

	Weeks										
	1	2	3	4	5	6	7	8	9	10	11
Pass tasks											
1.XP		S		D							
2.XP			S		D						
3.XP				S		D					
4.XP					S		D				
5.XP						S		D			
6.XP							S		D		
7.XP								S		D	
8.XP									S		D

All task discussions are compulsory and must be conducted in practical class (for campus student) or via OnTrack discussions (for Cloud students only). Please ensure that you are enrolled in the correct mode of study.

Tasks may be discussed with staff anytime within the submission period. It is strongly recommended that tasks are submitted well ahead of these deadlines, as completion of the tasks involve submitting work for assessment, responding to feedback, discussing the task with teaching staff, and ensuring work submitted demonstrates the required outcomes. In many cases work will need to be corrected and resubmitted, potentially more than once, as part of this process.

Summative Assessment during teaching period

(tasks that will be graded or marked)

From Monday 22 June 2020, Deakin has a universal assessment submission time of 8 pm AEST. A late penalty will apply to assessment submitted after 11.59 pm AEST.

NOTE: It is your responsibility to keep a backup copy of every assignment where it is possible (eg written/digital reports, essays, videos, images). In the unusual event that one of your assignments is misplaced, you will need to submit the backup copy. Any work you submit may be checked by electronic or other means for the purposes of detecting collusion and/or plagiarism.

When you are required to submit an assignment through your unit site (accessed in DeakinSync), you should receive an email to your Deakin email address confirming that it has been submitted. You should check that you can see your assignment in the Submissions view of the Assignment folder after upload, and check for, and keep, the email receipt for the submission.

- Summative assessment task 1

	Learning Portfolio
Brief description of assessment task	<p>In this unit, assessment is designed to encourage and reward you for demonstrating achievement of the unit learning outcomes; with higher grades representing better achievement of these outcomes.</p> <p>We will be using OnTrack, which is a web application designed specifically to support your completion of learning and assessment activities. Working regularly and completing the tasks on time will help you collect evidence for your portfolio. Your portfolio will consist of work that you complete in response to the unit's tasks.</p> <p>These tasks are designed to help you learn and demonstrate achievement of the unit learning outcomes. Tasks will consist of the following kinds of activities:</p> <ul style="list-style-type: none"> • Evaluating code in terms of professional practice and implementation of object-oriented principles • Modelling a program using object-oriented abstraction concepts • Develop class and sequence diagrams reflecting the intended program structure and behaviour • Develop programs using C# that realise the functionalities of a given case study • Write test cases using object-oriented testing libraries and frameworks • Compare different program designs, identify issues and choose the appropriate designs • Documenting designs, code, and test cases • Answering on-line quizzes covering theory and implementation issues studied in the unit
Detail of student output	<p>Students will produce a range of artefacts including program code, reports, concept maps, tests, and others. This work will be combined together with reflections on your learning into your learning portfolio for assessment.</p> <p>Task discussion</p> <p>You will be required to discuss your understanding of concepts, and demonstrate achievement of unit learning outcomes, with the teaching team for the unit tasks to be marked as Complete. On-campus students are required to do this face-to-face during practical sessions, while cloud students will be required to discuss this over OnTrack discussions, Bb Collaborate or Skype for Business.</p>

Grading and weighting (% total mark for unit)	<p>100% - marked and graded</p> <p>Each task in the unit is associated with a grade: either Pass, Credit, Distinction, or High Distinction. Each grade will be awarded based on completion of the tasks associated with that grade, and the lower grades. For this unit the following will set the minimum standard for each grade:</p> <ul style="list-style-type: none"> - Pass: Complete all critical pass tasks, including discussion with teaching staff - Credit: Complete all pass and credit tasks - Distinction: Complete all pass, credit and distinction tasks - High Distinction: Complete all pass, credit, distinction and at least one high distinction task. <p>In general, the graded tasks will provide the following challenge levels:</p> <ul style="list-style-type: none"> - Pass. Scaffolded tasks to help achieve minimum acceptable standard. Students will be able to implement object-oriented solutions that make use of abstraction, encapsulation, inheritance, and polymorphism if given a detailed design to work from. - Credit. Students will apply what they have learnt in the pass tasks to new problems with less guidance. Students will be able to implement object-oriented designs and propose changes to designs to meet changing requirements. - Distinction. Students will apply their advanced knowledge to design and build solutions to more challenging real-world scenarios. Students will be able to design and develop object-oriented programs to ill-defined problems. - High Distinction. Students will extend their understanding to demonstrate greater technical ability, more complex solution structures, advanced algorithms, or in other ways exceed the expectations of the unit.
This task assesses your achievement of these Unit Learning Outcome(s)	<p>The portfolio must demonstrate that you have achieved all unit learning outcomes by proving evidence and self-reflection against each outcome.</p> <p>ULO1 - Evaluate simple program code for correct use of coding conventions, and use code tracing and debugging techniques to identify and correct issues.</p> <p>ULO2 - Apply and explain the principles of object-oriented programming including abstraction, encapsulation, inheritance and polymorphism.</p> <p>ULO3 - Implement, and test small object-oriented programs that conform to planned system structures and requirements.</p> <p>ULO4 - Design, communicate, and evaluate solution structures using appropriate diagrams and textual descriptions.</p> <p>ULO5 - Justify meeting specified outcomes through providing relevant evidence and critiquing the quality of that evidence against given criteria.</p>
This task assesses your achievement of these Graduate Learning Outcome(s)	<p>GLO1 – Develop programming skills using modern programming languages and paradigms.</p> <p>GLO2 – Develop outcome and artefact written communication skills.</p> <p>GLO4 – Develop critical analysis skills comparing software designs quality.</p> <p>GLO5 – Develop problem solving skills by iteratively and incrementally realising requirements into software program.</p> <p>GLO6 – through student ability to reflect on their learning to determine areas of growth and areas that still require development.</p>

How and when you will receive feedback on your work	<p>You will be required to work on and submit tasks for feedback each week. The teaching team will then review your progress and provide you with individual face-to-face feedback during your practical session to assist you in completing the tasks and achieving your target grade for the unit.</p> <p>To ensure that there is sufficient time to staff to provide feedback, and to help manage the learning process, tasks will have set target dates and deadlines. The target date is the date that the task is considered to be due, however, as this may require additional fixes in order to incorporate feedback provided, the work can be resubmitted up to the deadline. Work submitted after the deadline will be checked in the portfolio, and additional formative feedback will not be provided.</p>
When and how to submit your work	<p>At the end of the unit you will use OnTrack to combine together the artefacts you have created and a learning summary report into a single portfolio for assessment by the end of Week 12.</p> <p>The due date is 8:00 pm (AEDT) Friday 9 October 2020 (end of Week 12).</p>

Your learning experiences in this Unit - and your expected commitment

To be successful in this unit, you must:

- Read all materials in preparation for your classes or seminars, and follow up each with further study and research on the topic;
- Start your assessment tasks well ahead of the due date;
- Read or listen to all feedback carefully, and use it in your future work;
- Attend and engage in all timetabled learning experiences as follows:

Scheduled learning activities - campus

1 x 2 hour live streamed class per week, weekly drop in support sessions.

Scheduled learning activities - cloud

1 x 2 hour live streamed class per week, weekly drop in support sessions

Students will on average spend 150 hours studying this unit. This includes engaging in online learning activities, assessment activities, readings and study time. Students are expected to complete all allocated learning and assessment tasks for each week and actively engage in discussions with other students and their teaching staff. This unit requires students to complete milestones as they progress through the unit. This requirement is to ensure that students engage with teaching staff throughout the unit.

This foundation unit has been designed to provide all students with a high level of interaction and feedback from teaching staff as a strategy to support student success. It includes the following resources organised for 10 weeks of learning activities to enable you achieve the unit learning outcomes:

In the unit site:

- Videos introducing unit content
- Links to readings and associated texts
- Recordings of seminars

- Web links
- Discussion forum
- In OnTrack
- Task sheets
- Task resources, as required
- Individual feedback
- Alignment of tasks to unit learning outcomes
- Visualisations of your progress to help keep you on track

Your work in this unit starts on Day 1 of the trimester. You are expected to complete the prescribed readings, watch the concept and demonstration videos, and complete unit tasks in OnTrack. As you complete the tasks, you will be able to collect evidence for justifying how you have met the unit learning outcomes through your portfolio. The process of developing your portfolio is simple and easy, so keep that in mind as you read the assessment instructions below.

In order to understand how assessment in this unit works, let's consider standard assessment practices. A typical unit has assignments and tests that you submit and get marks for. The problem is, you only get one chance to succeed, and any marks you lose are gone. This focuses your attention on marks, rather than on working to achieve good learning outcomes. To focus your attention on learning in this unit, we avoid having marks for tasks during the unit and instead assess your final work to see how well you have achieved the outcomes at the end of the unit. This is the summative assessment at the end of the unit, where your grade is determined by the evidence you present in your portfolio. We will work with you by providing formative feedback for these tasks as you submit them week by week. When you submit a task, we will review your work and provide you with feedback. Where your work does not correctly demonstrate the required outcomes, we will give you feedback to help enhance your learning and improve your work for your final portfolio submission. You then need to fix and resubmit the work, so we can check it again and sign it off as Complete when you have achieved the required standard.

We will keep track of all of this in OnTrack, which is where you submit work, receive feedback, resubmit it, and then finally see it signed off as Complete. The process for you is then just a matter of working through the required tasks week by week, and work with us to make sure they are ready for your final portfolio submission.

So, learning in this unit is as simple as setting your target grade, and completing the unit tasks associated with that grade in OnTrack. The teaching team will work with you in providing weekly feedback so that you can achieve the goals you set, demonstrate your ability to complete the unit tasks and discuss your performance with confidence.

Note

At Deakin,

- *Lectures* are referred to as *classes* (definition: a general meeting for all students, for which students do not need to register and where students are engaged through presentations and learning activities)
- *Tutorials, workshops and seminars* are referred to as seminars (definition: more interactive meetings for smaller groups of students).
- For the complete list of agreed definitions for learning experiences, see the [Course Design and Delivery Procedure](#).

UNIT LEARNING RESOURCES

Your unit learning resources are available in your unit site accessed in DeakinSync.

The texts and reading list for the unit can be found on the University Library via the link below: [SIT232](#) Note: Select the relevant trimester reading list. Please note that a future teaching period's reading list may not be available until a month prior to the start of that teaching period so you may wish to use the relevant trimester's prior year reading list as a guide only.

Essential learning resources

Learning resources for this unit will be available to students via the unit site. These resources include:

- audiovisual material;
- readings from: [C# Programming](#); [Beginning C# 2010](#); [C# Programming guide](#)
- software for download and use for learning in the unit - assessment tasks in OnTrack (<https://ontrack.deakin.edu.au/>)
- discussion on the unit site

KEY DATES FOR THIS TRIMESTER

Trimester begins (classes begin)	Monday 13 July 2020
Intra-trimester break (a short break during trimester)	Monday 17 August - Sunday 23 August 2020
Trimester ends (classes cease)	Friday 2 October 2020
Study period (examination preparation period)	Monday 5 October - Friday 9 October 2020
Examinations begin	Monday 12 October 2020
Examinations end	Friday 23 October 2020
Inter-trimester break (the period between trimesters)	Monday 26 October - Friday 6 November 2020
Unit results released	Friday 6 November 2020 (6pm)

UNIT WEEKLY ACTIVITIES

Week	Commencing	Topic	Assessment activity
1	13 July 2020	Unit Intro, Programming Languages, Introduction to Object Oriented Programming	
2	20 July	Abstract Data Type, Classes, Encapsulation, Information Hiding	Submit week 1 tasks.
3	27 July	Control Flow, Error handling, Scopes, Arrays and Lists	Submit week 2 tasks. Week 1 Tasks must be marked as Complete
4	3 August	Inheritance, Polymorphism, Abstract Classes, Interfaces	Submit week 3 tasks. Week 2 Tasks must be marked as complete
5#	10 August	Object Oriented Analysis with UML	Submit week 4 tasks. Week 3 tasks must be marked as complete.
6	24 August	Object Oriented Design Principles	Submit week 5 tasks. Week 4 tasks must be marked as complete.
7	31 August	Design Patterns	Submit week 6 tasks. Week 5 tasks must be marked as complete.
8	7 September	Delegates and Callback Functions	Submit week 7 tasks. Week 6 tasks must be marked as complete.

9	14 September	Extension Methods. Anonymous Types and Methods. Lambda Expressions, Generics	Submit week 8 tasks. Week 7 tasks must be marked as complete.
10	21 September	Guest Lecture	Week 8 tasks must be marked as complete
11	28 September	Final course review and portfolio preparation	Prepare portfolio for submission
12	5 October		Learning Portfolio due

#Intra-trimester break: **Monday 17 August - Sunday 23 August 2020** (between weeks 5 and 6)