THE INDEPENDENT
INSTITUTE OF
EDUCATION

PROGRAMMING 3B
PROG7312
MODULE OUTLINE 2022
(First Edition: 2019)

THE INDEPENDENT
INSTITUTE OF
EDUCATION

Table of Contents

Introduction

In PROG6212, you have learned the skills required to solve the functional requirements of business systems. In PROG7311, you have learned about the technologies and methodologies used to develop large enterprise systems.

In this module, you will learn to apply some advanced C# and programming concepts. The skills learned in this module can be used for almost any kind of programming task – from enterprise software to game development to mobile applications and everything in between. You will learn about recursion, advanced data structures, algorithms, pointers, generics and operator overloading.

Success can be ensured by practicing new knowledge and applying the more theoretical parts of this module to real-life situations.
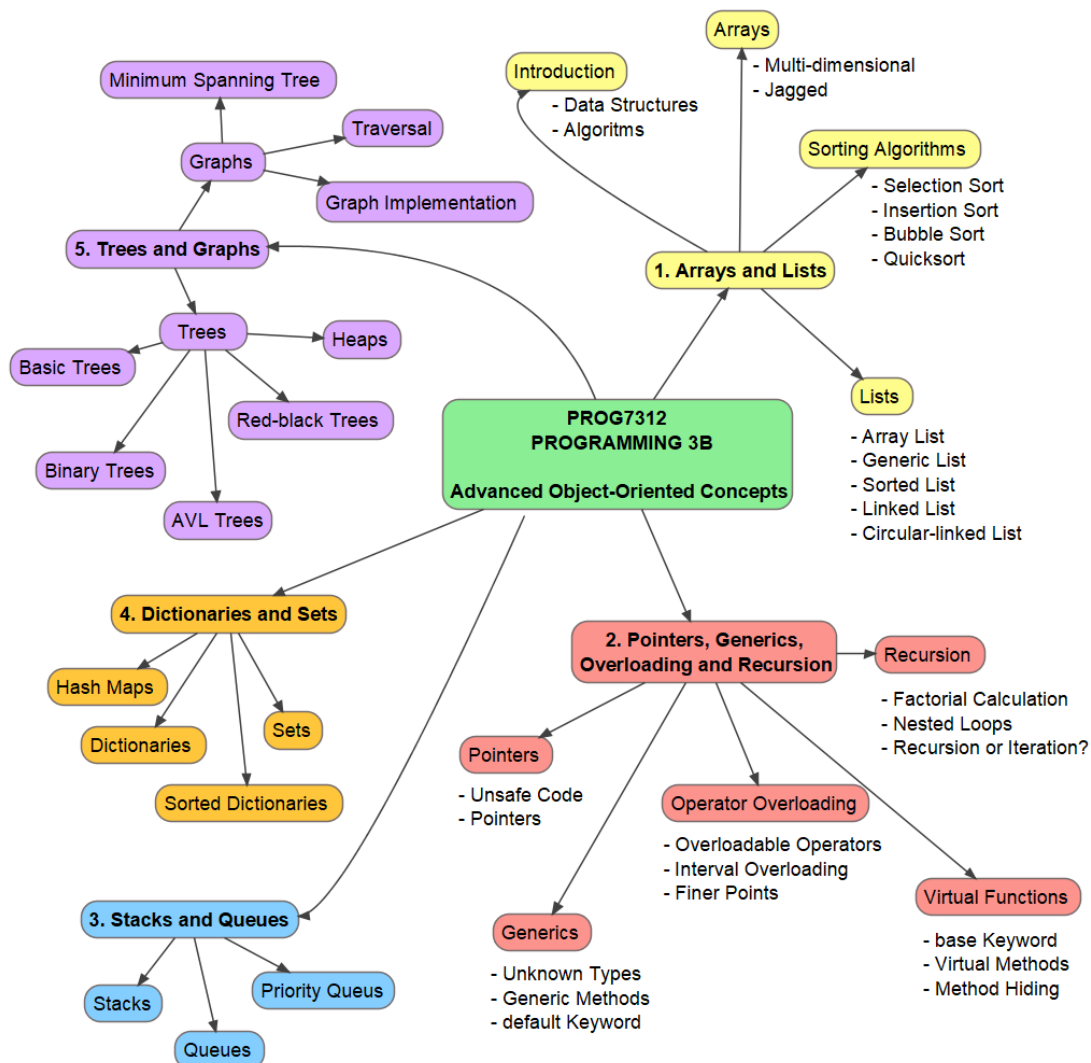
Using this Module Outline

This module outline has been developed to **support your learning**. Please note that the content of this module is on Learn as well as in the prescribed material. You will not succeed in this module if you focus on this document alone.

- This document does not reflect all the content on Learn, the links to different resources, nor the specific instructions for the group and individual activities.
- Your lecturer will decide when activities are available/open for submission and when these submissions or contributions are due. Ensure that you take note of announcements made during lectures and/or posted within Learn in this regard.

This Module on Learn

Learn is an online space, designed to support and maximise your learning in an active manner. Its main purpose is to **guide and pace** you through the module. In addition to the information provided in this document, you will find the following when you access Learn:

- A list of prescribed material;
- A variety of additional online resources (articles, videos, audio, interactive graphics, etc.) in each learning unit that will further help to explain theoretical concepts;
- Critical questions to guide you through the module's objectives;
- Collaborative and individual activities (all of which are gradable) with time-on-task estimates to assist you in managing your time around these;
- Revision questions, or references to revision questions, after each learning unit.

Kindly note:

- Unless you are completing this as a distance module, Learn does **not** replace your contact time with your lecturers and/or tutors.
- PROG7312 is a Learn module, and as such, you are required to engage extensively with the content on the Learn platform. Effective use of this tool will provide you with opportunities to discuss, debate, and consolidate your understanding of the content presented in this module.
- You are expected to work through the learning units on Learn in your own time – especially before class. Any contact sessions will therefore be used to raise and address any questions or interesting points with your lecturer, and **not** to cover every aspect of this module.
- Your lecturer will communicate **submission dates** for specific activities in class and/or on Learn.

Icons Used in this Document and on Learn

The following icons are used in all your modules on Learn:

Icon	Description
Objectives	A list of what you should be able to do after working through the learning unit.
Prescribed Work	Specific references to sections in the prescribed work.
ThinkAbout	Questions to help you recognise or think about theoretical concepts to be covered.
Active Learning	Sections where you get to grapple with the content/ theory. This is mainly presented in the form of questions which focus your attention and are aimed at helping you to understand the content better. You will be presented with online resources to work through (in addition to the textbook or manual references) and find some of the answers to the questions posed.
Connect the dots	Opportunities to make connections between different chunks of theory in the module or to real life.
That is life!	Real life or world of work information or examples of application of theory, using online resources for self-exploration.
REMEMBER: You need to log onto Learn to: • Access online resources such as articles, interactive graphics, explanations, video clips, etc. which will assist you in mastering the content; and • View instructions and submit or post your contributions to individual or group activities which are managed and tracked on Learn.	

Module Resources

Prescribed Material (PM) for this Module	Jamro, M. 2018. *C# Data Structures and Algorithms*. Packt Publishing. Specific chapters and sections from the following freely available e-book are also prescribed: Nakov, S. and Kolev, V. e. a., 2013. *Fundamentals of Computer Programming with C#*. Sofia: [ebook] Available at: https://www.free-ebooks.net/computer-internet/Fundamentals-of-Computer-Programming-with-C/pdf?dl&preview [Accessed 20 July 2022].
Recommended Readings, Digital and Web Resources	Please note that a number of additional resources and links to resources are provided throughout this module on the Learn platform. You are encouraged to engage with these as they will assist you in mastering the various objectives of this module. They may also be useful resources for completing any assignments. However, you will not be assessed under examination conditions on any additional or recommended reading material. *The following titles include information related to this module and may be consulted as additional resources. Please note, however, that you will not be tested on any content from these titles.* • Troelsen, A and Japikse, P. 2017. *Pro C# 7. With .NET and .NET Core*. 8th ed. Apress. • Peter Wentworth, 2015. *Think Sharply with C#.* [Online] Available at: http://www.ict.ru.ac.za/Resources/ThinkSharply/ThinkSharply/index.html [Accessed 20 July 2022]. The Internet is a valuable resource for all programming students as it provides up-to-date developments on the language and tools as the language and Microsoft.NET IDE get updated.
Software required	Microsoft Visual Studio 2019
System Requirements	7 GB – .iso is provided for the students on the FTP Server for downloading Run on Host Computer – Standalone Machine

Lab configuration settings	Microsoft Visual Studio 2019
Module Overview	You will find an overview of this module on Learn under the *Module Information* link in the Course Menu.
Assessments	Find more information on this module's assessments in this document and on the Student Portal.

Module Purpose

The purpose of this module is to cover advanced object-oriented programming topics such as data structures, pointers, overloading of operators, templates and recursion to create advanced programming solutions.

Module Outcomes

MO1	Demonstrate knowledge and understanding of recursion.
MO2	Create object-oriented programs using advanced data structures.
MO3	Write programs using dynamic arrays, pointers, virtual functions, templates and overloading of operators.

Assessments

Integrated Curriculum Engagement (ICE)	
Minimum number of ICE activities to complete	4
Weighting towards the final module mark	10%

Formative 1	Part 1
Weighting	25%
Duration	15 hours
Total marks	100
Open/Closed book	Open book
Resources required	• Prescribed textbooks; • Microsoft Visual Studio 2019 (C#); • Access to the Internet.
Learning Units covered	1-2
Period	3

Formative 2	Part 2
Weighting	30%
Duration	15 hours
Total marks	100
Open/Closed book	Open book
Resources required	• Prescribed textbooks; • Microsoft Visual Studio 2019 (C#); • Access to the Internet.
Learning Units covered	1-4
Period	6

Summative	POE
Weighting	35%
Duration	15 hours
Total marks	100
Open/Closed book	Open book
Resources required	• Prescribed textbooks; • Microsoft Visual Studio 2019 (C#); • Access to the Internet.
Learning Units covered	All

Assessment Preparation Guidelines

Format of the Assessment	Preparation Hints
POE	
The POE will assess all learning units in this module and will be application-type questions. The POE is composed of Task 1 and Task 2.	• Ensure that you work through all the activities, exercises and revision questions on Learn and in your module manual. • Make sure that you are comfortable in responding to all the objectives for all learning units. • Brainstorm possible questions based on the learning outcomes and objectives provided.

Module Pacer			
Code	Programme	Contact Sessions	Credits
PROG7312	BCAD3	64 60 lecture hours + 4 Learn hours	15
Learning Unit 1	Arrays and Lists		

Overview:

In this learning unit, you will learn the fundamentals of data structures and algorithms. With this background in place, you will then learn how to use different kinds of arrays. You will learn how different sorting algorithms work. And you will learn how and when to use the different types of list structures.

The learning unit relates to MO3.

If you are a contact student, you will likely spend 10 sessions on this learning unit.

Please work through Themes 1, 2, 3 and 4 on Learn, together with the relevant sections of your prescribed source/s. To ensure that you are working towards mastering the objectives for this learning unit, please also ensure that you complete the following activities on Learn:

Activity		Estimated Time on Task
1.2.1	Use multi-dimensional arrays to do vector algebra	2 hours
1.3.1	Investigate the performance of List<T>.Sort	2 hours
1.4.1	Use a circular linked list to store the notes in a looping song	2 hours
Estimated total time for this learning unit's activities		*6 hours*

Learning Unit 1: Theme Breakdown		
Sessions: 1-10	Theme 1: Introduction to Data Structures and Algorithms	Prescribed Material
	LO1: Compare the different types of data structures. LO2: Explain algorithmic complexity.	Nakov Chapter 16 & 19
	Theme 2: Arrays	Jamro Chapter 2
	LO3: Use multi-dimensional arrays. LO4: Use jagged arrays.	
	Theme 3: Sorting Algorithms	Jamro Chapter 2
	LO5: Compare different sorting algorithms.	
	Theme 4: Lists	Jamro Chapter 2
	LO6: Compare different types of lists. LO7: Use lists to solve programming problems.	

Learning Unit 2	Pointers, Generics, Overloading and Recursion

Overview:

In this learning unit, you will learn more detail about some of the advanced object-oriented features in C#. Some of the concepts will not be new, but more details will be provided here. You will learn about pointers, virtual methods, generics, operator overloading and recursion.

The learning unit relates to MO1 and MO3.

If you are a contact student, you will likely spend 15 sessions on this learning unit.

Please work through Themes 1, 2, 3, 4 and 5 on Learn, together with the relevant sections of your prescribed source/s. To ensure that you are working towards mastering the objectives for this learning unit, please also ensure that you complete the following activities on Learn:

Activity		Estimated Time on Task
2.1.1	Use pointers to pass data to a native DLL	3 hours
2.2.1	Use virtual methods to override character behaviour	2 hours
2.3.1	Write generic code that can move any simulation object	3 hours
2.4.1	Overload operators to make moving a simulation object around easier	3 hours
2.5.1	Use recursion to check whether a string is a palindrome	4 hours
Estimated total time for this learning unit's activities		*15 hours*

Learning Unit 2: Theme Breakdown		
Sessions: 11-25	**Theme 1: Pointers**	**Prescribed Material**
	LO1: Explain unsafe code. LO2: Use pointers to access data.	"Unsafe Code and Pointers" from the *C# Programming Guide*, online at https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/unsafe-code-pointers/ [Accessed 20 July 2022].
	Theme 2: Virtual Functions	Nakov Chapter 20, sections: • The "base" Keyword; • Constructors with Inheritance; • Virtual Methods; • Virtual Methods and Methods Hiding; • The Difference between Virtual and Non-Virtual Methods.
	LO3: Design applications using virtual methods. LO4: Explain method hiding.	
	Theme 3: Generics	Navkov Chapter 14 Generics
	LO5: Design applications using generics.	
	Theme 4: Operator Overloading	"Overloaded Operators", online at: http://people.cs.aau.dk/~normark/oop-csharp/html/notes/more-operations_themes-operator-sect.html [Accessed 20 July 2022].
	LO6: Explain the purpose of operator overloading. LO7: Use operator overloading to create applications.	
	Theme 5: Recursion	Nakov Chapter 10
	LO8: Decide when to use recursion instead of iteration. LO9: Use recursion to solve algorithmic problems	

Learning Unit 3	Stacks and Queues

Overview:

In this learning unit, you will learn about stacks (first-in-last-out (FILO) data structure) and queues (first-in-first-out (FIFO) data structure) and their applications.

The learning unit relates to MO2.

If you are a contact student, you will likely spend 7 sessions on this learning unit.

Please work through Themes 1, 2 and 3 on Learn, together with the relevant sections of your prescribed source/s. To ensure that you are working towards mastering the objectives for this learning unit, please also ensure that you complete the following activities on Learn:

Activity		Estimated Time on Task
3.1.1	Implement a stack that stores object transforms	4 hours
3.2.1	Implement a queue for multiple users connecting to a server	3 hours
3.3.1	Change the user queue to have high-priority users	2 hours
Estimated total time for this learning unit's activities		*9 hours*

Learning Unit 3: Theme Breakdown		
Sessions: 26 - 32	**Theme 1: Stacks**	Prescribed Material
	LO1: Explain what a stack is. LO2: Use a stack to solve programming problems.	Jamro Chapter 3
	Theme 2: Queues	Jamro Chapter 3
	LO3: Compare queues to stacks. LO4: Use a queue to solve programming problems.	
	Theme 3: Priority Queues	Jamro Chapter 3
	LO5: Compare a queue and a priority queue. LO6: Use a priority queue to solve programming problems.	

Learning Unit 4	Dictionaries and Sets

Overview:

In this learning unit, you will learn about data structures that are optimised for accessing information by keys.

The learning unit relates to MO2.

If you are a contact student, you will likely spend 10 sessions on this learning unit.

Please work through Themes 1, 2, 3 and 4 on Learn, together with the relevant sections of your prescribed source/s. To ensure that you are working towards mastering the objectives for this learning unit, please also ensure that you complete the following activities on Learn:

Activity		Estimated Time on Task
4.1.1	Use a hash table to quickly find simulation elements by their IDs	3 hours
4.2.1	Use a dictionary to quickly find a user's profile picture	3 hours
4.3.1	Use a sorted dictionary to store a high score list	3 hours
4.4.1	Use a set to store the goals for a level	2 hours
Estimated total time for this learning unit's activities		*11 hours*

Learning Unit 4: Theme Breakdown		
Sessions: 33 - 42	**Theme 1: Hash Tables**	Prescribed Material
	LO1: Explain the purpose of a hash table. LO2: Use a hash table to develop an application.	Jamro Chapter 4
	Theme 2: Dictionaries	Jamro Chapter 4
	LO3: Compare dictionaries and hash tables. LO4: Use a dictionary to develop an application.	
	Theme 3: Sorted Dictionaries	Jamro Chapter 4
	LO5: Explain sorted dictionaries.	
	Theme 4: Sets	Jamro Chapter 4
	LO6: Use hash sets to develop an application. LO7: Use sorted sets to develop an application.	

Learning Unit 5	Trees and Graphs

Overview:

In this learning unit you will learn about trees and graphs and their respective applications.

The learning unit relates to MO2.

If you are a contact student, you will likely spend 18 sessions on this learning unit.

Please work through Themes 1, 2, 3, 4, 5, 6, 7, 8 and 9 on Learn, together with the relevant sections of your prescribed source/s. To ensure that you are working towards mastering the objectives for this learning unit, please also ensure that you complete the following activities on Learn:

Activity		Estimated Time on Task
5.1.1	Use a tree to store a historical family tree	3 hours
5.2.1	Use a binary tree to represent choices in a simulation	4 hours
5.5.1	Use a red-black tree to sort objects according to distance	4 hours
5.8.1	Use graph traversal to create a patrol route	4 hours
5.9.1	Use a minimum spanning tree to calculate an optimal path	4 hours
Estimated total time for this learning unit's activities		*19 hours*

Learning Unit 5: Theme Breakdown		
Sessions: **43 - 60**	Theme 1: Basic Trees	Prescribed Material
	LO1: Design an application using a basic tree.	Jamro Chapter 5
	Theme 2: Binary Trees	Jamro Chapter 5
	LO2: Compare basic trees and binary trees.	
	Theme 3: Binary Search Trees	Jamro Chapter 5
	LO3: Explain the purpose of binary search trees.	
	Theme 4: AVL Trees	Jamro Chapter 5
	LO4: Explain the purpose of an AVL tree.	
	Theme 5: Red-black Trees	Jamro Chapter 5
	LO5: Explain the purpose of a red-black tree.	
	Theme 6: Heaps	Jamro Chapter 5
	LO6: Explain how a heap can be used in a sorting algorithm.	
	Theme 7: Graphs	Jamro Chapter 6
	LO7: Compare graphs and trees. LO8: Compare directed and undirected graphs.	
	Theme 8: Graph Traversal	Jamro Chapter 6
	LO9: Design an application that uses graph traversal.	
	Theme 9: Minimum Spanning Tree	Jamro Chapter 6
	LO10: Explain what a spanning tree is.	

Bibliography

IEEE. 1990. IEEE Standard Glossary of Software Engineering Terminology. New York: [Online]
Available at:
http://www.mit.jyu.fi/ope/kurssit/TIES462/Materiaalit/IEEE_SoftwareEngGlossary.pdf
[Accessed 20 July 2022].

Jamro, M. 2018. C# Data Structures and Algorithms. 1st ed. Birmingham: Packt.
Microsoft, 2018. Dictionary<TKey,TValue> Class. [Online]
Available at: https://docs.microsoft.com/en-
us/dotnet/api/system.collections.generic.dictionary-2?view=netframework-4.7.2
[Accessed 20 July 2022].

Nakov, S. and Kolev, V. e. a. 2013. Fundamentals of Computer Programming with C#. Sofia:
[ebook] Available at: https://www.free-ebooks.net/computer-internet/Fundamentals-of-
Computer-Programming-with-C/pdf?dl&preview [Accessed 20 July 2022].

Scottish Qualifications Authority. 2008. Traversing Linear Structures. [Online]
Available at: https://www.sqa.org.uk/e-learning/LinkedDS02CD/page_46.htm
[Accessed 20 July 2022].

Technopedia. 2018. Generics. [Online]
Available at: https://www.techopedia.com/definition/25616/generics-c-sharp
[Accessed 20 July 2022].

VS, R. 2005. Pointers in C#. [Online]
Available at: https://www.c-sharpcorner.com/article/pointers-in-C-Sharp/
[Accessed 20 July 2022].

Glossary of Key Terms for this Module

Term	Definition
Algorithm	"Any sequence of operations for performing a specific task." (IEEE, 1990)
Array	"An *n*-dimensional ordered set of data items identified by a single name and one or more indices, so that each element of the set is individually accessible." (IEEE, 1990)
Data Structure	"A physical or logical relationship among data elements, designed to support specific data manipulation functions." (IEEE, 1990)
Dictionary	"Represents a collection of keys and values." (Microsoft, 2018)
Generics	"Generics refer to a feature in C# that allows defining a class or method with type as a parameter." (Technopedia, 2018)
Graph	"A diagram or other representation consisting of a finite set of nodes and internode connections called edges or arcs." (IEEE, 1990)
Hash Table	"It allows mapping keys to particular values." (Jamro, 2018)
List	"[A]n ordered sequence (line) of elements." (Nakov & Kolev, 2013)
Operator	"Operators allow processing of primitive data types and objects. They take as an input one or more operands and return some value as a result." (Nakov & Kolev, 2013)
Pointer	"[A] variable that holds the memory address of another type." (VS, 2005)
Queue	"In queues we can add elements only on the back and retrieve elements only at the front." (Nakov & Kolev, 2013)
Recursion	"[A] powerful programming technique in which a method makes a call to itself from within its own method body." (Nakov & Kolev, 2013)
Set	"Sets are collections of unique elements (without any repeating elements inside)." (Nakov & Kolev, 2013)

Term	Definition
Stack	"[W]e could add elements on the top and remove the element, which has been added last, but no the previous ones (the ones that are below it)." (Nakov & Kolev, 2013)
Traversal	"Traversing a linear structure means moving through it sequentially, node by node." (Scottish Qualifications Authority, 2008)
Tree	A tree is a data structure that consists of a set of nodes which are linked to each other in a hierarchy. (Nakov & Kolev, 2013)