PROGRAMMING 2A
PROG6221
MODULE OUTLINE 2024
(First Edition: 2018)

The Independent Institute of Education (Pty) Ltd is registered with the Department of Higher Education and Training as a private higher education institution under the Higher Education Act, 1997 (reg. no. 2007/HE07/002). Company registration number: 1987/004754/07.

Table of Contents

Introduction

This module covers the introduction to C# and the Visual Studio IDE. You have already completed a module with similar content in Java. This module re-enforces your programming logic and introduces you to a different programming language.

The key purpose of the module is to provide a solid foundation in the C# programming language and Microsoft Visual Studio.

The module structure follows the layout of the textbook to ensure that you absorb all the foundation aspects of the language.

The module should be approached by re-enforcing the basics you learnt in your first programming language (Java). You are unfamiliar with Visual Studio IDE as a development environment, so you must practice and become familiar with IDE. For every new topic covered, it is recommended that you spend at least an additional five hours (in class and in your own time) doing examples and exercises.

The key to producing successful developers is in the amount of practice on the basic programming principles. Success can be ensured by doing many varied exercises to practice the new knowledge. Besides the additional exercises provided in this guide, you should complete the exercises at the end of each chapter of the textbook and the many examples and exercises available on the web.

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.
- PROG6221 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
	A list of what you should be able to do after working through the learning unit.
	Specific references to sections in the prescribed work.
	Questions to help you recognise or think about theoretical concepts to be covered.
	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.
	Opportunities to make connections between different chunks of theory in the module or to real 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	Troelsen, A. and Japikse, P. 2021. *Pro C# 9 with .NET 5: Foundational Principles and Practices in Programming*. 10th ed. Apress. ISBN: 978-1-4842-6938-1 (Paperback) or ISBN: 978-1-4842-6939-8 (eBook) Available on Ebscohost at: https://ezproxy.iielearn.ac.za/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=2917701&site=ehost-live&scope=site [Accessed 16 November 2022].
Recommended Readings, Digital, and Web Resources	Please note that several 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. You will not, however, be assessed under examination conditions on any additional or recommended reading material. Doyle, B. 2016. *C# programming: from problem analysis to program design*. 5th ed. Boston, (MA): Cengage Learning. Nakov, S. and Kolev, V. e. a., 2013. *Fundamentals of Computer Programming with C#*. Sofia: [ebook] Available at: http://www.introprogramming.info [Accessed 16 November 2022]. Farrell, J. 2011. *Microsoft Visual C# 2010: an introduction to object-oriented programming.* Mason, (OH): South-Western. Deitel, P.J. and Deitel H.M. *Visual C# 2010: how to program.* Upper Saddle River, (NJ): Pearson Prentice Hall. Bell, D. and Parr, M. 2009. *C# for students*. Harlow: Pearson;
Software required	Microsoft Visual Studio 2022
Software Licence requirements	
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 provide the students with an introduction to a multi-purpose object-oriented programming (OOP) computer programming language. Students are taught the language fundamentals as well as the more advanced OOP development features. An in-depth understanding of predefined structures, objects and classes as well as object-oriented programming techniques are covered.

Module Outcomes

MO1	Demonstrate comprehensive knowledge and understanding of concepts, terms, definitions and data manipulation within an object-oriented programming (OOP) language.
MO2	Use a variety of OOP software application tools and techniques to solve given problems.
MO3	Apply OOP principles to communicate information accurately and effectively with the end user.
MO4	Demonstrate correct OOP code testing and debugging.
MO5	Document and comment on OOP code correctly and efficiently to facilitate code maintenance.
MO6	Write simple Windows applications to communicate information accurately and effectively with the end user.

Assessments

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

Summative	POE
Weighting	90%
Duration	15 hours
Total marks	100
Open/Closed book	Open book
Resources required	• Prescribed textbook; • Microsoft Visual Studio 2022 (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 Part 1 and Part 2. Part 1 (weight = 25%) assesses LU 1 to LU2. Part 2 (weight = 30%) assesses LU1 to LU3. The final completed PoE must be submitted after LU4.	• Ensure that you work through all the activities, exercises, and revision questions on Learn and in your textbook. • Make sure that you are comfortable in responding to all the objectives for all learning units. • Ensure you know how to capture data into arrays and other relevant variables. • Develop a desktop application. • Ensure that you have covered all items listed in the marking rubric in Appendix A of the POE.

Module Pacer			
Code	Programme	Contact Sessions	Credits
PROG6221	BCA2; BCIS2; DIS2	72	15
Learning Unit 1	Introduction to C#		

Overview:

In this learning unit, we will explore the basics of programming in C#. We will start by looking at C# and the .NET Platform. And then we will look at how variables, arrays, methods, and other basic concepts work in C#. This learning unit is intended to build on your knowledge of programming in Java, so here the focus is to learn how C# differs from Java in terms of language features.

Please work through Themes 1 and 2 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 complete the following activities on Learn:

One aspect that you might find challenging is getting used to the differences between C# and Java. Chapters 3 and 4 of the textbook will be helpful in this regard.

Learning Unit 1: Theme Breakdown		
Sessions: 1-10	Theme 1: C# and the .NET Platform	Prescribed Material (PM)
Academic Week: 1-2	LO1: Identify the building blocks of the .NET Platform; LO2: Explain the purpose of the common type system; LO3: Differentiate between assemblies, namespaces, and types; LO4: Explain the platform independent nature of .NET.	PM: Chapter 1
Related Outcomes: MO1 MO5	Theme 2: Core C# Programming LO5: Write a console program that requires user input; LO6: Apply string manipulation to solve a programming problem; LO7: Use implicitly typed variables in a program; LO8: Explain the purpose of a nullable type.	PM: Chapters 3 and 4

Learning Unit 2	Object-Oriented Programming

Overview:

In this learning unit, we will look at object-oriented programming in C#. We will start by looking at encapsulation, then we will move on to inheritance and polymorphism. After that we will look at exception handling, and finally this learning unit will also cover interfaces. These concepts should all be familiar to you from first year, so here we are focussing on how these things work in C#.

Please work through Themes 1 to 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 complete the following activities on Learn:

One aspect that you might find challenging is making use of an interface. Chapter 8 in the textbook will be helpful in this regard.

Learning Unit 2: Theme Breakdown		
Sessions: **11-30**	**Theme 1: Encapsulation**	**Prescribed Material (PM)**
Academic Week: **2-5**	LO1: Identify the access modifiers in C#; LO2: Identify the pillars of object-oriented programming; LO3: Use automatic properties to solve a programming problem.	PM: Chapter 5
Related Outcomes: MO1 MO3	**Theme 2: Inheritance and Polymorphism**	PM: Chapter 6
	LO4: Use inheritance to solve a programming problem; LO5: Explain the casting rules in C#; LO6: Use methods from the System. Object class.	
	Theme 3: Structured Exception Handling	PM: Chapter 7
	LO7: Explain the purpose of exception handling.	
	Theme 4: Interfaces	PM: Chapter 8
	LO8: Differentiate between an interface and an abstract class; LO9: Use an interface to solve a programming problem.	

Learning Unit 3	Advanced C# Programming

Overview:

In this learning unit, we are going to explore more advanced concepts in C#. First, we will look at generic programming - using generic collections, as well as defining our own generic methods, classes, and structures. And then we will look at delegate, events, and lambda expressions.

Please work through Themes 1 and 2 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 complete the activities on Learn.

One aspect that you might find challenging is differentiating between anonymous methods and lambda expressions. Chapter 10 in the textbook will be helpful in this regard.

<table>
<tr><th colspan="3">Learning Unit 3: Theme Breakdown</th></tr>
<tr><th>Sessions:
31-50</th><th>Theme 1: Collections and Generics</th><th>Prescribed Material (PM)</th></tr>
<tr><td>Academic Week:
5-8</td><td>LO1: Differentiate between non-generic and generic collections;
LO2: Use a generic collection to solve a programming problem.</td><td>PM: Chapter 10</td></tr>
<tr><td rowspan="2">Related Outcomes:
MO2
MO3</td><td>Theme 2: Delegates, Events and Lambda Expressions</td><td rowspan="2">PM: Chapter 12</td></tr>
<tr><td>LO3: Differentiate between delegates and events;
LO4: Use delegates to solve a programming problem;
LO5: Use events to solve a programming problem;
LO6: Differentiate between anonymous methods and lambda expressions.</td></tr>
</table>

Learning Unit 4	Windows Presentation Foundation

Overview:

In this learning unit, we will learn how to create graphical user interfaces (GUIs) using Windows Presentation Foundation (WPF). WPF is a framework for creating Windows desktop applications, that illustrates the most important concepts of GUI development. We will start off by looking at the basics of WPF, then we will see how to use controls and layouts together with events and data binding. Then, we will look at graphics rendering services, and finally we will make use of resources, animations, styles, and templates. By the end of this learning unit, you will be able to create a GUI that works well and looks good.

Please work through Themes 1 to 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 complete the activities on Learn.

One aspect that you might find challenging is understanding how XAML is used to create user interfaces. Chapter 24 in the textbook will be helpful in this regard.

Learning Unit 4: Theme Breakdown		
Sessions: **51-72**	**Theme 1: Introduction to the Windows Presentation Foundation**	**Prescribed Material (PM)**
Academic Week: **9-12**	LO1: Explain the purpose of Extensible Application Markup Language; LO2: Use Extensible Application Markup Language to create graphical user interfaces.	PM: Chapter 24
Related Outcomes: MO2 MO4 MO6	**Theme 2: Controls, Layouts, Events, and Data Binding**	PM: Chapter 25
	LO3: Use controls to create a graphical user interface; LO4: Use panels to group controls on a graphical user interface; LO5: Use the Ink Application Programming Interface to solve a programming problem.	
	Theme 3: Graphics Rendering Services	PM: Chapter 26
	LO6: Use graphics rendering services to display graphical views of data.	
	Theme 4: Resources, Animations, Styles, and Templates	PM: Chapter 27
	LO7: Explain the purpose of resources. LO8: Use styles in a user interface; LO9: Use animations to display visualisations of data; LO10: Use a control template to solve a programming problem.	

Glossary of Key Terms for this Module

Term	Definition	*My Notes*
Delegate	"A .NET delegate is an object that points to other methods in your application." (Troelsen & Japikse, 2017: 1x)	
Encapsulation	"How does this language hide an object's internal implementation details and preserve data integrity?" (Troelsen & Japikse, 2017: 184)	
Exception	"Runtime anomalies that are difficult, if not impossible, to account for while programming your application." (Troelsen & Japikse, 2017: 255)	
Generics	"Generics are type safe because they can contain only the type of type you specify." (Troelsen & Japikse, 2017: 337)	
Inheritance	"Inheritance allows you to extend the behavior of a base (or parent) class by inheriting core functionality into the derived subclass (also called a child class)." (Troelsen & Japikse, 2017: 184)	
Interface	"An interface is nothing more than a named set of abstract members." (Troelsen & Japikse, 2017: 283)	
Lambda Expression	"Lambda expressions are nothing more than a concise way to author anonymous methods and ultimately simplify how you work with the .NET delegate type." (Troelsen & Japikse, 2017: 397)	
Namespace	"A namespace is a grouping of semantically related types contained in an assembly or possibly spear across multiple related assemblies." (Troelsen & Japikse, 2017: 22)	
Polymorphism	"How does this language let you treat related objects in a similar way?" (Troelsen & Japikse, 2017: 184)	