Course Outline

School:	Eng. Tech. & Applied Science
Department:	Information and Communication Engineering Technology (ICET)
Course Title:	Programming 3
Course Code:	COMP 212
Course Hours/Credits:	56
Prerequisites:	COMP 123
Co-requisites:	N/A
Eligible for Prior Learning, Assessment and Recognition:	Yes
Originated by:	Programming Languages Group
Creation Date:	Fall 2003
Revised by:	Narendra Pershad, Ilia Nika
Revision Date:	Winter 2017
Current Semester:	Winter 2017
Approved by:	*ppesikan c/o*

Chairperson/Dean

Students are expected to review and understand all areas of the course outline.

Retain this course outline for future transfer credit applications. A fee may be charged for additional copies.

This course outline is available in alternative formats upon request.

Course Description

The goal of this course is to enable students, already proficient in OOP, to build robust and more complex, data-driven Windows applications using the .NET languages. Coursework emphasizes data structures and collections in .NET, GUI programming, and database APIs. Entity Framework will be used to access and manipulate the data stored in various database servers. The language of instruction is C#.

Program Outcomes

Successful completion of this and other courses in the program culminates in the achievement of the Vocational Learning Outcomes (program outcomes) set by the Ministry of Advanced Education and Skills Development in the Program Standard. The VLOs express the learning a student must reliably demonstrate before graduation. To ensure a meaningful learning experience and to better understand how this course and program prepare graduates for success, students are encouraged to review the Program Standard by visiting http://www.tcu.gov.on.ca/pepg/audiences/colleges/progstan/. For apprenticeship-based programs, visit http://www.collegeoftrades.ca/training-standards.

Course Learning Outcomes

The student will reliably demonstrate the ability to:

1. Design, code and test the front end of a .NET application that uses advanced GUI components.
2. Develop database applications using entity framework.
3. Design, code and test a C# application that uses collections and data structures in .NET.
4. Design, code and test a C# application that uses generics and generic collections in .NET.
5. Design, code and test a C# application that uses asynchronous programming.
6. Design, code and test a WPF application.
7. Explain Functional Programming and use LINQ to demonstrate functional programming techniques.
8. Design, code and test a C# application that uses custom controls.

Essential Employability Skills (EES)

The student will reliably demonstrate the ability to*:

1. Communicate clearly, concisely and correctly in the written, spoken, and visual form that fulfills the purpose and meets the needs of the audience.
3. Execute mathematical operations accurately.
9. Interact with others in groups or teams in ways that contribute to effective working relationships and the achievement of goals.
10. Manage the use of time and other resources to complete projects.
11. Take responsibility for one's own actions, decisions, and consequences.

*There are 11 Essential Employability Skills outcomes as per the Ministry Program Standard. Of these 11 outcomes, the above will be assessed in this course.

Global Citizenship and Equity (GC&E) Outcomes

N/A

Methods of Instruction

Interactive Lecture, demonstrations, Hands-on lab sessions

Text and other Instructional/Learning Materials

Text Book(s):

Paul Deitel and Harvey Deitel. M. Visual C# How to Program, 6/e, Published by Pearson Education, ISBN-10:0-13-650154-8, ISBN-13:978-0-13-650154-0, 2017. Available on Safari IT Books online.

Classroom and Equipment Requirements

Computers with access to the Internet and college storage. Software image in Software engineering labs includes all tools used in this course.

Evaluation Scheme

⇨ Assignment 1: To be submitted at the end of week 2

⇨ Assignment 2: To be submitted at the end of week 4.

⇨ Assignment 3: To be submitted at the end of week 6.

⇨ Assignment 4: To be submitted at the end of week 9.

⇨ Assignment 5: To be submitted at the end of week 11

⇨ Assignment 6: To be submitted at the end of week 12.

⇨ Assignment 7: To be submitted at the end of Week 14.

⇨ Test 1 Hands-on: Test 1 will take place in week 7 and will cover material taught in weeks 1-6.

⇨ Test 2 Hands-On: Test 2 will take place in week 14 and will cover material taught in weeks 7-13.

Evaluation Name	CLO(s)	EES Outcome(s)	GCE Outcome(s)	Weight/100
Assignment 1	1, 8	3, 9, 11		5
Assignment 2	1, 2, 8	1, 3, 10		5
Assignment 3	3	1, 3, 10		15
Assignment 4	3, 4	3, 9, 10, 11		5
Assignment 5	5	3, 10		10
Assignment 6	7	3, 10		5
Assignment 7	6	1, 3, 9, 10		5
Test 1 Hands-on	1, 3, 4	1, 3		25
Test 2 Hands-On	1, 2, 8	1, 11		25
Total				100%

If students are unable to write a test they should immediately contact their professor or program Chair for advice. In exceptional and well documented circumstances (e.g. unforeseen family problems, serious illness, or death of a close family member), students may be able to write a make-up test.

All submitted work may be reviewed for authenticity and originality utilizing Turnitin®. Students who do not wish to have their work submitted to Turnitin® must, by the end of the second week of class, communicate this in writing to the instructor and make mutually agreeable alternate arrangements.

When writing tests, students must be able to produce official College photo identification or they may be refused the right to take the test or test results will be void.

Student Accommodation

Students with permanent or temporary accommodations who require academic accommodations are encouraged to register with the Centre for Students with Disabilities (CSD) located at Ashtonbee (L1-04), Progress (C1-03), Morningside (Rm 190), and Story Arts Campus (Rm 284). Documentation outlining the functional limitations of a disability is required; however, interim accommodations pending receipt of documentation may be possible. This service is free and confidential. For more information, please email csd@centennialcollege.ca.

Use of Dictionaries

- Any dictionary (hard copy or electronic) may be used in regular class work.

Program or School Policies

N/A

Course Policies

N/A

College Policies

Students should familiarize themselves with all College Policies that cover academic matters and student conduct.

All students and employees have the right to study and work in an environment that is free from discrimination and harassment and promotes respect and equity. Centennial policies ensure all incidents of harassment, discrimination, bullying and violence will be addressed and responded to accordingly.

Academic honesty is integral to the learning process and a necessary ingredient of academic integrity. Academic dishonesty includes cheating, plagiarism, and impersonation. All of these occur when the work of others is presented by a student as their own and/or without citing sources of information. Breaches of academic honesty may result in a failing grade on the assignment/course, suspension or expulsion from the college.

For more information on these and other policies, please visit www.centennialcollege.ca/about-centennial/college-overview/college-policies.

Students enrolled in a joint or collaborative program are subject to the partner institution's academic policies.

PLAR Process

This course is eligible for Prior Learning Assessment and Recognition (PLAR). PLAR is a process by which course credit may be granted for past learning acquired through work or other life experiences. The PLAR process involves completing an assessment (portfolio, test, assignment, etc.) that reliably demonstrates achievement of the course learning outcomes. Contact the academic school to obtain information on the PLAR process and the required assessment.

This course outline and its associated weekly topical(s) may not be reproduced, in whole or in part, without the prior permission of Centennial College.

Topical Outline (subject to change):

Week	Topics	Readings/Materials	Weekly Learning Outcome(s)	Instructional Strategies	Evaluation Name	Evaluation Date
1	Course Overview Advanced GUI Programming Delegates and event handling in C#	Chapter 14	Examine event-handling mechanism in C#. Use delegates to implement event handling in .NET, and create dynamic forms by registering Windows controls with their event handlers by code.	Interactive Lecture Demonstration Lab Session		
2	Advanced GUI Controls	Chapter 15	Design and implement complex GUI applications that use the following controls: • MenuStrip • StatusStrip • TabControl • TreeView • ListView • ToolStrip • SplitContainer • FlowPayoutPanel • TableLayoutPanel • OpenFileDialog • FontDialog • ColorDialog	Interactive Lecture Demonstration Lab Session	Assignment 1: Building complex GUI applications	Week 2
3	Visual Inheritance and User-Defined Controls	Chapter 15	Examine visual inheritance Design and implement a user-defined control.	Interactive Lecture Demonstration Lab Session		
4	Introduction to LINQ and the List Collection	Chapter 9	Explain LINQ Query an array of integers using LINQ. Query an array of objects using LINQ. Explain List collection. Query the generic List collection using LINQ.	Interactive Lecture Demonstration Lab Session	Assignment 2: Developing C# applications that utilize user-defined controls.	Week 4
5&6	Entity Framework	Chapter 22 and lecture handout	Examine Entity Framework. Use an ADO.NET Entity Data Model to create classes for interacting with a database via LINQ entities. Develop GUI applications that query a database using LINQ, provide data binding, navigation capabilities, and display the results in various .NET data controls. Use DataGridView and create	Interactive Lecture Demonstration Lab Session	Assignment 3: Developing Entity Framework database applications	Week 6

Week	Topics	Readings/Materials	Weekly Learning Outcome(s)	Instructional Strategies	Evaluation Name	Evaluation Date
			Use DataGridView and create Master/Detail views that enable user to select a record and display its details.			
7-8	Introduction to Data Structures	Chapter 19	Explain basic data structures such as linkedlists, stacks and queues. Create linked data structures using references, self-referential classes and recursion. Create and manipulate dynamic data structures, such as linked lists, queues, stacks and binary trees. Create reusable data structures with classes, inheritance and composition.	Interactive Lecture Demonstration	Test 1	
9-10	Generics and Generic Collections	Chapter 20, 21	Define generics. Use generics to maximize code reuse, type safety, and performance. Manipulate arrays with class Array's static methods. Iterate through a collection with enumerators. Create and work with generic dictionaries. Create and work with generic linklists. Create and work with generic queues. Create and work with generic stacks.	Interactive Lecture Demonstration Lab Session	Assignment 4: Developing C# applications that utilize custom data structures	Week 9
11	Functional Programming with LINQ	Chapter 21	Explain Functional Programming. Use lambda expressions to create anonymous methods and refer to those methods via delegate variables. Use LINQ method-call syntax and lambdas to demonstrate functional programming techniques. Parallelize LINQ operations with PLINQ for multicore performance.	Interactive Lecture Demonstration Lab Session	Assignment 5: Developing C# applications that utilize Generic Collections.	Week 11
12	Asynchronous Programming in C#	Chapter 23	Examine asynchronous programming and how it can improve the performance of applications. Use async and await to implement asynchronous calls. Use Task Parallel Library feature to take advantage of multicore processors. Determine the performance improvement of	Interactive Lecture Demonstration Lab Session	Assignment 6: Developing C# Applications that utilize Asynchronous programming	Week 12

Week	Topics	Readings/Materials	Weekly Learning Outcome(s)	Instructional Strategies	Evaluation Name	Evaluation Date
			multiple tasks. Call a web service asynchronously.			
13 & 14	GUI with Windows Presentation Foundation	Chapter 32	Explore WPF model. Define a WPF GUI with eXtensible Application Markup Language (XAML). Handle user interface events and use data binding to display data in WPF controls.	Interactive Lecture Demonstration Lab Session	Test 2 Assignment 7: Developing WPF Applications	