PROG32356
.NET Technologies using C#

Sheridan

I: Administrative Information **II: Course Details** **III: Topical Outline(s)** **Open in Word**

Retain during the course and for future use when applying for credit at other educational institutions

Land Acknowledgement

Sheridan College resides on land that has been, and still is, the traditional territory of several Indigenous nations, including the Anishinaabe, the Haudenosaunee Confederacy, the Wendat, and the Mississaugas of the Credit First Nation. We recognize this territory is covered by the Dish with One Spoon treaty and the Two Row Wampum treaty, which emphasize the importance of joint stewardship, peace, and respectful relationships.

As an institution of higher learning Sheridan embraces the critical role that education must play in facilitating real transformational change. We continue our collective efforts to recognize Canada's colonial history and to take steps to meaningful Truth and Reconciliation.

Section I: Administrative Information

Program(s): CST-Analysis & Design, CST-Software Dev & Network Eng, CST-Software Engineering, Computer Programming
Program Coordinator(s): Simon Hood, Ann Cadger, Walid Belal, Satyendra Narayan
Course Leader or Contact: Gursharan Tatla
Version: 20250505_00
Status: Approved (APPR)

Section I Notes: Class plans with evaluation details are provided to students during the first week of class. Fifty percent of the evaluation will be returned to students prior to the commencement of Week 9.

Total hours: 84.0
Credit Value: 6.0
Credit Value Notes: N/A
Effective: Spring/Summer 2025
Prerequisites: (PROG23863 AND DBAS27198)
Corequisites: N/A
Equivalents: N/A
Pre/Co/Equiv Notes: OR PROG24178

Section II: Course Details

Detailed Description

Students are introduced to .NET development using the C# language. They learn to build GUI-based applications, cross-platform applications and web applications. The course covers database programming alongside essential software design principles and object-oriented programming concepts.

Program Context

CST-Analysis & Design **Program Coordinator(s):** Walid Belal
This required course extends the student's previous programming knowledge and skills to include distributed software design and development, using .NET framework technologies and C# programming. This course builds on previous programming and database courses that students have taken.

CST-Software Dev & Network Eng **Program Coordinator(s):** Simon Hood
This required course extends the student's previous programming knowledge and skills to include distributed software design and development, using .NET framework technologies and C# programming. This course builds on previous

programming and database courses that students have taken.

CST-Software Engineering **Program Coordinator(s):** Satyendra Narayan

This required course extends the student's previous programming knowledge and skills to include distributed software design and development, using .NET framework technologies and C# programming. This course builds on previous programming and database courses that students have taken.

Computer Programming **Program Coordinator(s):** Ann Cadger, Satyendra Narayan

This required course extends the student's previous programming knowledge and skills to include distributed software design and development, using .NET framework technologies and C# programming. This course builds on previous programming and database courses that students have taken.

Course Critical Performance and Learning Outcomes

Critical Performance:

By the end of this course, students will have demonstrated the ability to work with the latest .NET technologies to create client-side .NET applications using C# and the ADO.NET, O/RM and Entity frameworks.

Learning Outcomes:

To achieve the critical performance, students will have demonstrated the ability to:

1. Develop .NET applications using the latest .NET technologies and C# language.
2. Apply software design principles and object-oriented mechanisms in software development.
3. Troubleshoot and debug software applications to identify and resolve issues effectively.
4. Demonstrate the effective use of generic data-structures to manage data in .NET applications.
5. Create advanced interactive applications using Microsoft GUI development technologies.
6. Develop cross-platform applications that work on multiple devices and operating systems, ensuring a smooth user experience.
7. Build interactive web applications for modern web development.
8. Construct data-driven applications using Microsoft technologies.
9. Apply Object Relational Mappers to develop database applications.

Evaluation Plan

Students demonstrate their learning in the following ways:

Evaluation Plan: IN-CLASS

Assignments 3 @ 10%	30.0%
Quizzes 4 @ 5%	20.0%
Midterm Exam 1 @ 25%	25.0%
Final Exam 1 @ 25%	25.0%
Total	100.0%

Evaluation Notes and Academic Missed Work Procedure:

To pass the course, students must achieve a 50% weighted average across the tests and the exams and at least 50% overall in the course.

Students must submit/complete all assignments, in-class activities and projects by the scheduled due date and write all tests on the specified date/time. Exceptions will only be made under extraordinary circumstances.

Refer to the School of Applied Computing's Academic Procedures for Evaluations for more details regarding missed work: Procedures for Evaluations

Provincial Context
The course meets the following Ministry of Colleges and Universities requirements:

Essential Employability Skills
Essential Employability Skills emphasized in the course:

- Communication Skills - Communicate clearly, concisely and correctly in the written, spoken, visual form that fulfills the purpose and meets the needs of the audience.
- Communication Skills - Respond to written, spoken, or visual messages in a manner that ensures effective communication.
- Critical Thinking & Problem Solving - Apply a systematic approach to solve problems.
- Critical Thinking & Problem Solving Skills - Use a variety of thinking skills to anticipate and solve problems.
- Personal Skills - Manage the use of time and other resources to complete projects.

Prior Learning Assessment and Recognition
PLAR Contact (if course is PLAR-eligible) - Office of the Registrar
Students may apply to receive credit by demonstrating achievement of the course learning outcomes through previous relevant work/life experience, service, self-study and training on the job. This course is eligible for challenge through the following method(s):

- Challenge Exam
  Notes:
- Interview
  Notes: Both Challenge Exam and Interview are required.

Section III: Topical Outline

Some details of this outline may change as a result of circumstances such as weather cancellations, College and student activities, and class timetabling.
Instruction Mode: In-Class
Professor: Multiple Professors
Resource(s): N/A
Applicable student group(s): Computer Programmer, Computer Systems Technology ? Analysis and Design, Computer Systems Technician - Software Engineering, Computer Systems Technology - Software Development and Network Engineering
Course Details:

Module 1 ? Introduction to .NET Development:
? Introduction to .NET, .NET Core and modern .NET versions
? .NET architecture and its key components such as CLR, FCL, and runtime environments
? Introduction to Visual Studio and VS Code
? Understanding the project structure including solutions, projects, and file organization
? Exploring debugging and troubleshooting techniques to identify and resolve errors
? Introduction to Git for version control and code management
? Basic Git commands such as clone, commit, push, pull, branch
? Using AI-assisted tools such as GitHub Copilot in Visual Studio to help with real-time code suggestions, debugging, and explanations

Module 2 ? C# Fundamentals:
? Overview of data types and type conversion
? Using control structures for decision-making and loops
? Defining and calling methods, method overloading, parameter passing techniques such as

passing by value, passing by reference, output parameters
? Working with classes, interfaces, structures, and enumerations
? Implementing properties for data encapsulation and controlled access to fields
? Exploring object-oriented programming concepts such as inheritance, polymorphism, virtual methods and abstract classes

Assignment #1 (10%)
Quiz #1 (5%)

Module 3 ? .NET Generics and Collections:
? Understanding generic classes and methods
? Working with built-in generic collections such as List, Dictionary, Stack and Queue
? Iterating through collections using loops and enumerators
? Introduction to Language Integrated Query (LINQ)
? Querying and filtering collections using LINQ methods and query syntax

Assignment #2 (10%)

Module 4 ? Introduction to WPF:
? Understanding XAML and its role in UI design
? Exploring layouts using Grid, StackPanel, Canvas, and other containers
? Working with various UI controls, including buttons, textboxes, and list controls
? Handling input events and applying styles
? Managing windows, dialogs, and message boxes

Quiz #2 (5%)
Review
Mid-Term Exam (25%)

Module 5 ? Database Programming:
? Introduction to ADO.NET and its role in data access
? Understanding connected and disconnected data access models
? Introduction to Object-Relational Mapping (O/RM) and Entity Framework
? Exploring Entity Framework Core for modern database interactions
? Working with Code-First and Database-First approaches in EF Core
? Using migrations in EF Core to manage schema changes efficiently

Assignment #3 (10%)

Module 6 ? Web Development with ASP.NET Core and Blazor:
? Introduction to ASP.NET Core and Blazor
? Understanding the differences between Blazor Server and Blazor WebAssembly
? Creating reusable UI components in Blazor
? Implementing navigation, routing, and layout management
? Utilizing data binding and event handling in Blazor
? Fetching and managing data from databases
? Building forms with validation to ensure data integrity

Quiz #3 (5%)

Module 7 ? Cross-Platform App Development:
? Introduction to .NET MAUI and cross-platform development
? Setting up .NET MAUI project
? Understanding UI and XAML
? Working with layouts and navigation
? Accessing platform-specific features
? Using emulators to test cross-platform applications on different devices

Quiz #4 (5%)
Review
Final Exam (25%)

Sheridan Policies

It is recommended that students read the following policies in relation to course outlines:

- **Academic Integrity**
- **Copyright**
- **Intellectual Property**
- **Respectful Behaviour**
- **Accessible Learning**

All Sheridan policies can be viewed on the Sheridan policy website.

Appropriate use of generative Artificial Intelligence tools: In alignment with Sheridan's Academic Integrity Policy, students should consult with their professors and/or refer to evaluation instructions regarding the appropriate use, or prohibition, of generative Artificial Intelligence (AI) tools for coursework. Turnitin AI detection software may be used by faculty members to screen assignment submissions or exams for unauthorized use of artificial intelligence.

Course Outline Changes: The information contained in this Course Outline including but not limited to faculty and program information and course description is subject to change without notice. Nothing in this Course Outline should be viewed as a representation, offer and/or warranty. Students are responsible for reading the Important Notice and Disclaimer which applies to Programs and Courses.