<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Hours</th>
<th>Course Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROG2002</td>
<td>Software Development Capstone Project</td>
<td>60.00</td>
<td>This course provides an opportunity to complete a significant programming project from the design phase through implementation with minimal instructor support. Emphasis is placed on project definition, testing, presentation, and implementation. Upon completion, students should be able to complete a project from the definition phase through implementation. Objectives: Gain practical experience by working on projects, Build up portfolio, Challenge and expand skills.</td>
</tr>
<tr>
<td>PROG2010</td>
<td>Software Development Portfolio Development</td>
<td>60.00</td>
<td>Using the skills developed through the course of their education, students will create a professional portfolio of their work. The resulting portfolio will be presented to prospective employers, as an interactive, graphic representation of their skills and accomplishments. Objectives: Develop materials for a professional quality portfolio, Design and develop packaging that enhances presentation of the portfolio, Demonstrate skills learned by presenting finished portfolio in mock interviews.</td>
</tr>
<tr>
<td>PROG1701</td>
<td>C# Programming I</td>
<td>120.00</td>
<td>This course utilizes the Visual Studio Integrated Development Environment (IDE) to develop and implement Visual C# .NET programs. Topics include data types, control structures, functions, pointers, arrays, and I/O streams. Students learn how to use classes, objects, properties, and methods to program in a Windows OS environment. Students learn about object-oriented encapsulation, inheritance, polymorphism, and database interfacing. Covered topics include how to input, store, and output data in C#.NET. Competencies: • Learn about Object Oriented programming (OOP) • Know the procedures, methods, and functions of C# .NET • Ability to connect to a remote database • Create and use classes • Use Controls and Event Handling procedures • Know and apply structured programming techniques</td>
</tr>
<tr>
<td>PROG2878</td>
<td>C# Programming II</td>
<td>120.00</td>
<td>This course teaches students to develop and implement intermediate level C# .NET programs. Topics include graphical user interfaces GUIs, files and streams, generics, functional programming with LINQ/PLINQ, searching and sorting, Databases using entity framework, Asynchronous programming, and more. Competencies: • Learn Graphical User Interfaces • Understand Generics and their uses • Ability to use functional programming with linq/plinq • Create and use classes • Use and understanding of asynchronous programming • Know and apply search and sorting techniques • Understand and use ADO.NET and the Entity Framework to work with databases</td>
</tr>
<tr>
<td>MDTC2136</td>
<td>JavaScript and jQuery</td>
<td>120.00</td>
<td>This course will demonstrate to students the fundamental concepts of JavaScript and JQuery. Students will develop a useful range of skills by learning the syntax of the JavaScript and JQuery languages. This course covers how client-side scripts interact with server-side programs using the Document Object Model (DOM), control program flow, validate forms, and animate images. At the end of this class, participants will have the knowledge necessary to utilize the power of JavaScript and JQuery to provide dynamic content on their Web sites. Competencies: • Demonstrate JavaScript and JQuery Event concepts • Understand the Document Object Model • Gain knowledge of how and where to find resources for help • Plan and design Web projects using JavaScript and JQuery</td>
</tr>
<tr>
<td>PROG1750</td>
<td>ASP.NET MVC</td>
<td>120.00</td>
<td>The students will learn to create web applications using Microsoft ASP.NET with C#. Course covers ASP.NET Web Forms, ASP.NET databases, validation, Ajax, JQuery, Web API services, and ASP.NET MVC. Students will gain a deep understanding of ASP.NET MVC and will be prepared to build highly interactive web applications. Objectives: Demonstrate an understanding of ASP.NET MVC, Demonstrate knowledge and use of RAZOR engine, Demonstrate understanding and use of ADO.NET to work with databases, Demonstrate and understand Source Control.</td>
</tr>
</tbody>
</table>
### Mobile App Development (480 hours required)

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROG2835</td>
<td>Java Programming II</td>
<td>120.00</td>
</tr>
</tbody>
</table>

This course adds to the skills taught in Java I course. The student will create Java Applets, GUI Components, networking and databases. Also the student will continue to work on concepts that they have learned in the Java I course.

Competencies:
- Learn how to use Structured Programming Techniques
- Learn how to build Java Applets made for World Wide Web Servers
- Learn how to implement Java clients and servers that communicate with one another
- Use a variety of Textbooks and Reference Material in order to solve programming problems

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROG2600</td>
<td>Android App Development</td>
<td>120.00</td>
</tr>
</tbody>
</table>

This course is an introduction to developing applications for Android mobile devices. Students will be using Android Studio and the Android SDK. Students will gain advanced experience in Java and XML as they develop mobile applications.

Competencies:
- Understand layouts, widgets, handle events and themes
- Knowledge of how projects are structured
- Skill using fragments, threads, adapters and intents
- Ability to work with services and notifications
- Understand SQLite databases use
- Show the use of locations and maps in Android
- Ability to deploy apps on Google Play Store

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROG2625</td>
<td>iOS App Development</td>
<td>120.00</td>
</tr>
</tbody>
</table>

This course will provide the student with the tools and skills necessary to develop an iOS App from scratch. By the end of this first course you will be able to demonstrate application programming in Swift. Guided by best practices you will become proficient with syntax, object oriented principles, memory management, functional concepts and more. Create appl in Swift to be ran on an iPhone or iPad.

Competencies:
- Develop an understanding of iOS app development
- Understand the Apple Xcode IDE
- Demonstrate a basic knowledge of Swift
- Develop knowledge of mobile applications
- Create and build iOS apps

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROG2030</td>
<td>Mobile: Cross-Platform Development</td>
<td>120.00</td>
</tr>
</tbody>
</table>

This course is designed to give students an introduction to the cross-platform environment using the Xamarin framework. Students will learn the basics of C# and Xamarin.

Students will build apps using the C# programming language that can run on Android and iOS devices. The course will deal with many different issues of the cross platform development process.

Objectives: Demonstrate understanding of C#, Demonstrate understanding of the cross-platform development process, Demonstrate knowledge of mobile applications, Build apps that run on Android and IOS.

### Embedded Development (480 hours required)

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROG2810</td>
<td>C++ Programming I</td>
<td>120.00</td>
</tr>
</tbody>
</table>

This course is an introduction to the C++ programming language. Topics will include data types, control structures, functions, pointers, arrays, I/O streams, classes, objects, encapsulation, overloading, inheritance and use of these concepts in problem solving. This course introduces the student to procedural programming concepts such as: declaring and using variables, screen input and output, if statements and other decision structures, loops, functions, strings and arrays, pointers, structures, and linked lists.

Competencies:
- Use of structured programming techniques
- Proficient with the C++ Editor and Compiler
- Understand Object Oriented Programming (OOP)
- Write C++ programs that can be used in a business

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROG2815</td>
<td>C++ Programming II</td>
<td>120.00</td>
</tr>
</tbody>
</table>

This course will increase student skill levels in C++ to include files, templates, container classes, introduction to the hierarchical class structure, exception handling and object embedding and linking. The course is designed to further develop a student’s understanding of C++ and the principles of OOP (object-oriented programming).

Competencies:
- Apply advanced programming skills that can be used in business
- Understand the development, compilation, and debugging of complete programs
- Understand scalar types, aggregate types, pointers, and reference types
- Be able to use statements, expressions, functions, and libraries

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROG2050</td>
<td>Python Programming I</td>
<td>60.00</td>
</tr>
</tbody>
</table>

This course utilizes the Visual Studio Integrated Development Environment (IDE) to develop and implement Python programs. Topics include data types, control structures, functions, lists, and I/O streams. Students learn about files and exception handling, Turtle Graphics and structured programming. Covered topics include how to input, store, and output data in Python.
PROG2070  Python Programming II  60.00
This course utilizes the Visual Studio Integrated Development Environment (IDE) to develop and implement Python programs. Topics include strings, dictionaries, classes, lists, and algorithms. Students learn about OOP and instantiation. Covered topics include inheritance, polymorphism, recursion in Python.

PROG1600  Embedded Programming  120.00
The course introduces the student to programming with C/C++ programming language on a microcontroller or in a Linux environment. Students will develop software for embedded microcontrollers, the ARM architecture, and on a Linux system.
Competencies:
• Understand basic microcontrollers and how to use them in embedded system design
• Basic knowledge of the Linux/Unix operating system on a user level
• Understand how to write a wide variety of applications using standard Linux system calls and library functions
• To develop a complete product from design to implementation to debugging
• To gain appreciation for software issues in embedded systems

Additional Electives (0 hours required)

PROG1801  Design Patterns  90.00
This course is designed to give users an understanding of software design patterns. Students will be provided a background of design patterns including good software design principles and practices. They will learn how to create templates to solve problems that can be used in multiple situations. Students will understand how design patterns fit into good software engineering practices. Covered topics include singletons, command, adapter, and state patterns.
Competencies:
• Ability to create repeatable solutions to common design issues
• Understand good architectural design patterns and principles
• Learn the classification of design patterns including: creational patterns, structural patterns, behavioral patterns and architectural patterns
• Understand how design patterns can be applied to the problem solving process
• Create a toolkit of tested solutions to common problems in software design

PROG2101  Intro to Agile  60.00
This course teaches you the basic agile development principles and techniques covering the entire software development process. Students will go through problem conception through development, testing and deployment. They will be able to effectively participate in agile software developments as a result of their successfully completing this course. For an analyst, programmer, tester, usability designer, or project manager, this course gives you the insight and foundation necessary to become a valuable agile team member.
Objectives: Basic Agile principles, Test-driven development, test-first design, and acceptance testing, Refactoring with unit testing, Agile design and design smells.

MDTC2116  PHP and MySQL  120.00
This advanced scripting class teaches techniques necessary to use PHP and MySQL to build professional quality, database-driven web sites. It uses open-source code to work with object oriented programming and incorporate authentication and security.
Competencies:
• Demonstrate an advanced understanding of scripting functions and applicability in Web site development.
• Demonstrate how to use data types, operators, functions, and control structures.
• Learning how to manipulating MySQL databases and PHP.
• Learning how Managing State information, error handling and debugging.

MDTC2125  PHP  120.00
This advanced scripting class teaches techniques necessary to use PHP to build professional quality, database-driven web sites. It uses open-source code to work with object oriented programming and incorporate authentication and security.

MDTC2145  JavaScript II  120.00
This course will teach students a continuation of Java Script

MDTC2415  UI Development  90.00
Students will learn skills in creating a web project that focuses on the fine detail of how users interact with the browser/device. They will develop skills which will enable them to create visual design using a graphical editor, and convert those layouts into useable code responsive to multiple browsers and devices.

PROG2905  Game Programming  60.00
This course covers both the concepts and the implementations necessary to get you started on bringing your own games to life with XNA. It does this by presenting four different games, each game introducing new concepts and techniques to build a solid foundation for your own ideas and creativity.
Competencies:
• Create a puzzle-style game exploring the concepts of game states, recursion, and 2D animation
• Implementation of sound effects and collisions
• Generate tile-based maps and path-finding
• Combine XNA and Windows Forms to create a map editor for a multi-layered tile map engine

PROG2910A  Programming Internship/Externship I  30.00
Prepares students for the Software Development industry by providing professional exposure to the skills and technologies they have studied in the program. This course is intended to be taken near the end of training.
Competencies:
• Student will gain broad, practical, experience directly relating to the coursework they have completed at the college
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROG2910B</td>
<td>Programming Internship/Externship II</td>
<td>30.00</td>
</tr>
</tbody>
</table>
|            | Prepares students for the Software Development industry by providing professional exposure to the skills and technologies they have studied in the program. This course is intended to be taken near the end of training. Competencies:  
|            | • Student will gain broad, practical, experience directly relating to the coursework they have completed at the college |
| PROG2910C  | Programming Internship/Externship III | 30.00   |
|            | Prepares students for the Software Development industry by providing professional exposure to the skills and technologies they have studied in the program. This course is intended to be taken near the end of training. Competencies:  
|            | • Student will gain broad, practical, experience directly relating to the coursework they have completed at the college |
| PROG2910D  | Programming Internship/Externship IV  | 30.00   |
|            | Prepares students for the Software Development industry by providing professional exposure to the skills and technologies they have studied in the program. This course is intended to be taken near the end of training. Competencies:  
|            | • Student will gain broad, practical, experience directly relating to the coursework they have completed at the college |