Course Code/Course Title:

FSW102-CS Programming Foundations

Course Description: The Programming Foundations course is an introduction to basic programming principles as expressed in the elective language. Students will develop a foundational knowledge of programming concepts, algorithms, design patterns and theory. Language-specific concepts will be covered, outlining the nuances associated with the elective language.

Course Length: 40 hours	**Prerequisites:** FSW101	**Proficiency Exam** ☐Yes ☒ No	
Course Start Date: **Course End Date:**	**Meeting Days/Times**		

Required Resources:

Online students are required to have a laptop or desktop and internet access.
Minimum: PC (Windows) or MacBook laptop. 4GB ram, 256GB HD, Core i5
Recommended: PC (Windows) or MacBook laptop. 8GB ram, 256GB SSD, Core i5

This will become your programming rig. Every student will need their own laptop/computer. It is required that you download programming resources to your laptop/computer, and it will also house your coding projects.

The web portal: https://wozu.exeterlms.com has the course material – this is the "textbook" for the course.

Additional Resources:

Students are expected to supply notebooks, pens, pencils, highlighters, folders, ring binders, calculators, USB storage devices and other general supplies as needed to aid in the collection and storage of information in their courses

A. **For Classes Delivered in an Online Format (for approved courses and campuses).** Online courses are delivered via **https://wozu.exeterlms.com** in an asynchronous format. Students enrolled in online courses/programs are expected to spend an equivalent amount of time on task, as campus-based students, in meeting course objectives. For Online Courses the total expected hours required for completion of course objectives are identified on the syllabus as **Total Contact Hours** and reflect the sum of theory, laboratory, and outside hours.

Educational Objectives:

Upon successful completion of this Program, students will be able to:

1. Learn about common programming concepts
2. Learn the basics of iteration, decision making, and code organization
3. Learn the basics of Object Oriented Programming
4. Learn how to handle errors and recover from failure states
5. Learn how to debug applications

Course Outline

Lessons:

Week 1

1. **Introduction to CSharp:** Includes What is C#, Visual Studio Code, extensions, .NET Core SDK, creating a new project, dotnet new command, structure, comments, variables, naming variables, data types, string, int, float, bool, if statement, else statement
2. **Performing Operations:** Includes performing operations, setup, boolean operations, equality, else if statements, inequality, logical operators, and operator, or operator, not operator, hierarchy, less than, greater than, arithmetic operations, string manipulation, concatenation, length, lowercase, uppercase,
3. **Collections and Loops:** Includes setup, Arrays, array indices, .length, loops, while loop, infinite loops, do/while loops, for loop, looping through arrays

Week 2

4. **Methods:** Includes Setup, Method example, parameters, returning different types
5. **Objects and Classes:** Includes object-oriented programming, objects, classes, constructor, constructor overloading, matching variables and parameters, properties, accessors, read only accessors, access modifiers, static classes
6. **Namespaces:** Includes namespaces, Nested namespaces, fully qualified names, using directives, using alias, .NET framework class library

Week 3

7. **Inheritance:** Includes basic inheritance, polymorphism, virtual and override keywords
8. **Interfaces:** Includes Interfaces, inherit from other interfaces, abstract, collections, lists, ArrayList
9. **Algorithms and Debugging:** Includes measuring algorithm performance, Big O notation, Sieve of Eratosthenes algorithm, Euclidean algorithm, Bubble Sort algorithm, Debugging, compile-time error, run-time error, logical error
10. **Final Project**

Outline:

- **L1 Hands On:** Create a page that lets users know if it is sunny outside and tells them to wear sunglasses or not. If it is sunny outside, then let the user know if they should wear sunblock or not. Lastly, if it is not sunny let the user choose if they want to still go outside or not and provide a message based on answers.
- **L2 Practice Hands On:** Create a program which prints the ticket prices for a user. If the user is a senior citizen (i.e., they are 65 years of age or older), they receive a discounted price of $7 per ticket. If they are 12 years of age or younger or a student, they receive a discounted price of $8 per ticket. Otherwise, they will pay $10, and if the value is a negative value it will print "Invalid age".
- **L3 Hands On:** Create a list of names in an array and use a loop to store the values in the array. Then print the values in order and then *reverse* the order.
- **L4 Practice Hands On:** Create a method that doubles and reuses the method to print the results of the operation. Then loop through to print a message including the final number based on if doubled, quadrupled, or doubled many times.

- **L5 Practice Hands On:** Using C#, create a Person class that instantiates 2 objects which will check the persons first name, last name and age and allows modification. If a person inputs a negative age it will set to 0 and prints an "invalid" message.
- **L6 Hands On:** Using namespace create a file that will print a statement to the console. Call each of the methods created from the namespaces within the suggested file.
- **L7 Practice Hands On:** Create a directory that stores employees information including their name, salary, and hire date. If the employee is an engineer also show which school they attended. If the employee is a software engineer do not print the salary information and print out that states this information is private.
- **L8 Hands On:** Create a directory that will return information regarding specific animals. You can choose the sound made by the animal or the top speed the animal can run. Add another area that will let the user know what the animal eats based on their selection and prints out all the animals and the sound they make, how fast they run, and what they eat.
- **L9 Practice Hands On:** For this Hands-On, your goal is to fix all 6 of the errors within the given starter project.
- **L10 Final Project:** Final Project

Final Project:

Using C#, build a game of BlingBlong. The game will have the numbers 1 -100 and will create a specific sequence. The game will print "Bling" if the number is a multiple of three, or "Blong" if the number is a multiple of five. For numbers multiples of three and five the game will print "BlingBlong".