Course Code/Course Title:

FSW102-JV 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:	**Prerequisites:**	**Proficiency Exam**	
40 hours	FSW101	☐Yes ☒ No	
Course Start Date: **Course End Date:**	**Meeting Days/Times**		

Required Resources:

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. We will be downloading programming resources to your laptop, 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 Java:** Includes What is Java, Setup, Eclipse install, Eclipse walkthrough, Data types and variables, string, int, float, boolean, comments, if statements, else clause,
2. **Performing Operations:** Includes boolean operations, equality, string equality, else-if, inequality, And, Or, Not, Hierarchy of operators, Less than, Greater than, arithmetic operations, string manipulation, concatenation, length, lowercase, uppercase
3. **Collections and Loops:** Includes code reusability, arrays, loops, while(), infinite loop, do-while, for

Week 2

4. **Methods:** Includes methods, parameters, type
5. **Objects and Classes:** Includes object-oriented programming, objects, classes, constructor, getters and setters, access modifier, static classes
6. **Java Built-in Classes:** Includes StringBuilder, exception handling, Hashtable, file I/O, FileReader, FileWriter

Week 3

7. **Inheritance:** Includes Inheritance, Polymorphism, Override and Final, Hashcodes, Nested classes
8. **Interfaces:** Includes interfaces, abstract, collections, lists, ArrayList, generic types
9. **Honing Programming:** Includes design patterns, factory, builder, visitor, debugging
10. **Final Project**

Outline:

- **L1 Practice Hands On:** Leverage what you have learned about data types and control flow to create a program which will inform the user to wear sunglasses if it's sunny outside.
- **L2 Hands On:** You should leverage what you have learned about boolean operations to 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, 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 Practice Hands On:** Use the newly added StudentHelper.ReadInputInt() method provided to read in the integer values and store them into the array. Then, use a for loop to store the values in the array. After you read in the user input and store it into the array, print the values in *reverse* order and print the values where the index is odd.
- **L4 Hands On:** Create a double method, then reuse the double method and print the results of the operation.
- **L5 Practice Hands On:** Using Java, create a Person class that has three constructors 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:** You should leverage what you have learned about StringBuilder and loop iteration to create a string composed of the songs in the array provided. Errors should print the stack trace along with the message "Error when writing songs.txt."

- **L7 Practice Hands On:** Create a company directory which holds all the employee's information. The employee's information should include their company name, and years working for the company while the manager's and engineer's profiles should also include their job title.
- **L8 Hands On:** Create a directory that will return information regarding specific animals or reptiles. You can choose the sound made by the animal, the top speed the animal can run, or the favorite food of the animal or reptile.
- **L9 Practice Hands On:** Create a project which contains a factory to produce animals: cat, dog, and cow and gives information about each animal.
- **L10 Final Project:** Final Project

Final Project:

Using Java, create a text-based task manager that will allow the user to add a task, remove a task, mark a task complete, and list the tasks based on the user.