COURSE OUTLINE: CSD226 - FULL-STACK APP. DEV.

Prepared: Computer Studies
Approved: Corey Meunier, Chair, Technology and Skilled Trades

Course Code: Title	CSD226: FULL-STACK APPLICATION DEVELOPMENT
Program Number: Name	2095: COMPUTER PROGRAMMING
Department:	COMPUTER STUDIES
Academic Year:	2023-2024
Course Description:	Modern web applications often require teams of developers to coordinate their efforts across a range of technologies. Students in this course learn how to integrate these technologies into a working system. Approaches to full-stack application development are explored using established development, testing, and deployment strategies, and application design patterns. Topics may include continuous integration and continuous delivery (CI/CD), MVC frameworks, cloud computing and development, and service-oriented architecture (SOA). Students apply their knowledge by building and deploying working web applications and services. Assignments and projects in this course are implemented using the Java platform.
Total Credits:	5
Hours/Week:	5
Total Hours:	70
Prerequisites:	CSD123, CSD213, CSD214
Corequisites:	There are no co-requisites for this course.
Vocational Learning Outcomes (VLO's) addressed in this course: **Please refer to program web page for a complete listing of program outcomes where applicable.**	**2095 - COMPUTER PROGRAMMING** VLO 1 Identify, analyze, develop, implement, verify and document the requirements for a computing environment. VLO 2 Contribute to the diagnostics, troubleshooting, documenting and monitoring of technical problems using appropriate methodologies and tools. VLO 4 Implement robust computing system solutions through validation testing that aligns with industry best practices. VLO 6 Select and apply strategies for personal and professional development to enhance work performance. VLO 7 Apply project management principles and tools when working on projects within a computing environment. VLO 9 Support the analysis and definition of software system specifications based on functional and non-functional requirements. VLO 10 Contribute to the development, documentation, implementation, maintenance and testing of software systems by using industry standard software development methodologies based on defined specifications and existing technologies/frameworks. VLO 11 Apply one or more programming paradigms such as, object-oriented, structured or functional programming, and design principles, as well as documented requirements,

		to the software development process.
	VLO 12	Model, design, implement, and maintain basic data storage solutions.
Essential Employability Skills (EES) addressed in this course:	EES 1	Communicate clearly, concisely and correctly in the written, spoken, and visual form that fulfills the purpose and meets the needs of the audience.
	EES 2	Respond to written, spoken, or visual messages in a manner that ensures effective communication.
	EES 4	Apply a systematic approach to solve problems.
	EES 5	Use a variety of thinking skills to anticipate and solve problems.
	EES 6	Locate, select, organize, and document information using appropriate technology and information systems.
	EES 7	Analyze, evaluate, and apply relevant information from a variety of sources.
	EES 9	Interact with others in groups or teams that contribute to effective working relationships and the achievement of goals.
	EES 10	Manage the use of time and other resources to complete projects.
	EES 11	Take responsibility for ones own actions, decisions, and consequences.
Course Evaluation:	Passing Grade: 50%, D A minimum program GPA of 2.0 or higher where program specific standards exist is required for graduation.	
Other Course Evaluation & Assessment Requirements:	Students are expected to be present to write all tests in class, unless otherwise specified. If a student is unable to write a test due to illness or a legitimate emergency, that student must contact the professor prior to class and provide reasoning. Should the student fail to contact the professor, the student shall receive a grade of zero on the test. If a student is not present 10 minutes after the test begins, the student will be considered absent and will not be given the privilege of writing the test. Students exhibiting academic dishonesty during a test will receive an automatic zero. Please refer to the College Academic Dishonesty Policy for further information. In order to qualify to write a missed test, the student shall have: a.) attended at least 75% of the classes to-date. b.) provide the professor an acceptable explanation for his/her absence. c.) be granted permission by the professor. NOTE: The missed test that has met the above criteria will be an end-of-semester test. Labs / assignments are due on the due-date indicated by the professor. Notice by the professor will be written on the labs / assignments and verbally announced in the class. Labs and assignments that are deemed late will have the following penalty: 1 day late - 10% reduction, 2 days late, 20% reduction, 3 days late, 30% reduction. After 3 days, no late assignments and labs will be accepted. It is the responsibility of the student who has missed a class to contact the professor immediately to obtain the lab / assignment. Students are responsible for doing their own work. Labs / assignments that are handed in and are deemed identical or near identical in content may constitute academic dishonesty and result in a zero grade.	

Students are expected to be present to write in-classroom quizzes. There are no make-up options for missed in-class quizzes.

Students have the right to learn in an environment that is distraction-free, therefore, everyone is expected to arrive on-time in class. Should lectures become distracted due to students walking in late, the professor may deny entry until the 1st break period, which is 50 minutes into the class or until that component of the lecture is complete.

Grade
Definition Grade Point Equivalent
A+ 90 - 100% 4.00
A 80 - 89%
B 70 - 79% 3.00
C 60 - 69% 2.00
D 50 - 59% 1.00
F (Fail) 49% and below 0.00

CR (Credit) Credit for diploma requirements has been awarded.
S Satisfactory achievement in field /clinical placement or non-graded subject area.
U Unsatisfactory achievement in field/clinical placement or non-graded subject area.
X A temporary grade limited to situations with extenuating circumstances giving a student additional time to complete the requirements for a course.
NR Grade not reported to Registrar`s office.
W Student has withdrawn from the course without academic penalty.

Books and Required Resources:

Only reference documentation and free education resources are used in this course

Course Outcomes and Learning Objectives:

Course Outcome 1	**Learning Objectives for Course Outcome 1**
1. Employ continuous integration and continuous delivery (CI/CD) in a team environment	1.1 Explain what CI/CD entails and describe its primary components 1.2 Use a CI/CD pipeline to test and deploy a web application as part of a team of software developers 1.3 Analyze build and test reports and implement resolutions to failed tests
Course Outcome 2	**Learning Objectives for Course Outcome 2**
2. Identify and describe common web application architectures	2.1 Discuss history of web technologies and common web architectures 2.2 Describe the Linux/Apache/MySQL/PHP (LAMP) stack and related technologies 2.3 Explain the Model/View/Controller (MVC) pattern and discuss how it applies to web applications 2.4 Describe and distinguish between monolith and microservice designs
Course Outcome 3	**Learning Objectives for Course Outcome 3**
3. Identify and use the common patterns and components of web applications	3.1 Distinguish front-end and back-end application programming 3.2 Implement secure user authentication, password, and session management 3.3 Use an ORM to connect a web application to a database

	3.4 Explain the role and design of models, views, and controllers in a typical web application 3.5 Create application views to present information to users 3.6 Create secure interactive web forms that allow authenticated and authorized users to manipulate application data 3.7 Use an MVC framework to implement a full-stack web application 3.8 Configure routing in a web application
Course Outcome 4	**Learning Objectives for Course Outcome 4**
4. Identify and mitigate common security threats in web applications	4.1 Prevent SQL injection using prepared statements 4.2 Prevent cross-site scripting using appropriate encoding and decoding of user-supplied data 4.3 Prevent Cross-Site Request Forgery (CSRF) using CSRF tokens
Course Outcome 5	**Learning Objectives for Course Outcome 5**
5. Design and create web services	5.1 Explain what a web API is 5.2 Design and create a JSON web service API 5.3 Implement authentication in a web service
Course Outcome 6	**Learning Objectives for Course Outcome 6**
6. Build cloud-based and serverless web applications	6.1 Define `the cloud` and describe how web applications are deployed in the cloud 6.2 Build and deploy an application on a Platform-as-a-Service (PaaS) 6.3 Describe what serverless functions are 6.4 Explain when serverless functions are appropriate vs PaaS 6.5 Build serverless functions that serve as a web API

Evaluation Process and Grading System:

Evaluation Type	Evaluation Weight
Lab Assignments	40%
Quizzes	10%
Test 1	25%
Test 2	25%

Date: May 31, 2023

Addendum: Please refer to the course outline addendum on the Learning Management System for further information.