

## COURSE SYLLABUS

### CSC11007 - DevOps Fundamentals

#### 1. GENERAL INFORMATION

Course name:	DevOps Fundamentals
Course name (in Vietnamese):	Nhập môn DevOps
Course ID:	
Knowledge block:	
Number of credits:	4
Credit hours for theory:	45
Credit hours for practice:	30
Credit hours for self-study:	90
Prerequisite:	DevOps Fundamentals
Prior course:	Programming 1, Computer Networking, Database, Linux Operating System
Instructors:	

#### 2. COURSE DESCRIPTION

The DevOps Fundamentals course is designed to provide students with a comprehensive understanding of DevOps principles, practices, and methodologies. DevOps, which stands for Development and Operations, is a collaborative approach to software development and IT operations that emphasizes automation, continuous integration, continuous delivery, and close collaboration between development and operations teams. This course serves as a solid foundation for individuals who are new to DevOps and seek to acquire the essential knowledge and skills needed to excel in this dynamic and rapidly evolving field.

#### 3. COURSE GOALS

At the end of the course, students are able to

ID	Description	Program LOs
----	-------------	-------------

G1	Use the specialized English terminology about information technology.	
G2	Explain DevOps culture and the benefits of applying DevOps culture to the software development process.	
G3	Explain the basic concepts of Continuous Integration (CI), Continuous Delivery and Continuous Deployment (CD), and DevOps.	
G4	Explain Git flow, how to use Git for source code management, and integrating it with Jenkins for building a CI/CD pipeline.	
G5	Building the Apps Dockerize using Docker Containers.	

#### 4. COURSE OUTCOMES

CO	Description	I/T/U
G1.1	Use specialized English terminology	T
G1.2	Explain English materials related to lectures	T, U
G2.1	Explain software development methodologies.	T
G2.2	Explain software architectures.	T
G3.1	Explain how to use CI tools.	T, U
G3.2	Explain how to use CD tools.	T, U
G4.1	Explain how to use Git tools.	T, U
G4.2	Explain how to use Jenkins to build CI/CD pipelines.	T, U
G5.1	Explain how to use Docker to set up a local development environment.	T, U

G5.2	Explain how to integrate Docker with DevOps tools.	T, U
------	--	------

## 5. TEACHING PLAN

ID	Topic	Course outcomes	Teaching/Learning Activities (samples)	Assessments
1	Software development methodologies <ul style="list-style-type: none"> <li>● Waterfall</li> <li>● V-Model</li> <li>● Incremental and Iterative Models</li> <li>● Agile</li> </ul>	G1.1, G1.2, G2.1	Lecturing, Demonstration, discussion	
2	Software architecture <ul style="list-style-type: none"> <li>● Client-Server (Front-end, Back-end)</li> <li>● Three-Tier</li> <li>● Model-View-Controller (MVC)</li> <li>● Monolithic</li> <li>● Microservices</li> </ul>	G1.1, G1.2, G2.2	Lecturing, Demonstration, discussion	
3	Components of Software system <ul style="list-style-type: none"> <li>● Database: SQL, NoSQL</li> <li>● Caching: Redis, Memcached</li> <li>● Message queue: RabbitMQ, Apache Kafka</li> <li>● Load Balancer</li> </ul>	G1.1, G1.2, G3.1	Lecturing, Demonstration, discussion	Lab#1
4	Version Control & Git	G1.1, G1.2, G4.1	Lecturing, Demonstration, discussion	Lab#2

5	Test Automation <ul style="list-style-type: none"> <li>● Test Automation Tools: Selenium</li> <li>● Test scripts</li> </ul>	G1.1, G1.2, G3.1	Lecturing, Demonstration, discussion	Lab#3
6	Container platform <ul style="list-style-type: none"> <li>● Containers: Docker, LXC</li> <li>● Docker Engine</li> <li>● Docker Images</li> <li>● Dockerfile</li> <li>● Docker Hub</li> <li>● Container Orchestration</li> <li>● Docker Compose</li> <li>● Docker Networking</li> <li>● Docker Volumes</li> <li>● Docker Registry</li> </ul>	G1.1, G1.2, G5.1	Lecturing, Demonstration, discussion	Lab#4
7	Ansible <ul style="list-style-type: none"> <li>● Inventory</li> <li>● Playbooks</li> <li>● Modules</li> <li>● Roles</li> </ul>	G1.1, G1.2, G3.2	Lecturing, Demonstration, discussion	Lab#5
8	Introduce DevOps	G1.1, G1.2, G3.1, G3.2	Lecturing, Demonstration, discussion	Lab#6
9	CI, CD Pipelines với Jenkins	G1.1, G1.2, G4.2	Lecturing, Demonstration, discussion	Lab#7
10	DevOps Use-Cases	G4.2, G5.2	Demonstration, discussion	Lab#8
11	Review		Q&A, Discussion	

## 6. ASSESSMENTS

ID	Topic	Description	Course outcomes	Ratio (%)
<b>A1</b>	<b>Exercise</b>			<b>60%</b>
<b>A1.1</b>	Lab#1	Practicing based on knowledge taught in class	G1.1, G1.2, G3.1	7.5%
<b>A1.2</b>	Lab#2	Practicing based on knowledge taught in class	G1.1, G1.2, G4.1	7.5%
<b>A1.3</b>	Lab#3	Practicing based on knowledge taught in class	G1.1, G1.2, G3.1	7.5%
<b>A1.4</b>	Lab#4	Practicing based on knowledge taught in class	G1.1, G1.2, G5.1	7.5%
<b>A1.5</b>	Lab#5	Practicing based on knowledge taught in class	G1.1, G1.2, G3.2	7.5%
<b>A1.6</b>	Lab#6	Practicing based on knowledge taught in class	G1.1, G1.2, G3.1, G3.2	7.5%
<b>A1.7</b>	Lab#7	Practicing based on knowledge taught in class	G1.1, G1.2, G4.2	7.5%
<b>A1.8</b>	Lab#8	Practicing based on knowledge taught in class	G4.2, G5.2	7.5%
<b>A2</b>	<b>Final Exam</b>			<b>40%</b>
<b>A2.1</b>	Final Project	This project is designed to strengthen the foundational knowledge gained in DevOps Fundamentals courses and provide practical experience in a DevOps environment	G2.1, G2.2, G3.1, G3.2, G4.1, G4.2, G5.1, G5.2	

## 7. RESOURCES

### ● Textbooks

- [1]. Gene Kim (Author), Patrick Debois (Author), John Willis (Author), Jez Humble (Author), John Allspaw (Foreword) (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*
- [2]. Emily Freeman (Author) (2019). *DevOps For Dummies (1th ed.)*
- [3]. Jez Humble (Author), David Farley (Author) (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (1th ed.)*

### ● Tools

- [4]. Ansible
- [5]. Apache Kafka
- [6]. Docker
- [7]. Jenkins
- [8]. Memcached
- [9]. RabbitMQ
- [10]. Redis
- [11]. Selenium
- [12]. K8s

## 8. GENERAL REGULATIONS & POLICIES

- All students are responsible for reading and following strictly the regulations and policies of the school and university.
- Students who are absent for more than 3 theory sessions are not allowed to take the exams.
- For any kind of cheating and plagiarism, students will be graded 0 for the course. The incident is then submitted to the school and university for further review.
- Students are encouraged to form study groups to discuss the topics. However, individual work must be done and submitted on your own.
- Students prepare lessons, preview documents according to regulations
- Students need to actively interact in online discussion environments

- All online accounts must be registered by student email, using the student-ID and full name, the real avatar in online workspace.
- The number of assignments may vary depending on the classroom situation