College of San Mateo
Official Course Outline

1. **COURSE ID:** CIS 145 **TITLE:** Introduction to DevOps
Units: 3.0 units **Hours/Semester:** 48.0-54.0 Lecture hours; and 96.0-108.0 Homework hours
Method of Grading: Grade Option (Letter Grade or Pass/No Pass)
Recommended Preparation:
CIS 117, and CIS 121

2. **COURSE DESIGNATION:**
Degree Credit
Transfer credit: CSU; UC

3. **COURSE DESCRIPTIONS:**
Catalog Description:
Introduction to the most common DevOps patterns used to develop, deploy and maintain applications. Covers the core principles of the DevOps methodology and examines a number of use cases applicable to startup, small-medium business, and enterprise development scenarios. Tools for configuration and deployment will be used, as well as common techniques for configuring and deploying systems. Also covered are operations, monitoring, testing, security, and Cloud features. Intended for students with previous programming experience.

4. **STUDENT LEARNING OUTCOME(S) (SLO'S):**
Upon successful completion of this course, a student will meet the following outcomes:
   1. Explain the principal concepts and practices behind the DevOps methodology
   2. Design and implement an infrastructure that supports DevOps development projects
   3. Deploy the infrastructure necessary to create development, test, and production environments
   4. Implement basic Security for infrastructure
   5. Practice the core principles of Continuous Integration and Continuous Deployment
   6. Employ DevOps tools and technologies to monitor the application and environment

5. **SPECIFIC INSTRUCTIONAL OBJECTIVES:**
Upon successful completion of this course, a student will be able to:
   1. Explain the principal concepts and practices behind the DevOps methodology
   2. Design and implement an infrastructure that supports DevOps development projects
   3. Deploy the infrastructure necessary to create development, test, and production environments
   4. Implement basic Security for infrastructure
   5. Practice the core principles of Continuous Integration and Continuous Deployment
   6. Employ DevOps tools and technologies to monitor the application and environment

6. **COURSE CONTENT:**
Lecture Content:
1 Introduction to DevOps
- Why DevOps?
- DevOps Perspective
- DevOps and Agile
- Team Structure
- Learning from the DevOps Transformation
    - Expanding Agile practices
    - Leveraging test automation
    - Building a delivery pipeline
    - Experimenting rapidly
    - Continuously improving
    - Looking at the DevOps Results

2 The Cloud as a Platform
- Introduction to the Cloud
- Features of the Cloud
- Choosing a Cloud Service Model for DevOps
- Infrastructure as a Service (IaaS)

  - Amazon Web Services
  - Rackspace
  - Cloud Foundry
- Platform as a Service (PaaS)
  - Amazon Web Services
  - OpenStackPaaS
  - Microsoft Azure
- Software as a Service (SaaS)
  - Amazon Web Services
  - Chef
  - Docker
  - New Relic

3 Operations

- Operations Services
- Service Operation Functions
- Continual Service Improvement
- Operations and DevOps

4 The Deployment Pipeline - Overall Architecture

- Overall Architecture Structure
- Microservice Architecture
- Amazon’s Rules for Teams
- Microservice Adoption for Existing Systems

5 Building and Testing

- The Deployment Pipeline
- Development and Pre-commit Testing
- Build and Integration Testing
- UAT/Staging/Performance Testing
- Tools
- Continuous Testing
- Performance Trending
- Incidents

6 Deployment

- Strategies for Managing a Deployment
- Logical Consistency
- Packaging
- Deploying to Multiple Environments
- Partial Deployment
- Rollback
- Tools
  - Chef
  - Puppet
  - Jenkins
  - Amazon OpsWorks
  - Google Cloud Platform Deployment Manager
  - Others?

7 Monitoring

- What to Monitor
- How to Monitor
- When to Change the Monitoring Configuration
- Interpreting Monitoring Data
- Tools
  - AWS CloudWatch
  - New Relic
  - Zabbix
  - Sensu
  - Others?
- Diagnosing an Anomaly from Monitoring Data

8 Security and Security Audits

- What Is Security?
- Threats

- Resources to Be Protected
- Security Roles and Activities
- Identity Management and Access Control
- Detection, Auditing, and Denial of Service
- Development
- Auditors
- Tools
    - Snort
    - Nessus
    - Wireshark PCAP
    - Others?

9 Other Considerations

- Repeatability
- Performance
- Reliability
- Recoverability
- Interoperability
- Testability
- Modifiability

10 Business Considerations

- Business Case
- Measurements and Compliance to DevOps Practices
- Points of Interaction between Dev and Ops

11 Advanced Topics - Microservices

- Introduction to Microservices
- Microservices Architecture
- Building a Platform for Deploying Microservices
- Development Process
- Container Technologies

7. **REPRESENTATIVE METHODS OF INSTRUCTION:**

Typical methods of instruction may include:

A. Lecture
B. Directed Study
C. Activity
D. Discussion
E. Other (Specify): Instructor will model problem solving techniques; teacher will create and manage an Internet conference for discussion of course topics; and students will work alone and in small groups to solve programming assignments.

8. **REPRESENTATIVE ASSIGNMENTS**

Representative assignments in this course may include, but are not limited to the following:

Writing Assignments:

The primary writing opportunity for students in this course is documentation supporting the various lab projects. This includes both technical documentation and other documentation, including that related to security, monitoring and log files. The technical documentation describes the problem to be solved, the scope of the project, an overview of the solution, and any limitations of the solution. Weekly exercises from the textbook along with continuous development and deployment operations in a cloud-based system will comprise the majority of the assignments. Various open-source tools will be employed for all steps in the DevOps cycle. In addition, students will create several substantial projects using the DevOps methodology.

Reading Assignments:

Weekly textbook readings support all learning objectives.

9. **REPRESENTATIVE METHODS OF EVALUATION**

Representative methods of evaluation may include:

A. Class Performance
B. Class Work
C. Exams/Tests

D. Group Projects
E. Homework
F. Projects
G. Quizzes
H. Bi-weekly quizzes (short answer--from textbook material) to provide feedback to students and teacher. Assessment of student contributions during class discussion and project time. Individual programming assignments. Midterm and Final exams (short answer from textbook material, general problem solving (similar to in-class work), short program segments (similar to programming assignments). Assessment of group participation on course projects, including peer-assessment of participation and contribution to the group effort.

10. **REPRESENTATIVE TEXT(S):**

Possible textbooks include:

A. Forsgren & Humble. *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*, 1st ed. IT Revolution Press, 2018
B. Pinheiro. *Mastering DevOps Automation*, 1st ed. Packt Publishing, 2018
C. Raheja & Borgese. *Effective DevOps with AWS*, 2nd ed. Packt Publishing, 2018
D. Benefield, R. *Lean DevOps: A Practical Guide to On Demand Service Delivery*, 1st ed. Addison-Wesley Professional, 2021

Origination Date: October 2020
Curriculum Committee Approval Date: November 2020
Effective Term: Fall 2021
Course Originator: Melissa Green