



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
KAKINADA – 533 003, Andhra Pradesh, India

DEPARTMENT OF INFORMATION TECHNOLOGY

III Year – I Semester		L	T	P	C
		0	0	4	2
CONTINUOUS INTEGRATION AND CONTINUOUS DELIVERY USING DevOps (Skill Oriented Course)					

Course Outcomes:

At the end of the Course, Student will be able to:

- Understand the why, what and how of DevOps adoption
- Attain literacy on Devops
- Align capabilities required in the team
- Create an automated CICD pipeline using a stack of tools

List of Exercises:

Note: There are online courses indicated in the reference links section. Learners need to go through the contents in order to perform the given exercises

Exercise 1:

Reference course name : Software engineering and Agile software development
Get an understanding of the stages in software development lifecycle, the process models, values and principles of agility and the need for agile software development. This will enable you to work in projects following an agile approach to software development.
Solve the questions given in the reference course name to gauge your understanding of the topic

Exercise 2:

Reference course name: Development & Testing with Agile: Extreme Programming
Get a working knowledge of using extreme automation through XP programming practices of test first development, refactoring and automating test case writing.
Solve the questions in the “Take test” module given in the reference course name to gauge your understanding of the topic

Exercise 3:

Module name : DevOps adoption in projects
It is important to comprehend the need to automate the software development lifecycle stages through DevOps. Gain an understanding of the capabilities required to implement DevOps, continuous integration and continuous delivery practices.
Solve the questions given in Quiz1, Quiz2, Quiz 3

Exercise 4:

Module name :Implementation of CICD with Java and open source stack
Configure the web application and Version control using Git using Git commands and version control operations.

Exercise 5:

Module Name: Implementation of CICD with Java and open source stack
Configure a static code analyzer which will perform static analysis of the web application code and identify the coding practices that are not appropriate. Configure the profiles and dashboard of the static code analysis tool.



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
KAKINADA – 533 003, Andhra Pradesh, India

DEPARTMENT OF INFORMATION TECHNOLOGY

Exercise 6:

Module Name: Implementation of CICD with Java and open source stack

Write a build script to build the application using a build automation tool like Maven. Create a folder structure that will run the build script and invoke the various software development build stages. This script should invoke the static analysis tool and unit test cases and deploy the application to a web application server like Tomcat.

Exercise 7:

Module Name: Implementation of CICD with Java and open source stack

Configure the Jenkins tool with the required paths, path variables, users and pipeline views.

Exercise 8:

Module name: Implementation of CICD with Java and open source stack

Configure the Jenkins pipeline to call the build script jobs and configure to run it whenever there is a change made to an application in the version control system. Make a change to the background color of the landing page of the web application and check if the configured pipeline runs.

Exercise 9:

Module name: Implementation of CICD with Java and open source stack

Create a pipeline view of the Jenkins pipeline used in Exercise 8. Configure it with user defined messages.

Exercise 10 :

Module name: Implementation of CICD with Java and open source stack

In the configured Jenkins pipeline created in Exercise 8 and 9, implement quality gates for static analysis of code.

Exercise 11:

Module name :Implementation of CICD with Java and open source stack

In the configured Jenkins pipeline created in Exercise 8 and 9, implement quality gates for static unit testing.

Exercise 12:

Module name :Course end assessment

In the configured Jenkins pipeline created in Exercise 8 and 9, implement quality gates for code coverage.

Reference Books:

1. Learning Continuous Integration with Jenkins: A beginner's guide to implementing Continuous Integration and Continuous Delivery using Jenkins - Nikhil Pathania ,Packt publication
[\[https://www.amazon.in/Learning-Continuous-Integration-Jenkins-Pathania/dp/1785284835\]](https://www.amazon.in/Learning-Continuous-Integration-Jenkins-Pathania/dp/1785284835)
2. Jenkins 2 – Up and Running: Evolve Your Deployment Pipeline for Next Generation Automation - Brent Laster, O'Reilly publication
[\[https://www.amazon.in/Jenkins-2-Running-Brent-Laster/dp/1491979593\]](https://www.amazon.in/Jenkins-2-Running-Brent-Laster/dp/1491979593)



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
KAKINADA – 533 003, Andhra Pradesh, India

DEPARTMENT OF INFORMATION TECHNOLOGY

Hardware and software configuration:

1. Git [GitHub or Gitlab]
2. Sonarqube
3. Jenkins
4. JUnit
5. Eclipse
6. Tomcat server
7. Maven
8. Cobertura or JaCoCo
9. Java SDK
10. All necessary drivers and jar files for connecting the software
11. Windows machine with 16GB RAM

Web Links: (Courses mapped to Infosys Springboard platform)

1. https://infyspringboard.onwingspan.com/en/app/toc/lex_auth_013382690411003904735_shared/overview [Software Engineering and Agile software development]
2. https://infyspringboard.onwingspan.com/en/viewer/html/lex_auth_01350157819497676810467 [Development & Testing with Agile: Extreme Programming]
3. https://infyspringboard.onwingspan.com/en/viewer/html/lex_auth_01353898917192499226_shared [DevOps CICD]