Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Proposed scheme for Honors in DevOps(Development and Operations)
(Academic Year 2022-2023)

Sr.	Course Code	Course	Teaching Scheme (hrs.)				Continuous Assessment (A) (marks)			Semester End Assessment (B) (marks)					(A+B)	Total Credits
			Th	P	T	Credits	Th	T/W	Total CA (A)	Th / Cb	O	P	O & P	Total SEA (B)		
Sem V																
1	DJ19ITHN1C1	Development Frameworks	4	--	--	4	25	--	25	75	--	--	--	75	100	**4**
Sem VI																
2	DJ19ITHN1C2	DevOps	4	--	--	4	25	--	25	75	--	--	--	75	100	**4**
3	DJ19ITHN1L1	DevOps Lab	--	2	--	1	--	25	25	--	--	--	25	25	25	**1**
Sem VII																
4	DJ19ITHN1C3	MLOps	4	--	--	4	25	--	25	75	--	--	--	75	100	**4**
5	DJ19ITHN1L2	MLOps Lab	--	2	--	1	--	25	25	--	--	--	25	25	25	**1**
Sem VIII																
6	DJ19ITHN1C4	Cloud Engineering	4	--	--	4	25	--	25	75	--	--	--	75	100	**4**
		Total	**16**	**4**	**0**	**18**	**100**	**50**	**150**	**300**	**0**	**0**	**50**	**350**	**450**	**18**

<table>
<tr><td colspan="9"><b>Program: Third Year (Honours in DevOps (Development and Operations))</b></td><td colspan="2"><b>Semester: V</b></td></tr>
<tr><td colspan="9"><b>Course: Development Frameworks</b></td><td colspan="2"><b>Course Code: DJ19ITHN1C1</b></td></tr>
<tr><td colspan="4" rowspan="2"><b>Teaching Scheme
(Hours / week)</b></td><td colspan="7"><b>Evaluation Scheme</b></td></tr>
<tr><td colspan="3"><b>Semester End Examination Marks (A)</b></td><td colspan="3"><b>Continuous Assessment Marks (B)</b></td><td><b>Total Mark (A+B)</b></td></tr>
<tr><td rowspan="2"><b>Lectures</b></td><td rowspan="2"><b>Practical</b></td><td rowspan="2"><b>Tutorial</b></td><td rowspan="2"><b>Total Credits</b></td><td colspan="3"><b>Theory</b></td><td><b>Term Test 1</b></td><td><b>Term Test 2</b></td><td><b>Avg.</b></td><td></td></tr>
<tr><td colspan="3"><b>75</b></td><td><b>25</b></td><td><b>25</b></td><td><b>25</b></td><td><b>100</b></td></tr>
<tr><td rowspan="3"><b>4</b></td><td rowspan="3"><b>-</b></td><td rowspan="3"><b>--</b></td><td rowspan="3"><b>4</b></td><td colspan="3"><b>Laboratory Examination</b></td><td colspan="2"><b>Term work</b></td><td rowspan="2"><b>Total Term work</b></td><td rowspan="3"><b>--</b></td></tr>
<tr><td><b>Oral</b></td><td><b>Practical</b></td><td><b>Oral &amp; Practical</b></td><td><b>Laboratory Work</b></td><td><b>Tutorial / Mini project / presentation/ Journal</b></td></tr>
<tr><td><b>-</b></td><td><b>--</b></td><td><b>--</b></td><td><b>-</b></td><td><b>-</b></td><td><b>-</b></td></tr>
</table>

Pre-requisite:

Knowledge of any programming language and Database Management System

Course Objectives: The objective of this course is to familiarize learners to different development frameworks. The course also introduces students to the principles and process of software engineering and design thinking.

Course Outcomes: On completion of the course, the learner will be able to:

1. Explore various frameworks for application development.
2. Apply software engineering principles for application development.

Detailed Syllabus: (unit wise)		
Unit	**Description**	**Duration**
1	**Introduction to Software Engineering and Process Model:** Software Engineering-process framework, Software Development Life Cycle (SDLC) Process Models: Incremental and Evolutionary models	**06**
2	**Fundamentals of Agile Process:** Need of Agile software development, Agile Manifesto and Principles, Stakeholders and Challenges, Overview of Agile Development Models: Scrum, Extreme Programming, Feature Driven Development, Crystal, Kanban, and Lean Software Development, Methods, Values, Roles, Artifacts, Stakeholders, and challenges. Business benefits of software agility. **Introduction to Scrum:** Agile Scrum Framework, Scrum Artifacts, Meetings, Activities and Roles, Scrum Team Simulation, Scrum Planning Principles, Product and Release Planning, Sprinting: Planning, Execution, Review and Retrospective; User story definition and Characteristics, Acceptance tests and Verifying stories, Burn down chart, Daily scrum, Scrum Case Study.	**10**
3	**Introduction to Architectures:** Introduction to Model View Controller (MVC) Framework: History of MVC, Features of MVC, MVC Architecture, MVC Examples, Popular MVC Frameworks, Advantages and Drawbacks of MVC, 3-Tier Architecture Vs MVC Architecture.	**10**

	The Reactive Manifesto: Introduction, Reactive Principles, Reactive Systems vs Reactive Programming **Clean architecture:** Introduction, The Dependency Rule, A Typical Scenario.	
4	**SOLID Design principles:** Introduction, The Single Responsibility Principle, The Open-Closed Principle, The Liskov Substitution Principle, The Interface Segregation Principle The Dependency Inversion Principle. **Reactive architecture**: Introduction, Design Principles of Reactive Systems, commands and Events, Commands, Events, Messages, Commands Versus Events: An Example Destinations and Space Decoupling, Time Decoupling, The Role of Nonblocking Input/Output, Blocking Network I/O, Threads, and Concurrency, How Does Nonblocking I/O Work? Reactor Pattern and Event Loop, Anatomy of Reactive Applications.	**12**
5	**Core Technologies of Spring Framework:** Introduction to Object oriented programming concept, Spring–Environment Setup, Spring beans and its scopes, Spring bean lifecycle, how to create a bean using Factory Bean? How to create a bean using static Factory Bean? Best Practices of spring Framework, Spring Dependency Injection and Inversion of Controls, Spring Java Configuration vs XML configuration.	**06**
6	**Spring Event Handling and Aspect Oriented Programming (AOP):** Event Handling in Spring, Custom Events in Spring, AOP Concepts, Types of AOP, AOP in Spring, AOP Spring Architecture, Framework Services for AOP, Using @AspectJ-Style Annotations, AspectJ Integration, Spring - Transaction Management, Spring Web MVC Framework, Spring - Logging with Log4J. **Spring Boot:** Introduction to spring boot, spring boot Build systems, spring boot Code structure, Springs and dependency injection, spring boot Runners, Spring Boot - Application Properties	**08**

Books Recommended:

Text Books:

1. Iuliana Cosmina Rob Harrop Chris Schaefer Clarence Ho,” An In-Depth Guide to the Spring Framework and Its Tools”, Apress Fifth Edition,2017.
2. Roger S Pressman, “Software Engineering: A Practitioner’s Approach”, 8th Edition, Mcgraw-Hill, 2015. 2. Ian Sommerville, “Software Engineering”, 9th Edition, Pearson Education, 2011.
3. Clement Escoffier , Ken Finnigan, “Reactive Systems in Java: Resilient, Event-Driven Architecture with Quarkus, 1st Edition, O'Reilly Media, 2021
4. Craig Walls. “Spring Boot in Action” 6th Edition, Manning,2016.

Reference Books:

1. Ashish Sarin J Sharma, “Getting Started with Spring Framework”, Second Edition, Createspace, 2012
2. Rod Johnson et al,” Professional Java Development with the Spring Framework”, John Wiley & Sons 2005.

Program: Final Year B.Tech. Information Technology (Honors)	**Semester: V**
Course: DevOps	**Course Code: DJ19ITHN1C2**
Course: DevOps Laboratory	**Course Code: DJ19ITHN1L1**

Teaching Scheme (Hours / week)				**Evaluation Scheme**						
				Semester End Examination Marks (A)			**Continuous Assessment Marks (B)**			**Total Mark (A+B)**
Lectures	**Practical**	**Tutorial**	**Total Credits**	**Theory**			**Term Test 1**	**Term Test 2**	**Avg.**	
				75			**25**	**25**	**25**	**100**
				Laboratory Examination			**Term work**		**Total Term work**	
4	**-**	**--**	**4**	**Oral**	**Practical**	**Oral & Practical**	**Laboratory Work**	**Tutorial / Mini project / presentation/ Journal**		**50**
				25	**--**		**15**	**10**	**25**	

Pre-requisite:

1. Knowledge of Linux Operating system, installation and configuration of services and command line basics.
2. Basics of Computer Networks and Software
3. Software Development Life cycle.

Course Objectives: The objective of this course is to understand the fundamentals of DevOps engineering and be fully proficient with DevOps terminologies, concepts, benefits, and deployment options to meet real world software development requirements.

Course Outcomes: On completion of the course, learner will be able to:

1. Apply DevOps principles to meet software development requirements.

Detailed Syllabus: (unit wise)		
Unit	**Description**	**Duration**
1	**Introduction to DevOps:** Phases of Software Lifecycle, Minimum Viable Product (MVP) & Cross-functional Teams, Lean, ITIL, Agile development methodologies, DevOps as a prominent culture to achieve agility in the software development process, History of DevOps, DevOps Stakeholders, Goals, Important terminology, DevOps and Agile, DevOps Tools, Configuration management, Continuous Integration and Deployment (CI/CD), DevOps application delivery.	**10**
2	**DevOps Principles and Practices:** 7 C's of DevOps Lifecycle for Business Agility, DevOps and Continuous Testing, How to Choose Right DevOps Tools, Challenges with DevOps Implementation, Must Do Things for DevOps, Mapping My App to DevOps - Assessment, Definition, Implementation, Measure and Feedback.	**10**
3	**Version Control:** Introduction, Overview of Version Control Systems, Role of Version Control System, Types of Control Systems and their Supporting Tools, Importance of version control in CICD pipeline.	**06**

4	**Continuous Integration:** Introduction to Jenkins (With Master –Slave Architecture), Choosing a launch method, Administering Jenkins slaves, Labels, groups and load balancing. Creating Views and Jobs in Jenkins: The Jenkins user interface, Jobs in Jenkins, Creating Views, Managing Views and Jobs in Jenkins: Managing Views in Jenkins, Navigating a job's project page, Job Execution, The Job Execution Configuration Panel, The Status Panel, Console Panel. **Continuous Deployment:** Overview of Docker, Benefits of Docker Workflow, Process Simplification, Architecture, Docker Containers, Docker Workflow, Anatomy of Dockerfile,Building an Image, Running an Image, Custom base Images, Storing Images.	**12**
5	**Continuous Testing** Introducing WebDriver and WebElements, Selenium Testing Tools, Differences between Selenium 2 and Selenium 3, Setting up a project in Eclipse with Maven and TestNG using Java, WebElements, Locating WebElements using WebDriver, Interacting with WebElements, Different Available WebDrivers, Using Java 8 Features with Selenium. Introducing Java 8 Stream API, Using Stream API with Selenium WebDriver.	**06**
6	**Continuous Management:** Overview of Infrastructure as a code, Benefits of Infrastructure as Code, The Four Key Metrics, Three Core Practices for Infrastructure as Code, The Parts of an Infrastructure System, Infrastructure Platforms, Infrastructure Resources, Compute Resources, Storage Resources, Network Resources **Puppet** Architecture, The Puppet Server, setting up the Puppet Agent, Performance Optimizations, Completing the stack with PuppetDB,The PuppetCA **Ansible** Ansible Architecture, Ansible and Infrastructure Management, Local Infrastructure Development: Ansible and Vagrant.	**08**

Suggested list of Laboratory Experiments (Tools):

1. To understand Version Control System / Source Code Management, install git and create a GitHub account.
2. To Perform various GIT operations on local and Remote repositories using GIT Cheat-Sheet.
3. To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job.
4. To Build the pipeline of jobs using Maven / Gradle / Ant in Jenkins, create a pipeline script to Test and deploy an application over the tomcat server.
5. To understand Jenkins Master-Slave Architecture and scale your Jenkins standalone implementation by implementing slave nodes.
6. To Setup and Run Selenium Tests in Jenkins Using Maven.
7. To understand Docker Architecture and Container Life Cycle, install Docker and execute docker commands to manage images and interact with containers.
8. To learn Dockerfile instructions, build an image for a sample web application using Dockerfile.
9. To install and Configure Pull based Software Configuration Management and provisioning tools using Puppet. /Ansible.
10. To learn Software Configuration Management and provisioning using Puppet Blocks (Manifest, Modules, Classes, Function).
11. To provision a LAMP/MEAN Stack using Puppet Manifest.

Any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.

Books Recommended

Textbooks

1. Mitesh Soni, “DevOps Bootcamp”, Packt Publishing Ltd, 2017.
2. Karl Matthias & Sean P. Kane, “Docker: Up and Running”, 3rd Edition, O'Reilly Publication, 2022.
3. Len Bass, Ingo Weber, Liming Zhu, DevOps, “A Software Architects Perspective”, Addison Wesley-Pearson Publication, 2015.
4. John Ferguson Smart, “Jenkins, The Definitive Guide”, 1st Edition, O'Reilly Publication,2011.
5. Ryan Russell-Yates, "Mastering Puppet 5: Optimize enterprise-grade environment performance with Puppet”, 1st Edition, Packt Publishing, 2018.
6. Jonathan McAllister, “Master Jenkins”, Packt Publishing, 2015.
7. Deepak Gaikwad, Viral Thakkar, “DevOps Tools from Practitioner's Viewpoint”, Wiley,2019.
8. Umesh Gundecha and Satya Avasarala, “Selenium Web Driver 3 Practical Guide”, 2nd Edition, Packt Publishing, 2014.
9. Jeff Geerling, “Ansible for DevOps”, 12th Edition, Midwestern Mac, LLC, 2015.
10. Mikael Krief, “Infrastructure as Code”, 1st Edition, O'Reilly Publication, 2016.
11. Stephane Jourdan, Pierre Pomès ,“ Infrastructure as Code (IAC) Cookbook”, 2nd Edition, Packt Publishing, 2017.
12. Martin Alfke , Felix Frank, “Puppet 5 Essentials", 3rd Edition, O'Reilly Publication,2017.

References:

1. Sanjeev Sharma and Bernie Coyne, “ DevOps for Dummies”, 3rd Edition, Wiley Publication,2017.
2. Httermann, Michael, “DevOps for Developers”, 1st Edition, APress Publication, 2012.
3. Joakim Verona, “Practical DevOps”, 2nd Edition Packt publication,2018.
4. Martin Alfke, “ Puppet 5 Essentials - Third Edition: A fast-paced guide to automating your infrastructure”, 3rd Revised Edition, Packt Publishing, 2017.

Online References:

1. Continuous Delivery Using Build Pipelines with Jenkins and Ant”, https://www.methodsandtools.com/archive/archive.php?id=121.(accessed on Dec.16, 2022).
2. DevOps Bootcamp – SYBGEN, DevOps Bootcamp - SYBGEN, (accessed on Dec.17, 2022).

Prepared by Checked by Head of the Department Principal

Program: Final Year B.Tech. Information Technology (Honors)	**Semester: VII**
Course: MLOps	**Course Code: DJ19ITHN1C3**
Course: MLOps Laboratory	**Course Code: DJ19ITHN1L2**

Teaching Scheme (Hours / week)				**Evaluation Scheme**						
				Semester End Examination Marks (A)			**Continuous Assessment Marks (B)**			**Total Mark (A+B)**
Lectures	**Practical**	**Tutorial**	**Total Credits**	**Theory**			**Term Test 1**	**Term Test 2**	**Avg.**	
				75			25	25	25	100
				Laboratory Examination			**Term work**		**Total Term work**	50
4	2	--	5	**Oral**	**Practical**	**Oral & Practical**	**Laboratory Work**	**Tutorial / Mini project / presentation/ Journal**		150
				25	--	--	15	10	25	

Pre-requisite:

1. Knowledge of Linux Operating system, installation and configuration of services and command line basics,
2. Basics of Machine Learning
3. Knowledge Development Life cycle, development frameworks and DevOps

Course Objectives: The objective of this course is to understand the fundamentals of MLOps and its significance in the ML lifecycle. Students will Learn various tools and technologies used in MLOps to design and build scalable ML pipelines. Students will get exposure to deploy ML models. Students will learn techniques for monitoring, debugging, and optimizing ML systems. Finally, students will explore methods for reproducibility, version control, and model governance.

Course Outcomes: On completion of the course, learner will be able to:

1. Automate the deployment of ML models into the core software system or as a service component.

Detailed Syllabus: (unit wise)

Unit	**Description**	**Duration**
1	**Introduction to Machine Learning Operations** Overview of MLOps and its importance, Understanding the challenges in deploying and managing ML models, ML development lifecycle, Role of MLOps in the ML development lifecycle, Introduction to DevOps and its application to ML	**06**
2	**Data Management, Model Development and Training for MLOps** Data versioning and reproducibility, Data preprocessing and feature engineering pipelines	

	Data validation and monitoring, Data quality assurance and governance, Model versioning and tracking, Model training pipelines and automation, Hyperparameter tuning and model selection, Model evaluation and validation techniques	**06**
3	**Model Deployment and Serving, Continuous Integration and Delivery (CI/CD) for ML** Model packaging and containerization (e.g., Docker), Infrastructure provisioning and orchestration (e.g., Kubernetes), Deploying models as scalable services, managing model endpoints and versioning, Version control and collaboration (e.g., Git), Building reproducible ML pipelines, Automated testing and code quality checks, Continuous integration and deployment strategies	**06**
4	**Monitoring and Performance Optimization** Monitoring model performance and behavior, Real-time and batch monitoring techniques, Logging and error tracking in ML systems, Performance optimization and scalability considerations	**06**
5	**Governance and Compliance in MLOps** Data privacy and protection in ML systems, Access control and authentication mechanisms, Security considerations for model deployment, Compliance with industry regulations (e.g., GDPR, HIPAA)	**07**
6	**Model Lifecycle Management and Infrastructure for MLOps** Model versioning and governance, Retraining and revalidation strategies, Model deployment and retirement, Ensuring fairness, transparency, and accountability. **Cloud Platforms and Infrastructure for MLOps** Introduction to cloud platforms (e.g., AWS, Azure, GCP), Deploying ML models on cloud infrastructure, Managing resources and scaling ML workloads, Cost optimization strategies for ML systems.	**08**

Suggested List of Laboratory Experiments:

Any 10 experiments from the below given topics or any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.

1. **Setting up a Version Control System (VCS) for ML Projects:**
    - Experiment with popular VCS tools like Git and create a repository for ML projects.
    - Learn to track code changes, collaborate with team members, and manage different branches.
2. **Creating a Continuous Integration (CI) Pipeline:**
    - Build a CI pipeline using tools like Jenkins, Travis CI, or GitLab CI.
    - Automate the process of building, testing, and validating ML models with each code commit.
3. **Containerization with Docker:**
    - Containerize ML models and their dependencies using Docker.
    - Experiment with Docker images, containers, and Dockerfile configurations.
4. **Orchestrating ML Workflows with Kubernetes:**
    - Deploy ML models as scalable and resilient services using Kubernetes.
    - Experiment with deploying, managing, and scaling ML workloads in Kubernetes clusters.
5. **Model Packaging and Deployment with TensorFlow Serving:**
    - Package trained ML models using TensorFlow Serving.
    - Experiment with deploying and serving models as RESTful APIs.
6. **Experiment Tracking and Management:**

- Use tools like MLflow or Neptune.ai to track experiments, log metrics, and manage model versions.
- Explore features like hyperparameter tuning, model registry, and experiment reproducibility.

7. **Continuous Deployment (CD) for ML Models:**
   - Implement a CD pipeline to automate the deployment of ML models to production.
   - Experiment with different deployment strategies, such as blue-green deployment or canary releases.
8. **Monitoring and Alerting:**
   - Set up monitoring and alerting systems to track model performance, data drift, and anomalies.
   - Experiment with tools like Prometheus, Grafana, or DataDog to visualize and monitor ML system metrics.
9. **Model Performance Optimization:**
   - Explore techniques for optimizing model performance, such as model quantization, pruning, or distillation.
   - Experiment with different optimization approaches and measure their impact on model efficiency.
10. **A/B Testing and Experimentation:**
    - Design and conduct A/B tests to compare the performance of different ML models or algorithms.
    - Experiment with statistical analysis and hypothesis testing to evaluate model improvements.
11. **Model Governance and Compliance**:
    - Understand the importance of model governance and compliance in regulated industries.
    - Experiment with model explainability, bias detection, and fairness assessment techniques.
12. **Infrastructure as Code (IaC) for ML:**
    - Use tools like Terraform or AWS CloudFormation to manage ML infrastructure.
    - Experiment with provisioning and automating the setup of ML environments.
13. **Case Studies and Best Practices**
    - Real-world MLOps case studies
    - Best practices and lessons learned
    - Industry trends and emerging technologies in MLOps
    - Future directions and challenges in the field

Suggested list of tools:

MLOps tools for model development, deployment, and monitoring to standardize, simplify, and streamline the machine learning process.

1. Tools for performing AutoML.[AutoGluon, AutoKeras, AutoPyTorch]
2. Data and Pipeline Versioning Tools [Pachyderm/ Data Version Control (DVC)]
3. Experiment Tracking and Model Metadata Management Tools (MLFlow/ Comnet ML/ Weights & Biases
4. End-to-End MLOps Platforms [AWS SageMaker/ DagsHub/ Kubeflow]
5. CI/CD for Machine Learning (clearML, CML)
6. Orchestration and Workflow Pipelines MLOps Tools [Prefect/ Metaflow/ Kedro]
7. Model Deployment and Serving Tools [TensorFlow Extended (TFX) Serving/ BentoML/ Cortex]
8. Model Testing & Validation [deepchecks/ trubrics]
9. Model Monitoring in Production ML Ops Tools [Evidently/ Fiddler/ Censius AI]

Books Recommended

Textbooks

- Noah Gift , "Practical MLOps: A Guide to Building Real-World Machine Learning Systems", O'Reilly, First Edition, September 2021.
- Mark Treveil, Nicolas Omont, “Introducing MLOps: How to Scale Machine Learning in the Enterprise”, O'Reilly Media, First Edition, January 5, 2021
- Emmanuel Raj, “Engineering MLOps: Rapidly build, test, and manage production-ready machine learning life cycles at scale”, Packt Publishing Limited, 1st edition, 19 April 2021

Reference Books:

- Hannes Hapke and Catherine Nelson, "Building Machine Learning Pipelines: Automating Model Life Cycles with TensorFlow", O’Reilly, First Edition, 19 July 2020.
- Chris Fregly, Antje Barth, "Data Science on AWS: Implementing End-to-End Continuous Machine Learning Pipelines", O’Reilly, First Edition, 9 May 2021.
- Sridhar Alla, Suman Kalyan Adari, “Beginning MLOps with MLFlow: Deploy Models in AWS SageMaker, Google Cloud, and Microsoft Azure”, Apress publication, 1st edition, 8 December 2020

Web Resources

Blogs and Websites:

- MLflow Blog: MLflow is an open-source platform for managing the ML lifecycle. The blog covers topics related to MLOps, model deployment, and reproducibility.
- Towards Data Science: A popular online publication with a dedicated section on MLOps, featuring articles and tutorials on topics like model deployment, monitoring, and CI/CD pipelines.

Online Courses and Tutorials:

- Coursera: "Machine Learning Engineering for Production (MLOps)" by deeplearning.ai. This course provides a comprehensive introduction to MLOps, covering topics like data and model versioning, deployment, monitoring, and more.
- Udacity: "Machine Learning Deployment" by Google Cloud. This course focuses on deploying and scaling machine learning models using Google Cloud technologies and covers MLOps principles.
- YouTube: You can find numerous tutorials and talks on MLOps from conferences and industry experts. Look for channels like TensorFlow, PyTorch, and DevOps-related channels.

<table>
<tr><td colspan="8"><b>Program: Final Year B.Tech. Information Technology</b></td><td colspan="3"><b>Semester: VIII</b></td></tr>
<tr><td colspan="8"><b>Course: Cloud Engineering</b></td><td colspan="3"><b>Course Code: DJ19ITHN1C4</b></td></tr>
<tr><td colspan="4" rowspan="2"><b>Teaching Scheme (Hours / week)</b></td><td colspan="7"><b>Evaluation Scheme</b></td></tr>
<tr><td colspan="3"><b>Semester End Examination Marks (A)</b></td><td colspan="3"><b>Continuous Assessment Marks (B)</b></td><td><b>Total Mark (A+ B)</b></td></tr>
<tr><td rowspan="3"><b>Lectures</b></td><td rowspan="3"><b>Practical</b></td><td rowspan="3"><b>Tutorial</b></td><td rowspan="3"><b>Total Credits</b></td><td colspan="3"><b>Theory</b></td><td><b>Term Test 1</b></td><td><b>Term Test 2</b></td><td><b>Avg.</b></td><td></td></tr>
<tr><td colspan="3"><b>75</b></td><td><b>25</b></td><td><b>25</b></td><td><b>25</b></td><td><b>100</b></td></tr>
<tr><td colspan="3"><b>Laboratory Examination</b></td><td colspan="2"><b>Term work</b></td><td rowspan="2"><b>Total Term work</b></td><td rowspan="3"><b>--</b></td></tr>
<tr><td rowspan="2"><b>4</b></td><td rowspan="2"><b>-</b></td><td rowspan="2"><b>--</b></td><td rowspan="2"><b>4</b></td><td><b>Oral</b></td><td><b>Practical</b></td><td><b>Oral & Practical</b></td><td><b>Laboratory Work</b></td><td><b>Tutorial / Mini project / presentation/ Journal</b></td></tr>
<tr><td><b>-</b></td><td><b>--</b></td><td><b>--</b></td><td><b>-</b></td><td><b>-</b></td><td><b>-</b></td></tr>
</table>

Pre-requisite: Fundamentals of Distributed System.

Course Objectives: The course aims to familiarize students with cloud engineering concepts and principles. The objective of this course is to realize the importance of Cloud Virtualization, Abstractions and Enabling Technologies. This course aims students to understand the hardware, software concepts and architecture of cloud Engineering.

Course Outcomes: On completion of the course learners will be able to:

1. Identify the architecture, infrastructure and delivery models of cloud engineering.
2. Apply suitable virtualization concepts and address the core issues of cloud engineering such as security, privacy and interoperability.

Detailed Syllabus: (unit wise)		
Unit	**Description**	**Duration**
1	**Cloud Engineering Introduction:** Cloud Engineering, Layers and Types of Clouds, Cloud Infrastructure Management, Challenges and Applications. Virtualization: Virtualization of Computing, Storage and Resources. Cloud Services: Introduction to Cloud Services IaaS, PaaS and SaaS	**05**
2	**Software as a Service (SaaS) :** Evolution of SaaS, Challenges of SaaS Paradigm, SaaS Integration Services, SaaS Integration of Products and Platforms. Infrastructure As a Services (IaaS): Introduction, Background & Related Work, Virtual Machines Provisioning and Manageability, Virtual Machine Migration Services, VM Provisioning and Migration in Action. Platform As a service (PaaS): Integration of Private and Public Cloud, Technologies and Tools for Cloud Engineering, Resource Provisioning services	**09**
3	**Abstraction and Virtualization:** Introduction to Virtualization Technologies, Load Balancing and Virtualization, Understanding Hyper visors, Understanding Machine Imaging, Porting Applications, Virtual Machines Provisioning and Manageability Virtual Machine Migration Services, Virtual Machine Provisioning and Migration in Action, Provisioning in the Cloud Context, Virtualization of CPU, Memory, I/O Devices, Virtual Clusters and Resource management, Virtualization for Data Center Automation	**10**

4	**Cloud Infrastructure and Cloud Resource Management:** Architectural Design of Compute and Storage Clouds, Layered Cloud Architecture Development, Design Challenges, Inter Cloud Resource Management, Resource Provisioning and Platform Deployment, Global Exchange of Cloud Resources. Administrating the Clouds, Cloud Management Products, Emerging Cloud Management Standards.	**09**
5	**Security:** Security Overview, Cloud Security Challenges and Risks, Software-as-a Service Security, Cloud Engineering security architecture: Architectural Considerations, General Issues Securing the Cloud, Securing Data, Data Security, Application Security, Virtual Machine Security, Identity and Presence, Identity Management and Access Control, Autonomic Security Establishing Trusted Cloud Engineering, Secure Execution Environments and Communications, Autonomic Security Storage Area Networks, Disaster Recovery in Clouds, Cloud Disaster Recovery.	**09**
6	**Cloud Middleware and Cloud Based Case-Studies:** OpenStack, Eucaluptus, Windows Azure, CloudSim, EyeOs, Aneka, Google App Engine. Overview of Cloud services, Designing Solutions for the Cloud, Implement & Integrate Solutions, Emerging Markets and the Cloud, Tools for Building Private Cloud: IaaS using Eucalyptus, PaaS on IaaS - AppScale	**10**

Books Recommended:

Textbooks:

1. Rajkumar Buyya, James Broberg, Andrzej M Goscinski, “Cloud Computing: Principles and Paradigms”, Wiley publication, 2013.
2. George Reese, “Cloud Application Architectures: Building Applications and Infrastructure in the Cloud”, O'Reilly Publication, 2011.
3. Toby Velte, Anthony Velte, Cloud Computing: A Practical Approach, McGraw-Hill Osborne Media, 2013.
4. John Rhoton, Cloud Computing Explained: Implementation Handbook for Enterprises, Recursive Press, 2009.

Reference Books:

1. Rishabh Sharma: Cloud Computing Fundamentals, Industry Approach and Trends: Wiley Publication,2015.(ISBN: 978-81-265-5306-8)
2. Gautam Shroff, Enterprise Cloud Computing Technology Architecture Applications,2010.[ISBN: 978-0521137355]