Sanjay Ghodawat University, Kolhapur

School of Technology

Department of Computer Science and Engineering

Structure and Contents for Final Year B. Tech. Computer Science and Engineering Program

(AY 2020-21) R0

AME/P/80/0

Structure for B. Tech Final Year Semester VII									
Course Code	**Course Title**	**L**	**T**	**P**	**C**		**Evaluation Scheme**		
						Component	**Exam**	**WT %**	**Min. Pass %**
CST401 (PC | ST) Version: 1.0	Distributed and Parallel Computing	3	-	-	3	Theory 100 Marks	FET	20	40
							CAT I	15	
							CAT II	15	
							ESE	50	40
CST403 (PC | ST) Version: 1.0	Distributed and Parallel Computing Laboratory	-	-	2	1	Practical 50 Marks	FEP	100	40
CST405 (PC | ST) Version: 1.0	Agile Software Development	3	-	-	3	Theory 100 Marks	FET	20	40
							CAT I	15	
							CAT II	15	
							ESE	50	40
CST407 (PC | ST) Version: 1.0	Agile Software Development Laboratory	-	-	2	1	Practical 100 Marks	FEP	50	40
							POE	50	40
CST409 (PC | ST) Version: 1.0	Mobile Application Development	3	-	-	3	Theory 100 Marks	FET	20	40
							CAT I	15	
							CAT II	15	
							ESE	50	40
CST411 (PC | ST) Version: 1.0	Mobile Application Development Laboratory	-	-	4	2	Practical 100 Marks	FEP	50	40
							POE	50	40
CST413_ (PE | ST) Version: 1.0	Program Vertical III	3	-	-	3	Theory 100 Marks	FET	20	40
							CAT I	15	
							CAT II	15	
							ESE	50	40
CST415_ (PE | ST) Version: 1.0	Program Vertical III Laboratory	-	-	2	1	Practical 50 Marks	FEP	100	40
CST417 (PC|ST) Version: 1.0	Software Proficiency Program III	-	-	4	2	Practical 100 Marks	FEP	50	40
							POE	50	40
Total		**12**		**14**	**19**	**Total Hours: 26, Total Credits: 19**			

FET – Faculty Evaluation Theory; FEP - Faculty Evaluation Practical; CAT – Continuous Assessment Test; ESE – End Semester Examination; Au - Audit Course

CST401 : Distributed and Parallel Computing							
Ver. 1.0, Program Core, School of Technology							
Lect.	**Tut.**	**Pract.**	**Credits**	**Evaluation Scheme**			
				Component	**Exam**	**Weightage %**	**Pass%**
3	-	-	3	Theory (100 Marks)	FET	20	40
					CAT I	15	
					CAT II	15	
					ESE	50	40

Prerequisite: Operating systems, Computer Networks

Course Description: This course covers a broad range of topics related to parallel and distributed computing, including parallel and distributed architectures and systems, parallel and distributed programming paradigms, parallel algorithms, and scientific and other applications of parallel and distributed computing.

Course Objective: Students will be able to List architectural elements of modern processors and explain their impact on performance.

Course Outcomes: After the end of this course students will be able to

CO1 **Evaluate**[4] applicability of distributed and parallel systems for various applications.

CO2 **Interpret**[3] requirements used to implement distributed system.

CO3 **Experiment**[3] the synchronization in distributed systems.

CO4 **Summarize**[4] concepts of parallel computing.

Syllabus (Theory)

Units	**Description**	**Hours**
I.	**Introduction** Definition, Goals, Types of distributed systems: Distributed Computing System, Distributed Information System, Architecture: Architectural, Styles, System Architecture	**7**

II.	**Processes** Threads, Threads in distributed system, Virtualization, Clients, Servers, Code Migration.	**7**
III.	**Communication** Fundamentals, Remote Procedure Call, Message Oriented Communication, Stream Oriented Communication ,Multicast communication Synchronization, Logical Clock, Mutual exclusion, Election Algorithms	**7**
IV.	**Synchronization** Clock sysncronization,Logical clocks, Mutual exclusion,Centarlized algorithm, decentralized algorithm, distributed algorithm, token ring algorithm	**7**
V.	**Parallel programming** Paradigms And Programmability – Parallel Programming Models – Shared Memory Programming.	**7**
VI.	**Message passing programming** Message Passing Paradigm – Message Passing Interface – Parallel Virtual Machine.	**7**

Textbooks:

1. Tanenbaum, Steen, *Distributed Systems: Principles and Paradigms*, Pearson Prentice Hall Publications, 2007.
2. Kai Hwang and Zhi. Wei Xu, *Scalable Parallel Computing*, Tata McGraw-Hill Publications, 1998

References :

1. Liu M.L ,*Distributed Computing, Principles and Applications*, Pearson Education Publications, 2004
2. Barry Wilkinson and Michael Allen, *Parallel Programming–Techniques and applications Using Networked Workstations and Parallel Computers*, Prentice Hall Publications, 2005.

CST403 : Distributed and Parallel Computing Laboratory							
Ver.1.0, Program Core, School of Technology							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage %	Pass %
-	-	2	1	Practical (50 Marks)	FEP	100	40

Prerequisite: Programming concepts, JAVA programming, concepts of computer networks

Course Description: This course includes programming Assignments using different programming paradigms, and students will have the opportunity to examine one course topic in depth.

Course Objective: To introduce the advanced concepts of Parallel and Distributed Computing and its implementation for assessment of understanding the course by the students.

Course Outcomes: After the end of this course students will be able to

CO1 **Design**[5] algorithm for distributed system concepts.

CO2 **Demonstrate**[3] the implantation of algorithms in distributed computing environment.

CO3 **Develop**[5] programs to implement concepts of distributed and parallel programming.

Practical

Two hours per week per batch practical is to be utilized for writing programs to ensure that students have properly learnt the topics covered in the lectures. It should comprise of minimum 10-12 experiments. Students of different batches should implement different programs based on the following guidelines-

1. Write a program using java to implement the Remote Procedure call in distributed system.
2. Write a program using java to implement the Remote Invocation Method in distributed system.
3 Demonstrate the concept of mutual exclusion in distributed system. Implement Bully election algorithm for mutual exclusion in distributed systems. For conducting on-line

examination.

4 Demonstrate the concept of mutual exclusion in distributed system and Implement centralized algorithm for mutual exclusion in distributed systems
5. Write a program using java to implement the Berkely algorithm in distributed system.
6. Write a client server program using java to learn about client-server programming, and how distributed Java applications can communicate with each other using sockets.
7. Write a java application l learn how to write distributed applications in the Single Program Multiple Data (SPMD) model, specifically by using the Message Passing Interface (MPI) library.
8. Write an application to learn the fundamentals of task parallelism. Tasks are the most basic unit of parallel programming. An increasing number of programming languages (including Java and C++) are moving from older thread-based approaches to more modern task-based approaches for parallel programming.
9. Write a program to learn about approaches to parallelisms that have been inspired by functional programming.
10. Write a program to learn about approaches to parallelisms that have been inspired by loop parallelism.
11. Implement a program to give you an idea on how to implement parallelism in Java with the Fork/Join Framework.

Textbooks:

1. Jennifer Welch Hagit Attiya, *Distributed Computing: Fundamentals, Simulations and Advanced Topics*, Second Edition, Wiley Publications, 2004.

2. Tanenbaum, Steen, *Distributed Systems: Principles and Paradigms*, Pearson Prentice Hall Publications, 2016

References :

1. Liu M.L, *Distributed Computing, Principles and Applications*, Pearson Education Publications, 2004

2. Barry Wilkinson and Michael Allen, *Parallel Programming – Techniques and applications Using Networked Workstations and Parallel Computers*, Prentice Hall Publications, 2005.

CST405 : Agile Software Development							
Ver.1.0, Program Core | School of Technology							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage %	Pass %
3	-	-	3	Theory (100 Marks)	FET	20	40
					CAT I	15	
					CAT II	15	
					ESE	50	40

Prerequisite: Knowledge of software development, project management

Course Description: This course emphasizes the quick realization of system value through disciplined, iterative, and incremental software development techniques and the elimination of wasteful practices. It will study and cover the full spectrum of agile methods, including Scrum, Extreme Programming, Kanban, Dynamic Systems Development Method, and Feature-Driven Development.

Course Objective:

1. Know the fundamental issues that agile software engineering attempts to address.
2. Analyze the potential pitfalls of agile practices.
3. Discuss the benefits and risks of agile software engineering.
4. Apply agile planning principles to team project.

Course Outcomes: After the end of this course students will be able to

CO1 **Understand**[2] basic concept of agile software development and tools

CO2 **Describe**[2] Scrum framework for project management

CO3 **Illustrate**[2] Agile lifecycle and its effect on testing.

CO4 **Make use of**[3] Agile methodology in software projects.

Syllabus (Theory)

Units	Description	Hours
I.	**Introduction to Agile development** Agile development, Classification of methods, Introduction and background, Agile Manifesto and Principles, Overview of Scrum, Extreme Programming, Feature Driven development, Lean Software Development, Agile project	7

management, Design and development practices in Agile projects, Test Driven Development, Continuous Integration, Refactoring, Pair Programming, Simple Design, User Stories, Agile Testing, Agile Tools

II. Introduction to Scrum

Introduction to Scrum, Project phases, Agile Estimation, Planning game, Product backlog, Sprint backlog, Iteration planning, Initial Stages of Building a Requirement Document, Techniques for Requirements Elicitation, Burn down chart, Sprint planning and retrospective, Daily scrum, Scrum roles – Product Owner, Scrum Master, Scrum Team, Tools for Agile project management **7**

III. User Stories

User story definition, Characteristics and content of user stories, Agile design practices, Role of design Principles including Single Responsibility Principle, Open Closed Principle, Need and significance of Refactoring, Refactoring Techniques, Continuous Integration **7**

IV. Agile lifecycle and its impact on testing

The Agile lifecycle and its impact on testing, Core Testing Concepts, Functional and Non-Functional Testing Integration Testing and System Testing ,User Acceptance Testing and End-to-End Testing The agile alliances, Test-Driven Development (TDD), Testing user stories - acceptance tests and scenarios **7**

V. Test automation

Planning and managing testing cycle, Test automation, Tools to support the Agile tester, Agile testing – Nine principles and six concrete practices for testing on agile teams. **7**

VI. Agile Project management

Market scenario and adoption of Agile, Roles in an Agile project, Agile applicability, Agile in Distributed teams, Business benefits, Challenges in Agile, Risks and Mitigation, Agile projects on Cloud, Balancing Agility with Discipline, Agile rapid development technologies. Selenium Automation **7**

Text Books :

1. Ken Schawber, Mike Beedle, *Agile Software Development with Scrum,* Beedle Publisher: Pearson Publications, 2002
2. Robert C. Martin, *Agile Software Development, Principles, Patterns and Practices,* Publisher: Prentice Hall Publications, 2011

3. Lisa Crispin, Janet Gregory, *Agile Testing: A Practical Guide for Testers and Agile Team*, Addison Wesley Publications, 2009

References :

1. Alistair Cockburn, *Agile Software Development: The Cooperative Game*, Addison Wesley Publications, 2006

2. Mike Cohn ,*User Stories Applied: For Agile Software*, Addison-Wesley Publications, 2004

CST407: Agile Software Development Laboratory							
(Ver.1.0, Program Core | School of Technology)							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage %	Pass %
-	-	2	1	Practical (100 Marks)	FEP	50	40
					POE	50	40

Prerequisite: Knowledge of software development, project management

Course Description: This lab course emphasizes the quick realization of system value through disciplined, iterative, and incremental software development techniques and the elimination of wasteful practices. It will study and cover the full spectrum of agile methods, including Scrum, Extreme Programming, Kanban, Dynamic Systems Development Method, and Feature-Driven Development.

Course Objective: This lab course emphasizes the quick realization of system value through disciplined, iterative, and incremental software development techniques and the elimination of wasteful practices. It will study and cover the full spectrum of agile methods, including Scrum, Extreme Programming, Kanban, Dynamic Systems Development Method, and Feature-Driven Development.

Course Outcomes: After the end of this course students will be able to

CO1 **Explain**[2] Agile Manifesto with Lean & Kanban Method

CO2 **Make use of**[3] the common agile development practices and methods

CO3 **Demonstrate**[4] all stages of an agile software process in a team, to produce working software

Practical

Two hours per week per batch practical is to be utilized for writing to ensure that students have properly learnt the topics covered in the lectures. This shall include extra problem statements and there implementations to strengthen the programming logic. Students of different batches should implement different programs based on following guidelines.

1. Introduction of Agile Manifesto with Lean & Kanban Method.
2. Application of agile methodology characteristic to create an iterative model.
3. Need and importance of Product Backlog for a Project.
4. Create and explain a Agile team structure
5. Demonstrate common Agile Practices using - Retrospective, Backlog Preparation & Backlog refinement.
6. Create a sprint plan for Car rental system.
7. Define and create user stories for a Mobile banking application.
8. Create test plan and test case document for Car rental system.
9. Create a wireframes for online shopping website (login to Add to cart)
10. Use of Agile Application development in miniproject.
11. Create a DFD and use case diagram for any project.

Text Books :

1. Ken Schawber, Mike Beedle, *Agile Software Development with Scrum,* Pearson Publications, 2002.
2. Robert C. Martin, *Agile Software Development, Principles, Patterns and Practices*, Prentice Hall Publications, 2011.
3. Lisa Crispin, Janet Gregory, *Agile Testing: A Practical Guide for Testers and Agile Team*, Addison Wesley Publications, 2009.

References :

1. Alistair Cockburn, *Agile Software Development: The Cooperative Game*, Addison Wesley Publications, 2006.

2. Mike Cohn ,*User Stories Applied: For Agile Software*, Addison-Wesley Publications, 2004.

CST409 : Mobile Application Development							
Ver.1.0, Program Core, School of Technology							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage	Pass%
3	-	-	3	Theory 100 Marks	FET	20	40
					CAT I	15	
					CAT II	15	
					ESE	50	40

Prerequisite: Basics of Operating System, XML, Database Engineering

Course Description: This course will investigate application development for the Android mobile platform. We will look at techniques for building applications that adapt to the ways in which mobile apps differ from traditional desktop or web-based apps, including constrained resources, small screen sizes, varying display resolutions, intermittent network connectivity, specialized sensors, and security restrictions.

Course Objective: To develop problem solving abilities using mobile applications and study procedure to develop applications using Mobile Operating System.

Course Outcomes: After the end of this course students will be able to

CO1 **Compare**[2] mobile application framework and application development environment.
CO2 **Explain**[2] android architecture and the tools for developing android Applications.
CO3 **Select**[3] user interfaces used in android applications.
CO4 **Demonstarte**[3] deployment of android application.

Syllabus (Theory)

Units	Description	Hours
I.	**Android Overview:** Overview of Android, History, Android Versions, Android OS stack: Linux kernel, Native Libraries/DVM, Application Framework, Applications, Activity, Activity lifecycle, Fragments, Activity Back Stack, Process and Threads	7
II.	**Android Development Environment:** Introduction to Android SDK, Android Emulator, Creating a Project, Project Directory Structure, DDMS, Logging in Android (Logcat), Android Manifest File, Permissions	7

III.	**Intents and Layouts:** XML, Android View Hierarchies, Linear Layouts, Relative Layout, Table Layout, Frame Layout Sliding, Using Padding and Margins with Layouts.	**7**
IV.	**Input Controls, Input Events, Dialogs:** Buttons, Text Fields, Checkboxes, Radio Buttons, Toggle Buttons, Spinners, Event Listeners, Event Handlers, Touch Mode, Handling Focus, Dialogs: Alerts, Popup, Toasts	**7**
V.	**Menus, Notification and Action Bar:** Menus, Options menu, Context menu, Popup menu, Handling menu click events, Creating a Notification, Notification actions, Notification priority, Managing Notifications, Removing notifications.	**7**
VI.	**Android Database and App Market:** Installing SQLite plug-in, DbHelper, The Database Schema and Its Creation, Four Major Operations, Cursors, Example, publish app to the Android Market	**7**

Textbooks:

1. Bill Phillips, Chris Stewart, & Kristin Marsicano, *Android Programming: The Big Nerd Ranch Guide*, Third Edition, Big Nerd Ranch Guides Publisher, 2017.
2. Reto Meier, *Professional Android 4 Application Development*, Wiley India Publications, 2012
3. Wallace Jackson, *Android Apps for Absolute Beginners*, Second Edition, Apress Publishers, 2012

References :

1. W.Frank Ableson, Robi Sen, Chris King, C. Enrique Ortiz, *Android in Action*, Third Edition, Manning Publications, 2011

CST411 : Mobile Application Development Laboratory							
Ver.1.0, Program Core, School of Technology							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage %	Pass %
-	-	4	2	Practical 100 Marks	FEP	50	40
					POE	50	40

Prerequisite: Basics of Operating System, XML, Database Engineering

Course Description: In this course, the student will build the code, compile, execute, and debug mobile applications using the Java for Android programming language and Eclipse to develop programs using advanced programming concepts.

Course Objective: To implement different concepts of mobile applications using GUI, Layouts, Event Listener and Database so that student will be able to develop project based on mobile application.

Course Outcomes: After the end of this course students will be able to

CO1 **Analyze**[4] and discover own mobile app for simple needs.

CO2 **Develop**[3] mobile applications using GUI, Layouts, Event Listener and Database.

Practical

Two hours per week per batch practical is to be utilized for writing programs to ensure that students have properly learnt the topics covered in the lectures. It should comprise of minimum 10-12 experiments. Students of different batches should implement different programs based on the following guidelines-

5. Develop an application that draws basic graphical primitives on the screen.
6. Develop an application that makes use of databases.
7. Develop an application that makes use of Notification Manager.
8. Implement Date & Time application that uses Multi-threading.
9. Develop a native application that uses GPS location information.
10. Implement an application that creates an alert upon receiving a message.

11. Write a mobile application that makes use of RSS feed.
12. Develop a mobile application to send an email.
13. Developing & understanding of augmented reality (Group assignment).
14. Comparison of real & virtual environment of Ball.
15. Develop a Mobile application for simple needs (Mini Project).

Textbooks:

1. Bill Phillips, Chris Stewart, & Kristin Marsicano, *Android Programming: The Big Nerd Ranch Guide*, Third Edition, Big Nerd Ranch Guides Publisher, 2017.
2. Reto Meier, *Professional Android 4 Application Development*, Publisher: Wiley India, 2012
3. Wallace Jackson, *Android Apps for Absolute Beginners*, Second Edition, Apress Publishers, 2012

References :

1. W.Frank Ableson, Robi Sen, Chris King, C. Enrique Ortiz, *Android in Action*, Third Edition, Manning Publications, 2011

Sanjay Ghodawat University, Kolhapur

School of Technology

Department of Computer Science and Engineering

Structure and Contents for Final Year B. Tech. Computer Science and Engineering Program

(AY 2020-21) R0

AME/P/80/0

CST417: Software Proficiency Program III							
(Ver.1.0, Program Core, School of Technology)							
Lect	**Tut**	**Pract**	**Credits**	**Evaluation Scheme**			
				Component	**Exam**	**Weightage %**	**Pass %**
-	-	4	2	Practical (100 Marks)	FEP	50	40
					POE	50	40

Pre-requisites: Programming skills

Course Description: Julia is a high-level, high-performance dynamic programming language developed specifically for scientific computing.

Course Objective: To enable the students to program in Julia language.

Course Outcomes: After the end of this course students will able to

CO1 **Apply**[3] data structures , object oriented programming, networking concepts in Julia

CO2 **Examine**[4] Data Frames ,plots and make use of R and Python libraries in Julia

Practical/Experiments:

Four hours per (week/batch) practical is to be utilized for learning Python. This shall include extra problem statements and there implementations to strengthen the programming logic. It should comprise of minimum of 16-17 experiments. Students of different batches should implement different programs based on following guidelines

	Experiment Title
1	Basics of Julia for Data Analysis-Running your first program
2	Basics of Julia for Data Analysis-Loops, Conditionals, Functions in Julia
3	Programming in Julia- Object Oriented Features, type system
4	The Julia Ecosystem-Packages (common scientific computing packages, and how to use them) - graphics, math packages, graph theory, optimization, etc.
5	Metaprogramming in Julia
6	I/O, Networking, parallel computing

7	Exploratory analysis with Julia-Introduction to DataFrames.
8	Exploratory analysis with Julia-Visualisation in Julia using Plots.
9	Exploratory analysis with Julia-Bonus – Interactive visualizations using Plotly
10	Visualizing Data with Scatterplots, Histograms, and Box Plots
11	Data Munging in Julia
12	Building a predictive ML model-Logistic Regression
13	Building a predictive ML model- Decision Tree
14	Building a predictive ML model- Random Forest
15	Calling R libraries in Julia-Using pandas with Julia
16	Calling python libraries in Julia-Using ggplot2 in Julia

Text Books:

1. Ivo Balbaert , Adrian Salceanu, *Julia 1.0 Programming Complete Reference Guide*, Packt Publishing, May 20, 2019

References:

1. Paul D. McNicholas, Peter Tait, *Data Science with Julia*, 1st Edition, CRC Press Publications, January 11, 2019.
2. Ivo Balbaert, *Getting Started with Julia*, Kindle Edition, Packt Publishing, 2015

CST4131: Network Simulation and Modeling							
(Version 1.0, Program Elective, School of Technology)							
Lect.	**Tut.**	**Pract.**	**Credits**	**Evaluation Scheme**			
				Component	**Exam**	**Weightage %**	**Pass%**
3	-	-	3	Theory	FET	20	40
					CAT I	15	
					CAT II	15	
					ESE	50	40

Prerequisite: Basic knowledge of computer networks, Programming skills

Course Description: This course contains different methods for random number generation also the need for the development process to initiate the real problem. This course also focuses on principle and techniques of simulation methods informed by research direction.

Course Objective: The aim of this course is to introduce various system modeling and simulation techniques, and highlight their applications in different areas. It includes modeling, design, simulation, planning, verification and validation. After learning the simulation techniques, the students are expected to be able to solve real world problems which cannot be solved strictly by mathematical approaches. This course begins by demonstrating the usefulness of simulation as a tool for problem solving in business, industry, government, and society.

Course Outcomes: After the end of this course students will be able to

CO1 **Describe** [1] the components of continuous and discrete systems and simulate them
CO2 **Model** [3] any system from different fields
CO3 **Discuss**[3] the simulation methods and select the suitable technique on the problems.
CO4 **Analyze**[4] different Simulation Programming techniques

Syllabus (Theory)

Units	Description	Hours
I.	**Module 1: Introduction to Simulation** Introduction Simulation Terminologies, Application areas, Model Classification, Types of Simulation, Steps in a Simulation study, Concepts in Discrete Event Simulation, Simulation Examples	**7**

II. Module 2: Mathematical models

Statistical Models, Concepts Discrete Distribution- Continuous Distribution, Poisson Process, Empirical Distributions, Queuing Models, Characteristics- Notation, Queuing Systems, Markovian Models, Generation of Pseudo Random numbers- Properties of random numbers, Techniques for generating random numbers, Testing random number generators, Generating Random Variates, Inverse Transform technique, Acceptance- Rejection technique, Composition & Convolution Method — **7**

III. Module 3: Analysis of Simulation Data

Input Modeling, Data collection, Assessing sample independence, Hypothesizing distribution family with data, Parameter Estimation, Goodness-of fit tests, Selecting input models in absence of data, Output analysis for a Single system, Terminating Simulations, Steady state simulations — **7**

IV. Module 4: Verification and Validation

Model Building , Verification of Simulation Models**,** Calibration and Validation of Models**,** Validation of Model Assumptions**,** Validating Input – Output Transformations — **7**

V. Module 5: Simulation Of Computer Systems and Case Studies

Simulation Tool**,** Model Input**,** High level computer system simulation**,** CPU Memory Simulation**,** Comparison of systems via simulation**,** Simulation Programming techniques**,** Development of Simulation models. — **7**

VI. Module 1: Introduction to Simulation

Introduction Simulation Terminologies, Application areas, Model Classification, Types of Simulation, Steps in a Simulation study, Concepts in Discrete Event Simulation, Simulation Examples — **7**

Textbooks:

1. Jerry Banks and John Carson, *Discrete Event System Simulation*, Fourth Edition, PHI, 2005.
2. Geoffrey Gordon, *System Simulation*, Second Edition, PHI Publications, 2006 (Unit – V).
3. Frank L. Severance, *System Modeling and Simulation*, Wiley Publications, 2001.

References :

1. Sheldon M. Ross, *Introduction to Probability Models*, 7th Edition, Academic Press Publications, 2002
2. Donald E. Knuth, *The Art of Computer Programming - Volume 2, Semi Numerical Algorithms*, 2nd Edition, PEARSON Publication, 1997

Sanjay Ghodawat University, Kolhapur

School of Technology

Department of Computer Science and Engineering

Structure and Contents for Final Year B. Tech. Computer Science and Engineering Program

(AY 2020-21) R0

AME/P/80/0

4132: Cloud Computing							
Ver. 1.0, Program Elective, School of Technology							
Lect.	**Tut.**	**Pract.**	**Credits**	**Evaluation Scheme**			
				Component	**Exam**	**Weightage**	**Pass %**
3	--	--	3	Theory (100)	FET	20	40
					CAT I	15	
					CAT II	15	
					ESE	50	40

Prerequisite: Basics cloud computing and Information security

Course Description: Course Introduces the concept of cloud computing and describes the IT infrastructure security capabilities it also describes IT infrastructure security capabilities at the network, host, and application levels. Examines the current state of data security and the storage of data in the cloud Explains the identity and access management (IAM) practice and reveals the importance of audit and compliance functions within the cloud.

Course Objective:

1. To describe the physical and virtual components of and identify the principle technologies of cloud based systems.
2. To Evaluate and implement the security controls necessary to ensure confidentiality, integrity and availability in cloud computing
3. To Conduct risk assessments of existing and proposed cloud-based environments
4. To explain importance of Identity and Access Management(IAM) and audit and compliance functions within the cloud

Course Outcomes: After the end of this course students will be able to

CO1 **Analyze**[4] basic concepts and services of cloud computing.
CO2 **Demonstrate**[2] large scale distributed systems and cloud applications
CO3 **List**[1] the importance of cloud security
CO4 **Explain**[2] Ubiquitous Computing and applications

Syllabus (Theory)

Units	Description	Hours
I.	**Introduction to Cloud Computing:** Defining Cloud computing, Essential characteristics of Cloud computing, Cloud deployment model, Cloud service models, Multitenancy, Cloud cube model, Cloud economics and benefits, Cloud types and service scalability	7

over the cloud, challenges in cloud NIST guidelines.

II.	**Virtualization, Server, Storage And Networking**: Virtualization concepts, types, Server virtualization, Storage virtualization, Storage services, Network virtualization, Service virtualization, Virtualization management, Virtualization technologies and architectures, Internals of virtual machine, Measurement and profiling of virtualized applications. Hypervisors: KVM, Xen, Hyper V Different hypervisors and features.	**7**
III.	**Monitoring And Management:** Architecture for federated cloud computing, SLA management in cloud computing: Service provider's perspective, performance prediction for HPC on Clouds, Monitoring Tools.	**7**
IV.	**Security:** Cloud Security risks, Security, Privacy, Trust, Operating system security, Security of virtualization, Security risks posed by shared images, Security risk posed by a management OS, Trusted virtual machine monitor.	**7**
V.	**Cloud Implementation And Applications:** Cloud Platforms: Amazon EC2 and S3, Cloudstack, Intercloud, Google App Engine, Open Source cloud Eucalyptus, Open stack, Open Nebulla, etc., Applications.	**7**
VI.	**Ubiquitous Computing:** Basics and Vision, Applications and Requirements, Smart Devices and Services, Human Computer Interaction, Tagging, Sensing and controlling, Context-Aware Systems, Ubiquitous Communication, Management of Smart Devices, Ubiquitous System Challenge and outlook	**7**

Textbooks:

1. Barrie Sosinsky, *Cloud Computing Bible*, Wiley Publications, 2011
2. Gautham Shroff, *Enterprise Cloud Computing*, Cambridge University Press, 2010.
3. Stefan Poslad, *Ubiquitous Computing: Smart Devices, Environments and Interactions*, John Wiley & Sons Publications, 2011.

References:

1. Rajkumar Buyya, J.Broberg, A. Goscinski, *Cloud Computing Principles and Paradigms*, First Edition, Wiley Publications, 2013.
2. Ronald Krutz and Russell Dean Vines, *Cloud Security: Comprehensive guide to Secure Cloud Computing*, Wiley Publications, 2010.

CST4133: Data Visualization							
(Ver. 1.0, Program Elective, School of Technology)							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage %	Pass %
3	-	-	3	Theory	FET	20	40
					CAT I	15	
					CAT II	15	
					ESE	50	40

Prerequisite: : Basic programming skills

Course Description: This course is designed to provide students with the foundations necessary for understanding and extending the current state of the art in data visualization.

Course Objective: An understanding of the key techniques and theory used in visualization, including data models, graphical perception and techniques for visual encoding and interaction.

Course Outcomes: After the end of this course students will be able to

CO1 **Explain**[2] the basics of data visualization

CO2 **Select**[2] the techniques of the visualization process

CO3 **Choose**[3] different techniques for data visualization.

CO4 **Apply**[3] appropriate data visualization techniques from visualization systems.

Syllabus (Theory)

Units	Description	Hours
I.	**Introduction:** Introduction of visual perception, History of visualization, visual representation of data, Gestalt principles, information overloads. Creating visual representations, visualization reference model, visual mapping, and visual analytics.	7
II.	**Visualization Techniques** Classification of visualization systems, Interaction and visualization techniques misleading, Visualization of one, two and multi-dimensional data, text and text documents	7

III.	**Visualization Techniques for Tree, graph and Networks** Visualization of groups, trees, graphs, clusters, networks, software, Metaphorical visualization	**7**
IV.	**Visualization of spatial data for field based GIS** Visualization of volumetric data, vector fields, processes and simulations, Visualization of maps, geographic information, GIS systems, collaborative visualizations, Evaluating visualizations	**7**
V.	**Comparing and evaluating Visualization Techniques** User task, User characteristics, Data characteristic, Visualization characteristic, Structure for evaluating visualization, Benchmarking Procedure	**7**
VI.	**Visualization Systems** System based on Data Type, System based on Analysis Types, Text analysis and visualization, Modern integrated visualization system.	**7**

Textbooks:

1. Ben Fry, *Visualizing Data: Exploring and Explaining Data with the Processing Environment*, 1st Edition, O'Reilly Media Publications, 2008.
2. Chun-houh Chen, Wolfgang Härdle, Antony Unwin, *Handbook of Data Visualization*, Springer Publications, 2007.

References :

1. Thomas Strothotte, *Computer Visualization–Graphics Abstraction and Interactivity*, Springer Publications, 2011.
2. Edward R. Tufte, *The Visual Display of Quantitative Information*, Second Edition, Graphics Press Publications, 2001
3. Charles D. Hansen and C. R. Johnson, *Visualization Handbook*, Academic Press Publications, 2007

CST4134: Soft Computing							
Ver. 1.0, Program Elective, School of Technology							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage	Pass%
3	--	--	3	Theory (100)	FET	20	40
					CAT I	15	
					CAT II	15	
					ESE	50	40

Prerequisite: Linear algebra and calculus, Proficiency with algorithms, Critical thinking and problem solving skills, Programming skills

Course Description: Soft computing is the use of approximate calculations to provide imprecise but usable solutions to complex computational problems. The approach enables solutions for problems that may be either unsolvable or just too time-consuming to solve with current hardware. Soft computing is an important branch of computational intelligence, where fuzzy logic, probability theory, neural networks, and genetic algorithms are synergistically used to mimic the reasoning and decision making of a human.

Course Objective:

Be exposed to various soft computing applications.
Understand soft computing problems and solve it
Be exposed to concept of fuzzy logic and use it in various industrial applications.
Understand different multi objective optimization problems and solve them.
Learn deep learning concepts and apply its techniques to various applications.

Course Outcomes: After the end of this course students will be able to

CO1 **Identify**[3] applications of soft computing to solve real life problems
CO2 **Make Use of**[3] concepts of Fuzzy Logic in industrial applications.
CO3 **Integrate**[3] and Solve multi-objective optimization problems
CO4 **Apply**[3] suitable Deep Learning techniques to various applications.

Syllabus (Theory)

Units	Description	Hours
I.	**Introduction:** Concept of computing systems, "Soft" computing versus "Hard" computing, Characteristics of Soft computing, Some applications of Soft computing techniques	7

II.	**Fuzzy Logic-I**: Basic concepts of fuzzy logic, Fuzzy sets and Crisp sets, Fuzzy set theory and operations, Properties of fuzzy sets, Fuzzy and Crisp relations, Fuzzy to Crisp conversion.	**7**
III.	**Fuzzy Logic –II:** Membership functions, interference in fuzzy logic, fuzzy if-then rules, Fuzzy implications and Fuzzy algorithms, Fuzzyfications & Defuzzifications, Fuzzy Controller, Industrial applications.	**7**
IV.	**Genetic Algorithm:** Concept of "Genetics" and "Evolution" and its application to probabilistic search techniques, Basic GA framework and different GA architectures, GA operators: Encoding, Crossover, Selection, Mutation, etc., Solving single-objective optimization problems using GAs.	**7**
V.	**Multi-objective Optimization Problem Solving:** Concept of multi-objective optimization problems (MOOPs) and issues of solving them, Multi-Objective Evolutionary Algorithm (MOEA), Non-Pareto approaches to solve MOOPs, Pareto-based approaches to solve MOOPs, Some applications with MOEAs.	**7**
VI.	**Deep Learning:** Study of Neural Networks, Simple implementation of Artificial Neural Network, Introduction to Deep Learning, Recent Trends in Deep Learning, various classifiers and techniques.	**7**

Textbooks:

1. S. Rajasekaran, G. A.Vijayalakshmi Pai, *Neural Networks, Fuzzy Logic and Genetic Algorithm, Synthesis and Applications*, PHI Learning Pvt. Ltd., 2017.
2. Melanie Mitchell , *An Introduction to Genetic Algorithms*, MIT Press Publications, 2000.
3. D. K. Pratihar, *Soft Computing : Fundamentals and Applications*, Second Edition, Narosa Publications, 2013.

References :

1. Timothy J. Ross, *Fuzzy Logic with Engineering Applications*, Third Edition, Wiley India Publications, 2011.
2. Kwang H. Lee, *First course on Fuzzy Theory and Applications*, Springer Publications, 2005.
3. George J. Klir and Bo Yuan, *Fuzzy Sets and Fuzzy Logic-Theory and Applications*,

Prentice Hall Publications, 1996.

CST4135: SOFTWARE ARCHITECTURE							
Ver. 1.0, Program Elective, School of Technology							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage	Pass%
3	-	-	3	Theory 100 Marks	FET	20	40
					CAT I	15	
					CAT II	15	
					ESE	50	40

Prerequisite: Object Oriented Modeling and Design, Software Engineering

Course Description: This course introduces the Importance of software architecture while designing and developing software models. This Course also gives quality attributes and design patterns of software. This course will give you the brief knowledge about how to use software architecture in the context of cloud.

Course Objective: Students will learn computer architecture design principles.

Course Outcomes: After the end of this course students will be able to

CO1 **Explain**[2] software architectures and its importance.
CO2 **Outline**[2] quality attribute in software architecture.
CO3 **Develop**[3] Attribute Driven Design software architecture.
CO4 **Develop**[3] cloud based Software Architecture.

Syllabus (Theory)

Units	Description	Hours
I.	**Introduction:** What Is Software Architecture, What Software Architecture Is and What It Isn't, Architectural Structures and Views, Architectural Patterns, What Makes a "Good" Architecture.	7
II.	**Why Is Software Architecture Important:** Inhibiting or Enabling a System's Quality Attributes, Reasoning About and Managing Change, Predicting System Qualities, Enhancing Communication among Stakeholders, Carrying Early Design Decisions, Defining Constraints on an Implementation, Improving Cost and Schedule	7

Estimates

III. Software Quality Attributes:

Quality Classes, Understanding Quality Attributes: Usability, Availability, Interoperability, Testability, Performance, Modifiability, Security, Other Quality Attributes. **7**

IV. Architecture and Designing:

How Much Architecture, Agility and Architecture Methods, A Brief Example of Agile Architecting , Guidelines for the Agile Architect, Design Strategy , The Attribute-Driven Design Method, The Steps of ADD **7**

V. Architecture Patterns

Layering Patterns, Pipe & Filter, Blackboard, Distributed Systems, Reflection **7**

VI. Architecture in the Context of Cloud

Introduction to Cloud Computing, Service Models, Case Studies Software Architecture: Past, Present, and Future **7**

Textbooks:

1. Len Bass, Paul Clements, Rick Kazman, *Software Architecture in Practice*, Second Edition, Pearson, 2003, ISBN 978-81-775-8996-2
2. Erich Gamma, *Design Patterns*, First Edition, Pearson Publications, 1994, ISBN 0-201-63361-2.

References :

1. Richard N.Taylor, Nenad M., *Software Architecture Foundation Theory and practice*, Wiley Publication, 2010, ISBN: 978-81-265-2802-8.
2. Paul Clements, Felix Bachmann, Len Bass, David Garlan, *Documenting Software Architectures: Views and Beyond*, Addison-Wesley Professional Publications, 2003, ISBN-10:0201703726, ISBN-13: 9780201703726

CST4136 : Social Network Technologies							
Ver. 1.0, Program Elective, School of Technology							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage %	Pass %
3	-	-	3	Theory 100 Marks	FET	20	40
					CAT I	15	
					CAT II	15	
					ESE	50	40

Prerequisite: Social Network Analysis

Course Description: This course is an introduction to the concepts and methods of social network analysis. Students will learn to extract and manage data about network structure and dynamics, and to analyze, model and visualize such data.

Course Objective: To understand theoretical and empirical issues in current research on social network analysis

Course Outcomes: After the end of this course students will able to

CO1 **Understand**[2] basic concept of social network and evolution.
CO2 **Explain**[2] knowledge Extraction for Social Networks analysis.
CO3 **Compare**[4] Modeling and visualization technique in online social network.
CO4 **Relate**[2] the ontology in social network & semantic web

Syllabus (Theory)

Units	Description	Hours
I.	**Introduction** Introduction to Web, Limitations of current Web, Development of Semantic Web, Emergence of the Social Web, Statistical Properties of Social Networks, Network analysis, Development of Social Network Analysis-Key concepts and measures in network analysis, Discussion networks-Blogs and online communities-Web-based networks	**7**

II. Evolution **7**

Evolution in Social Networks–Framework, Tracing Smoothly Evolving Communities, Models and Algorithms for Social Influence Analysis, Influence Related Statistics, Social Similarity and Influence, Influence Maximization in Viral Marketing, Link Prediction in Social Networks, Feature based Link Prediction

III. Mining Communities and Opinion Mining **7**

Applications of Community Mining Algorithms, Node Classification in Social Networks Opinion Extraction–Sentiment Classification and Clustering, Temporal Sentiment Analysis-Irony Detection in Opinion Mining-Wish Analysis–Product Review Mining–Review Classification

IV. Modeling and Visualization Visualizing **7**

Online Social Networks, A Taxonomy of Visualizations, Graph Representation-Centrality-Clustering-Node-Edge Diagrams-Visualizing Social Networks with Matrix-Based Representations-Node-Link Diagrams, Hybrid Representations, Modeling and Aggregating Social Network Data, Random Walks and their Applications, Ontological representation of Social Individuals and Relationships

V. Ontology Engineering **7**

Ontology's and their role in the semantic web, Ontology, Ontology Web Language (OWL), UML, XML/XML Schema, Constructing Ontology, Ontology Methods, Ontology Sharing and Merging, Ontology Libraries and Ontology Mapping, Logic, Rule and Inference Engines.

VI. Social Network and semantic web **7**

What is social Networks analysis, development of the social networks analysis, Electronic Sources for Network Analysis, Electronic Discussion networks, Blogs and Online Communities, Web Based Networks.

Textbooks:

1. Charu C. Aggarwal, *Social Network Data Analytics*, Springer Publications, 2011
2. Peter Mika, *Social Networks and the Semantic Web*, First Edition, Springer Publications, 2007.
3. BorkoFurht, *Handbook of Social Network Technologies and Applications*, First Edition, Springer Publications, 2010.

References :

1. Valente, Thomas W. 2010. *Social Networks & Health: Models, Methods, & Applications*, Oxford University Press Publications, 2010, ISBN 9780195301014
2. Ajith Abraham, Aboul Ella Hassanien, VáclavSnášel, *Computational Social Network Analysis: Trends, Tools and Research Advances*, Springer Publications, 2012

CST4137 : Blockchain Technologies							
Ver. 1.0, Program Elective, School of Technology							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage %	Pass %
3	-	-	3	Theory	FET	20	40
					CAT I	15	
					CAT II	15	
					ESE	50	40

Prerequisite: Understanding of basic programming languages like Golang, Java, or Javascript

Course Description: This course introduces basic concepts and functionality of Blockchain technology. This course describes about how to write smart contracts. This course also introduces ethereum and hyperledger platforms.

Course Objective: To learn basic concepts of Blockchain technology along with ethereum & hyperedger platforms.

Course Outcomes: After the end of this course students will able to

CO1 **Discuss**[4] applications of Blockchain
CO2 **Assess**[5] Blockchain applications in a structured manner
CO3 **Explain**[2] Blockchain concepts clearly and persuasively
CO4 **Summarize**[2] Crypto currencies

Syllabus (Theory)

Units	Description	Hours
I.	**Introduction To Blockchain** Digital Trust, Asset Transactions, Distributed Ledger Technology, Types of network, components of Blockchain or DLT, Ledger -Blocks, Blockchain, PKI and Cryptography - Private keys, Public keys, Hashing, Digital Signature	7
II.	**Blockchain Working** Consensus - Byzantine Fault, Proof of Work, Poof of Stake, Security – DdoS, Cryptocurrency, Digital Token, How Blockchain Works, Structure	7

	of Blockchain, Block Hash Blockchain, Distributed Lifecycle of Blockchain, Smart Contract, Consensus Algorithm, Proof of Work, Proof of Stake, Practical Byzantine Fault Tolerance, Actors of Blockchain	
III.	**Introduction to Bitcoin** Currency, Double Spending, Cryptocurrency, P2P Payment Gateway, Wallet Mining	**7**
IV.	**Ethereum** Ethereum network, EVM, Transaction fee, Mist, Ether, gas, Solidity-Smart contracts, Truffle, Web3 Design and issue, Cryptocurrency, Mining, DApps	**7**
V.	**Hyperledger Fabric V1.1** Introduction to Hyperledger, What is Hyperledger, Why Hyperledger, Where can Hyperledger be used, Hyperledger Architecture, Membership, Blockchain, Transaction, Chaincode, Hyperledger Fabric, Features of Hyperledger, Fabric Installation of prerequisite	**7**
VI.	**Hyperledger Explorer** Introduction To Hyperledger Explorer, Block Details Peer List,Chaincode List, Transaction Details, Installation of Hyperledger Explorer, Starting the Explorer App	**7**

Textbooks:

1. EladElrom, *The Blockchain Developer: A Practical Guide for Designing, Implementing, Publishing, Testing, and Securing Distributed Blockchain-based Projects*, Apress publication, 2019.
2. Horst Treiblmaier, Roman Beck, *Business Transformation through Blockchain*, Macmillan publication, 2019.

References :

1. Vincenzo Morabito, *Business Innovation Through Blockchain The B³ Perspective* Springer publications, 2017.
2. Daniel Drescher, *Blockchain Basics*, Apress publication, 2017.

CST4138 : Cloud Security							
Ver. 1.0, Program Elective , School of Technology							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage	Pass%
3	-	-	3	Theory	FET	20	40
					CAT I	15	
					CAT II	15	
					ESE	50	40

Prerequisite: Information security basics and computer network basics

Course Description: The course will describe the Cloud security architecture and explore the guiding security design principles, design patterns, industry standards, applied technologies and addressing regulatory compliance requirements critical to design, implement, deliver and manage secure cloud based services.

Course Objective:

1. To describe the physical and virtual components of and identify the principle technologies of cloud based systems.
2. To Evaluate and implement the security controls necessary to ensure confidentiality, integrity and availability in cloud computing
3. To Conduct risk assessments of existing and proposed cloud-based environments
4. To explain importance of Identity and Access Management(IAM) and audit and compliance functions within the cloud

Course Outcomes: After the end of this course students will be able to

CO1 **Evaluate**[5] the security various layers of cloud infrastructure
CO2 **Analyze**[4] encryption and identity management services in a cloud environment
CO3 **Perform**[3] vulnerability assessments in a cloud environment
CO4 **Develop**[3] a cloud disaster recovery and business continuity plan

Syllabus (Theory)

Units	Description	Hours
I.	**Introduction to Cloud Computing**	**7**
	The Evolution of Cloud Computing, What is Cloud computing? , SPI framework of Cloud Computing, Traditional Software Model, Cloud	

Service Delivery model, Cloud Deployment Models, Key Drivers to Adopting the Cloud, The Impact of Cloud Computing on Users, Governance in the Cloud, Barriers to Cloud Computing Adoption in the Enterprise

II. Fundamentals and Risk Issues in the Cloud **7**

Cloud Information Security Objectives, Cloud Security Services, Cloud Security Design Principles, Secure Cloud Software Requirements, Security Policy Implementation and decomposition, Cloud Computing and Business Continuity/Disaster Recovery, CIA triad, Privacy and compliance risk

III. Infrastructure and data Security **7**

Infrastructure Security: The Network Level, Infrastructure Security: The Host Level, Infrastructure Security: The Application Level 4 15% 4 Data Security and Aspects of Data Security, Data Security Mitigation, Provider Data and Its Security

IV. Identity and Access Management **7**

Trust Boundaries and IAM, Why IAM? , IAM Challenges, IAM Definitions, IAM Architecture and Practice, Getting Ready for the Cloud, Relevant IAM Standards and Protocols for Cloud Services, IAM Practices in the Cloud. Cloud Authorization Management, Cloud Service Provider IAM Practice and use cases for IAM with Amazon EC2 and Amazon S3. IAM access management, policies and permissions.

V. Security Management in the Cloud **7**

Security Management Standards, Security Management in the Cloud, Availability Management, SaaS Availability Management, PaaS Availability Management, IaaS Availability Management, Access Control, Security Vulnerability, Patch, and Configuration Management

VI. Audit and Compliance **7**

Internal Policy Compliance, Governance, Risk, and Compliance (GRC) Illustrative Control Objectives for Cloud Computing ,Incremental CSP-Specific Control Objectives Additional Key Management Control Objectives, Control Considerations for CSP Users Regulatory/External Compliance, Cloud Security Alliance Auditing the Cloud for Compliance

Textbooks:

1. Tim Mather, Subra Kumaraswamy and Shahed Latif, *Cloud Security and Privacy*, O'Reilly Publications, 2009.

References :

1. Raghu Yeluri and Enrique Castro-Leon, *Building the Infrastructure for Cloud Security A Solutions view*, Apress open Publications, 2014.

2. Ronald L. Krutz and Russell Dean Vines, *Cloud Security A Comprehensive Guide to Secure Cloud Computing*, Wiley Publications, 2010.

CST4151: Network Simulation and Modeling							
(Version 1.0, Program Elective, School of Technology)							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weight-age	Pass %
-	-	2	1	Practical 50 Marks	FEP	100	40

Prerequisite: Programming skills

Course Description: The aim of this course is to introduce various system modeling and simulation techniques, and highlight their applications in different areas. It includes modeling, design, simulation, planning, verification and validation. After learning the simulation techniques, the students are expected to be able to solve real world problems which cannot be solved strictly by mathematical approaches. This course begins by demonstrating the usefulness of simulation as a tool for problem solving in business, industry, government, and society.

Course Objective: To study the communication networks characteristics and to analyze various MAC and routing layer Protocols.

Course Outcomes: After the end of this course students will be able to

CO1 **Experiment** [4] on CSMA/CD and CSMA/CA protocols, Stop and Wait protocol, Go back N, Selective Repeat protocols by using NETSIM simulator.

CO2 **Demonstrate**[2] network protocols by using tools available in Petri-nets, Uppaal, SMV modeling tools.

CO3 **Apply**[3] numerical algorithm to meet the simple requirements

Practical

Two hours per week per batch practical is to be utilized for writing programs to ensure that students have properly learnt the topics covered in the lectures. It should comprise of minimum 10-12 experiments. Students of different batches should implement different programs based on the following guidelines-

Implement following programs by using NETSIM Software and LAN Trainer kit.

1. To create scenario of CSMA/CD and study the performance of the CSMA/CD protocol through simulation

2. To create scenario of Token Bus and Token Ring protocols and study the performance through simulation
3.To create scenario and study the performance of network with CSMA/CA protocol and compare with CSMA/CD protocols
4. Implementation of Stop and Wait protocol
5. Implementation of Go back N and Selective Repeat protocols
6. Implementation of Distance Vector Routing algorithm
7. Implementation of Link state routing algorithm
8. Implementation of data encryption and decryption
9. Transfer of files from PC to PC using Windows/ UNIX socket processing
10. Implementation tools available in Petri-nets, Uppaal, SMV tools.
11. Implementation and Modeling with Petri Nets
12.Experiment on primitives for Programming Constructs
13. Implementation of sub-structures of Petri Nets and Model Checking.

Textbooks:

1. Jerry Banks and John Carson, *Discrete Event System Simulation*, Fourth Edition, PHI Publications, 2005.
2. Averill M. Law and W. David Kelton, *Simulation Modeling and Analysis*, Third Edition, McGraw Hill Publications, 2006.
3. Jerry Banks, *Handbook of Simulation: Principles, Methodology, Advances, Applications and Practice*, Wiley Publications, 1998.

References :

1. Sheldon M. Ross, *Introduction to Probability Models*, 7th Edition, Academic Press Publications, 2002
2. Donald E. Knuth, *The Art of Computer Programming*, Volume 2: Semi Numerical Algorithms, 2nd Edition, PEARSON Education Publications, Reading MA, USA, 2000

CST4152: Cloud Computing Laboratory							
Ver. 1.0, Program Elective, School of Technology							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage %	Pass %
-	-	2	1	Practical 50 Marks	FEP	100	40

Prerequisite: Information security basics and programming Languages C,C++ Java

Course Description: Cloud computing is the delivery of computing services including servers, storage, databases, networking, software, analytics, and intelligence over the Internet ("the cloud") to offer faster innovation, flexible resources, and economies of scale. The purpose is to earn concept such as virtual machines, open stack, cloud features and security, cloud services etc.

Course Objective:

1. To Develop basic cloud instance on public/ private cloud
2. To Perform the Identity , Access management
3. To perform risk analysis in cloud computing

Course Outcomes: After the end of this course students will be able to

CO1 **Create**[3] and run virtual machines on open source OS

CO2 **Install**[3] security features for cloud

Practical

Two hours per week per batch practical is to be utilized for writing programs to ensure that students have properly learnt the topics covered in the lectures. It should comprise of minimum 10-12 experiments. Students of different batches should implement different programs based on the following guidelines-

1. Creating and running virtual machines on open source OS
2. Installing OpenStack and use it as Infrastructure as a Service
3. Installation and understanding features of own Cloud as SaaS.
4. Installing and using identity management feature of OpenStack

5. Write a program for web feed
6. Installing and using JOSSO
7. Installing and using security feature of own Cloud
8. Installing and using Administrative features of own Cloud
9. Case study on Microsoft Azure
10. Creating a cloud like social site for institute

Textbooks:

1. Gautam Shroff, *Enterprise Cloud Computing*, Cambridge Publications,2010
2. Ronald Krutz and Russell Dean Vines, *Cloud Security*, Wiley – India Publications, 2010, ISBN:978-0-470-58987-8
3. Aditya Patawar, *Getting Started with Own Cloud*, Packt Publishing Ltd, 2013

References :

1. Rajkumar Buyya, J.Broberg, A. Goscinski, *Cloud Computing Principles and Paradigms*, Wiley Publications, 2013.
2. Ronald Krutz, *Cloud Security: Comprehensive guide to Secure Cloud Computing*, Wiley Publications, 2010.
3. Anthony T. Velte, *Cloud Computing: Practical Approach*, McGraw Hill Publications, 2009.
4. Tim Mather, *Cloud Security and Privacy*, O'REILLY Media Publications, 2009.

CST4153: Data Visualization Laboratory							
(Ver. 1.0, Program Elective, School of Technology)							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	WEIGHTAGE %	Pass%
-	-	2	1	Practical 50 Marks	FEP	100	40

Pre-Requisite Courses: Basic programming skills

Course Description: This courses shows you how to better understand your data, present clear evidence of your findings to your intended audience, and tell engaging data stories that clearly depict the points you want to make all through data graphics.

Course Objectives :

1. Understand the role of visualization in the processing and analysis of data coming from a broad range of sources
2. Develop software and tools to create visualizations of data that are effective for Analysis

Course Outcomes: At the end of this course students will able to

CO1 **Identify**[3] appropriate techniques for data visualization
CO2 **Model**[3] different types of data using visualization tools
CO3 **Make use of**[3] library functions to visualize data

Practicals

Two hours per week per batch practical is to be utilized for writing to ensure that students have properly learnt the topics covered in the lectures. This shall include extra problem statements and there implementations to strengthen the programming logic. It should comprise of minimum of 10-12 experiments. Students of different batches should implement different programs based on following guidelines :

List of experiments to meet the requirements of the syllabus

Experiment Title

1 Implementation of Basic plots in Data Visualization
2 Use of Different Data Visualization Libraries using R Programming

3	Customization or Styling the plot by setting different colors with different shapes sizes for different plots
4	Visualizing data with Data Visualization Functions Implementation of Density Plot for given Dataframe
5	Categorical composition of the total population using waffle chart and pie chart
6	Implementation of Tree Map for given dataset
7	Implementation of Time Series Plots from data formats
8	Program to define function in R programming.
9	Use of Correlogram plot for correlating multiple variables in given data frames
10	Implementation of Seasonal Plot for time series data frames
11	Program to implement matrix operations in R
12	Load the iGraph library and create an undirected star graph with 5 nodes and plot the graph.

Textbooks:

1. Ben Fry, *Visualizing Data: Exploring and Explaining Data with the Processing Environment*, 1st Edition, O'Reilly Media Publications, 2008.
2. Chun-houh Chen, Wolfgang Härdle, Antony Unwin, *Handbook of Data Visualization*, Springer Publications, 2007.

References :

1. Thomas Strothotte, *Computer Visualization–Graphics Abstraction and Interactivity*, Springer Publications, 2011.
2. Edward R. Tufte, *The Visual Display of Quantitative Information*, 2nd ed. Graphics Press Publications, 2001
3. Charles D. Hansen and C. R. Johnson, *Visualization Handbook*, Academic Press Publications, 2007

CST4154: Soft ComputingLaboratory							
Ver. 1.0, Program Elective, School of Technology							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage %	Pass %
-	-	2	1	Practical 50 Marks	FEP	100	40

Prerequisite: students must know basic concepts of mathematics and set theory

Course Description: Soft Computing is a discipline that deals with the design of intelligent systems which are in contrast to classical hard computing techniques. The principal objective of this course is to introduce students to soft computing techniques from computer science perspective. It covers fuzzy logic, neural networks and evolutionary algorithms. These techniques help to achieve tractable, robust, and low cost solutions to real-world problems.

Course Objective: Understand the underlying principle of soft computing with its usage in various applications.

Course Outcomes: After the end of this course students will be able to

CO1 **Apply**[3] fuzzy logic technique to solve fuzzy problems

CO2 **Apply**[3] genetic algorithm to find out optimized solution to given problem

CO3 **Classify**[4] given data using artificial neural network

Practical

Two hours per week per batch practical is to be utilized for writing programs to ensure that students have properly learnt the topics covered in the lectures. It should comprise of minimum 10-12 experiments. Students of different batches should implement different programs based on the following guidelines-

1. Study of Soft computing concepts
2. Implementation of Fuzzy Operations
3. Implementation of fuzzy relations (Max-Min Composition)

4. Implementation of fuzzy controller
5. Implementation Genetic Application – Match Word Finding
6. Study of ANFIS Architecture
7. Study of Derivative-free Optimization
8. Solving routing problem of mass transit system using GA
9. Implementation of Supervised Learning Algorithm
10. Implementation of unsupervised learning algorithm – Hebbian Learning
11. Problems based on ANN and its applications in transportation
12. Implementation of Single layer Perceptron Learning Algorithm

Textbooks:

1. S.Rajasekaran, G.A.VijayalakshmiPai, *Neural Networks, Fuzzy Logic and Genetic Algorithm, Synthesis and Applications*, PHI Learning Pvt. Ltd., 2017.
2. Melanie Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, 2000 Soft Computing: Fundamentals and Applications (2nd Ed.) D. K. Pratihar (Narosa, 2013)

References :

1. Timothy J. Ross, *Fuzzy Logic with Engineering Applications*, Third Edition, Wiley India Publications, 2011.
2. Kwang H. Lee, *First course on Fuzzy Theory and Applications*, Springer Publications, 2005.

CST4155 : Software Architecture Lab							
Ver. 1.0, Program Elective, School of Technology							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage %	Pass %
-	-	2	1	Practical 50 Marks	FEP	100	40

Prerequisite: Object Oriented Modeling and Design, Software Engineering

Course Description: Practical approaches and methods for creating and analyzing software architecture are presented. The emphasis is on the interaction between quality attributes and software architecture.

Course Objective: Students will also gain experiences with examples in design pattern application and case studies in software architecture.

Course Outcomes: After the end of this course students will be able to

CO1 **Develop**[3] architecture view for target System.

CO2 **Develop**[3] software applications in a development environment

CO3 **Test**[4] system performance for target case study.

Practical

Two hours per week per batch practical is to be utilized for writing programs to ensure that students have properly learnt the topics covered in the lectures. Students of different batches should implement different programs based on the following guidelines-

1. To narrate Requirement Definition Document for the target system with following three areas: Problem Identification, Problem Definition, and Problem Statement
2. To narrate System Requirements Specification Document for target system with reference to the IEEE 610.12.1990 Std guidelines.
3. To narrate System Architecture Requirement Specification Document for target system

with stakeholder and roles description.

4. To select appropriate Architectural View and Style and prepare Architecture Diagram for the target system.
5. To prepare Architecture Decision document describing Architectural Decisions, Software Interfaces, and behaviors along with Architectural Review.
6. To implement the target system using the Technical Architecture conforming to technology availability and scalability.
7. To create Test Plan, Test Cases and apply them to test the performance adequacy of the system implemented.

Textbooks:

1. Len Bass, Paul Clements, Rick Kazman, *Software Architecture in Practice*, Second Edition, Pearson Publications, 2003, ISBN 978-81-775-8996-2
2. Erich Gamma, *Design Patterns*, First Edition, Pearson Publications, 1994, ISBN 0-201-63361-2.

References :

1. Richard N.Taylor , Nenad M., *Software Architecture Foundation Theory and practice*, Wiley Publication, 2010, ISBN: 978-81-265-2802-8.
2. Paul Clements, Felix Bachmann, Len Bass, David Garlan, *Documenting Software Architectures: Views and Beyond* Addison-Wesley Professional 2003, ISBN-10:0201703726, ISBN-13: 9780201703726

CST4156 : Social Networks Laboratory							
Ver. 1.0, Program Elective, School of Technology							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage %	Pass %
-	-	2	1	Practical 50 Marks	FEP	100	40

Prerequisite: Data Structures

Course Description: This course is an introduction to the theory, methods and procedures of network analysis with emphasis on applications to communication and social behavior.

Course Objective: To provide a working knowledge of the concepts and methods used to describe and analyze social networks so that professionals and researchers can understand the results and implications of this body of research.

Course Outcomes: After the end of this course students will able to

CO1 **Apply**[3] different clustering techniques on big data.

CO2 **Develop**[3] social network based on ontology.

CO3 **Analyze**[3] the impact of Social Networks.

Practical

Two hours per week per batch practical is to be utilized for writing programs to ensure that students have properly learnt the topics covered in the lectures. It should comprise of minimum 10-12 experiments. Students of different batches should implement different programs based on the following guidelines-

1. Implement Neurons &Social Actor-Advanced techniques in Link Prediction
2. Implement MapReduce using Hadoop.
3. Develop Ontology in protégé editor using OWL.
4. Prepare a small network ~ 15-25 nodes representing your friendship network or another network if you prefer not to disclose your personal info
   Write a short description indicating the nature of nodes and edges
   Prepare a table and a drawing representing the network
5. Solve Degree Centrality, Between Centrality and Cliques.
6. Develop random graphs using Erdos Reini Model

7. Design Weak Ties, Traidic closure for given graph
8. Develop Spectral Clustering steps for given graph.
9. Case Studies of Social Network sites like Facebook-Twitter-Linkedin
10. Case Studies of Web based networks and Erdos Reini Model

Textbooks:

1. John Scott, *Social network Analysis*, Third edition, SAGE Publications, 2012.

References :

1. Christina Prell, *Social Network Analysis: History, Theory and Methodology*, SAGE Publications, 2011

CST4157 : Blockchain Technologies Laboratory							
Ver. 1.0, Program Elective, School of Technology							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage %	Pass %
-	-	2	1	Practical 50 Marks	FEP	100	40

Prerequisite: Understanding of basic programming languages like Golang, Java, or Javascript

Course Description: With this course students will be able to generate crypto currency, Blockchain node and write smart contract.

Course Objective: Students should be able to develop Blockchain components.

Course Outcomes: After the end of this course students will able to

CO1 **Make use of**[3] crypto currency exchanges and wallets safely

CO2 **Implement**[3] wallet, crypto currency, tokens, channel, ICO & smart contract.

Practical

Two hours per week per batch practical is to be utilized for writing programs to ensure that students have properly learnt the topics covered in the lectures. It should comprise of minimum 10-12 experiments. Students of different batches should implement different programs based on the following guidelines-

1. Creation of wallets and sending crypto currency
2. Starting a Wordpress website
3. Experimenting Blockchain explorer
4. Creation of your own crypto currency
5. Tokenization and trading crypto currencies

6. Generation of crypto material for the various participants
7. Generation of genesis block for the ordered node and start ordering service (solo node).
8. Generation of configuration transaction block to create a new channel.
9. Signing the configuration block and create the new channel.
10. Making peers of all the organizations join the channel
11. Starting your own ICO
12. Writing a smart contract

Textbooks:

1. EladElrom, *The Blockchain Developer: A Practical Guide for Designing, Implementing, Publishing, Testing, and Securing Distributed Blockchain-based Projects*, Apress publication, 2019.
2. Horst Treiblmaier, Roman Beck, *Business Transformation through Blockchain*, Macmillan publication, 2019

References :

1. Vincenzo Morabito, *Business Innovation Through Blockchain The B³ Perspective* Springer publications, 2017.
2. Daniel Drescher, *Blockchain Basics*, Apress publication, 2017.

CST4158 : Cloud Security Laboratory							
Ver. 1.0, Program Elective, School of Technology							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage %	Pass %
-	-	2	1	Practical 50 Marks	FEP	100	40

Prerequisite: Information security basics and programming Languages C,C++ Java

Course Description: The course delves deep into the secure cloud architectural aspects with regards to identifying and mitigating risks, protection and isolation of physical & logical infrastructures

Course Objective: To Design security architectures that assures secure isolation of physical and logical infrastructures including compute, network and storage, comprehensive data protection at all layers, end-to-end identity and access management, monitoring and auditing processes and compliance with industry and regulatory mandates.

Course Outcomes: After the end of this course students will be able to

CO1 **Develop**[3] basic cloud instance on public/ private cloud infrastructure

CO2 **Discuss**[2] Identity, Access management in cloud risk analysis

Experiments

1. Create a basic cloud instance on public cloud infrastructure and a Security baseline
2. Encrypt public cloud data: learn about cloud storage options and encrypt the data for the public cloud deployment
3. Create a basic federated identity infrastructure to support cloud application.
4. Create IAM user group and manage users access and permissions
5. Create IAM (Identity and Access Management) web service to manage users and user services in AWS(Amazon Web services)
6. Developing the procedures to handle security breaches and other incidents

7. Create Attribute Based Access Control in IAM AWS
8. Create and Secure Private Cloud
9. Design based experiment Amazon Web Services (IaaS)
10. Create Linux Firewall (Netfilter)

Text Books :

1. Tim Mather, Subra Kumaraswamy and Shahed Latif, *Cloud Security and Privacy*, O'Reilly Publications, 2009.
2. Raghu Yeluri and Enrique Castro-Leon, *Building the Infrastructure for Cloud Security A Solutions view*, Apress open Publications, 2014.

References :

1. Ronald L. Krutz and Russell Dean Vines, *Cloud Security A Comprehensive Guide to Secure Cloud Computing*, Wiley Wiley Publications, 2010.