SANJAY GHODAWAT UNIVERSITY
KOLHAPUR

(Established as a State University under Government of Maharashtra Act No XL dated 3rd May 2017)

Third Year B. Tech. Computer Science and Engineering

Curriculum Structure

Sanjay Ghodawat University, Kolhapur

School of Technology

Department of Computer Science and Engineering

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

AY 2024-25

Semester - V											
Course Code	Course Title	L	T	P	Cr	Evaluation Scheme					
						Component	Exam	Marks	Wt.%	Min Pass (%)	Pass (%)
UCS501 (PC)	Theory of Computation	3	1	-	4	Theory & Practical	FA	50	50	40	40
							SA	100	50	40	
UCS502 (PC)	Design and Analysis of Algorithms	3	-	2	4	Theory & Practical	FA	20	20	40	40
								30	30		
							SA	100	50	40	
UCS503 (PC)	Software Proficiency I	2	-	4	4	Practical	FA	100	50	40	40
							SA	50	50	40	
UCS5X (PE)	Program Elective II	3	-	2	4	Theory & Practical	FA	20	20	40	40
								30	30		
							SA	100	50	40	
UST5XX	Open Elective I	3	-	2	4	Theory & Practical	FA	20	20	40	40
								30	30	40	
							SA	100	50	40	
UNM009	Yoga for Physical Health and Mental Health	-	-	-	NC	-	-		-	-	-
Total		**14**	**1**	**10**	**20**	**Total Hours: 25 Hrs.; Total Credits: 20**					

FAT – Formative Assessment Theory; FAP - Formative Assessment Practical; ESE - End Semester Examination; FC – Fundamental Core; PC – Program Core ; PE - Program Elective; UC - University Core; UNCMC- University Mandatory Non- Credit Course

Program Electives (SEM – V)

Course Code UCS5X	Program Elective - II
UCS555	Wireless Sensor Networks
UCS556	Introduction to AI
UCS557	Computer Vision
UCS558	Digital Forensics

Open Elective Courses			
Sr.No	**Semester**	**Course Code**	**Course Name**
1	V	UST555	Network Security
2	VI	UST565	Cloud Security
3	VII	UST575	Cybersecurity Law and Ethics
4	VIII	UST585	Digital forensics
5.	VIII	UST586	Ethical Hacking

Sanjay Ghodawat University, Kolhapur

School of Technology

Department of Computer Science and Engineering

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

AY 2024-25

Semester - VI											
Course Code	**Course Title**	**L**	**T**	**P**	**Cr**	**Evaluation Scheme**					
						Component	**Exam**	**Marks**	**Wt.%**	**Min Pass (%)**	**Pass (%)**
UCS601 (PC)	Advance Database ManagementSystems	3	-	2	4	Theory & Practical	FA	20	20	40	40
								50	30		
							SA	100	50	40	
UCS602 (PC)	Compiler Design	3	-	2	4	Theory & Practical	FA	20	20	40	40
								30	30		
							SA	100	50	40	
UCS603 (PC)	Machine Learning	2	-	4	4	Theory & Practical	FA	20	20	40	40
								50	30		
							SA	100	50	40	
UCS6X (PE)	Program Elective III	3	-	2	4	Practical	FA	20	20	40	40
								30	30		
							SA	100	50	40	
UST5XX	Open Elective II	3	-	2	4	Theory & Practical	FA	20	20	40	40
								30	30		
							SA	100	50	40	
UNM010	Creativity and Innovations	-	-	-	NC	-	-		-	-	-
	Total	**14**	**-**	**12**	**20**	**Total Hours: 26 Hrs.; Total Credits: 20**					

FAT – Formative Assessment Theory; FAP - Formative Assessment Practical; ESE - End Semester Examination; FC – Fundamental Core; PC – Program Core; PE - Program Elective; UC - University Core; UNCMC- University Mandatory Non- Credit Course

Program Electives (SEM – VI)

Course Code UCS6X	**Program Elective - III**
UCS659	Mobile Computing
UCS660	Intelligent Systems
UCS661	High Performance Computing
UCS662	Ethical Hacking

Index

SANJAY GHODAWAT UNIVERSITY
KOLHAPUR

Curriculum Contents

UCS501: Theory of Computation								
Program Core, School of Technology								
Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	Weightage %	Min Pass %	
3	1	-	4	Theory & Practical	FA	50	40	40
					SA	50	40	

Course Description:
This course is at even semester of third year B Tech computer science and engineering. It is a core course in computer science and may be pre requisites for other courses in computer science. It covers basics of regular languages, finite automata, grammar, languages and conceptsTuring machines.

Prerequisite: Discrete structures and basics of mathematics

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

CLO1 **Design**[3] finite automata to recognize a regular language.

CLO2 **Design**[3] a Simplified Context Free Grammar for any language.

CLO3 **Describe**[2] Push Down Automata and Context Free Language.

CLO4 **Construct**[5] Turing Machine to recognize a given language.

Syllabus (Theory)

Units	Description	Hrs.
I	**Mathematical Induction, Regular Languages & Finite Automata:** Introduction to Finite Automata, Types of Proofs, The Principle of Mathematical Induction Type of Proof, Descriptive and Recursive Definitions of Language, Regular Languages and Regular Expressions, Deterministic Finite Automata (DFA): Definition and Representation, Unions, Intersection & Complements of Regular Languages, Equivalence of FA's.Non-Deterministic Finite Automata: Definition, Representation, Acceptance of Language, Equivalence of NFA and FA. Non-Deterministic Finite Automata (NFA) and NFA with null transitions: Need, Definition, Representation, Acceptance of Language, NFA and NFA - Null.	**07**

II	**Kleene's Theorem:** Kleene's Theorem Part I and II Statements and Proofs, Minimum State of FA for a Regular Language : Minimizing Number of States In Finite	**07**
III	**Grammars and Languages Part - I** Introduction to Grammar, Types of Grammars, Regular Grammar and DFA, Context Free Grammar: Definition, Representation, Types of Derivation, Derivation of String and Ambiguity. Simplification of Context FreeGrammar: Eliminating useless and dead variables from a CFG, Eliminating Null Production and Unit productions from CFG.	**07**
IV	**Grammars and Languages Part - II** Context Free Language (CFL) - Definition, Closure Properties of CFL - Union, Concatenation and *'s, of CFLs, Writing CFG for a given language. Chomsky Normal Form (CNF): Definition, Need, Convert the Given CFG to CNF Form: Algorithm and Solving Examples.	**07**
V	**Push Down Automata and Context Free Language** Definition, Representation, Types of PDA: Deterministic PDA and Non-Deterministic PDA, Types of Acceptance, Instantons Descriptions, Design of PDA, Equivalence of CFG's and PDAs, **CFL's and non CFL's:** Pumping Lemma: Statement and Proof, Solving Examples, Properties of Context Free Languages	**07**
VI	**Turing Machines:** Introduction, Definition, Representation, A String Acceptance by TM, Instantons Descriptions for TM, Design of Turing Machine for given Language, Variations in Turing Machines, Turing Machine and Computing Function.	**07**

Tutorial

One hour per week per batch tutorial is to be utilized to ensure that students have properly learntthe topics covered in the lectures. It should comprise of minimum 8-10 tutorials. Students of different batches should perform different tutorials based on the following guidelines-

1. Writing decrypting and recursive definition of language, solving problems on DFA and regular expressions.
2. Solving problems on equivalence of NFA and DFA, NFA and NFA - Null.
3. Kleene's Theorem Part - I and Part - II.
4. Minimum State of FA for a regular language, minimizing number of states in FA

5. Derivation, Ambiguity and Simplification of CFG.
6. Designing of a CFG for a given language and convert the given CFG to ChomskyNormal Form.
7. Design a PDA for a given CFL and explain types of acceptance.
8. Pumping Lemma and examples, CFL's and non CFL's
9. Designing a Turing Machine (for given language / problems) – I
10. Designing a Turing Machine (for given language / problems) - II

Text Book

1. John C. Martin, "Introduction to Languages & Theory of Computations", 2nd Edition, McGraw-Hill Education, May 2010.

References

1. Hopcroft J.E., Motwani R. and Ullman J.D, "Introduction to Automata Theory,Languages and Computations", Second Edition, Pearson Education, 2008.
2. Michael Sipser, "Introduction to the Theory of Computation" Cengage Learning,2012.
3. Peter Linz, "An introduction to formal languages and automata", Jones & BartlettLearning, 2001.

****

UCS502: Design and Analysis of Algorithms								
Program Core, School of Technology								
Lect.	**Tut.**	**Pract.**	**Credits**	**Evaluation Scheme**				
				Component	**Exam**	**Weightage %**	**Mini Pass %**	
3	-	2	4	Theory & Practical	FA	20%	40%	40%
						30%		
					SA	50%	40%	

Course Description:

The objective of this course is to provide third-year Computer Science and Engineering students with a comprehensive understanding of the principles, techniques, and methodologies involved in the design, analysis, and evaluation of algorithms. Through theoretical discussions, practical implementations, and problem-solving exercises, students will develop a strong foundation in algorithmic thinking, enabling them to tackle complex computational challenges effectively

Prerequisite: - Data Structures.

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

CLO1 Solve[3] real time problems based on different algorithmic strategies.

CLO2 Analyse[4] the complexity of different algorithms based on different techniques

CLO3 Classify[4] real time problems into different algorithmic techniques.

CLO4 Compare[4] different algorithms based on different techniques

Syllabus (Theory)

Units	**Description**	**Hours**
I.	**Introduction to Algorithms:** Introduction to Algorithm, Growth of **Functions**- Solving Recursive Equation: Substitution method, Iteration Method and Master Method. Divide and Conquer: Finding maximum and Minimum, Selection, Stassen's matrix Multiplication.	**0 7**
II.	**Greedy Algorithms :** Greedy Approach-General Method, Knapsack Problem, Minimum cost spanning tree- Prim's and Kruskal's algorithm, Single Source Shortest Path.	**0 7**

III. **Dynamic Programming :** Principle of Optimality, All Pair Shortest Path, Longest Common Sequence, Optimal binary search algorithm, Travelling Salesman Problem, Reliability Design **07**

IV. **Backtracking:** General Method, 8-Queen Problem, Sum-of-Subnet Problem, Hamilton Cycle, Branch and Bound Knapsack Problem, Travelling Salesman Problem. **07**

V. **String Matching and Parallel Algorithm: Simple** String matching, The naive string-matching algorithm, The Rabin-Karp algorithm, PRAM Computation Model, Fundamental techniques, MESH-Computation model, Packet Routing, Fundamental techniques, HYPERCUBE- Computation model, PPR Routing, Fundamental techniques. **07**

VI. **NP-Hard and NP-Complete Problems :** Basic concept of N, NP, NP-Hard,NP-Complete, NP-Hard Graph Problems-Clique Decision Problem(CDP), Node Cover Decision Problem (NCDP), Chromatic number decision problem (CNDP), Directed Hamiltonian Cycle (DHC), Traveling salesman Problem (TSP), AND/OR Graph decision Problem (AOG). **07**

Practical Syllabus

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 programs based on recursive algorithms.
2. Write program for selection problem.
3. Write program to implement Knapsack problem using greedy technique.
4. Write program to implement Prim's/Kruskal's algorithm.
5. Write program to implement Single-Source Shortest Path.
6. Write program to implement All- Pair Shortest Path.
7. Write program to implement Travelling Salesman problem.
8. Write program to implement 8-Queens problem.
9. Write program to implement Hamiltonian cycle problem.
10. Write program to implement Knapsack using Branch and Bound technique.
11. Write program to implement Naive string-matching algorithm.
12. Write program to implement the Rabin-Karp algorithm.
13. Write program to implement List ranking in PRAM.
14. Write program to implement Prefix Computation in MESH.

Textbooks:

1. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Second Edition, Universities Press, Hyderabad, 2008.
2. Thomas H Cormen, Charles E Leiserson, Ronald L Rivest and Clifford Stein, "Introduction to Algorithms", Second Edition, Prentice Hall of India, New Delhi, 2007.

References:

1. Kenneth A. Berman and Jerome L. Paul, "Algorithms", Cengage learning India Edition, New Delhi, 2002.
2. Sara Baase and Allen Van Gelder, "Computer Algorithms – Introduction to Design & Analysis", Third Edition, Pearson Education, New Delhi, 2000.

****

UCS503: Software Proficiency I								
Program Core, School of Technology								
Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	Weightage %	Mini Pass %	
2	-	4	4	Practical	FA	50	40	40%
					SA	50	40	

Course Description:

The objective of this short course is to introduce third-year Computer Science and Engineering students to the fundamental concepts, syntax, and applications of the Python programming language. Through a combination of theoretical explanations, hands-on coding exercises, and practical projects, students will acquire proficiency in Python programming and develop the skills necessary to leverage Python for various computational tasks

Prerequisite: - Knowledge of Object oriented Programming Concepts.

Syllabus (Theory)

Units	Description	Hours
I.	**Basic concepts and functions in Python:** Introduction to Python, the application areas of Python, Python Basics, Data types and variables, Data input, Comments, String, List, Tuple, Dictionary, Set, Control Statements, Range, Strings, introduction to function, define a function, pass arguments, different type for arguments, Local and global variables, return a value from function, Return multiple values, Lambda Functions.	07
II.	**Object Oriented concepts in Python:** Class definition, creating objects, Constructors, accessing attributes, Built-in class attributes, Destructors, Inheritance, Overriding, Overloading, what is exception? Python built-in exceptions, Try-Except-Finally, raise exceptions, User defined exception.	08
III.	**File and Database Handling:** Reading, Writing, File manipulations, Directories, performing different database operations like Select, Insert, Update, Delete using MySql and Python connectivity	07
IV.	**NumPy and Pandas:** Create, access, modify, and sort multidimensional NumPy arrays (ndarrays), Load	08

and save ndarrays, Use slicing, Boolean indexing, and set operations to select change subsets of an ndarray, understand difference between a view and a copy ndarray, perform element-wise operations on ndarrays, Use broadcasting perform operations on ndarrays of different sizes, Create, access, and modify t main objects in Pandas, Series and DataFrames, Perform arithmetic operations Series and DataFrames, Load data into a DataFrame, Deal with Not a Numb (NaN) values.

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

CLO1	**Explain[2]** data types and control statements in Python.
CLO2	**Make use of[3]** functions and object oriented concepts to develop basic programs in Python.
CLO3	**Apply3** file handling techniques and CRUD operations for given problems in Python**.**
CLO4	**Construct[3]** solutions to different problems using NumPy and pandas.

Syllabus (Practical)

Four hours per week per batch practical is to be utilized for writing programs to ensure that students have properly learnt the topics. It should comprise of minimum of 13-15 experiments. Students of different batches should implement different programs based on following guidelines.

1. Introduction to Python, Setup, working in Python shell and Jupyter Notebook, Basic I/O.
2. Implement list, tuple, set and dictionary.
3. Implement String methods.
4. Implement Iterator, Conditionals and Loops.
5. Implement functions and recursive functions.
6. Implement lambda functions.
7. Implement a basic Object Oriented program using Classes, Objects.
8. Implement Inheritance and Method Overriding.
9. Implement Exception handling.
10. Implement File handling: Open, close, read, write and append data using file.
11. Implement programs based on Python and MySql Database connectivity.
12. Implement Python programs using NumPy. (e. g. Multiplication of two Matrices in Single line)
13. Implement Python programs based on Pandas library.

Textbooks:

1. Charles R. Severance," Python for Everybody: Exploring Data using Python 3",1st edition, CreateSpace Independent Publishing Platform, 2016, (http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf)(chapters 1-13,15)

2. Sandeep Rakshit, “R for beginners”, Mcgraw Hill, Edition: 1, 2017, Pages:424, Manav Book Distributors

References:

1. Allen B.Downey, ”Think python: How to Think like a computer Scientist” 2nd edition, Green Tea Press,2015

****

UCS555: Wireless Sensor Network							
Program Core, School of Technology							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage	Pass %
3	-	2	4	Theory & Practical	FA	20	40
						30	
					SA	50	40

Course Description:

A wireless sensor network is an emerging area that employs a network of sensor nodes that are tiny in size and battery operated. They are required to be used very power efficiently so that network lifetime is optimized and provide reliable communication to the base station. By learning this subject the students will gain the recent trends of wireless sensor networks, their design limitations and challenges.

Prerequisite: Computer Networking

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

CLO1 **Demonstrate** [3] basic concept of WSNs, supportive technologies and radio transmission issues

CLO2 **Compare**[4] requirements of access control protocols and media access control techniques

CLO3 **Apply** [2] different strategies used to develop routing protocols for WSN

CLO4 **Determine**[4] operating systems for wireless sensor networks and design.

Syllabus (Theory)

Units	Description	Hours
I.	**Introduction and Overview of Wireless Sensor Networks** Introduction, Basic Overview of the Technology, Applications of Wireless Sensor Networks, Another Taxonomy of WSN Technology	7
II.	**Basic Wireless Sensor Technology** Sensor Node Technology, Hardware and Software, Sensor Taxonomy, WN Operating environment, , Wireless Transmission Technology and Systems, Radio Technology Primer, Available Wireless Technologies, Medium Access Control Protocols for Wireless Sensor Networks	7
III	**Routing Protocols for Wireless Sensor Networks**	

	Data Dissemination and Gathering, Routing Challenges and Design Issues in Wireless Sensor Networks, Routing Strategies ,Transport Control Protocols for Wireless Sensor Networks, Traditional Transport Control Protocols, Transport Protocol Design Issues, Examples of Existing Transport Control Protocols	**07**
IV	**WSN Middleware Principles** Middleware Architecture, Network Management for Wireless Sensor Networks, Network Management Requirements, Traditional Network Management Models, Network Management Design Issues, Example of Management Architecture: MANNA	**07**
V	**Operating Systems for Wireless Sensor Networks** Operating System Design Issues, Examples of Operating Systems, TinyOS, Mate, MagnetOS, MANTIS, OSPM, EYES OS, SenOS, EMERALDS, PicOS	**07**
VI	**Performance and Traffic Management** Introduction, Background, WSN Design Issues, MAC Protocols, Routing Protocols, Transport Protocols, Performance Modeling of WSNs, Performance Metrics, Basic Models, Network Models, Case Study: Simple Computation of the System Life Span	**07**

Syllabus (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. Exploring Wireless Sensor Networks: An Introductory Journey
2. Demystifying Basic Wireless Sensor Technology
3. Navigating Routing Protocols in Wireless Sensor Networks
4. Understanding WSN Middleware Principles
5. Unveiling Operating Systems for Wireless Sensor Networks
6. Optimizing Performance and Traffic Management in WSNs
7. Wireless Sensor Networks: A Comprehensive Overview
8. Essential Components of Wireless Sensor Technology
9. Mastering Routing Protocols for Efficient Data Handling
10. Harnessing Middleware Principles for WSN Management
11. Selecting Operating Systems for Your Wireless Sensor Network
12. Enhancing Performance and Traffic Management in WSN Environments

Textbooks

1. Waltenegus Dargie, Christian Poellabauer, "Fundamentals of Wireless Sensor Networks, Theory and Practice", 1st Edition, John Wiley & Sons, 2007.
2. Kazem Sohraby, Daniel Manoli "Wireless Sensor Networks- Technology, Protocols and Applications", 1st Edition ,Wiley InterScience Publications, 2010.

References

1. Bhaskar Krishnamachari, "Networking Wireless Sensors", 1st Edition, Cambridge University Press, 2005
2. C. S Raghavendra, Krishna M. Sivalingam, Taieb Znati," Wireless Sensor Networks", 1st Edition, Springer Science, 2004
3. Edgar H. Callaway, Jr. and Edgar H. Callaway," Wireless Sensor Networks: Architectures and Protocols", 1st Edition, CRC Press, 2003.

****

UCS556:Introduction to AI								
University Core (Non-Credit), School of Technology								
Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	Weightage	Mini. Pass %	
3	-	2	4	Theory & Practical	FA	20	40	40%
						30		
					Sa	50	40	

Course Description:

This course serves as an introduction to the field of Artificial Intelligence (AI) for computer engineering students. It covers foundational concepts, algorithms, and applications of AI, providing students with the necessary knowledge and skills to understand, design, and implement AI systems.

Course Learning Outcomes:

At the end of this course students will able to

CLO1 **Explain[2]** concepts and principles of Artificial Intelligence.

CLO2 **Apply [3]** common AI techniques and algorithms for given problem.

CLO3 **Apply [3]** AI techniques to solve engineering problems

CLO4 **Explain[4]** real-world applications of AI across various domains.

SYLLABUS

UNITS	DESCRIPTION	HOURS
I	Definition of AI, birth of AI, brief history, Turing test, Types of environment, Types of agents, PEAS(Performance measure , Environment, Actuators, Sensors),	7
II	**Introduction to searching**, State Space, SAGP (State, Action, Goal test, Path cost), DFS, BFS (Completeness, Time complexity, Space complexity, Optimality), Heuristics, Local Search Algorithm, Hill Climbing. Applications of Artificial Intelligence in real word.	**7**

III	**SP, Game Playing and Logics** Constrain Satisfaction Problems examples, Approaches to solve CSPs, Test and generate method, back tracking. Game Playing, Optimal decision in games, Min Max algorithm, Evaluation functions,	7
IV	**Introduction Logic** Propositional Logic and First Order Logic, Syntax, Substitution, Unification, Deduction, Soundness, Completeness, Consistency, Satisfiability, Expert Systems.	7
V	Probabilistic Reasoning, Review of Probability Theory, Probabilistic Inference Rules, Bayes Theorem, examples of Bayes theorem, Introduction to Learning, Taxonomy of Learning Systems, Concept Learning, Find-S algorithm, Candidate Elimination Algorithm.	7
VI	**Introduction to Neural Networks** Biological Neural Networks, Artificial Neural Networks, Perceptron, Perceptron Learning Rule, Delta Rule, Applications of Neural Networks.	7

Syllabus (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

Sr. No	Experiment Description
1	Implement a simple program to simulate the Turing test using user input.
2	Implement Depth-First Search (DFS) and Breadth-First Search (BFS) algorithms in Python.
3	Implement a simple game (e.g., Sudoku) and solve it using the backtracking algorithm.
4	Develop a Python program to demonstrate the Min-Max algorithm for optimal decision-making in games.
5	Implement Python code to perform syntax checking, substitution, unification, deduction, and other logic operations.
6	Implement simple expert system using Python to make logical deductions based on given facts and rules.
7	Implement Python functions to calculate probabilities and apply Bayes' theorem.
8	Implement Python algorithms for concept learning using the Find-S algorithm and the Candidate Elimination Algorithm.
9	Implement a basic perceptron model in Python using the perceptron learning rule and the delta rule..
10	Explore applications of neural networks in Python, such as pattern recognition or

classification tasks.

Textbook:

1. Stuart Russell and Peter Norvig – Artificial Intelligence A Modern Approach, PEARSON Education.
2. Simon Haykin -Neural Networks PHI.
3. Python Crash Course: A Hands-On, Project-Based Introduction to Programming by Eric Matthes

References

1. N. P. Padhy – Artificial Intelligence and Intelligence Systems, OXFORD publication.
2. B. YagnaNarayana - Artificial Neural Networks, PHI

****

UCS557 Computer Vision								
Program Core, School of Technology								
Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	Weightage	Mini Pass %	
3	-	2	4	Theory & Practical	FA	20	40	40%
						30		
					SA	50	40	

Course Description: This course is at odd semester of Third year B Tech Technology for Computer Science Engineering. It is a foundation course in Numerical method and Fuzzy Mathematics and may be pre requisites for other courses. It covers solution of algebraic and transcendental equation by different methods, Numerical differentiation and Numerical integration, Introduction to fuzzy set and Fuzzy equations.

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

CLO1 **Explain[2]** fundamental concepts of digital image processing require for image enhancement.

CLO2 **Choose[3]** algorithms to build solutions to the real world computer vision problems.

CLO3 **Demonstrate[3]** different techniques of image segmentation with justification.

CLO4 **Anlyze[4]** different Image Processing and text processing techniques**.**

Syllabus (Theory)

Units	Description	Hours
I	**Digital Image Fundamental** Introduction: Concept, Fundamental S t e p s and Components of Image Processing System, Image Acquisition, A simple image model, Sampling and Quantization, Imaging Geometry, Different types of digital images	**07**
II	**Image Transforms** 2D systems and Necessary Mathematical preliminaries, 2D Orthogonal and Unitary Transforms, 1-D DFT, KL-Transforms, Cosine, Hadamard Transforms, Introduction to Wavelet transforms.	**07**
III	**Image Enhancement** Point Processing, Basic Gray Level Transformations, Histogram processing, Spatial domain Filtering, Frequency domain filtering.	**07**

IV	**Image Segmentation and Analysis** Edge Detection – using first and second order derivatives, LoG, Canny edge detector, Boundary Extraction – Connectivity, Heuristic Graph Search, Hough Transform, Active Contour, Watershed Transform Region-based Segmentation – region growing, region splitting and merging, Feature Extraction	**07**
V	**Color Image Processing** Color Fundamentals, Color models, Gray level to color transformations Basics of Color Image Processing, Color Transformations, Smoothing and Sharpening, Color Segmentation.	**07**
VI	**Texture Analysis:** Definition, Types of texture, Texel's, Texture analysis concept and categories, Approaches to texture analysis, Statistics, Texture descriptors - statistical - Auto-correlation, co-occurrence matrices and features, edge density and direction, local binary partition, Law's texture energy measures, Wavelets and texture analysis.	**07**

Syllabus (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

Sr. No	Experiment Description
1	Demonstrate the concept of sampling and quantization through practical examples
2	Implement 2D orthogonal and unitary transforms using MATLAB or similar software. Compute 1-D Discrete Fourier Transform (DFT) of images
3	Implement basic gray level transformations and histogram processing techniques
4	Implement edge detection algorithms using first and second-order derivatives, LoG, and Canny edge detector
5	Implement active contour and watershed transform algorithms
6	Implement region growing and region splitting and merging algorithms for segmentation
7	Implement gray level to color transformations.
8	Implement color transformations, smoothing, and sharpening techniques to color images
9	Implement texture analysis techniques using statistical methods such as auto-correlation and co-occurrence matrices.
10	Calculate edge density and direction in textures

Textbooks:

1. Gonzalez R. C., Woods R. E., "Digital Image Processing", PHI, Second Edition. 2002
2. Sonka Milan, Vaclav Hlavac, Boyle, "Digital Image Processing and Computer Vision", Cengage Learning, Third edition, 2013

References:

1. S. Jayaraman, S. Esakkirajan, T. Veerkumar, "Digital Image Processing", Tata McGraw Hill, Third edition, 2010
2. D. A. Forsyth, J. Ponce, "Computer Vision – A Modern approach", Pearson Education, Prentice Hall, 2005
3. Linda Shapiro, George C. Stockman, "Computer Vision", Prentice Hall, 2000

****

UCS558: Digital Forensics								
Program Core, School of Technology								
Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	Weightage	Mini Pass %	
3	-	2	4	Theory & Practical	FA	50	40	40%
					SA	50	40	

Course Description:
The objective of this course is to provide third-year Computer Science and Engineering students with a foundational understanding of digital forensics principles, techniques, and methodologies. Through theoretical discussions, practical exercises, and case studies, students will develop the skills necessary to investigate digital crimes, analyze digital evidence, and preserve the integrity of digital information.

Prerequisite: - Students should have knowledge of Networking.

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

CLO1 **Explain**[2] the role of digital forensics.

CLO2 **Demonstrate** [3] the computing requirements appropriate to solve given problem.

CLO3 **Classify** [4] the network forensics and methods of investigation using digital forensics techniques.

CLO4 **Adapt**[6] professional, ethical, legal, security and social issues and responsibilities.

Syllabus (Theory)

Units	Description	Hours
I	**Introduction** Introduction of Cybercrime: Types, The Internet spawns crime, Worms versus viruses, Computers' roles in crimes, Introduction to digital forensics, Introduction to Incident - Incident Response Methodology – Steps - Activities in Initial Response, Phase after detection of an incident.	07

II	**Initial Response and forensic duplication** Initial Response & Volatile Data Collection from Windows system - Initial Response & Volatile Data Collection from Unix system – Forensic Duplication: Forensic duplication: Forensic Duplicates as Admissible Evidence, Forensic Duplication Tool Requirements, Creating a Forensic. Duplicate/Qualified Forensic Duplicate of a Hard Drive.	**07**
III	**Preserving and Recovering Digital Evidence:** File Systems: FAT, NTFS - Forensic Analysis of File Systems – Storage Fundamentals: Storage Layer, Hard Drives Evidence Handling: Types of Evidence, Challenges in evidence handling, Overview of evidence handling	**07**
IV	**Network Forensics:** Intrusion detection; Different Attacks in network, analysis Collecting Network Based Evidence - Investigating Routers - Network Protocols - Email Tracing- Internet Fraud procedure	**07**
V	**System investigation:** Data Analysis Techniques - Investigating Live Systems (Windows & Unix) Investigating, Hacker Tools - Ethical Issues – Cybercrime.	**07**
VI	**Bodies of law:** Constitutional law, Criminal law, Civil law, Administrative regulations, Levels of law: Local laws, State laws, Federal laws, International laws, Levels of culpability: Intent, Knowledge, Recklessness, Negligence Level and burden of proof: Criminal versus civil cases, Vicarious liability, Laws related to computers: CFAA, DMCA, CAN Spam, etc.	**07**

Syllabus (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

Sr. No	Experiment Description
1	Acquisition of volatile memory using various tools like Volatility, etc.
2	Setup of memory forensic environment and extract various artifacts from memory dump and analyze the memory dump, Using different tools like Volatility, LiME, etc
3	Windows artifact analysis using different forensic tools, which includes MRU, link file, USB analysis, Prefetch analysis, shell bag, web cache etc.,

4	Analysis of UNIX based and other operating systems using EnCase, FTK, Autopsy and similar tools.
5	Perform forensic analysis of Multimedia Files and CCTV Footage analysis.
6	Using APKTool to reverse an Android application, Auditing Android applications.
7	Perform the following on different Android Image files: • Using a custom recovery android image. • Using AFLogical to extract contacts, calls, and text messages. • Dumping application databases manually. • Logging the logcat and using backup to extract an application's data.
8	Developing your first iOS app and running apps on iDevice.
9	Pentest using iOS Simulator and identifying the Flaws in Local Storage
10	Perform traffic analysis for iOS Applications by: Intercepting traffic over HTTP, HTTPS, iOS Simulator
11	Perform physical acquisition of iOS devices physical and Data backup acquisition using Elcomsoft iOS Forensic Toolkit (EIFT), Open source and free tools.

Textbook:

1. Kevin Mandia, Chris Prosise, “Incident Response and computer forensics”, Tata McGraw Hill, 2006
2. Peter Stephenson, "Investigating Computer Crime: A Handbook for Corporate Investigations", Sept 1999
3. Eoghan Casey, "Handbook Computer Crime Investigation's Forensic Tools and Technology", Academic Press, 1st Edition, 2001

References:

1. Skoudis. E., Perlman. R. “Counter Hack: A Step-by-Step Guide to Computer Attacks and Effective Defenses”, Prentice Hall Professional Technical Reference. 2001
2. Norbert Zaenglein, "Disk Detective: Secret You Must Know to Recover Information from a Computer", Paladin Press, 2000
3. Bill Nelson, Amelia Philips and Christopher Steuart, “Guide to computer forensics investigation”, Cengage learning , 4th edition, 2013

****

UCS505(UOE501):Distributed Parallel Computing							
Program Elective, School of Technology							
Lect.	**Tut.**	**Pract.**	**Credits**	**Evaluation Scheme**			
				Component	**Exam**	**Weightage**	**Pass %**
3	-	2	4	Theory & Practical	FA	50	40
					SA	50	

Course Description:
The course aims to introduce students to the principles, techniques, and challenges of Parallel and Distributed Computing. It covers various aspects such as parallel programming platforms, parallel algorithm design principles, and CUDA programming model, analytical modeling of parallel programs, dense matrix algorithms, graph algorithms, and search algorithms for discrete optimization problems.

Prerequisite: Computer Networking, Wireless Adhoc Networks

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

CLO1 Describe[2] the Scope and Challenges of Parallel and Distributed Computing.
CLO2 Implementing[3] parallel algorithms, including decomposition techniques, load balancing strategies.
CLO3 Develop[4] parallelized code for GPUs (Graphics Processing Units), including isolating data, allocating memory, transferring data between host and device.
CLO4 Analyze[4] Performance and Scalability of Parallel Programs

Syllabus (Theory)

Units	**Description**	**Hrs.**
I	**Introduction:** Scope , issues, applications and challenges of Parallel and Distributed Computing, Parallel Programming Platforms: Implicit Parallelism: Trends in Microprocessor Architectures, Dichotomy of Parallel Computing Platforms, Physical Organization, Communication Costs in Parallel Machines, Routing Mechanisms for Interconnection Networks, GPU, co-processing	**07**
II	**Principles of Parallel Algorithm Design**: Decomposition Techniques, Characteristics of Tasks and Interactions, Mapping Techniques for Load Balancing.	**07**

III	**CUDA programming model:**	**07**
	Overview of CUDA, Isolating data to be used by parallelized code,API function to allocate memory on parallel computing device, to transfer data, Concepts of Threads, Blocks, Grids, Developing a kernel function to be executed by individual threads, Execution of kernel function by parallel threads, transferring data back to host processor with API Function.	
IV	**Analytical Modeling of Parallel Programs:**	**07**
	Sources of Overhead in Parallel Programs, Performance Metrics for Parallel Systems, The Effect of Granularity on Performance, Scalability of Parallel Systems, Minimum Execution Time and Minimum Cost- Optimal Execution Time	
V	**Dense Matrix Algorithms:**	**07**
	Matrix-Vector Multiplication, Matrix-Matrix Multiplication, Issues in Sorting on Parallel Computers, Bubble Sort and Variants, Quick Sort,Other Sorting Algorithms **Graph Algorithms**: Minimum Spanning Tree: Prim's Algorithm, Single-Source Shortest Paths: Dijkstra's Algorithm, All-Pairs Shortest Paths, Transitive Closure, Connected Components, Algorithms for Sparse Graph	
VI	**Search Algorithms for Discrete Optimization Problems**: Sequential Search Algorithms, Parallel Depth-First Search, Parallel Best-First Search, Speedup Anomalies in Parallel Search Algorithms **Laboratory Work :**To Implement the algorithms with the help of CUDA programming using parallel and distributed programming techniques	**07**

Syllabus (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. Parallel Odd-Even Transposition Sort
2. Mandelbrot Set Computation
3. N-Body Simulation
4. Heat Distribution Simulation
5. Sample MPI program in C
6. Implement simple cluster using master and slave node
7. Divide and conquer parallel computing program using C

8. Implement sample program for load balancing
9. OpenMP concepts implementation
10. Implement distributed shared memory

Textbooks:

1. A Grama, A Gupra, G Karypis, V Kumar. Introduction to Parallel Computing (2nd ed.). Addison Wesley, 2003.
2. C Lin, L Snyder. Principles of Parallel Programming. USA: Addison-Wesley Publishing Company, 2008.

References:

1. J Jeffers, J Reinders. Intel Xeon Phi Coprocessor High-Performance Programming. Morgan Kaufmann Publishing and Elsevier, 2013.
2. T Mattson, B Sanders, B Massingill. Patterns for Parallel Programming. Addison-Wesley Professional, 2004.

****

UNCMC: Yoga for Yoga for Physical Health and Mental Health and Mental Health								
Program Elective, School of Technology								
Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	Weightage	Mini Pass %	
-	-	-	-	Theory & Practical	FA	50	40	40%
					SA	50	40	

Course Description:

The main objective of this course is to enhance students understanding of the Yoga for Physical Health and Mental Health to create a better-informed person, which will lead to a more batter action by all and for all.

Prerequisite:

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

CLO1 Describe[2] the different Challenges of Physical Health and Mental Health in students.
CLO2 Implementing[3] yoga for human physical fitness.
CLO3 Develop[4] routine for the body and mind by following a better lifestyle
CLO4 Analyze[4] career option and hence a source of income due to the growing demands of Healthy and fit body and a stable mental health among all age groups.

Syllabus (Theory)

Units	Description	Hrs.
I	**Introduction:** Health definition according to W.H.O., Mental Health (Its meaning, determinates and application). Concept of normality in Yoga and modem psychology Personality theories with special reference to the Freudian, Malso, Mere	**10**
II	**Concept of personality** - Indian approach to personality. Samkhya – Yoga.Upanished, Pancha kosha. Sanskara.	**10**
III	**Personality integration** from the view point of Yoga. Satva, Rajas, Tamas, Deviya & Asuri Pravrati.Attitude formation through Yamas and Niyamas & Yogic Philosophy.Rationale selected yogic practices and their contribution to physical & Mental Health	**09**

IV	**Tackling ill effects of frustration**; General Introduction, Meaning, Cause & Yogic method Anxiety. General Introduction, Meaning, Causes Yogic method. Conflict. Other methods contribution of Yoga in personal. Interpersonal adjustment.	**09**
V	**Prayer**; Meaning, Introduction & its significance in yogic practices. Role of Prayer in daily life. Concept of Prayer in all religion. Concept & Define. Role & effects of Mantra in Yogic Practices.	**09**

References:

1. Rath, S.S. Physical Fitness and Wellness
2. Gore, M.M. Anatomy & Physiology of Yogic Practices
3. Yatendra, A.Yoga & Stress management
4. International Day of YOGA, common protocol: Ministry of Ayurveda,Yoga & Naturepathy, Unani, Siddha and Homeopathy (AYUSH

****

Sanjay Ghodawat University, Kolhapur

Third Year Department of Computer Science and Engineering

AME/P/00

Structure and Contents for T.Y. B. Tech. Computer Science and Engineering Program

Semester - VI											
Course Code	**Course Title**	**L**	**T**	**P**	**Cr**	**Evaluation Scheme**					
						Component	**Exam**	**Marks**	**Wt.%**	**Min Pass (%)**	**Pass (%)**
UCS601 (PC)	Advance Database ManagementSystems	3	-	2	4	Theory & Practical	FA	20	20	40	40
								50	30		
							SA	100	50	40	
UCS602 (PC)	Compiler Design	3	-	2	4	Theory & Practical	FA	20	20	40	40
								30	30		
							SA	100	50	40	
UCS603 (PC)	Machine Learning	2	-	4	4	Theory & Practical	FA	20	20	40	40
								50	30		
							SA	100	50	40	
UCS6___ (PE)	Program Elective III	3	-	2	4	Practical	FA	20	20	40	40
								30	30		
							SA	100	50	40	
UCS605 **(UOE601)**	Advance Cloud Computing	3	-	2	4	Theory & Practical	FA	20	20	40	40
								30	30		
							SA	100	50	40	
UNCMC	Creativity and Innovations	-	-	-	NC	-	-		-	-	-
Total		**14**	**-**	**12**	**20**	**Total Hours: 26 Hrs.; Total Credits: 20**					

FAT – Formative Assessment Theory; FAP - Formative Assessment Practical; ESE - End Semester Examination; FC – Fundamental Core; PC – Program Core ; PE - Program Elective; UC - University Core; UNCMC- University Mandatory Non- Credit Course

Program Electives (SEM – VI)

Course Code UCS6X	**Program Elective - III**
UCS659	Mobile Computing
UCS660	Intelligent Systems
UCS661	High Performance Computing
UCS662	Ethical Hacking

UCS601:Advance Database Management Systems								
Program Elective, School of Technology								
Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	Weightage	Mini Pass %	
3	-	2	4	Theory & Practical	FA	20	40	40%
						30		
					SA	50	40	

Course Description:

The objective of this course is to provide third-year Computer Science and Engineering students with an advanced understanding of database management systems (DBMS) and related concepts. Building upon foundational knowledge acquired in previous database courses, this course delves into advanced topics and emerging trends in the field of database management

Prerequisite: Database System and Programming Experience

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

CLO1: Classify[2] various types of database architectures based on the domain of application.

CLO2: Implement [3] advanced query writing techniques with PL/SQL to solve database problems

CLO3: Analyze4 the use of MongoDB and XML databases for information retrieval.

CLO4: Elaborate[6] the effectiveness of NoSQL and Big Data in specific database applications

Syllabus (Theory)

Units	Description	Hrs.
I	**ORDBMS and OODBMS** **S**tructured data types, Operations on structured data, Encapsulation and ADTs, Inheritance, Objects, OIDS and Reference types, Database design for an ORDBMS. Object identity, Nested collections. Introduction to Object Database Management Group (ODMG), Object Definition Language (ODL) and Object Query Language (OQL)- SELECT and sub queries. Comparison of RDBMS, ORDBMS and OODBMS.	**07**
II	**Advanced SQL** PL/SQL- A Basic introduction, Functions and Procedure, Packages, Synonyms, Database Links, Embedded SQL and Dynamic SQL. Database	**07**

Design: systems development life cycle, database life cycle, DBMS Software Selection: top- down versus bottom-up design

III	**Information Retrieval & XML data**	**07**

Introduction to information retrieval, Indexing for Text search. Overview and structure of XML data, Data model for XML, XMLDTD's, Domain specific DTD's, Querying XML data, XML Applications

IV	**NoSQL & Introduction to Big Data**	**07**

Types of digital data: Structured, unstructured and semi-structured. Definition of Big Data. Why big data. Traditional Business Intelligence (BI) vs Big Data. NoSQL: Types, advantages and use of NoSQL in industry. Comparison of SQL, NoSQL and NewSQL. Introduction to Hadoop: Features and key advantages of Hadoop. The Hadoop ecosystem: HDFS and map reduce. Introduction to interacting with Hadoop eco system: Pig, Hive, Scoop, HBase.

V	**MongoDB - A NoSQL Database**	**07**

Definition and need of MongoDB. JavaScript programming, Datatypes in MongoDB. MongoDB Query Language: Insert, save methods, add/remove fields to/from documents, arrays, aggregate functions, Mapreduce function, Mongo import, Mongo export.

VI	**Current Trends In Advanced Databases**	**07**

Introduction to different types of databases and application areas: Multimedia Database, Cloud Databases, Spatial Databases, Temporal Databases, Mobile Databases, Deductive databases

Textbooks:

1. Elmasri Ramez, Navathe Shamkant, "Fundamentals of Database System", Pearson publications, 6th Edition, 2013.
2. P.S. Deshpande, "SQL & PL/SQL for Oracle Black Book", Dreamtechpress, 1st edition, 2012.
3. Seema Acharya and Subhashini Chellappan, "Big data and Analytics", Wiley publications, 1st Edition, 2015.

References:

1. Mario Piattini, Oscar Diaz, "Advanced Database Technology and Design", Artech House Inc, 1st edition, 2000.
2. Raghu Ramkrishanan, Johannes Gehrke, "Database Management Systems", McGraw Hill Publications, 3rd edition, 2003.

Syllabus (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. Develop an application using multi-valued attributes, complex types, procedures and functions in ORDBMS.
2. Implement a database application to study PL/SQL concepts and features.
3. Embed PL/SQL in a high-level host language such as C and demonstrate a transaction application.
4. Create XML, XML schemas, DTD for any database application and implement queries using XQuery and XPath. This can be done in two experiments.
5. Installation of Hadoop.
6. Performing queries to demonstrate use of HDFS and Mapreduce.
7. Design database schemas and implement queries using Hive/ Hbase.

Textbooks:

1. Abraham Silberschatz, Henry Korth, S, Sudarshan , “Database System Concepts”, McGraw Hill International, 6th Edition, 2015.
2. Raghu Ramkrishanan, Johannes Gehrke, “Database Management Systems”, McGraw Hill International, 3rd Edition, 2003.

References:

1. Tom White, “Hadoop: The Definitive Guide”, O’Reilly Publications, 4th edition, 2015.
2. Silberschatz A., Korth H., Sudarshan S, "Database System Concepts”, McGraw Hill Publishers,6th Edition, 2015.
3. Jiawei Han, Micheline Kamber, Jian Pei, “Data Mining: Concepts and Techniques”, Morgan Kaufmann Publishers, 3rd Edition, 2011.
4. http://nosql-database.org/
5. http://www.objectdb.com/database/jdo

****

UCS602: Compiler Design								
Program Elective, School of Technology								
Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	Weightage	Mini Pass %	
3	-	2	4	Theory & Practical	FA	20	40	40%
						30		
					SA	50	40	

Course Description:

The objective of this course is to provide third-year Computer Science and Engineering students with a comprehensive understanding of compiler design principles, techniques, and methodologies. Compiler design plays a crucial role in translating high-level programming languages into executable code, making it an essential aspect of software development and system programming. Through theoretical discussions, practical exercises, and hands-on projects, students will develop the skills necessary to design, implement, and optimize compilers for various programming languages.

Prerequisite: Basics of programming in C, Fundamentals of Theory of Computation:

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

CLO1 **Describe**[2] generation of tokens during the compilation process.

CLO2 **Apply**[3] parsing techniques to parse a string

CLO3 **Simplify**[4] intermediate code using semantic analysis

CLO4 **Apply**[3] different code generation techniques to optimize the code.

Syllabus (Theory)

Units	Description	Hrs.
I	**Introduction** Introduction, Structure of a Compiler, Role of Lexical Analyzer, Lexical Errors, Input Buffering, Sentinels, Specification of Tokens, Recognition of Tokens.	07
II	**Syntax Analysis and Role of Parsers** Role of Parser, Elimination of Left Recursion and Left Factoring. **Top-Down Parser –** Top-Down Parser**,** LL (1) Class of Grammar and Types of Parsers - Recursive Descent Parser with Backtracking, Calculating First and Follow, Predictive Parser: Introduction, formation of parsing table and parsing of a string, Error Recovery in Predictive Parsing.	07

III LR Parsers - I

Bottom-Up Parsing – Introduction to Bottom-Up parsers and types**,** LR (K) Class of Grammar, Handle, Handle Pruning, Shift- Reduce parser and Parsing of String, Conflicts in Shift – Reduce Parser, LR (0) Items, Augmented Grammar, Kernel Items, Viable Prefixes, Closure and Goto function, Canonical Construction of LR (0) Items, Construction of LR (0) parser & SLR parser and parsing of string

IV LR Parsers – II

LR (1) Items and Canonical Construction of LR (1) Items, Construction of More Powerful Parsers – Canonical LR Parser, LALR Parser, Parsing of String, Conflicts in LR Parsers.

V Semantic Analysis and Intermediate Code Generation

Syntax – Directed Definition – Semantics, Attributes and Attributed Grammar, Types of Attributes, Writing Syntax Directed Definition, Annotated Tree, Construction of Syntax Tree, Dependency Graphs, S - Attributed Definition, L- Attributed Definition

VI Code Optimization & Code Generation

Representation of Intermediate Code Using Three Address Code: Quadruples, Triples, Indirect Triples, Back Patching, Principal Sources of Optimization, Control Flow Graphs - Basic Blocks and Flow Graphs, Loops in Flow Graphs, Peephole Optimization, Data Flow Analysis and Equation of Data Flow Analysis.

Textbooks:

1. A.V. Aho, R. Shethi and J.D. Ullman,"Compilers - Principles, Techniques and Tools", Pearson Education

References:

1. D.M. Dhamdare, "Compiler Construction" , Mc-Millan Publication, 2nd Edition.
2. D.M. Dhamdhere, "System Programming and Operating Systems", Tata McGraw HillPublication, 2nd Edition.

Syllabus (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. Remove white spaces (space, tab) and comments from the C/ C++ Program.
2. Implement a program to construct a Symbol Table for the tokens.
3. Demonstrate the working of Lex Tool used in Lexical Analyzer.
4. Implement a program to eliminate left recursion and Left factor of the given input grammar to remove the ambiguity.
5. Implement a program to apply First and Follow functions of LL(1) class of a grammar.
6. Implement a program to demonstrate the working of Top Down parsers.
7. Implement a program to demonstrate the working of LR(0) or SLR
8. Implement a program to demonstrate working of LALR or Canonical LR.
9. Demonstrate the working of Yacc Tool used in Syntax Analyzer.
10. Implement a three address code generator program for the given expression.
11. Implement a program to construct DAG (Directed Acyclic Graph) for expressions
12. Implement a program to construct DAG (Directed Acyclic Graph) for basic blocks.
13. Implement a program to identify the Basic blocks, and loops in a given three address code

Textbooks:

1. Compilers – principles, Techniques and Tools – A. V Aho, R. Shethi and J. D. Ullman (Pearson Education.)
2. Crafting A Compiler with C - Charles Fischer, Richard LeBlanc (Pearson publication)

References:

1. Modern Compiler Design - D. Grune , H. Bal , C. Jacobs , K. Langendoen , Wiley publication, 2nd Edition.
2. Modern Compiler Implementation in Java - Andrew W. Appel , Cambridge University Press 1998,2nd Edition.
3. Compiler Construction by D.M. Dhamdare, Mc-Millan Publication, 2nd Edition.

UCS603: Machine Learning								
Program Elective, School of Technology								
Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	Weightage	Mini Pass %	
2	-	4	4	Theory & Practical	FA	20	40	40%
						30		
					SA	50	40	

Course Description:
The objective of this course is to provide third-year Computer Science and Engineering students with a comprehensive understanding of machine learning principles, algorithms, and applications. Machine learning has emerged as a transformative technology with wide-ranging applications in various domains, including computer vision, natural language processing, robotics, healthcare, finance, and more. Through theoretical discussions, practical exercises, and hands-on projects, students will develop the skills necessary to apply machine learning techniques to solve real-world problems and lay the foundation for further study or research in the field
Prerequisite: Basic concepts of computer algorithm

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

CLO1 **Classify**[3] machine learning techniques.

CLO2 **Implement**[3] machine learning techniques suitable for problem solving.

CLO3 **Solve**[3] the problem using various Deep learning techniques

CLO4 **Describe** [2] various basic concept of Artificial Neural Networks

Syllabus (Theory)

Units	Description	Hrs.
I	**Introduction** Supervised and unsupervised learning, Hypothesis space, Applications of machine learning, Feature selection and extraction, Principal component analysis.	**06**
II	**Supervised learning:** Bias-Variance Dichotomy, Linear regression in one variable: Cost Function, Gradient descent; Linear Regression with Multiple Variables: Gradient descent; Logistic regression, KNN. Bayesian Learning and Decision Trees, SVM, Ensemble Methods	**06**

III	**Unsupervised learning:** Clustering algorithms: K-means, hierarchical clustering, DBSCAN Dimensionality reduction techniques: Principal Component Analysis (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE) Association rule learning: Apriori algorithm Evaluation metrics for unsupervised learning models: silhouette score, Davies-Bouldin index	**06**
IV	**Deep Learning:** Introduction to neural networks: perceptron's, activation functions Building blocks of deep learning: layers, neurons, weights, biases Popular deep learning architectures: Convolutional Neural Networks (CNNs) for image classification, Recurrent Neural Networks (RNNs) for sequential data, and their variants	**06**
V	**Evaluation of Learning Algorithms:** Cross-validation, learning curves, and statistical hypothesis testing. □ Ensemble learning methods: bagging, boosting, stacking Hyper parameter tuning techniques: grid search, random search, Bayesian optimization	**06**
VI	**Machine Learning based Artificial Neural Networks:** Fundamentals of Artificial Neural Networks, Perceptron, Model of Neuron in an ANN, Backpropagation.	**06**

Textbooks:

1. Coursera online course byAndre NG, on Machine Learning.
2. http://www.stanford.edu/class/cs229/materials.html
3. T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning (2ed.), 2008.

References:

1. Christopher Bishop, "Pattern Recognition and Machine Learning", 2016
2. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, "Introduction to Statistical Learning", Springer, 2013
3. Richard Duda, Peter Hart, David Stork, "Pattern Classification", John Wiley & Sons, Second edition 2001.
4. NPTEL online course by Prof. Balaraman Ravindran on Introduction to Machine Learning.

Syllabus (Practical)

Four 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 12-15

experiments. Students of different batches should implement different programs based on the following guidelines:

1. Implementation of feature selection and extraction algorithm.
2. Implementation of linear regression
3. Implementation of logistic regression.
4. Implementation of KNN algorithm
5. Implementation of decision tree.
6. Implementation of Naïve Bayesian classifier.
7. Implementation of Bayesian network.
8. Clustering Based on EM algorithm.
9. Clustering Based on K-Means algorithm.
10. Implementation of evaluation techniques.
11. Implementation of back propagation for ANN.

Textbooks:

1. T. Hastie, R. Tibshirani, J. Friedman,"The Elements of Statistical Learning", 2e, 2008
2. Christopher Bishop, Pattern Recognition and Machine Learning, Springer 2006

References:

1. Christopher Bishop, "Pattern Recognition and Machine Learning", 2016
2. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani , "Introduction to Statistical Learning", Springer, 2013
3. Richard Duda, Peter Hart, David Stork, "Pattern Classification", John Wiley & Sons, 2e,2001

***

UCS659:Mobile Computing								
Program Elective, School of Technology								
Lect.	**Tut.**	**Pract.**	**Credits**	**Evaluation Scheme**				
				Component	**Exam**	**Weightage**	**Mini Pass %**	
3	-	2	4	Theory & Practical	FA	50	40	40%
					SA	50	40	

Course Description:

The objective of this course is to provide third-year Computer Science and Engineering students with a comprehensive understanding of mobile computing principles, technologies, and applications. Mobile computing has become increasingly pervasive in modern society, with smartphones, tablets, and other mobile devices playing a central role in communication, commerce, and entertainment. Through theoretical discussions, practical exercises, and hands-on projects, students will explore the key concepts, challenges, and opportunities in mobile computing, enabling them to design, develop, and deploy mobile applications effectively.

Prerequisite: Network Technologies

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

CLO1 **Illustrate[2]** the basics concept of mobile telecommunication systems

CLO2 **Identify[3]** the generations of telecommunication systems in wireless networks

CLO3 **Categorize[4]** the functionality of MAC, network layer, Transport and Application layers

CLO4 **Construct[3]** a mobile application using Android/blackberry/ios/Windows SDK

Syllabus (Theory)

Units	Description	Hrs.
I	**Introduction and Overview of Wireless Sensor Networks** Introduction, Basic Overview of the Technology, Applications of Wireless Sensor Networks, Another Taxonomy of WSN Technology	07
II	**Basic Wireless Sensor Technology:** Sensor Node Technology, Hardware and Software, Sensor Taxonomy, WN Operating environment, , Wireless Transmission Technology and Systems, Radio Technology Primer, Available Wireless Technologies, Medium Access Control Protocols for Wireless Sensor Networks	07

III	**Routing Protocols for Wireless Sensor Networks:** Data Dissemination and Gathering, Routing Challenges and Design Issues in Wireless Sensor Networks, Routing Strategies, Transport Control Protocols for Wireless Sensor Networks, Traditional Transport Control Protocols, Transport Protocol Design Issues, Examples of Existing Transport Control Protocols.	**07**
IV	**WSN Middleware Principles:** Middleware Architecture, Network Management for Wireless Sensor Networks, Network Management Requirements, Traditional Network Management Models, Network Management Design Issues, Example of Management Architecture: MANNA	**07**
V	**Operating Systems for Wireless Sensor Networks:** Operating System Design Issues, Examples of Operating Systems, TinyOS, Mate, MagnetOS, MANTIS, OSPM, EYES OS, SenOS, EMERALDS,PicOS	**07**
VI	**Performance and Traffic Management:** Introduction, Background, WSN Design Issues, MAC Protocols, Routing Protocols, Transport Protocols, Performance Modeling of WSNs, Performance Metrics, Basic Models, Network Models, Case Study: Simple Computation of the System Life Span	**07**

Textbooks:

1. Waltenegus Dargie, Christian Poellabauer, "Fundamentals of Wireless Sensor Networks, Theory and Practice", 1st Edition, John Wiley & Sons, 2007.

2. Kazem Sohraby, Daniel Manoli "Wireless Sensor Networks- Technology, Protocols and Applications", 1st Edition ,Wiley Inter Science Publications, 2010.

References :

1. Bhaskar Krishnamachari, "Networking Wireless Sensors", 1st Edition, Cambridge University Press, 2005

2. C. S Raghavendra, Krishna M. Sivalingam, Taieb znati," Wireless Sensor Networks", 1st Edition, Springer Science, 2004

3. Edgar H. Callaway, Jr. and Edgar H. Callaway," Wireless Sensor Networks: Architectures and Protocols", 1st Edition, CRC Press, 2003.

Syllabus (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. To check orthogonality of two codes. Generation of Walsh codes.
2. To implement Code Division Multiple Access (CDMA).
3. To study frequency reuse.
4. To create a MIDletsuite with two MIDlets.
5. To study ChoiceGroup class and its implementation in J2ME.
6. To study Canvas class and its implementation in J2ME.
7. Write WML page using various tags such as select and option tags.
8. Write a WML page to display an image and to accept input from the user.
9. Study Assignment 1: Detailed study of Bluetooth
10. Study Assignment 2 : Detailed study of Wireless Application Protocol

Textbooks:

1. Jochen Schiller, "Mobile Communications", PHI, Second Edition, 2003.
2. Prasant Kumar Pattnaik, Rajib Mall, "Fundamentals of Mobile Computing", PHILearning Pvt.Ltd, New Delhi – 2012

References:

1. Dharma PrakashAgarval, Qing and An Zeng, "Introduction to Wireless and Mobile systems",Thomson Asia Pvt Ltd, 2005.
2. William.C.Y.Lee, "Mobile Cellular Telecommunications-Analog and Digital Systems", Second Edition,TataMcGraw Hill Edition ,2006.
3. C.K.Toh, "AdHoc Mobile Wireless Networks", First Edition, Pearson Education, 2002.
4. Android Developers : http://developer.android.com/index.html
5. Apple Developer : https://developer.apple.com/
6. Windows Phone DevCenter : http://developer.windowsphone.com
7. BlackBerry Developer : http://developer.blackberry.com

UCS660:Intelligent Systems								
Program Elective, School of Technology								
Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	Weightage	Mini Pass %	
3	-	2	4	Theory & Practical	FA	50	40	40%
					SA	50	40	

Course Description:

The objective of this course is to provide third-year Computer Science and Engineering students with a comprehensive understanding of intelligent systems, including the theories, algorithms, and applications of artificial intelligence (AI) and machine learning (ML). Intelligent systems play a crucial role in various domains, including robotics, natural language processing, computer vision, and expert systems. Through theoretical discussions, practical exercises, and hands-on projects, students will develop the skills necessary to design, implement, and evaluate intelligent systems for solving complex computational problems

Prerequisite: Programming & Data Structures.

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

CLO1 **Make use of[3]** different types of Agents in Artificial Intelligence to solve problems

CLO2 **Implement [4]** different Search Strategies for appropriate real-world problem-Solving

CLO3 **Analyze[4]** different Knowledge, Reasoning and Learning and planning scheme for appropriate Problem-Solving

CLO4 **Differentiate[4]** between different Expert systems and there Applications.

Syllabus (Theory)

Units	Description	Hrs.
I	**Introduction** Definition, Introduction to AI,- Future of Artificial Intelligence Characteristics of Intelligent Agents, Structure of Intelligent agents, Types of Agents, Agent Environments PEAS representation for an Agent	07
II	**Problem Solving:** Solving problems by searching, Problem formulation, Search Strategies, Uninformed Search Techniques-DFS, BFS, Uniform cost search, Informed search methods-Best First Search, heuristic Functions, Hill Climbing, A*,CSP	07

III	**Knowledge and Reasoning:** A knowledge Based Agent, WUMPUS WORLD Environment, Propositional Logic, First Order Predicate Logic Syntax and Semantics, Unification, Forward and backward chaining.	**07**
IV	**Uncertain Knowledge and Reasoning:** Uncertainty, Representing Knowledge in an Uncertain Domain, Probability, Bays Theorem, Belief Networks, Simple Inference in Belief Networks	**07**
V	**Learning and Planning:** General Model of Learning Agents, Types of Learning-Supervised, Unsupervised, Reinforcement Learning, Planning: A Simple Planning Agent, Planning in Situation calculus, Basic representation for planning, A Partial Order Planning	**07**
VI	**Expert systems:** Expert systems – Architecture of expert systems, Roles of expert systems, Knowledge Acquisition, Meta knowledge, Heuristics. Typical expert systems– MYCIN, DART.	**07**

Text Book

1. Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach", 2nd Edition, Pearson Publication, 2009.
2. Robin R Murphy, "Introduction to AI Robotics", ISBN-81-203-2458-7, 2nd Edition, PHI Publication, 2000

References

1. Patrick H. Winston, "Artificial Intelligence", 3 rd Edition, Pearson Publication, 1992.
2. George Lugar, "AI-Structures and Strategies for Complex Problem Solving", 6th Edition, Pearson Educations, 2002.
3. Nils J. Nilsson, "Principles of Artificial Intelligence", 1st Edition, Elsevier Publication, 1982.

Syllabus (Practical)

Two hours per week per batch practical is to be utilized for writing programs to ensure thatstudents 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-

Suggested List of practical

1. Write a program to make use of pre-processing libraries-standardization & normalization and predefined libraries e.g., PANDAS, NumPI
2. Write a program to implement Single Player Game (Using Heuristic Function)
3. Write a program for diagnosis the diseases
4. Implementation of A* Algorithm
5. Implementation of Tic-Tac-Toe game problem
6. Implementing WUMPUS world problem.
7. Implementation of Water jug Problem
8. Implementation of 8 puzzle problem
9. Implementation of Traveling salesman problem
10. Solve any problem using depth first search
11. Write a Program to find factorial of given number
12. Write a Program to detect species of Animal

Textbooks:

1. Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach", 2nd Edition, Pearson Publication, 2009.
2. Robin R Murphy, "Introduction to AI Robotics", ISBN-81-203-2458-7, 2nd Edition, PHI Publication, 2000

.

References:

1. Patrick H. Winston, "Artificial Intelligence", 3 rd Edition, Pearson Publication, 1992.
2. George Lugar, "Al-Structures and Strategies for Complex Problem Solving", 6th Edition, Pearson Educations, 2002.
3. Nils J. Nilsson, "Principles of Artificial Intelligence", 1st Edition, Elsevier Publication, 1982.

***

UCS661:High Performance Computing								
University Open Elective, School of Technology								
Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	Weightage	Mini Pass %	
3	-	2	4	Theory & Practical	FA	50	40	40%
					SA	50	40	

Course Objective:
The objective of this course is to provide third-year Computer Science and Engineering students with a comprehensive understanding of high-performance computing (HPC) principles, techniques, and methodologies. High-performance computing plays a crucial role in solving computationally intensive problems efficiently, spanning various domains such as scientific simulations, numerical analysis, data analytics, and machine learning.

Prerequisite: - Computer Algorithm

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

CLO1 **Distinguish[4]** algorithms in the computational area for efficient programming in modern computer architectures

CLO2 **Make use of[3]** suitable algorithms for scientific computations

CLO3 **Make use of[3]** tools for performance optimization and debugging

CLO4 **Compare[4]** parallel processing architectures based on their performance

Syllabus (Theory)

Units	Description	Hrs.
I	**Introduction:** Introduction to Parallel Computing , Scope of Parallel Computing, Organization and Contents of the Text, Parallel Programming Platforms, Implicit Parallelism, Trends in Microprocessor & Architectures, and Limitations of Memory System Performance.	07
II	**Parallel Processing:** Dichotomy of Parallel Computing Platforms, Physical Organization of Parallel Platforms, Communication Costs in Parallel Machines Levels of parallelism , Parallelism models SIMD, MIMD, SIMT, SPMD, Demand-driven Computation, Architectures: N-wide superscalar Architectures, multi-core and multi-threaded.	07

III	**Parallel Programming Techniques:** Principles of Parallel Algorithm Design: Preliminaries, Decomposition Techniques, Characteristics of Tasks and Interactions, Mapping Techniques for Load Balancing, Methods for Containing Interaction Overheads, Parallel Algorithm Models, Processor Architecture, Interconnect, Communication, Memory Organization, and Programming Models in high performance computing architecture ,micro-architecture Memory hierarchy and transaction specific memory design, Thread Organization.	07
IV	**Parallel Programming Paradigm:** Programming Using the Message-Passing Paradigm: Principles of Message-Passing Programming, Send and Receive Operations, the Message Passing Interface, Topology and Embedding, Overlapping Communication with Computation, Collective Communication and Computation Operations, One- Dimensional Matrix-Vector Multiplication, Two-Dimensional Matrix-Vector Multiplication.	
V	**Scheduling:** Scheduling, Job Allocation, Job Partitioning, Dependency Analysis Mapping Parallel Algorithms onto Parallel Architectures, Performance Analysis of Parallel Algorithms.	
VI	**Synchronization:** Programming Shared Address Space Platforms: Thread Basics, need, The POSIX Thread API, Thread Basics: Creation and Termination, Synchronization Primitives in Pthreads, Controlling Thread and Synchronization Attributes, Thread Cancellation, Composite Synchronization Constructs, Tips for Designing Asynchronous Programs, OpenMP: a Standard for Directive Based Parallel Programming.	

Textbooks:

1. Kai Hwang, "Advanced Computer Architecture: Parallelism, Scalability, programmability", McGraw Hill, 1993.
2. David Culler, Jaswinder, Pal Singh, "Parallel Computer Architecture: A hardware/Software Approach", Morgan Kaufmann, 1999.
3. Rezaur Rahman, "Intel Xeon Phi Coprocessor Architecture and Tools" Apress Open, 2013

References:

1. Kai Hwang, "Scalable Parallel Computing", 2004.

Syllabus (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 a program for vector addition
2. Program for matrix multiplication and tiled matrix multiplication
3. Demonstrate Picture scaling, image great scaling, image blur
4. Program using 1D, 2D, and 3D stencil operations.
5. Demonstrate convolution, scan and reduction
6. OpenMP thread-based programming
7. Calculation of pi using work sharing and reduction
8. Demonstrate producer consumer problem
9. Demonstrate Molecular dynamics simulation problem.
10. Calculate π - MPI Bcast and MPI Reduce
11. Solve ocean kernel, reduction problem
12. Perform matrix multiplication on a Cartesian grid using Cannon's algorithm

Textbooks:

1. Wen-Mei W Hwu, David B Kirk, Programming Massively Parallel Processors a Hands- on Approach (3ed.).Morgann Kaufmann, 2016
2. Rezaur Rahman, Intel Xeon Phi Coprocessor Architecture and Tools. Apress Open, 2013

***

UCS662:Ethical Hacking							
University Core, School of Technology							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage	Pass %
3	-	2	4	Theory & Practical	FA	50	40
					SA	50	40

Course Description:

The objective of this course is to provide third-year Computer Science and Engineering students with a comprehensive understanding of ethical hacking principles, techniques, and methodologies. Ethical hacking, also known as penetration testing or white-hat hacking, involves identifying and exploiting vulnerabilities in computer systems, networks, and applications with the permission of the system owner, to assess and improve their security posture

Prerequisite: - Computer Network

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

CLO1 **Identify[2]** how intruders escalate privileges
CLO2 **Implement[3]** Intrusion Detection, Policy Creation, Social Engineering
CLO3 **Evaluate[4]** different types of Attacks and their protection mechanisms
CLO4 **Demonstrate[3]** hacking of web server.

Syllabus (Theory)

Units	Description	Hrs.
I	**Introduction** Types of Data Stolen From the Organizations, Elements of Information Security, Authenticity and Non-Repudiation, Security Challenges, Effects of Hacking, Hacker – Types of Hacker, Ethical Hacker	**07**

II	**Penetration Testing** Hacktivism - Role of Security and Penetration Tester, Penetration Testing Methodology, Networking & Computer Attacks – Malicious Software (Malware), Protection Against Malware, Intruder Attacks on Networks and Computers	07
III	**Foot Printing and Social Engineering** Web Tools for Foot Printing, Conducting Competitive Intelligence, Google Hacking, Scanning, Enumeration, Trojans & Backdoors, Virus & Worms, Proxy & Packet Filtering, Denial of Service, Sniffer, Social Engineering – shoulder surfing, Dumpster Diving, Piggybacking	07
IV	**Data Security** Physical Security – Attacks and Protection, Steganography – Methods, Attacks and Measures	
V	**Network Protection System** Routers, Firewall & Honeypots, IDS & IPS, Web Filtering, Vulnerability, Penetration testing, Session Hijacking	
VI	**Hacking Web Servers** Web Server, Email Hacking, Incident Handling & Response, Bluetooth Hacking, Mobiles Phone Hacking	

Textbooks:

1. Michael T. Simpson, Kent Backman, James E, "Corley, Hands-On Ethical Hackingand Network Defense", Second Edition, CENGAGE Learning, 2010

References:

1. Steven DeFino, Barry Kaufman, Nick Valenteen, "Official Certified Ethical Hacker Review Guide", CENGAGE Learning, 2009-11-01.
2. Whitaker & Newman, "Penetration Testing and Network Defense", Cisco Press, Indianapolis, IN, 2006.
3. Patrick Engebretson, "The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy", Syngress Basics Series – Elsevier, August 4, 2011.

Syllabus (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. To experiment Email Servers-mail Forgery and Spamming.
2. To provide security to Anonymous Mailing, Attacks on E-mail Password.
3. To implement securing the E-Mail Passwords, Email Forensics.
4. To experiment Windows login, password and other security measures.
5. To experiment Linux login, password and other security measures.
6. To use steganography and backdoor types and tools.
7. To apply detection of Trojans, viruses and apply security.
8. To experiment registry tweaks and Tricks.
9. To apply Back-Track Penetration Tool.
10. To implement Secure Network Configuration

Textbooks:

1. Michael T. Simpson, Kent Backman, James E. "Corley, Hands-On Ethical Hacking and Network Defense", Second Edition, CENGAGE Learning, 2010.

References:

1. Steven DeFino, Barry Kaufman, Nick Valenteen, "Official Certified Ethical Hacker Review Guide", CENGAGE Learning, 2009-11-01.
2. Whitaker & Newman, "Penetration Testing and Network Defense", Cisco Press, Indianapolis, IN, 2006.
3. Patrick Engebretson, "The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy", Syngress Basics Series – Elsevier, August 4, 2011.

***

UCS605 (UOE601): Advance Cloud Computing							
University Core, School of Technology							
Lect.	**Tut.**	**Pract.**	**Credits**	**Evaluation Scheme**			
				Component	**Exam**	**Weightage**	**Pass %**
3	-	2	4	Theory & Practical	FA	50	40
					SA	50	40

Prerequisite: - Basic knowledge of Operating System

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

CLO1 **Analyze[4]** the basic concepts and services of cloud computing.
CLO2 **Demonstrate[2]** large scale distributed systems and cloud applications
CLO3 **Apply[3]** security in cloud applications
CLO4 **Design[5]** Cloud services for IOT Application

Syllabus (Theory)

Units	Description	Hrs.
I	**Introduction to Cloud Computing** Importance of Cloud Computing, Characteristics, Pros and Cons of Cloud Computing, Migrating into the Cloud, Seven-step model of migration into a Cloud, Trends in Computing. Cloud Service Models: SaaS, PaaS, IaaS, Storage. Cloud Architecture: Cloud Computing Logical Architecture, Developing Holistic Cloud Computing Reference Model, Cloud System Architecture, and Cloud Deployment Models.	07
II	**Data Storage and Cloud Computing** Data Storage: Introduction to Enterprise Data Storage, Direct Attached Storage, Storage Area Network, Network Attached Storage, Data Storage Management, File System, Cloud Data Stores, Using Grids for Data Storage. Cloud Storage: Data Management, Provisioning Cloud storage, Data Intensive Technologies for Cloud Computing. Cloud Storage from LANs to WANs: Cloud Characteristics, Distributed Data Storage.	07
III	**Virtualization in Cloud Computing** Introduction: Definition of Virtualization, Adopting Virtualization, Types of Virtualization,	07

Virtualization Architecture and Software, Virtual Clustering, Virtualization Application, Pitfalls of Virtualization. Grid, Cloud and Virtualization: Virtualization in Grid, Virtualization in Cloud, Virtualization and Cloud Security. Virtualization and Cloud Computing: Anatomy of Cloud Infrastructure, Virtual infrastructures, CPU Virtualization, Network and Storage Virtualization.

IV	**Cloud Platforms and Cloud Applications** Amazon Web Services (AWS): Amazon Web Services and Components, Amazon Simple DB, Elastic Cloud Computing (EC2), Amazon Storage System, Amazon Database services (Dynamo DB). Microsoft Cloud Services: Azure core concepts, SQL Azure, Windows Azure Platform Appliance. Cloud Computing Applications: Healthcare: ECG Analysis in the Cloud, Biology: Protein Structure Prediction, Geosciences: Satellite Image Processing, Business and Consumer Applications: CRM and ERP, Social Networking, Google Cloud Application: Google App Engine. Overview of OpenStack architecture	**07**
V	**Security in Cloud Computing** Risks in Cloud Computing: Risk Management, Enterprise-Wide Risk Management, Types of Risks in Cloud Computing. Data Security in Cloud: Security Issues, Challenges, advantages, Disadvantages, Cloud Digital persona and Data security, Content Level Security. Cloud Security Services: Confidentiality, Integrity and Availability, Security Authorization Challenges in the Cloud, Secure Cloud Software Requirements, Secure Cloud Software Testing.	**07**
VI	**Advanced Techniques in Cloud Computing** Future Tends in cloud Computing, Mobile Cloud, and Automatic Cloud Computing: Comet Cloud. Multimedia Cloud: IPTV, Energy Aware Cloud Computing, Jungle Computing, Distributed Cloud Computing Vs Edge Computing, Containers, Docker, and Kubernetes, Introduction to DevOps. IOT and Cloud Convergence: The Cloud and IoT in your Home, The IOT and cloud in your Automobile, PERSONAL: IoT in Healthcare.	**07**

Syllabus(Practical)

Two Hrs. 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 experiments. Students of different batches should implement different programs based on the following guidelines-

1. Introduction to Amazon Web services AWS.
2. To create a free tire AWS account.
3. To launch an EC2 to instance.
4. To configure private and public key using Putty or Git Bash.
5. To connect EC2 instance using Putty / GitBash
6. To install Linux on EC2 instance and demonstrate Linux basic commands
7. To launch S3 bucket from AWS management console.
8. To configure S3 bucket public policy
9. To deploy a HTML sample page using S3 bucket
10. To install GitHub on EC2 instance and demonstrate GitHub basic operations.

Textbooks:

1. A. Srinivasan, J. Suresh, “Cloud Computing: A Practical Approach for Learning and Implementation”, Pearson, ISBN: 978-81-317-7651-3
2. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, “Mastering Cloud Computing”, McGraw Hill Education, ISBN-13:978-1-25-902995-0.

References:

1. James Bond ,“The Enterprise Cloud”, O'Reilly Media, Inc. ISBN: 9781491907627
2. Dr. Kris Jamsa, “Cloud Computing: SaaS, PaaS, IaaS, Virtualization and more”, Wiley Publications, ISBN: 978-0-470-97389-9
3. Anthony T. Velte Toby J. Velte, Robert Elsenpeter, “Cloud Computing: A Practical Approach”, 2010, The McGraw-Hill.
4. Gautam Shrof, “ENTERPRISE CLOUD COMPUTING Technology Architecture, Applications”, Cambridge University Press, ISBN: 9780511778476
5. Tim Mather, Subra K, Shahid L.,”Cloud Security and Privacy”, Oreilly, ISBN-13 978-81-8404-815-5.

UNCMC Creativity and Innovations							
University Core, School of Technology							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage	Pass %
-	-	-	-	Theory	FA	50	40
					SA	50	40

Course Description:

This course is intended for students from any discipline who require an understanding of design

Thinking for brand, product, and service development. Students will learn a series of design thinking concepts, methods and techniques that are used to bring about innovation in business and in the social sector.

Prerequisite:

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

CLO1 **Illustrate[2]** the art perspectives, ideas, concepts, and solutions related to the design and execution of projects using design thinking principles and IPR.

CLO2 **Prepare[6]** the mindset and discipline of systemic inspiration driven by a desire to identify new sources of ideas, and new models especially outside their regular working.

CLO3 **Categorize[4]** different Indian Designs Law in creativity and Innovations practices. Transport and Application layers

CLO4 **Propose[6]** a concrete, feasible, viable and relevant innovation project/challenge

Syllabus (Theory)

Units	Description	Hrs.
I	**Introduction** What is Different About Design thinking? Design Thinking Skills Principles of Design Thinking, The Basis for Design Thinking, The Design Thinking Team, Design Thinking Workshops and Meetings – Exercises and case based discussions	07
II	**Listening and Empathizing Techniques:** Listening and Empathizing Techniques – observation – structured open ended approach - , Design Thinking Frameworks, Ideation tools – brainstorming, innovation heuristics, behaviour models, overcoming cognitive fixedness – Exercises and case based discussions	07

III	**Design Thinking – Empathy map and Story telling:** Use of Diagrams and Maps in Design Thinking – Empathy map. Affinity diagram, mind map, journey map, combining ideas into complex innovation concepts. Story telling – improvisation, scenario planning, development of scenarios, evaluation tools, frog design and prototyping - – Exercises and case-based discussions Assess developer and user perspectives for bias – apply frameworks to strengthen communication – sustain a culture of innovation..	**07**
IV	**Intellectual Property Management:** Intellectual Property Management. Market Capitalization, Intellectual Capital (IC), Components of Intellectual Capital, Tangible and Intangible Assets of Firms, Goodwill, Linkage between IC, Corporate Strategy, and Profits, Relationship between Intellectual Capital and Intellectual Property, Knowledge Economy and the need for Intellectual Property Management, Enforcement of IPRs – IP and constitution of India, World intellectual Property Organization (WIPO) –WTO/TRIPS Agreement – India and the TRIPS Agreement – Patent law in India- IP & Start up policies in India.	**07**
V	**Types of Intellectual Property trademarks:** Various Types of Intellectual Property trademarks, Copyrights, Patents, Trade Secrets, and Industrial Design, International IP Treaties/Agreements on IP Rights, Types of Patents, Patenting Advantage, Offensive and Defensive IP Strategies, Case studies – Procedure for obtaining patent – Rights of a patentee – Limitations on Particular's Rights – Revocation of patent for Non – working Transfer of patent – Infringement of patent – Global Innovation Indexs and IP Management, Intellectual Property Strategies in Indian Context Universities, CSIR and Commercial Firms	**07**
VI	**Indian Designs Law:** Indian Designs Law – Meaning of Design Registration and Prohibitions – Copyright in Designs –Design and Penalties – Steps for filing an Application – Copyright law in India –Owner of the copyright – Rights of Broad Casters and Performers – Registration of Copyright – Assignment, Licensing and Transmission – Infringement – International Copyright and Copyright Societies -Trade Mark Law in India – Functions of a Trade Mark – Registration of Trade Mark Exploiting Trade Mark – Infringement – Offenses and Penalties	**07**

Textbooks:

1. Roger Martin, "The Design of Business: Why Design Thinking is the Next Competitive Advantage",Harvard Business Press , 2009.
2. Burgelman, Christensen, and Wheelwright, "Strategic Management of Technology and Innovation"5th Edition, McGraw Hill Publications, 2017
3. Prabuddha Ganguli (2017). Intellectual Property Rights: Unleashing the knowledge economy, McGraw Hill Education India (Pvt) Ltd

References:

1. Hasso Plattner, Christoph Meinel and Larry Leifer (eds), "Design Thinking: Understand – Improve– Apply", Springer, 2011
2. Idris Mootee, "Design Thinking for Strategic Innovation: What They Can't Teach You at Business or Design School", John Wiley & Sons 2013
3. Jeanne Liedtka , Andrew King, Kevin Bennett , "Book - Solving Problems with Design Thinking
- Ten Stories of What Works" (Columbia Business School Publishing), 2013
4. Maurício Vianna, Ysmar Vianna, Isabel K. Adler, Brenda Lucena, Beatriz Russo, "Design thinking: Business Innovation" MJV Press, 2011

Online courses:

1. https://www.hss.iitb.ac.in/en/hs-438-intellectual-property-rights-technology-development-and-management/
2. https://cde.nus.edu.sg/isem/graduate/coursework/masters-of-science-intellectual-property-management-programme/

***