Semester 6 (3rd year 2nd Semester)

Course Code	Course Title	Credit	Theory	Lab
CSE 3201	Distributed Systems	1	1	0

Course Outline: Foundations **-** Characterization of DS, System Models, Networking and Internetworking, Interprocess Communication, Remote Invocation, Indirect Communication and Operating System Support **Middleware** - Dist. Objects and Components, Web Services and Peer-to-Peer Systems System services – Security, Distributed File Systems and Name Services Distributed algorithms - Time and Global States, Coordination and Agreement Shared data, Transactions and Concurrency Control, Distributed Transactions, and Replication, New challenges -Mobile and Ubiquitous Computing

References:

1. Distributed Systems: Concepts and Design (5th Edition). George Coulouris (Author), Jean Dollimore (Author), Tim Kindberg (Author), Gordon Blair (Author)

Course Code	Course Title	Credit	Theory	Lab
CSE 3202	Distributed Systems Lab	2	0	2

Lab: Introduction to Message passing technology and its applications, Sockets Programming, Remote Procedure Calls code implementation, Synchronization assignments, Group Communication code implementation, Distributed mutual exclusion assignment, Implementation of Election Algorithms, Implementation of Distributed File system: MapReduce, Spanner, Distributed Systems Design assignments: Cloud Services and Content Delivery Networks configuration.

Course Code	Course Title	Credit	Theory	Lab
SE 3203	Software Metrics	2	2	0

Course Outline: Overview of Software Metrics, The basics of Measurement, Goal based framework for software measurement, Empirical Investigation, Measuring Internal Attributes : Size, Measuring Internal Attributes : Structure, Measuring Cost and Effort, Measuring External product attributes : Quality, Measuring Software Reliability, Object Oriented Metrics, For hands-on experiences: Students will implement different software metrics calculation related algorithms, utilize existing industry related tools for measuring software metrics and compare it with their implementations to gain concrete idea.

References:

1. Software metrics- A Rigorous and Practical Approach, (3rd Edition) Norman Fenton, and Jones

Bieman.

2. Software Measurement and Estimation: A practical Approach (1st Edition) Linda M. Laird, and M. Carol Brennan

Course Code	Course Title	Credit	Theory	Lab
SE 3204	Software Metrics Lab	1	0	1

Course Outline: Overview of Software Metrics, The basics of Measurement, Goal based framework for software measurement, Empirical Investigation, Measuring Internal Attributes : Size, Measuring Internal Attributes : Structure, Measuring Cost and Effort, Measuring External product attributes : Quality, Measuring Software Reliability, Object Oriented Metrics, For hands-on experiences: Students will implement different software metrics calculation related algorithms, utilize existing industry related tools for measuring software metrics and compare it with their implementations to gain concrete idea.

References:

1. Software metrics- A Rigorous and Practical Approach, (3rd Edition) Norman Fenton, and Jones Bieman.
2. Software Measurement and Estimation: A practical Approach (1st Edition) Linda M. Laird, and M. Carol Brennan

Course Code	Course Title	Credit	Theory	Lab
SE 3205	Software Security	2	2	0

Course Outline: Introduction: Security principles, concept of computer security, security services and policies Security risks: Database security, operating systems security, secure coding Countermeasures: methodologies and tools for identifying and eliminating security vulnerabilities, techniques to prove the absence of vulnerabilities, and ways to avoid security holes in new software. Secure software design: essential guidelines for building secure software, information security standards

Suggested Readings:

1. Security in Computing, 4th Edition, by Charles P. Pfleeger , Publisher: Prentice Hall; 4th edition
2. Computer security: principles and practices, by William Stallings and Lawrie Brown, 2nd Edition,
3. Brian Chess and Jacob West, Secure Programming with Static Analysis (required)
4. David A. Wheeler, Secure Programming for Linux and Unix HOWTO Version 3.5, Aug 2004 (required)
5. Goertzel et al, Software Security Assurance State of the Art Report, May 2007.
6. Aleph One, Smashing the Stack for Fun and Profit. Phrack Vol 7, Nr. 49

7. Tim Newsham, Format String Attacks, Guardent tech report, Sept 2000

Course Code	Course Title	Credit	Theory	Lab
SE 3206	Software Security Lab	1	0	1

Course Outline: Introduction: Security principles, concept of computer security, security services and policies Security risks: Database security, operating systems security, secure coding Countermeasures: methodologies and tools for identifying and eliminating security vulnerabilities, techniques to prove the absence of vulnerabilities, and ways to avoid security holes in new software. Secure software design: essential guidelines for building secure software, information security standards

Suggested Readings:

1. Security in Computing, 4th Edition, by Charles P. Pfleeger , Publisher: Prentice Hall; 4th edition
2. Computer security: principles and practices, by William Stallings and Lawrie Brown, 2nd Edition,
3. Brian Chess and Jacob West, Secure Programming with Static Analysis (required)
4. David A. Wheeler, Secure Programming for Linux and Unix HOWTO Version 3.5, Aug 2004 (required)
5. Goertzel et al, Software Security Assurance State of the Art Report, May 2007.
6. Aleph One, Smashing the Stack for Fun and Profit. Phrack Vol 7, Nr. 49
7. Tim Newsham, Format String Attacks, Guardent tech report, Sept 2000

Course Code	Course Title	Credit	Theory	Lab
CSE 3207	Artificial Intelligence	2	2	0

Course Outline: Intelligent Agents and their Environments - The concept of a Rational Agent, Specifying the Task environment (PEAS description), Different characteristics of environments (Fully vs Partially observable, Static vs Dynamic, Episodic vs Sequential etc.) and Different types of agents (Reflex, Goal-based, Utility-based etc.),Search - Formulating a search problem , Uninformed Search strategies: BFS, DFS, DLS, ID-DFS, their working principles, complexities, relative advantages and disadvantages, Informed (heuristic) Search strategies: Greedy Best-first search, A* search: Working principle, Characteristics of heuristics (admissibility and consistency), Proof of A*'s optimality, Local search: Hill Climbing, Searching with non-deterministic actions: AND-OR search trees and Searching with partial observability: Belief state-space search, Adversarial Search - Formulation of a Game tree, The minimax algorithm, Alpha-Beta pruning: Its rationale, working principle and Additional techniques such as Move ordering and Search cut-off, Probabilistic Reasoning - Bayes' rule and its uses, Bayesian Network: Building a Bayes-net and making inference from it, Markov Chains and Hidden Markov Models: Transition and Sensor models, Building and HMM, applications of HMM, Inference in temporal models: Filtering, Prediction, Most Likely explanations (Viterbi algorithm) etc. and Particle Filters: basic working principle, Making Decisions - Decision theory and Utility theory: Lottery, Utility functions, Maximum Expected Utility principle, Constraints of Utility (Orderability, Transitivity etc) and Markov Decision Processes: Policies, Rewards, Optimal policies and the Utility of States, Value Iteration, Supervised Learning - Basic concepts of classification and supervised learning: Training set, Test set, Overfitting, Underfitting etc., Decision trees: Basic understanding, Learning a Decision tree through entropy calculation, Nearest Neighbor classifier: Basic working principle, Relative advantages and disadvantages, Naive Bayes classifier: Basic working principle, Calculating classification procedures, Relative advantages and disadvantages, Artificial Neural Network: Basic working principle, Basic structure and calculation of a perceptron, Basics of backpropagation algorithm and Support Vector Machines: Basic working principle, Unsupervised Learning (Clustering) - Basic concepts and applications of Clustering, Different types of Clustering: Partitional vs. Hierarchical, Exclusive vs Overlapping vs Fuzzy, Complete vs Partial, K-means Clustering: Basic working principle, characteristics, advantages, disadvantages, Agglomerative Hierarchical Clustering: Basic concepts, Representations (Dendrograms and Nested cluster diagrams), Different techniques to define cluster proximity: Single link, Complete link, Group average, Centroid method, their relative advantages and disadvantages and DBSCAN: Basic principle and applications, Classification of points (Core, Border and Noise), Reinforcement Learning - Understanding basics of Reinforcement Learning: MDPs, Policies, Rewards, Utilities etc., Passive and Active Reinforcement Learning, Exploration and Exploitation, Adaptive Dynamic Programming, Temporal Difference Learning and Q-Learning.

References:

1. Russell, Stuart, and Peter Norvig. "Artificial intelligence: a modern approach." (1995).

Course Code	**Course Title**	**Credit**	**Theory**	**Lab**

CSE 3208	Artificial Intelligence Lab	1	0	1

Course Outline: Intelligent Agents and their Environments - The concept of a Rational Agent, Specifying the Task environment (PEAS description), Different characteristics of environments (Fully vs Partially observable, Static vs Dynamic, Episodic vs Sequential etc.) and Different types of agents (Reflex, Goal-based, Utility-based etc.),Search - Formulating a search problem , Uninformed Search strategies: BFS, DFS, DLS, ID-DFS, their working principles, complexities, relative advantages and disadvantages, Informed (heuristic) Search strategies: Greedy Best-first search, A* search: Working principle, Characteristics of heuristics (admissibility and consistency), Proof of A*'s optimality, Local search: Hill Climbing, Searching with non-deterministic actions: AND-OR search trees and Searching with partial observability: Belief state-space search, Adversarial Search - Formulation of a Game tree, The minimax algorithm, Alpha-Beta pruning: Its rationale, working principle and Additional techniques such as Move ordering and Search cut-off, Probabilistic Reasoning - Bayes' rule and its uses, Bayesian Network: Building a Bayes-net and making inference from it, Markov Chains and Hidden Markov Models: Transition and Sensor models, Building and HMM, applications of HMM, Inference in temporal models: Filtering, Prediction, Most Likely explanations (Viterbi algorithm) etc. and Particle Filters: basic working principle, Making Decisions - Decision theory and Utility theory: Lottery, Utility functions, Maximum Expected Utility principle, Constraints of Utility (Orderability, Transitivity etc) and Markov Decision Processes: Policies, Rewards, Optimal policies and the Utility of States, Value Iteration, Supervised Learning - Basic concepts of classification and supervised learning: Training set, Test set, Overfitting, Underfitting etc., Decision trees: Basic understanding, Learning a Decision tree through entropy calculation, Nearest Neighbor classifier: Basic working principle, Relative advantages and disadvantages, Naive Bayes classifier: Basic working principle, Calculating classification procedures, Relative advantages and disadvantages, Artificial Neural Network: Basic working principle, Basic structure and calculation of a perceptron, Basics of backpropagation algorithm and Support Vector Machines: Basic working principle, Unsupervised Learning (Clustering) - Basic concepts and applications of Clustering, Different types of Clustering: Partitional vs. Hierarchical, Exclusive vs Overlapping vs Fuzzy, Complete vs Partial, K-means Clustering: Basic working principle, characteristics, advantages, disadvantages, Agglomerative Hierarchical Clustering: Basic concepts, Representations (Dendrograms and Nested cluster diagrams), Different techniques to define cluster proximity: Single link, Complete link, Group average, Centroid method, their relative advantages and disadvantages and DBSCAN: Basic principle and applications, Classification of points (Core, Border and Noise), Reinforcement Learning - Understanding basics of Reinforcement Learning: MDPs, Policies, Rewards, Utilities etc., Passive and Active Reinforcement Learning, Exploration and Exploitation, Adaptive Dynamic Programming, Temporal Difference Learning and Q-Learning.

References:

1. Russell, Stuart, and Peter Norvig. "Artificial intelligence: a modern approach." (1995).

Course Code	Course Title	Credit	Theory	Lab
SE 3209	Software Testing and Quality Assurance	2	2	0

Course Outline: The Psychology and Economics of Software Testing, Software Testing Life Cycle (STLC), Software Testing Terminology and Methodology, V&V Model, Dynamic Black Box Testing – Boundary Value Analysis, Equivalence Partitioning, State Transition based Testing, Decision Table based Testing, Cause-Effect Graphing based Testing and Error Guessing, Dynamic White Box Testing

– Basis Path Testing, Data Flow Testing and Mutation Testing, Inspections, Walkthroughs, Technical Reviews, Unit Testing, Integration Testing, Function Testing, System Testing, Acceptance Testing, Regression Testing, Test Management – Test Organization, Test Plan, Test Design and Specifications, Software Metrics, Software Quality, Quality Control and Quality Assurance, Quality Management and Project Management, Software Quality Metrics, Testing Internet Applications - Security and Performance Testing, Debugging, Test Driven Development (TDD), Behavior Driven Development (BDD). **Tools and Project** - The students will be divided into small groups having at most 3 members and a class project will be given to them for preparing a system test case. They must validate the requirements and create Mock UIs during the preparation of test cases. Besides, each of the students will relate their learnings on unit, regression, performance and security testing, debugging, behavior driven development via different tools like JUnit, Selenium, Apache JMeter, Sprajax, Sqlninja, Bugzilla, Cucumber

References:

1. Naresh Chauhan, Software Testing: Principles and Practices, 1st or higher Edition, Oxford University Press.
2. Glenford J. Myers, Corey Sandler, and Tom Badgett. The Art of Software Testing, 3rd or higher Edition, John Wiley & Sons.
2. Lisa Crispin and Janet Gregory. Agile Testing: A Practical Guide for Testers and Agile Teams, 1st or higher Edition, Pearson Education.

Course Code	**Course Title**	**Credit**	**Theory**	**Lab**
SE 3210	Software Testing and Quality Assurance Lab	1	0	1

Course Outline: The Psychology and Economics of Software Testing, Software Testing Life Cycle (STLC), Software Testing Terminology and Methodology, V&V Model, Dynamic Black Box Testing – Boundary Value Analysis, Equivalence Partitioning, State Transition based Testing, Decision Table based Testing, Cause-Effect Graphing based Testing and Error Guessing, Dynamic White Box Testing – Basis Path Testing, Data Flow Testing and Mutation Testing, Inspections, Walkthroughs, Technical Reviews, Unit Testing, Integration Testing, Function Testing, System Testing, Acceptance Testing, Regression Testing, Test Management – Test Organization, Test Plan, Test Design and Specifications, Software Metrics, Software Quality, Quality Control and Quality Assurance, Quality Management and Project Management, Software Quality Metrics, Testing Internet Applications - Security and Performance Testing, Debugging, Test Driven Development (TDD), Behavior Driven Development (BDD). **Tools and Project** - The students will be divided into small groups having at most 3 members and a class project will be given to them for preparing a system test case. They must validate the requirements and create Mock UIs during the preparation of test cases. Besides, each of the students will relate their learnings on unit, regression, performance and security testing, debugging, behavior driven development via different tools like JUnit, Selenium, Apache JMeter, Sprajax, Sqlninja, Bugzilla, Cucumber

References:

1. Naresh Chauhan, Software Testing: Principles and Practices, 1st or higher Edition, Oxford University Press.
2. Glenford J. Myers, Corey Sandler, and Tom Badgett. The Art of Software Testing, 3rd or higher Edition, John Wiley & Sons.
3. Lisa Crispin and Janet Gregory. Agile Testing: A Practical Guide for Testers and Agile Teams, 1st or higher Edition, Pearson Education.

Course Code	Course Title	Credit	Theory	Lab
SE 3211	Software Design and Architecture	2	2	0

Course Outline: Design Concept - The Design Process, Design Concepts, The Design Model; Architectural Design: Software Architecture, Architectural Genres, Architectural Styles, Architectural Design, Assessing, Alternative Architectural Designs, Architectural Mapping Using Data Flow; Component-Level Design: What Is a Component, Designing Class-Based Components, Conducting Component-Level Design, Component-Level Design for WebApps, Designing Traditional Components, Component-Based Development; User Interface Design: The Golden Rules, User Interface Analysis and Design, Interface Analysis, Interface Design Steps, Web App Interface Design, Design Evaluation.

References:

1. Software Engineering – A Practitioner's Approach. 7th Edition, Roger S. Pressman
2. Software Engineering. 9th Edition, Ian Sommerville

Course Code	Course Title	Credit	Theory	Lab
SE 3212	Software Design and Architecture Lab	1	0	1

Course Outline: Design Concept - The Design Process, Design Concepts, The Design Model; Architectural Design: Software Architecture, Architectural Genres, Architectural Styles, Architectural Design, Assessing, Alternative Architectural Designs, Architectural Mapping Using Data Flow; Component-Level Design: What Is a Component, Designing Class-Based Components, Conducting Component-Level Design, Component-Level Design for WebApps, Designing Traditional Components, Component-Based Development; User Interface Design: The Golden Rules, User Interface Analysis and Design, Interface Analysis, Interface Design Steps, Web App Interface Design, Design Evaluation.

References:

1. Software Engineering – A Practitioner's Approach. 7th Edition, Roger S. Pressman
2. Software Engineering. 9th Edition, Ian Sommerville