<table>
<thead>
<tr>
<th>Course Outcomes:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.</strong> Understand how to apply the knowledge to gain insight of distributed systems in solving real world problems.</td>
<td></td>
</tr>
<tr>
<td><strong>2.</strong> Identify and formulate the broader domain areas where the concept of distributed systems can be used.</td>
<td></td>
</tr>
<tr>
<td><strong>3.</strong> Develop various web applications and automate the real time problems.</td>
<td></td>
</tr>
<tr>
<td><strong>4.</strong> Enhance the concepts of failure recovery in distributed systems and also develop software to recover from failure.</td>
<td></td>
</tr>
<tr>
<td><strong>5.</strong> Utilize the modern software and technical skills in order to control concurrency in distributed transactions.</td>
<td></td>
</tr>
<tr>
<td><strong>6.</strong> Implement the concepts of distributed deadlock in order to avoid the deadlock situation.</td>
<td></td>
</tr>
</tbody>
</table>

**Syllabus: As per AKTU**

**Unit 1**  

**Unit 2**  
Distributed Mutual Exclusion: Classification of distributed mutual exclusion, requirement of mutual exclusion theorem, Token based and non token based algorithms, performance metric for distributed mutual exclusion algorithms. Distributed Deadlock Detection: system model, resource Vs communication deadlocks, deadlock prevention, avoidance, detection & resolution, centralized dead lock detection, distributed dead lock detection, path pushing algorithms, edge chasing algorithms.

**Unit 3**  
Agreement Protocols: Introduction, System models, classification of Agreement Problem, Byzantine
<p>| <strong>Unit 5</strong> | Transactions and Concurrency Control: Transactions, Nested transactions, Locks, Optimistic Concurrency control, Timestamp ordering, Comparison of methods for concurrency control. Distributed Transactions: Flat and nested distributed transactions, Atomic Commit protocols, Concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery. Replication: System model and group communication, Fault - tolerant services, highly available services, Transactions with replicated data. |</p>
<table>
<thead>
<tr>
<th>Course Outcomes:</th>
<th>Subject Name: Artificial Intelligence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. To understand the difference between normal and Artificial Intelligence</td>
<td>1. To understand the difference between normal and Artificial Intelligence</td>
</tr>
<tr>
<td>2. To understand what are intelligent drivers</td>
<td>2. To understand what are intelligent drivers</td>
</tr>
<tr>
<td>3. To learn what are intelligent agents and their application.</td>
<td>3. To learn what are intelligent agents and their application.</td>
</tr>
<tr>
<td>4. To learn how AI impacts on reasoning and machine learning</td>
<td>4. To learn how AI impacts on reasoning and machine learning</td>
</tr>
<tr>
<td>5. To understand the AI impacts on Pattern recognition</td>
<td>5. To understand the AI impacts on Pattern recognition</td>
</tr>
<tr>
<td>6. To learn about various AI applications and their impact on today’s society</td>
<td>6. To learn about various AI applications and their impact on today’s society</td>
</tr>
</tbody>
</table>

**Syllabus: As per AKTU**

**Unit 1**
Introduction: Introduction to Artificial Intelligence, Foundations and History of Artificial Intelligence, Applications of Artificial Intelligence, Intelligent Agents, Structure of Intelligent Agents, Computer vision, Natural Language Possessing.

**Unit 2**

**Unit 3**
Knowledge Representation & Reasoning: Propositional logic, Theory of first order logic, Inference in First order logic, Forward & Backward chaining, Resolution, Probabilistic reasoning, Utility theory, Hidden Markov Models (HMM), Bayesian Networks.

**Unit 4**

**Unit 5**
# Course Outcomes:


2. To understand different types of software testing (i.e. Functional Testing, Structural Testing, Regression Testing etc.)

3. To apply different types of testing with tools

4. To understand different types of Software Testing Activities (i.e Levels of Testing, Debugging, Testing techniques and their Applicability, Exploratory Testing)

5. To understand Object oriented Testing

6. To understand Testing Web Applications

## Syllabus: As per AKTU

### Unit 1: Review of Software Engineering:

**Verification:**
Verification methods, SRS verification, Source code reviews, User documentation verification, Software project audit, Tailoring Software Quality Assurance Program by Reviews, Walkthrough, Inspection, and Configuration Audits.

### Unit 2: Functional Testing:
Boundary Value Analysis, Equivalence Class Testing, Decision Table Based Testing, Cause Effect Graphing Technique.

**Structural Testing:**
Control flow testing, Path testing, Independent paths, Generation of graph from program, Identification of independent paths, CyclomaticComplexity, Data Flow Testing, Mutation Testing.

### Unit 3: Regression Testing:
What is Regression Testing? Regression Test cases selection, Reducing the number of test cases, Code coverage prioritization technique.

**Reducing the number of test cases:**
Prioritization guidelines, Priority category, Scheme, Risk Analysis.
| Unit 4 | **Software Testing Activities:** Levels of Testing, Debugging, Testing techniques and their Applicability, Exploratory Testing  
**Automated Test Data Generation:**  
| --- | --- |
| Unit 5 | **Object oriented Testing:** Definition, Issues, Class Testing, Object Oriented Integration and System Testing.  
| Course Outcomes: | 1. To understand security issues, services, goals and mechanism along with various types cipher like DES  
 | | 2. To understand mathematical foundation required for various cryptographic Algorithms (RSA, AES)  
 | | 3. To apply message Authentication Codes and digital Signature Techniques  
 | | 4. To apply Key Management and distribution in cryptography  
 | | 5. To understand IP security and system security  
 | | 6. Identify and classify computer and security threats and develop a security model to prevent, detect and recover from attacks  

**Syllabus: As per AKTU**

| Unit 1 | Introduction to security attacks, services and mechanism, Classical encryption techniques-substitution ciphers and transposition ciphers, cryptanalysis, steganography, Stream and block ciphers. Modern Block Ciphers: Block ciphers principles, Shannon’s theory of confusion and diffusion, fiestal structure, Data encryption standard(DES), Strength of DES, Idea of differential cryptanalysis, block cipher modes of operations, Triple DES  
 | Unit 2 | Introduction to group, field, finite field of the form GF(p), modular arithmetic, prime and relative prime numbers, Extended Euclidean Algorithm, Advanced Encryption Standard (AES) encryption and decryptionFermat’s and Euler’s theorem, Primarily testing, Chinese Remainder theorem, Discrete Logarithmic Problem, Principals of public key crypto systems, RSA algorithm, security of RSA  
 | Unit 3 | Message Authentication Codes: Authentication requirements, authentication functions, message authentication code, hash functions, birthday attacks, security of hash functions, Secure hash algorithm (SHA) Digital Signatures: Digital Signatures, Elgamal Digital Signature Techniques, Digital signature standards (DSS), proof of digital signature algorithm,  
 | Unit 4 | Key Management and distribution: Symmetric key distribution, Diffie-Hellman Key Exchange, Public key distribution, X.509 Certificates, Public key Infrastructure. Authentication Applications: Kerberos, Electronic mail security: pretty good privacy (PGP), S/MIME.  

### Branch: Computer Science and Engineering
### Year: IV
### Semester: Odd

#### Subject Code: NOE 077

**Subject Name:** Software Project Management

**Course Outcomes:**

1. To understand the basic concept of software project management, different Models.

2. To understand the concept of Project Organization and Scheduling.

3. To understand the concept of Project Monitoring and Control.

4. To understand the concept of Quality Assurance and Testing.


6. An ability to identify, formulate, and solve software engineering problems.

**Syllabus: As per AKTU**

**Unit 1**

**Introduction and Software Project Planning**

**Unit 2**

**Project Organization and Scheduling**

**Unit 3**

**Project Monitoring and Control**

**Unit 4**

**Software Quality Assurance and Testing**

**Unit 5**

**Project Management and Project Management Tools**
<table>
<thead>
<tr>
<th>Branch: Computer Science and Engineering</th>
<th>Year: IV</th>
<th>Semester: Odd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject Code: NCS 751</td>
<td>Subject Name: Distributed Systems Lab</td>
<td></td>
</tr>
<tr>
<td>Course Outcomes:</td>
<td>1. Understand and remember fundamentals of distributed networking approaches.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Understand and remember the certain algorithms approaches in distributed computing.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Implementation of these advance computing algorithms and execute.</td>
<td></td>
</tr>
</tbody>
</table>

**Syllabus: As per AKTU**

The following programs may be developed preferably on ‘UNIX’ platform:- A part from the above other problems may be given as per Course Instructor.
1. Simulate the functioning of Lamport’s Logical Clock in ‘C’.
2. Simulate the Distributed Mutual Exclusion in ‘C’.
3. Implement a Distributed Chat Server using TCP Sockets in ‘C’.
4. Implement RPC mechanism for a file transfer across a network in ‘C’.
5. Implement ‘Java RMI’ mechanism for accessing methods of remote systems.
7. Implement CORBA mechanism by using ‘C++’ program at one end and ‘Java program on the other.'
<table>
<thead>
<tr>
<th>Branch: Computer Science and Engineering</th>
<th>Year: IV</th>
<th>Semester: Odd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject Code: NCS 752</td>
<td>Subject Name: Project</td>
<td></td>
</tr>
</tbody>
</table>

**Course Outcomes:**

1. Able to develop a design solution for a set of requirements
2. Able to test and validate the conformance of the developed prototype against the original requirements of the problem
3. Work as a responsible member and possibly a leader of a team in developing software solutions

---

**GUIDELINES FOR STUDENTS**

**Introduction:** As per AKTU guidelines, Project has to be initiated in 7th semester beginning and completed by the end of 8th semester with proper report and demonstration.

Following Milestone to be followed in 7th Semester

1. Preliminary SRS, Base Paper
2. SRS, Time Plan, Reference Papers
3. Literature Survey based on SRS
4. Design Documentation: HLD, LLD, Test plan
5. System Setup, Tools, UI Design,
6. Initial Implementation
7. Writing Project Report
Branch: Computer Science and Engineering  
Year: IV  
Semester: Odd

Subject Code: NCS 753  
Subject Name: Industrial Training

Course Outcomes:
1. An ability to work in actual working environment.
2. An ability to utilize technical resources
3. An ability to write technical documents and give oral presentations related to the work completed

GUIDELINES FOR STUDENTS

1. As per AKTU guidelines, Industrial Training done after 6th Semester would be evaluated in 7th semester through Report and Viva-voce; the students undergo the industrial training for 1 month training.

2. Before proceeding on Industrial Training, student must seek instructions from the Class in-charge or the Teacher, who is the in-charge of Industrial Training.

3. The student must go through the syllabus of the subject. Carry a copy of the syllabus with him/her.

4. The purpose of the Industrial Training is not to observe the processes being performed on the industry but to develop the work process being performed and apprise them of the industry problems.

5. Industrial Training must give exposure in solving the open ended problems in real work setting so as to cause college base knowledge and skill into practical problems as envisaged during the training.

6. During the training, students will be given practical problems by the industry in which they are undergoing training. In case the industry do not give them the problems, the students will themselves formulate problems and carry out detailed study on them and recommend the optimum solution based on their theory knowledge.

7. The practical training report must include the following:
   
   (a) The basic history/introduction of the industry.

   (b) The software packages being used for the designing and assembly of the objects.

   (c) The sequence of operations followed/ systems introduced for the production.

   (d) The layout of various workshop/floors or the labs and admin section of the industry.
(e) Computer configuration required for the loading the used software’s.

(f) The formulation of practical problems.

(g) Data required formulating the problems.

(h) Analysis of the data, steps required and commands used in software industry.

(i) Suggestions made based on the analysis of the data.

(j) Certificate from the industry for the period of training undergone.

(k) The final report must be at-least 80 to 100 pages for the student undergoing 30 days training with 1.5 line spacing of font size 10 to 12.

(l) In case no. of students undergoing training in the same industry are more than one, each student will prepare his/her report separately and it should be copy of the same report.

11. The report and the required documents must be submitted to training coordinator on the date and time announced. The above information will be displaced on notice board.