**PANDIT DEENDAYAL PETROLEUM UNIVERSITY, GANDHINAGAR**
**SCHOOL OF TECHNOLOGY**

| COURSE STRUCTURE FOR B TECH IN COMPUTER ENGINEERING | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Semester V** | | | **B Tech in Computer Engineering** | | | | | | | | | | |
| **Sr. No.** | **Course/Lab Code** | **Course/Lab Name** | **Teaching Scheme** | | | | | **Examination Scheme** | | | | | |
| | | | | | | | | **Theory** | | | **Practical** | | **Total** |
| | | | **L** | **T** | **P** | **C** | **Hrs/Wk** | **CE** | **MS** | **ES** | **CE** | **ES** | **Marks** |
| 1 | 18CP301T | Operating Systems | 4 | 0 | 0 | 4 | 4 | 25 | 25 | 50 | -- | -- | 100 |
| 2 | 18CP302 | Theory of Computation | 3 | 1 | 0 | 4 | 4 | 25 | 25 | 50 | -- | -- | 100 |
| 3 | 18CP303T | Web Technology | 3 | 0 | 0 | 3 | 3 | 25 | 25 | 50 | -- | -- | 100 |
| 4 | 18CP304 | Software Engineering | 3 | 0 | 0 | 3 | 3 | 25 | 25 | 50 | -- | -- | 100 |
| 5 | 18CP305T | Information Security | 3 | 0 | 0 | 3 | 3 | 25 | 25 | 50 | -- | -- | 100 |
| 6 | 18CP306 | Principles of Economics | 3 | 0 | 0 | 3 | 3 | 25 | 25 | 50 | -- | -- | 100 |
| 7 | 18CP301P | Operating Systems Lab | 0 | 0 | 2 | 1 | 2 | -- | -- | -- | 25 | 25 | 50 |
| 8 | 18CP303P | Web Technology Lab | 0 | 0 | 2 | 1 | 2 | -- | -- | -- | 25 | 25 | 50 |
| 9 | 18CP305P | Information Security Lab | 0 | 0 | 2 | 1 | 2 | -- | -- | -- | 25 | 25 | 50 |
| | | Total | 19 | 1 | 6 | 23 | 26 | -- | -- | -- | -- | -- | 750 |

CE- Continuous Evaluation, MS-Mid Semester; ES – End Semester Exam

| **Course Code:** 18CP301T | | | | | **Course Name: Operating Systems** | | | |
|---|---|---|---|---|---|---|---|---|
| **Teaching Scheme** | | | | | **Examination Scheme** | | | |
| **L** | **T** | **P** | **C** | **Hrs/ Wk** | **Theory** | | | **Total** |
| | | | | | **Continuous Evaluation** | **Mid Semester** | **End Semester** | **Marks** |
| 4 | 0 | 0 | 4 | 4 | 25 | 25 | 50 | 100 |

**Prerequisites: Computer Programming**

**Learning objectives:**

- To understand the OS role in the overall computer system, To study the operations performed by OS as a resource manager
- To understand the scheduling policies of OS, to understand the different memory management techniques
- To understand process concurrency and synchronization
- To understand the concepts of input/output, storage and file management
- To introduce system call interface for file and process management
- To study different OS and compare their features.

**Unit wise allocation of course content**

**UNIT 1 (14 L)**

Overview-Introduction-Operating system objectives, User view, System view, Operating system definition, Computer System Organization, Computer System Architecture, OS Structure, OS Operations, Process Management, Memory Management, Storage Management, Protection and Security, Computing Environments. Operating System services, User and OS Interface, System Calls, Types of System Calls, System Programs, Operating System Design and Implementation.

Process and CPU Scheduling - Process concepts-The Process, Process State, Process Control Block, Threads, Process Scheduling-Scheduling Queues, Schedulers, Context Switch, Operations on Processes, System calls, Inter-process communication-ordinary pipes and named pipes in Unix.

**UNIT 2 (12 L)**

Process Scheduling-Basic concepts, Scheduling Criteria, Scheduling algorithms, Multiple- Processor Scheduling, Real-Time Scheduling, Thread scheduling, Linux scheduling and Windows scheduling. Process Synchronization, Background, The Critical Section Problem, Peterson's solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization.

Deadlocks - System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock.

**UNIT 3 (14 L)**

Memory Management and Virtual Memory – Memory Management Strategies- Background, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of Page Table, IA-32 Segmentation, IA-32 Paging. Virtual Memory Management - Background, Demand Paging, Copy-on-Write, Page Replacement, Page Replacement Algorithms, Allocation of Frames, Thrashing.

**UNIT 4 (12 L)**

Storage Management-File System- Concept of a File, System calls for file operations - open(), read(),

write(), close(), seek(), unlink(), Access methods, Directory and Disk Structure, File System Mounting, File Sharing, Protection. File System Implementation - File System Structure, File System Implementation, Directory Implementation, Allocation methods, Free-space Management, Efficiency, and Performance. Mass Storage Structure – Overview of Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management, Swap space Management

**Student centering learning: (**The student centering learning contents should be declared at the commencement of semester. It should be maximum 10% ; however exact contents is left to faculty)

**Lecture: 52 Hrs**
**Tutorial: 0 Hrs**
**Approximate Total: 52 Hrs**

**Texts and References:**


1. Operating System Concepts , Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Wiley.
2. Operating Systems – Internals and Design Principles, W. Stallings.
3. Unix Concepts and Applications by Sumitabha Das, TMH.
4. Modern Operating Systems, Andrew S Tanenbaum, PHI
5. Operating Systems: A concept-based Approach, D.M. Dhamdhere, TMH.
6. Principles of Operating Systems, B. L. Stuart, Cengage learning.
7. An Introduction to Operating Systems, P.C.P. Bhatt, PHI.
8. Principles of Operating systems, Naresh Chauhan, Oxford University Press.
9. Unix System Programming Using C++, Terrence Chan, PHI/Pearson.


**Course Outcomes (COs):**

At the end of this course students will be able to

1. Apply concepts of process management for the improvement of system performance.
2. Ability to understand and solve synchronization problems.
3. Assess about minimization of turnaround time, waiting time and response time and also maximization of throughput by keeping CPU as busy as possible.
4. Analyze important parameters for memory management to handle memory optimally.
5. Judge efficacy of Page replacement algorithms
6. Design and create new techniques for process, synchronization, Memory and file management.

| **Lab Code** 18CP301P | | | | | **Lab Name: Operating Systems Lab** | | |
|---|---|---|---|---|---|---|---|
| **Teaching Scheme** | | | | | **Examination Scheme** | | |
| | | | | | **Practical** | | **Total** |
| **L** | **T** | **P** | **C** | **Hrs/Wk** | **Continuous Evaluation** | **End Semester Exam** | **Marks** |
| 0 | 0 | 2 | 1 | 2 | 25 | 25 | 50 |

**Prerequisites: Computer Programming**

**Course objectives:**

- To write programs in Linux environment using system calls.
- To implement the scheduling algorithms.
- To implement page replacement algorithms
- To implement file allocation methods.
- To understand and implement IPC mechanism using named and unnamed pipes.
- To develop solutions for synchronization problems using semaphores.

**List of Experiments:**

1. Basic Shell commands and Scripting.
2. Write C programs to simulate the following CPU scheduling algorithms:
   a) Round Robin b) SJF
3. Write C programs to simulate the following CPU scheduling algorithms:
   a) FCFS b) Priority
4. Write C programs to simulate the following File organization techniques:
   a) Single level directory b) Two level c) Hierarchical
5. Write C programs to simulate the following File allocation methods:
   a) Contiguous b) Linked c)Indexed
6. Write a C program to copy the contents of one file to another using system calls.
7. Write a C program to simulate Bankers Algorithm for Dead Lock Avoidance
8. Write a C program to simulate Bankers Algorithm for Dead Lock Prevention
9. Write C programs to simulate the following page replacement algorithms:
   a) FIFO b) LRU c) LFU
10. Write C programs to simulate the following techniques of memory management:
    a) Paging b) Segmentation
11. Write a C program to implement the ls | sort command. (Use unnamed Pipe)
12. Write a C program to solve the Dining- Philosopher problem using semaphores.
13. Write C programs to implement IPC between two unrelated processes using named pipe.

**Details of Assessment Instruments under LW Practical Component:**

- Experiments during lab sessions and record-keeping of lab work (Term Work)
- Assignments / Mini project / Quiz / Practical Test

**Course Outcomes (COs):**

At the end of this course students will be able to

1. Develop application programs using system calls in Unix.
2. Implement inter-process communication between two processes.
3. Design and solve synchronization problems.
4. Simulate process scheduling and resource scheduling,

5. Implement Deadlock management, file management, and memory management.
6. Apply I/O management and process management.

| **Course Code:** 18CP302 | | | | | **Course Name: Theory of Computation** | | | |
|---|---|---|---|---|---|---|---|---|
| **Teaching Scheme** | | | | | **Examination Scheme** | | | |
| **L** | **T** | **P** | **C** | **Hrs/ Wk** | **Theory** | | | **Total** |
| | | | | | **Continuous Evaluation** | **Mid Semester** | **End Semester** | **Marks** |
| 3 | 1 | 0 | 4 | 4 | 25 | 25 | 50 | 100 |

**Prerequisites: Nil**

**Learning objectives:**

- To provide introduction to some of the central ideas of theoretical computer science from the perspective of formal languages.
- To introduce the fundamental concepts of formal languages, grammars and automata theory.
- Classify machines by their power to recognize languages.
- Employ finite state machines to solve problems in computing.
- To understand deterministic and non-deterministic machines.
- To understand the differences between decidability and undecidability

**Unit wise allocation of course content**

**UNIT 1 (10 L + 4T)**

Introduction to Finite Automata, Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory – Alphabets, Strings, Languages, Problems. Deterministic Finite Automata, Nondeterministic Finite Automata, Finite Automata with Epsilon-Transitions.

Regular Expressions, Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Properties of Regular Languages-Pumping Lemma for Regular Languages, Applications of the Pumping Lemma, Closure Properties of Regular Languages, Decision Properties of Regular Languages, Equivalence and Minimization of Automata.

**UNIT 2 (9 L + 3T)**

Context-Free Grammars**:** Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Sentential Forms, Parse Tress, Applications of Context-Free Grammars, Ambiguity in Grammars and Languages. Push Down Automata,: Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's and CFG's, Deterministic Pushdown Automata.

**UNIT 3 (10 L + 4T)**

Normal Forms for Context- Free Grammars, the Pumping Lemma for Context-Free Languages, Closure Properties of Context-Free Languages. Decision Properties of CFL's - Complexity of Converting among CFG's and PDA's, Chomsky Normal Form. Introduction to Turing Machines-Problems That Computers Cannot Solve, Programming Techniques for Turing Machines, Extensions to the basic Turing machine, Restricted Turing Machines

**UNIT 4 (10 L + 2T)**

Undecidability**:** A Language that is Not Recursively Enumerable, An Undecidable Problem That is RE,

Undecidable Problems about Turing Machines, Post Correspondence Problem, Other Undecidable Problems, Intractable Problems**:** The Classes P and NP, An NP-Complete Problem.

**Student centering learning: (**The student centering learning contents should be declared at the commencement of semester. It should be maximum 10% ; however exact contents is left to faculty)

**Lecture: 39 Hrs**
**Tutorial: 13 Hrs**
**Approximate Total: 52 Hrs**

**Texts and References:**


1. Introduction to Automata Theory, Languages, and Computation, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Theory of Computer Science – Automata languages and computation, Mishra and Chandrashekaran, PHI.
3. Introduction to the Theory of Computation, Michael Sipser, Cengage Learning.
4. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
5. Introduction to Computer Theory, Daniel I.A. Cohen, John Wiley.
6. A Text book on Automata Theory, P. K. Srimani, Nasir S. F. B, Cambridge University Press.


**Course Outcomes (COs):**

At the end of this course students will be able to

1. Understand the basic concepts and application of Theory of Computation
2. Learn the concept of abstract machines and its power to recognize the languages.
3. Employ finite state machines for modeling and solving computing problems.
4. Design context free grammars for formal languages.
5. Implement the concepts of Turing machine to understand decidability and undecidability.
6. Apply the knowledge of Theory of Computation to solve computational problems.

| **Course Code:** 18CP303T | | | | | **Course Name: Web Technology** | | | |
|---|---|---|---|---|---|---|---|---|
| **Teaching Scheme** | | | | | **Examination Scheme** | | | |
| | | | | | **Theory** | | | **Total** |
| **L** | **T** | **P** | **C** | **Hrs/Wk** | **Continuous Evaluation** | **Mid Semester** | **End Semester** | **Marks** |
| 3 | 0 | 0 | 3 | 3 | 25 | 25 | 50 | 100 |

**Prerequisites:** Basic web scripting programming concepts.

**Learning objectives:**

- Learn fundamentals of web development.
- To introduce Client side scripting with Javascript and AJAX.
- To introduce Server side programming with PHP, Java Servlets and JSP.
- Learn Database connectivity with Web pages.
- Use of web design framework and tools.
- Introduction and Implementation of Web services
- To introduce XML and processing of XML Data.

**Unit wise allocation of course content**

**UNIT 1 (09 L)**

**Introduction to PHP**: Declaring variables, data types, arrays, strings, operators, expressions, control structures, functions, Reading data from web form controls like text boxes, radio buttons, lists etc., Handling File Uploads, Connecting to database (MySQL as reference), executing simple queries, handling results, Handling sessions and cookies.

**File Handling in PHP**: File operations like opening, closing, reading, writing, appending, deleting etc. on text and binary files, listing directories

**UNIT 2 (10 L)**

**XML:** Introduction to XML, Defining XML tags, their attributes and values, Document Type Definition, XML Schemas, Document Object Model, XHTML

**Parsing XML Data** - DOM and SAX Parsers in java.

**UNIT 3 (10 L)**

**Introduction to Servlets**: Common Gateway Interface (CGI), Lifecycle of a Servlet, deploying a servlet, The Servlet API, Reading Servlet parameters, Reading Initialization parameters, Handling Http Request & responses, Using Cookies and Sessions, connecting to a database using JDBC.

**UNIT 4 (10 L)**

**Introduction to JSP:** The Anatomy of a JSP Page, JSP Processing, Declarations, Directives, Expressions, Code Snippets, implicit objects, Using Beans in JSP Pages, Using Cookies and session for session tracking, connecting to database in JSP.

**Client side Scripting:** Introduction to Javascript: Javascript language - declaring variables, scope of variables, functions, event handlers (onclick, onsubmit etc.), Document Object Model, Form validation. Simple AJAX application.

**Student centering learning: (**The student centering learning contents should be declared at the commencement of semester. It should be maximum 10% ; however exact contents is left to faculty)

**Lecture: 39 Hrs**
**Tutorial: 0 Hrs**
**Approximate Total: 39 Hrs**

**Texts and References:**


1. Web Technologies, Uttam K Roy, Oxford University Press
2. The Complete Reference PHP – Steven Holzner, Tata McGraw-Hill
3. Web Programming, building internet applications, Chris Bates, Wiley Dreamtech
4. Java Server Pages –Hans Bergsten, SPD O'Reilly
5. Java Script, D. Flanagan, O'Reilly, SPD.
6. Beginning Web Programming - Jon Duckett, WROX.
7. Programming World Wide Web, R. W. Sebesta, Pearson.
8. Internet and World Wide Web – How to program, Dietel and Nieto, Pearson.


**Course Outcomes (COs):**

At the end of this course students will be able to

1. Learn the Web Design Concepts including WWW, HTTP protocol and Browser.
2. Create Interactive web pages using client side scripting, validation of forms and AJAX programming.
3. Implement server side scripting with PHP language, Java Servlets and JSP.
4. Develop the modern Web applications using the MVC framework.
5. Create Web services.
6. Parse and Use XML data in web pages

| Lab Code : 18CP303P | | | | | Lab Name: Web Technology Lab | | |
|---|---|---|---|---|---|---|---|
| Teaching Scheme | | | | | Examination Scheme | | |
| | | | | | Practical | | Total |
| L | T | P | C | Hrs/Wk | Continuous Evaluation | End semester exam | Marks |
| 0 | 0 | 2 | 1 | 2 | 25 | 25 | 50 |

**Prerequisites:** Basic scripting and some computer programming concepts.

**Course objectives:**

- To enable the student to install and work with web server.
- To enable the student to create interactive web pages using Java Script and AJAX technologies
- To enable the student to design the web pages using Server side scripting languages such as PHP, Servlet and JSP
- To enable the student to work with modern web design tools and MVC framework
- To enable the student to process XML data in web pages.
- To enable the student to implement Web Services.

**List of Experiments:**

**Note:**

1. Use LAMP Stack (Linux, Apache, MySQL and PHP) for the Lab Experiments. Though not mandatory, encourage the use of Eclipse platform wherever applicable
2. The list suggests the minimum program set. Hence, the concerned staff is requested to
3. One **Mini Project** related web development considering software engineering practices like SDLC, etc. to be incorporated.

**Add more problems to the list as needed**

1. Install the following on the local machine
   Apache Web Server (if not installed)
   Tomcat Application Server locally
   Install MySQL (if not installed)
   Install PHP and configure it to work with Apache web server and MySQL (if not already configured)
2. Create an XML document that contains 10 users information. Write a Java program, which takes User Id as input and returns the user details by taking the user information from the XML document using (a) DOM Parser and (b) SAX parser
   Implement the following web applications using (a) PHP, (b) Servlets and (c) JSP:
3. A user validation web application, where the user submits the login name and password to the server. The name and password are checked against the data already available in Database and if the data matches, a successful login page is returned. Otherwise a failure message is shown to the user.
4. Modify the above program to use an xml file instead of database.
5. Modify the above program to use AJAX to show the result on the same page below the submit button.
6. Modify the above program such that it stores each query in a database and checks the database first for the result. If the query is already available in the DB, it returns the value that was

previously computed (from DB) or it computes the result and returns it after storing the new query and result in DB.

7. A web application takes a name as input and on submit it shows a hello <name> page where <name> is taken from the request. It shows the start time at the right top corner of the page and provides a logout button. On clicking this button, it should show a logout page with Thank You <name> message with the duration of usage (hint: Use session to store name and time).
8. A web application for implementation:
   The user is first served a login page which takes user's name and password. After submitting the details the server checks these values against the data from a database and takes the following decisions.
   If name and password matches, serves a welcome page with user's full name.
   If name matches and password doesn't match, then serves "password mismatch" page
   If name is not found in the database, serves a registration page, where user's full name is asked and on submitting the full name, it stores, the login name, password and full name in the database (hint: use session for storing the submitted login name and password)
9. A web application that lists all cookies stored in the browser on clicking "List Cookies" button. Add cookies if necessary.

**Course Outcomes (COs):**

At the end of this course students will be able to

1. Install and use web servers and application servers such as Tomcat
2. Write Interactive web applications with Technologies like HTML, Javascript, AJAX, PHP, Servlets and JSPs
3. Connect to Database and get results
4. Working with Web design tools and MVC framework.
5. Create Web services and client to use them.
6. Parse XML files using Java (DOM and SAX parsers)

| Course Code: 18CP304 | | | | | Course Name: Software Engineering | | | |
|---|---|---|---|---|---|---|---|---|
| Teaching Scheme | | | | | Examination Scheme | | | |
| L | T | P | C | Hrs/Wk | Theory | | | Total |
| | | | | | Continuous Evaluation | Mid Semester | End Semester | Marks |
| 3 | 0 | 0 | 3 | 3 | 25 | 25 | 50 | 100 |

**Prerequisites: Nil**

**Learning objectives:**

- To understand the concept of software engineering and software process models.
- To create functional and non-functional requirement and SRS document.
- To understand software design patterns and styles.
- To generate software test cases.
- To ensure software quality and designing risk mitigation strategies..

**Unit wise allocation of course content**

**UNIT 1 (09 L)**

**Introduction to Software Engineering**: The evolving role of software, Changing Nature of Software, legacy software, Software myths. A Generic view of process: Software engineering- A layered technology, a process framework, The Capability Maturity Model Integration (CMMI), Process patterns, process assessment, personal and team process models. Process models: The waterfall model, Incremental process models, Evolutionary process models, Specialized process models, The Unified process.

**UNIT 2 (10 L)**

**Software Requirements:** Functional and non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document. Requirements engineering process: Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management. System models: Context Models, Behavioral models, Data models, Object models, structured methods.

**UNIT 3 (10 L)**

**Design Engineering**: Design process and Design quality, Design concepts, the design model, pattern based software design. Creating an architectural design: software architecture, Data design, Architectural styles and patterns, Architectural Design, assessing alternative architectural designs, mapping data flow
into a software architecture. Modeling component-level design: Designing class-based components, conducting component-level design, object constraint language, designing conventional components. Performing User interface design: Golden rules, User interface analysis, and design, interface analysis, interface design steps, Design evaluation. Introduction to Agile Software Design

**UNIT 4 (10 L)**

**Testing Strategies:** A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, System testing, the art of Debugging. Product metrics: Software Quality, Frame work for Product metrics, Metrics for Analysis Model, Metrics for Design Model, Metrics for source code, Metrics for testing, Metrics for maintenance. Metrics for Process and Products: Software Measurement, Metrics for software quality. **Risk management:** Reactive vs Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM, RMMM Plan.

Quality Management: Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews, Statistical Software quality Assurance, Software reliability, The ISO 9000 quality standards.

**Student centering learning: (**The student centering learning contents should be declared at the commencement of semester. It should be maximum 10% ; however exact contents is left to faculty)

**Lecture: 39 Hrs**
**Tutorial: 0 Hrs**
**Approximate Total: 39 Hrs**

**Texts and References:**

1. Software engineering A practitioner's Approach, Roger S Pressman, McGraw Hill International Edition.
2. Fundamentals of Software Engineering, Rajib Mall, PHI
3. Software Engineering, Ian Sommerville, Pearson education.
4. Software Engineering, A Precise Approach, Pankaj Jalote, Wiley India.

**Course Outcomes (COs):**

At the end of this course students will be able to

1. Understand the concept of software engineering and software process models.
2. Determine the requirements of Software system.
3. Identify the software architectural styles to the suitable applications.
4. Generate the test cases and maintenance metrics.
5. Create Risk Mitigation strategies during software development process.
6. Adapt quality management in Software Development

| Course Code: CP 18CP305T | | | | | Course Name: Information Security | | | |
|---|---|---|---|---|---|---|---|---|
| Teaching Scheme | | | | | Examination Scheme | | | |
| L | T | P | C | Hrs/Week | Theory | | | Total Marks |
| | | | | | Continuous Evaluation | Mid Semester | End Semester | |
| 3 | 0 | 0 | 3 | 3 | 25 | 50 | 25 | 100 |

**Prerequisites: Nil**

**Learning objectives:**

- To understand the concept of security requirements, security attacks, and security policy.
- To understand the mathematical concepts for cryptographic algorithms
- To understand the security mechanisms available to protect the data.
- To understand the security analysis of cryptographic algorithms.

**UNIT 1 (10 L)**

Information Security Requirements, Security Attacks, Security Services, Security Mechanism, Substitution Ciphers, Transposition Ciphers, Cryptanalysis of Ciphers, One Time Pad, Introduction to Steganography. Introduction to Number Theory, Divisibility, Division Algorithm, Binary Operations, Euclidean Algorithm, Extended Euclidean Algorithm, Modular Arithmetic, Matrices, Linear Congruence, Groups, Rings, and Fields, Finite Fields of the form GF(p), Polynomial Arithmetic.

**UNIT 2 (10 L)**

Introduction to Symmetric Key Cryptography, Feistel Structure, Advanced Encryption Standard (AES), Security Analysis of AES, Data Encryption Standard (DES), Multiple DES, Security Analysis of DES, Modern Block Ciphers, Modes of Operation, Synchronous and Asynchronous Stream Ciphers, Use of Modern Block Ciphers and Stream Ciphers. Attacks under Message Indistinguishability: Chosen Plaintext Attack, Chosen Ciphertext Attacks, Attacks under Message Non-malleability.

**UNIT 3 (10 L)**

Introduction to Public Key Cryptography, Diffie-Hellman Key Exchange, One-way and trapdoor one-way functions, RSA Cryptosystem, Attacks on RSA. Introduction of cryptography hash function. Application of Cryptographic Hash Functions.

**UNIT 4 (09 L)**

Properties of Hash Functions, Message Authentication, MD5, SHA, Message Authentication Code (MAC), MAC based on Hash Function (HMAC), Key Management and Distribution. Digital Signature, Authentication Protocols.

**Student centering learning: (**The student centering learning contents should be declared at the commencement of semester. It should be maximum 10% ; however exact contents is left to faculty)

**Lecture: 39 Hrs**
**Tutorial: 0 Hrs**
**Approximate Total: 39 Hrs**

**Texts and References:**

1. William Stallings, "Cryptography and Network Security Principles and Practice", Pearson Education
2. Atul Kahate, “Cryptography and Network Security”, Tata McGraw-Hill Education
3. Charlie Kaufman, Radia Perlman, Mike Speciner, “Network Security: Private Communication in a Public World”, Prentice Hall
4. Behrouz A. Forouzan, “Cryptography and Network Security”, McGraw-Hill Education
5. Menezes, Oorschot, Vanstone : "Handbook of Applied Cryptography", CRC Press
6. Douglas Stinson, "Cryptography: Theory and Practice", Chapman& Hall
7. Menezes Bernard, "Network Security and Cryptography", Cengage Learning India
8. Wenbo Mao, “Modern Cryptography : Theory and Practice”, Prentice Hall

**Course Outcomes (COs):**

At the end of this course students will be able to

1. Define the importance of security policies, basic concepts of cryptography, and cryptanalysis.
2. Demonstrate the mathematical concepts for cryptographic algorithms.
3. Apply appropriate encryption techniques to secure data in transit across data networks.
4. Analyze the different cryptography algorithms with the knowledge of security requirements and security attacks.
5. Evaluate the authentication and hash algorithms as per security requirements.
6. Discuss the security analysis of framework and policies applied in real life applications.

| Lab Code 18CP305P | | | | | Lab Name: Information Security Lab | | |
|---|---|---|---|---|---|---|---|
| Teaching Scheme | | | | | Examination Scheme | | |
| | | | | | Practical | | Total |
| L | T | P | C | Hrs/Wk | Continuous Evaluation | End Semester Exam | Marks |
| 0 | 0 | 2 | 1 | 2 | 25 | 25 | 50 |

**Prerequisites: Computer Programming C or C++**

**Course objectives:**

- To Implement Substitution and Transposition Ciphers.
- To Implement Cryptanalysis on Cipher.
- To implement Public Key Cryptography Algorithm.
- To implement Digital Signature and Hash Function.
- To understand the concept of Steganography
- To understand the concept of various block cipher modes.

**List of Experiments:**

1. Download and Practice Cryptool

**Note: For all cipher implementation, consider file as an input in program. The program should work for large and variable length input text.**

2. Implement a menu driven program for Ceaser Cipher with Encryption, Decryption, Brute Force Attack, and Frequency Analysis functions. Note: For all cipher implementation,
3. Implement a program for Transposition (Columnar) Cipher to encrypt and decrypt the message.
4. Implement a program for Rail Fence Transposition Cipher to encrypt and decrypt the message.
5. Implement a program for Vigenère Cipher to encrypt and decrypt the message.
6. Implement a program for 6x6 Playfair Cipher.
7. Implement a program for n-gram Hill Cipher.
8. Use Crypto++ library to implement encryption and decryption functions for following block ciphers.
   - Electronic Codebook (ECB)
   - Cipher Block Chaining Mode (CBC)
   - Note: Use AES or DES to encrypt each block of plaintext.
9. Implement RSA Encryption and Decryption function.
10. Use RSA for generation and verification of digital signature on file.
11. Study Steganography Tool - Embed secret message in grayscale image.

**Details of Assessment Instruments under LW Practical Component:**

- Experiments during lab sessions and record-keeping of lab work (Term Work)
- Assignments / Quiz / Practical Test

**Course Outcomes (COs):**

At the end of this course students will be able to

1. Define the concept and use of Steganography.
2. Demonstrate the Brute Force and frequency analysis attacks on ciphers.

3. Experiment with cryptographic library and use the different cryptographic algorithms in applications.
4. Analyze encoding algorithm for encryption and decryption based on cryptanalysis for real life application.
5. Compare the different ciphers based on the security analysis.
6. Develop the digital signature and hash function based applications.

| **Course Code:** 18CP306 | | | | | **Course Name: Principles of Economics** | | | |
|---|---|---|---|---|---|---|---|---|
| **Teaching Scheme** | | | | | **Examination Scheme** | | | |
| **L** | **T** | **P** | **C** | **Hrs/ Wk** | **Theory** | | | **Total** |
| | | | | | **Continuous Evaluation** | **Mid Semester** | **End Semester** | **Marks** |
| 3 | 0 | 0 | 3 | 3 | 25 | 25 | 50 | 100 |

**Prerequisites: Nil**

**Learning objectives:**

- Enable them to make day to day decision making.
- Students will learn how markets establish price, production, wage and employment levels, and the likely consequences of government attempts to alter market outcomes.
- Enable students to better understand and evaluate economic issues and events presented by Government and policymakers.
- To build logical abilities and reasoning of the students so as to understand the real-world phenomena and mechanism tactfully.

**Unit wise allocation of course content**

**UNIT 1 (9L)**

**Meaning of Economics**, **The Economic Problem:** Scarcity and Choice; Allocation; Problem of Economics; Role of Assumptions; Meaning and differences of Micro and Macro Economics----

**UNIT 2 (10 L)**

**Derivation of Demand and Supply Curve**: Equilibrium; Externality; Elasticity; Factors affecting Demand and Supply; Market Equilibrium; Government Interventions in Market

**Market Structures**: Perfect and Imperfect competition, Assumptions, Price and output determination in perfect completion, monopoly, monopolistic competition and oligopoly,

**UNIT 3 (10 L)**

**Nature and Scope of Macro Economics:** Circular Flow of Product and Income**;** Four Sector Economy Model; Business Cycle ; National Income : Exchange Rates

**UNIT 4 (10 L)**

**Macroeconomic policies**: Monetary and Fiscal policy- Budget

**International Trade**: Gains from trade, Trade Barriers

**Student centering learning:** (The student centering learning contents should be declared at the commencement of semester. It should be maximum 10% ; however exact contents is left to faculty)

**Lecture: 39 Hrs**
**Tutorial: 0 Hrs**
**Approximate Total: 39 Hrs**

**Texts and References:**

1. Principles of Economics, Gregory Mankiw, Cengage Learning.
2. Principles of Economics, Karl E Case, Ray C Fair and Sharon Oster, Pearson Press.
3. Economics by Paul A Samuelson (Author), William D Nordhaus (Author)

**Course Outcomes (COs):**

At the end of this course students will be able to

1. Argue with reason about the state of economy and factors leading to the same
2. Make rational decisions based on choice
3. Understand the reasons behind changes in National economy and Micro economy
4. Apply economics knowledge to day to day events
5. Understand business cycle.
6. Understand Monetary and Fiscal policy