PANDIT DEENDAYAL PETROLEUM UNIVERSITY, GANDHINAGAR
SCHOOL OF TECHNOLOGY

COURSE STRUCTURE FOR B TECH IN COMPUTER ENGINEERING

Semester VI			B Tech in Computer Engineering										
Sr. No.	Course/Lab Code	Course/Lab Name	Teaching Scheme					Examination Scheme					
			L	T	P	C	Hrs/Wk	Theory			Practical		Total
								CE	MS	ES	CE	ES	Marks
1	18CP311T	Artificial Intelligence	4	0	0	4	4	25	25	50			100
2	18CP312T	System Software & Compiler Design	4	0	0	4	4	25	25	50			100
3	18CP313T	Distributed System	3	0	0	3	3	25	25	50			100
4	18CP314	Wireless Technology & Mobile Computing	3	0	0	3	3	25	25	50			100
5	18CP315X	Elective I	3	0	0	3	3	25	25	50			100
6	18CP315X	Elective II	3	0	0	3	3	25	25	50			100
7	18CP311P	Artificial Intelligence Lab	0	0	2	1	2				25	25	50
8	18CP312P	System Software Lab	0	0	2	1	2				25	25	50
9	18CP313P	Distributed System Lab	0	0	2	1	2				25	25	50
		Total	20	0	6	23	26						750

Subject Code	Elective I & Elective II	Credit	L-T-P	Category
18CP3151	Advanced Computer Architecture	3	3-0-0	System
18CP3152	Embedded System	3	3-0-0	System
18CP3153	Computer Graphics	3	3-0-0	Application
18CP3154	Natural Language Processing	3	3-0-0	Data Science
18CP3155	Data Warehousing and Data Mining	3	3-0-0	Data Science
18CP3156	Principles of Programming language	3	3-0-0	Theory & Algorithm
18CP3157	Data Communication & Coding	3	3-0-0	Communication System
18CP3158	Wireless Sensor Networks	3	3-0-0	Communication System
18CP3159	Cryptography & Network Security	3	3-0-0	Security

Course Code: 18CP311T					**Course Name: Artificial Intelligence**			
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 Fundamentals

Learning objectives:

- Identify the problems where AI is required and the different methods available
- Compare and contrast different AI techniques available.
- Define and explain learning algorithms
- Identify problems in game playing
- Learn Neural Networks
- Learn Expert systems

Unit wise allocation of course content

UNIT 1 (13 L)

Introduction to AI: AI Problems, Intelligent Agents, Problem Formulation, Basic Problem Solving Methods. Searching : Search strategies, Uniformed Search Strategies, State-Space Search, Bi-Directional Search, BFS, DFS, Heuristic Search Strategies, Local Search Algorithms, Hill Climbing, Greedy Best First Search, A* Search, Simulated Annealing, Measure of performance and analysis of search algorithms.

UNIT 2 (14 L)

Knowledge Representation and Inference: Game playing, Knowledge representation using-Predicate logic, Introduction to predicate calculus, Resolution, Use of predicate calculus, Knowledge representation using other logic, Structured representation of knowledge. Production based system, Frame based system. First order logic. Inference in first order logic, propositional Vs. first order inference, unification & lifts forward chaining, Backward chaining, Resolution.

UNIT 3 (13 L)

Neural Networks: Characteristics of Neural Networks, Historical Development of Neural Networks Principles, Artificial Neural Networks: Terminology, Models of Neuron, Topology, Basic Learning Laws, Pattern Recognition Problem, Basic Functional Units, Pattern Recognition Tasks by the Functional Units.

UNIT 4 (12 L)

Expert Systems: Introduction to Expert systems - Architecture of expert systems, Roles of expert systems - Knowledge Acquisition –Meta knowledge, Heuristics. Example of expert systems - MYCIN, DART, XOON, Expert systems shells, Introduction to Planning.

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. Russell, S.J. and Norvig, P., Artificial Intelligence: A Modern Approach, Pearson Education.
2. Kevin Night and Elaine Rich, Nair B., “Artificial Intelligence (SIE)”, McGraw Hill.
3. Dan W. Patterson, “Introduction to AI and ES”, Pearson Education.
4. G.Luger, W.A. Sttubblefield, “Artificial Intelligence”, Addison-Wesley Longman.
5. N.J.Nilson, “Principles of Artificial Intelligence”, Narosa Publishing House.
6. K. Boyer, L. Stark, H. Bunke, "Applications of AI, Machine Vision and Robotics" World Scientific Pub Co.
7. Artificial Intelligence and Expert Systems Development by D W Rolston-Mc Graw hill.
8. N.P. Padhy “Artificial Intelligence and Intelligent Systems”, Oxford University Press.

Course Outcomes (COs):

At the end of this course students will be able to

1. Identify the AI based problems
2. Apply techniques to solve the AI problems
3. Define learning and explain various learning techniques
4. Discuss on Neural Networks
5. Able to understand Game playing
6. Able to learn expert systems

Lab Code 18CP311P					**Lab Name: Artificial Intelligence Lab**		
Teaching Scheme					**Examination Scheme**		
L	**T**	**P**	**C**	**Hrs/Wk**	**Practical**		**Total**
					Continuous evaluation	**End semester exam**	**Marks**
0	0	3	1.5	3	25	25	50

Prerequisites: Computer Programming

Course objectives:

- Understand data structure and its applications like DFS and Best FS
- Apply the concepts to solve game theory problems.
- Understand learning algorithms
- Implement Neural Networks
- Implement and explore tools for Expert systems

List of Experiments:

1. Write a program for Depth First Search
2. Write a program for Best First Search
3. Write a program to generate the output for A* algorithm
4. Write a program to solve water Jug problem using Heuristic functions
5. Write a program to show the Tic Tac Toe game from 0 and X
6. Write a program for expert system using Forward Chaining
7. Hands-on on Matlab/Python for AI related problems like Neural Network, Genetic Algorithm, etc.
8. Project work as decided by Tutor. (all tools related to AI can be explored)

Course Outcomes (COs):

At the end of this course students will be able to

1. Perform collaborative activities in complex and dynamic settings.
2. Exploit and integrate information coming from different sources.
3. Design and implement distributed cognitive systems for information exploitation and collaboration.
4. Understand Neural Network.
5. Learn Genetic Algorithms.
6. Work on Intelligent systems.

Course Code: 18CP312T					**Course Name: System Software & Compiler Design**			
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: Operating System, Theory of Computation

Learning objectives:

- Define System Software such as Assemblers, Loaders, Linkers.
- Learn Macroprocessors.
- Familiarize with source file, object file and executable file structures and libraries.
- Describe the front-end and back-end phases of compiler and their importance to students.
- Learn Lexical Analysis, Syntax Analysis and Semantic Analysis.
- Learn to generate Intermediate Code and code optimization.

Unit wise allocation of course content

UNIT 1 (10 L)

Introduction to System Software, Machine Architecture machine architecture, machine level representation of programs, assembly language programming and optimizing program performance. **Assemblers:** Basic assembler functions, machine dependent assembler features, machine independent assembler features, assembler design options.

UNIT 2 (10 L)

Loaders and Linkers: Basic Loader Functions**,** Machine Dependent Loader Features, Machine Independent Loader Features, Loader Design Options, Implementation Examples.

UNIT 3 (15 L)

Lexical Analysis: Introduction, Alphabets And Tokens In Computer Languages, Representation, Token Recognition And Finite Automata, Implementation, Error Recovery. **Syntax Analysis:** Introduction, Role Of Parsers, Context Free Grammars,

UNIT 4 (17 L)

Top Down Parsers, Bottom-Up Parsers, Operator-Precedence Parsing, Syntax Directed Translation, Intermediate code generation, Code generation

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. System Software by Leland. L. Beck, D Manjula.
2. Compilers-Principles, Techniques and Tools by Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. Pearson.
3. Systems programming – Srimanta Pal , Oxford university press.
4. System programming and Compiler Design, K C Louden, Cengage Learning
5. System software and operating system by D. M. Dhamdhere TMG
6. Compiler Design, K Muneeswaran, Oxford University Press.

Course Outcomes (COs):

At the end of this course students will be able to

1. Understand the data structures and design of system software such as assemblers, loaders,
2. Understand linkers and macro-processors.
3. Apply the system software development tools like Lex, YACC, ANTLR, etc..
4. Analyze different parsing techniques.
5. Develop optimized target code.
6. Design and develop system software.

Lab Code 18CP312P					**Lab Name: System Software Lab**		
Teaching Scheme					**Examination Scheme**		
L	**T**	**P**	**C**	**Hrs/Wk**	**Practical**		**Total**
					Continuous Evaluation	**End Semester Exam**	**Marks**
0	0	2	1	2	25	25	50

Prerequisites: Computer Organization and Programming, C Programming

Course objectives:

- Implement Pass1 and Pass2 assemblers.
- Implement all data structures related to macro processing
- To make students familiar with Lexical Analysis and Syntax Analysis phases of Compiler.
- Design and implement programs on these phases using LEX & YACC tools.
- Implement programs related to NFA, DFA.
- To learn implementation of compiler for new languages.

List of Experiments:

1. Write a LEX program to recognize valid ***arithmetic expression.*** Identifiers in the expression could be only integers and operators could be + and *. Count the identifiers & operators present and print them separately.
2. Write YACC program to evaluate ***arithmetic expression*** involving operators: +, -, *, and /
3. Develop, Implement and Execute a program using YACC tool to recognize all strings ending with ***b*** preceded by ***n a's*** using the grammar ***an b*** (note: input ***n*** value)
4. Design, develop and implement YACC/ C program to construct ***Predictive / LL(1) Parsing Table*** for the grammar rules: ***A □aBa , B □bB | □***. Use this table to parse the sentence: ***abba$***
5. Design, develop and implement YACC/C program to demonstrate ***Shift Reduce Parsing*** technique for the grammar rules: ***E □E+T | T, T □T*F | F, F □(E) | id*** and parse the sentence: ***id + id * id***.
6. Design, develop and implement a C/Java program to generate the machine code using

   Triples for the statement ***A = -B * (C +D)*** whose intermediate code in three-address form:
   T1 = -B
   T2 = C + D
   T3 = T1 + T2
   A = T3
7. Write a LEX program to eliminate ***comment lines*** in a ***C*** program and copy the resulting program into a separate file.
8. Write YACC program to recognize valid ***identifier, operators and keywords*** in the given text (*C program*) file.

One **Mini Project** related to lexer, parser, device driver, etc. using compiler design tools such as LEX, YACC, ANTLR, etc..

Course Outcomes (COs):

At the end of this course students will be able to

1. Implement Assembler
2. Implement macro processing
3. Implement and demonstrate Lexer’s and Parser’s
4. Implement basic compiler
5. Implement different address codes
6. Implement optimization techniques related to target code generation.

Course Code: 18CP313T					**Course Name: Distributed System**			
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: Operating System

Learning objectives:

- Understand foundations of Distributed Systems.
- Introduce the idea of peer to peer services and file system.
- Illustrate the system level and support required for distributed system.
- Identify issues involved in distributed process and resource management.
- Understand distributed algorithms for clock synchronization and logical clocks and leader election.
- Understand current research issues in distributed system.

Unit wise allocation of course content

UNIT 1 (09 L)

Characterization of Distributed Systems: Introduction, Examples of Distributed Systems, Resource Sharing and the Web, Challenges. Time and Global States: Introduction, Clocks Events and Process States, Synchronizing Physical Clocks, Logical Time and Logical Clocks, Global States, Distributed Debugging.

UNIT 2 (10 L)

Coordination and Agreement: Introduction, Distributed Mutual Exclusion, Elections, Multicast Communication, Consensus and Related Problems. Inter Process Communication: Introduction, The API for the Internet Protocols, External Data Representation and Marshalling, Client-Server Communication, Group Communication, Case Study: IPC in UNIX.

UNIT 3 (10 L)

Distributed Objects and Remote Invocation: Introduction, Communication between Distributed Objects, Remote Procedure Call, Events and Notifications, Case Study: JAVA RMI.
Distributed File Systems: Introduction, File Service Architecture,
Case Study 1: Sun Network File System, Case Study 2: The Andrew File System.
Name Services: Introduction, Name Services and the Domain Name System, Directory Services,
Case Study of the Global Name Services.

UNIT 4 (10 L)

Distributed Shared Memory: Introduction, Design and Implementation Issues, Sequential Consistency and IVY Case study, Release Consistency, Munin Case Study, Other Consistency Models. Transactions and Concurrency Control: Introduction, Transactions, Nested Transactions, Locks, Optimistic Concurrency Control, Timestamp Ordering, Comparison of Methods for Concurrency Control.
Distributed Transactions: Introduction, Flat and Nested Distributed Transactions, Atomic Commit Protocols, Concurrency Control in Distributed Transactions, Distributed Deadlocks, Transaction Recovery.

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. Distributed Systems, Concepts and Design, George Coulouris, J Dollimore and Tim Kindberg, Pearson Education.
2. Distributed Systems, Principles and Paradigms, Andrew S. Tanenbaum, Maarten Van Steen, PHI.
3. Distributed Systems, An Algorithm Approach, Sukumar Ghosh, Chapman&Hall/CRC, Taylor & Fransis Group.

Course Outcomes (COs):

At the end of this course students will be able to

1. Understand design of distributed systems.
2. Comprehend knowledge of the core architectural aspects of distributed systems and communication
3. Understand distributed algorithms on synchronization, coordination and agreement.
4. Analyze Distributed objects and remote invocation.
5. Identify techniques to support scalability and fault tolerance in distributed systems.
6. Analyze Concurrency protocols for Distributed transactions and recovery

Lab Code 18CP313P					Lab Name: Distributed System Lab		
Teaching Scheme					Examination Scheme		
L	T	P	C	Hrs/Wk	Practical		Total
					Continuous Evaluation	End Semester Exam	Marks
0	0	2	1	2	25	25	50

Prerequisites: - Operating System

Course objectives:

- To examine the fundamental principles of distributed systems, and provide students hands-on experience in developing distributed protocols.
- To emphasize on distributed communication, process, naming, and synchronization.
- To address consistency and replication, and fault tolerance in distributed systems.

List of Experiments:

1. Write a Program to implement Concurrent Echo Client Server Application.
2. Write the Programs for Remote Procedure call.
3. Write the Programs for Remote Method Invocation.
4. Write the Programs for Thread Programming in JAVA.
5. Implementation of Clock Synchronization (logical/physical)
6. Implementation of Mutual Exclusion algorithms
7. Implementation of Election algorithm.
8. Program to demonstrate process/code migration.
9. Write a distributed application using EJB
10. Write a program using CORBA to demonstrate object brokering.
11. Mini Project : e.g. using SOA

    Study practical:
12. Study of Web service programming.
13. Study of Grid Services using various Tools.

Course Outcomes (COs):

At the end of this course students will be able to

1. Implement the distributed Communication between Client and Server.
2. Apply the concept of Remote Procedure call and Remote Method Invocation.
3. Apply Shared Data access and Files concepts.
4. Design a distributed system that fulfills requirements with regards to key distributed systems properties.
5. Analyze Distributed File Systems and Distributed Shared Memory.
6. Apply Distributed web-based system.

Course Code: 18CP314					**Course Name: Wireless Technology & Mobile Computing**			
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

Learning Objective

- Understand fundamentals of wireless technology
- Learn the design of the IEEE standards for wireless technology
- Understand the concepts of mobile computing and cellular technolgies
- Understand the effect of mobility and other constraints
- Understand various communication protocol design
- Explore tools and technologies for adhoc networks

UNIT I (09 L)
Introduction
Fundamentals of Wireless Communication : Mobile Radio Propagation: Large-Scale Path Loss, Small-Scale Fading and Multipath ; Protocol Stack for Wireless Networking

UNIT II (11 L)
Wireless LAN: Fundamentals and Architecture of WiFi 802.11, Bluetooth Standard, Zigbee 802.15.4; Multiple Access Techniques for Wireless Communications; Wireless Communication in High Mobility Scenario: 802.11p, WAVE

Unit III (15 L)
Cellular Communication: Cellular Networks Design; GSM Architecture; Introduction to 3G,4G, 5G technology;

Unit IV (17L)
Mobile IP and variants of TCP for wireless communication; Wireless communication without Infrastructure support and in constrained environment- Routing and other issues

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. T. Rappaport, Wireless Communications – Principles and Practice, Prentice Hall
2. Jochen Schiller, Mobile Communications, Pearson Education
3. C. Siva Ram Murthy and B.S. Manoj, Ad Hoc Wireless Networks: Architectures and Protocols, Prentice Hall.

Course Outcomes (COs):
At the end of this course students will be able to

1. Remember concepts of wireless communication, technologies, networking and mobile computing
2. Understand about different wireless technologies, standards, networking and computing
3. Apply concepts of wireless communication technologies, networking and computing to solve problems
4. Analyze different wireless communication standards, Adhoc networking protocols and mobile communication and computing
5. Evaluate core components of wireless communication technologies, networking and computing
6. Create different scenarios of wireless communication technologies, networking and routing for resource constrained computing

Course Code: 18CP3151					Course Name: Advanced Computer Architecture			
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

Learning Objective

- To study contemporary computer architecture issues and techniques.
- To provide knowledge of advanced hardware-based techniques for exploiting instruction level parallelism.
- To learn architectures and techniques used for building high performance scalable multithreaded and multiprocessor systems.
- Apply the learned knowledge to conduct computer architecture research using performance simulators.

UNIT I (9 L)

Architecture and Machines: Some definition and terms, interpretation and microprogramming. The instruction set, Basic data types, Instructions, Addressing and Memory. Virtual to real mapping. Basic Instruction Timing. Time, Area And Instruction Sets: Time, cost-area, technology state of the Art, The Economics of a processor project: A study, Instruction sets, Professor Evaluation Matrix

UNIT II (10 L)

Cache Memory Notion: Basic Notion, Cache Organization, Cache Data, adjusting the data for cache organization, write policies, strategies for line replacement at miss time, Cache Environment, other types of Cache. Split I and D-Caches, on chip caches, Two level Caches, write assembly Cache, Cache references per instruction, technology dependent Cache considerations, virtual to real translation, overlapping the Tcycle in V-R Translation, studies. Design summary.

UNIT III (10 L)

Memory System Design: The physical memory, models of simple processor memory interaction, processor memory modeling using queuing theory, open, closed and mixed-queue models, waiting time, performance, and buffer size, review and selection of queuing models, processors with cache.

UNIT IV(10 L)

Concurrent Processors: Vector Processors, Vector Memory, Multiple Issue Machines, Comparing vector and Multiple Issue processors. Shared Memory Multiprocessors: Basic issues, partitioning, synchronization and coherency, Type of shared Memory multiprocessors, Memory Coherence in shared Memory Multiprocessors.

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. Advance computer architecture by Hwang & Briggs, TMH.
2. Pipelined and Parallel processor design by Michael J. Fiynn, Narosa

Course Outcomes (COs):

At the end of this course students will be able to

1. Understand and apply concept and principle of cache memory and virtual memory to high-performance computer architecture.
2. Understand pipelining and its speed advantage and design pipelined logic.
3. Analyze the design of multiprocessing systems.
4. Apply the concept of memory management in multiprocessing systems.
5. Design the overall organization of cache and virtual memories, and pipelined processors.
6. Compare the features of advanced processors.

Course Code: 18CP3152					**Course Name: Embedded Systems**			
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: Digital Electronics, Computer Organization and Programming

Learning objectives:

- To get familiarize with different embedded system design techniques.
- To facilitate the understanding of the microcontrollers used in embedded systems.
- To facilitate the understanding of interfacing of different devices with microcontroller to have interactive embedded system.

Unit wise allocation of course content

UNIT 1 (03 L)

Introduction to Embedded Systems:

Modern definition of embedded systems, Embedded system and general purpose computers, Characteristics, Classifications, Applications, Purposes and Examples of embedded systems, Embedded system components: Core of the Embedded System, Memories, Embedded Firmware, PCB and Passive Components, Embedded System Design Process.

UNIT 2 (15 L)

Designing Embedded Systems with Microcontrollers:

Factors to be considered in selecting microcontroller, An exemplary Microcontroller (MCS-51/AVR/PIC/ARM etc.): CPU Architecture and Organization, Pin diagram, Packages, Oscillator and Clock circuit, Reset Circuits: Power on reset and Brown out protection, Power Supply, Register set, On chip memory organization, Program and data memory buses, Instruction Set and Assembly language programming, Typical Data transfer, Arithmetic, Logic and Branch instructions, Addressing Modes, T-state, Machine cycle and Instruction Cycle, Looping, Branching, Indexing, Counting and Time Delays using instructions, Stack and Stack operations, Subroutines and Procedures, Timers and Counters Circuits and Programming, Power down modes.

UNIT 3 (15 L)

Interfacing and Communicating with Microcontrollers:

I/O Ports, Interfacing LEDs, 7 segment LED display, Opto coupler, Drivers and Buffers, Stepper Motor, DC Motors, PWM ports and Speed Control, Relays, Buzzers, Push button, Toggle and Proximity switches, Matrix keypad, Text and Graphics LCD, Touch Screen Unit, Interfacing using ADC, Interfacing of Sensors and Actuators, Obtaining DAC and waveform generation, Parallel and Serial Communication, Serial Communication Interfaces: UART, SPI and I2C, RS-232, RS-422 and RS-485 interfaces, RTC interfacing, USB and Firewire, Interfacing Wireless Communication Devices: IrDA, Bluetooth, WiFi, Zig bee, GPRS, Emulation of Various Interfaces on USB, Use of Interrupts in real time interfacing of serial and parallel I/O devices, Concept of device drivers, Watchdog Timer and its Applications.

UNIT 4 (06 L)

Embedded System Design Tools

Hardware development, Electronic Design Automation (EDA) Tools, Schematic Design using EDA Tools, PCB layout design and fabrication, Embedded Firmware Design Approaches, Development Languages, Programming in Embedded C, Integration of Hardware and Firmware, Integrated Development Environment (IDE), Types of File Generation and Cross compilation, Simulators, Emulators, Target Hardware Debugging, Programmers: Out of System, In System and In Application Programming, Boot loaders, Boundary Scan, JTAG, Recent Trends in Embedded System Design..

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. Shibu K. V, Introduction to Embedded Systems, TMH.
2. Frank Vahid, Tony Givargis, Embedded system design: A unified Hardware/Software introduction, Wiley.
3. Rajkamal, Embedded System: Architecture, Programming and Design, TMH.
4. Wayne Wolf, Morgan, Computer as Components: Principles of Embedded Computing System Design, Kaufmann Publication.
5. Muhammad Ali Mazidi, 8051 Microcontroller And Embedded Systems: Using Assembly And C, Pearson education India.
6. Muhammad Ali Mazidi, AVR Microcontroller And Embedded Systems: Using Assembly And C, Pearson education India.
7. Muhammad Ali Mazidi, PIC Microcontroller And Embedded Systems: Using Assembly And C, Pearson education India.
8. Steve Furber, ARM System on Chip Architecture, Pearson Education.
9. Data Sheet of the Devices.

Course Outcomes (COs):

At the end of this course students will be able to

1. Understand the concepts related to embedded system design.
2. Implement microcontroller programming in assembly and C.
3. Understand the interfacing of input-output device and controlling them using microcontroller.
4. Understand the interfacing and programming of sensors and actuators.
5.Analyze the complexity of embedded systems.
6. Design a small scale embedded systems.

Course Code: 18CP3153					**Course Name: Computer Graphics**			
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: Computer Programming

Learning objectives:

- To illustrate hardware, software and OpenGL Graphics Primitives.
- Understand interactive computer graphic using the OpenGL.
- Design and implementation of algorithms for 2D graphics Primitives and attributes.
- Demonstrate geometric transformations, viewing on both 2D and 3D objects.
- Infer the representation of curves, surfaces, color.
- Able to understand Illumination models.

Unit wise allocation of course content

UNIT 1 (10 L)

Computer Graphics: Basics of computer graphics, Application of Computer Graphics, Video Display Devices: Random Scan and Raster Scan displays, color CRT monitors, Flat panel displays. Raster-scan systems: video controller, raster scan Display processor, graphics workstations and viewing systems, Input devices, graphics networks, graphics on the internet, graphics software.

OpenGL: Introduction to OpenGL, coordinate reference frames, specifying two-dimensional world coordinate reference frames in OpenGL, OpenGL point functions, OpenGL line functions, point attributes, line attributes, curve attributes, OpenGL point attribute functions, OpenGL line attribute functions, Line drawing algorithms(DDA, Bresenham’s), circle generation algorithms (Bresenham’s).

UNIT 2 (9 L)

Fill area Primitives: Polygon fill-areas, OpenGL polygon fill area functions, fill area attributes, general scan line polygon fill algorithm, OpenGL fill-area attribute functions. 2DGeometric Transformations: Basic 2D Geometric Transformations, matrix representations and homogeneous coordinates. Inverse transformations,

2DComposite transformations, other 2D transformations, raster methods for geometric transformations, OpenGL raster transformations, OpenGL geometric transformations function, 2D viewing: 2D viewing pipeline, OpenGL 2D viewing functions.

UNIT 3 (10 L)

Clipping: clipping window, normalization and viewport transformations, clipping algorithms,2D point clipping, 2D line clipping algorithms: cohen-sutherland line clipping only -polygon fill area clipping: Sutherland-Hodgeman polygon clipping algorithm only.3DGeometric Transformations: 3D translation, rotation, scaling, composite 3D transformations, other 3D transformations, affine transformations, OpenGL geometric transformations functions. Color Models: Properties of light, color models, RGB and CMY color models. Illumination Models: Light sources, basic illumination models-Ambient light, diffuse reflection, specular and phong model, Corresponding openGL functions.

UNIT 4 (10 L)

3D Viewing and Visible Surface Detection: 3DViewing:3D viewing concepts, 3D viewing pipeline, 3D viewing coordinate parameters, Transformation from world to viewing coordinates, Projection transformation, orthogonal projections, perspective projections, The viewport transformation and 3D screen coordinates. OpenGL 3D viewing functions. Visible Surface Detection Methods: Classification of visible surface Detection algorithms, back face detection, depth buffer method and OpenGL visibility detection functions. Curved surfaces, quadric surfaces, OpenGL Quadric-Surface and Cubic-Surface Functions, Bezier Spline Curves, Bezier surfaces, OpenGL curve functions. Corresponding openGL functions.

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. Donald Hearn & Pauline Baker: Computer Graphics with OpenGL Version, Pearson Education.
2. Edward Angel: Interactive Computer Graphics- A Top Down approach with OpenGL, Pearson Education.
3. James D Foley, Andries Van Dam, Steven K Feiner, John F Huges Computer graphics with OpenGL: pearson education
4. Xiang, Plastock, Computer Graphics , schaum's outline series, TMH.
5. Kelvin Sung, Peter Shirley, steven Baer : Interactive Computer Graphics, concepts and applications, Cengage Learning
6. M M Raiker, Computer Graphics using OpenGL, Filip learning/Elsevier

Course Outcomes (COs):

At the end of this course students will be able to

1. Understand the design and algorithms for 2D graphics primitives and attributes.
2. Implement Geometric transformations on both 2D and 3D objects.
3. Apply concepts of clipping and visible surface detection in 2D and 3D viewing, and Illumination Models.
4. Analyze the suitable hardware and software for developing graphics packages using OpenGL.
5. Implement Interactive games using multimedia contents.
6. Discuss the application of computer graphics concepts in the development of computer games, information visualization, and business applications.

Course Code: 18CP3154					**Course Name: Natural Language Processing**			
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: Computer Programming

Learning objectives:

- To understand the structure of natural language for processing
- To understand the basic operations of NLP such as Tokenization, Stemming, POS tagging
- To understand the concepts of linguistic rules and machine learning approaches for classification
- To understand the syntax of Natural languages for grouping local words for parsing
- To understand the concept of shallow parsing
- To study the various applications of NLP- machine translation, sentiment analysis, summarization.

Unit wise allocation of course content

UNIT 1 (9 L)

Introduction to NLP. Language Structure and Analyzer - Overview of language, requirement of computational grammar. Words and their Analysis. Tokenization. Stemming. Morphological Analysis. POS tagging.

UNIT 2 (9 L)

Classical approaches to NLP with knowledge bases and linguistic rules; Data Driven and Machine Learning Approaches to NLP; Efficient, Robust and Scalable NLP

UNIT 3 (9 L)

Linguistics Fundamentals: Syntax and Parsing: Meaning: Empirical or Statistical NLP: Probabilistic Methods on Introductory Graphical Models for NLP: Shallow Parsing: Probabilistic Parsing

UNIT 4 (12 L)

Applications: Machine Translation, Information Retrieval, Sentiment Analysis, Summarization, Information Extraction

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. Jurafsky, Daniel, and James H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics, Prentice Hall, 2000.
2. Christopher D. Manning and Hinrich Schütze, Foundations of Statistical Natural Language Processing. Cambridge, MIT Press, 1999.
3. James Allen, Natural Language Understanding, Benjamin/Cummings, 1995.
4. Eugene Charniak, Statistical Language Learning, MIT Press, 1996.
5. Martin Atkinson, David Britain, Harald Clahsen, Andrew Redford, Linguistics, Cambridge University Press, 1999.
6. P. Lieberman, toward an evolutionary biology of language, Harvard University Press, 2006.
7. Natural Language Processing: A Paninian Perspective by Akshar Bharati, Vineet Chaitanya and Rajeev Sangal

Course Outcomes (COs):

At the end of this course students will be able to

1. Analyse the natural language text and speech
2. Generate the natural language
3. Do machine translation, text summarization, sentiment analysis
4. Apply information retrieval techniques to build search engine, question answering system
5. Process the Natural Language based on structure.
6. Develop POS tagger, parsers and shallow parser for different languages.

Course Code: 18CP3155					**Course Name: Data Warehousing & Data Mining**			
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: Database Management System

Learning Objective

- Study mathematical foundations of data ware housing and data mining.
- Understand and implement data warehouse and data mining models.
- Apply data mining techniques for solving problems.

UNIT I (09 L)

Data warehouse: Introduction to Data warehouse, Difference between operational database systems and data warehouses. Data warehouse Characteristics, Data warehouse Architecture and its Components, Extraction – Transformation – Loading, Logical (Multi – Dimensional), Data Modelling, Schema Design, Star and Snow – Flake Schema, Fact Consultation, Fact Table, Fully Addictive, Semi – Addictive, Non Addictive Measures; Fact Consultation, Fact Table, Fully Addictive, Semi – Addictive, Non Addictive Measures; Fact – Less – Facts, Dimension Table Characteristics; OLAP Cube, OLAP Operations, OLAP Server Architecture – ROLAP, MOLAP and HOLAP.

UNIT II (10 L)

Introducing to Data Mining : Introduction, What is Data Mining, Definition, KDD, Challenges, Data Mining Tasks, Data Preprocessing, Data Cleaning, Missing data, Dimensionality Reduction, Feature Subset Selection, Discretization and Binaryzation, Data Transformation; Measures of Similarity and Dissimilarity – Basics.

UNIT III (10 L)

Association Rules : problems Definition, Frequent Item Set Generation, The APRIORI Principle, Support and Confidence Measures, Association Rule Generation; APRIOIRI Algorithm, The Partition Algorithms, FP- Growth Algorithms, Compact Representation of Frequent Item set- Maximal Frequent Item Set, Closed Frequent Item Sets.

Classification : Problem Definition, General Approaches to solving a classification problem, Evaluation of classifiers, Classification Techniques, Decision Tree – Decision tree Construction, Methods for Expressing attribute test conditions, Measures for Selecting the Best Split, Algorithm for Decision tree Induction;

UNIT IV (10 L)

Naive Bayes Classifier, Bayesian Belief Networks; K – N earnest neighbour classification – Algorithm and Characteristics. Clustering: Problem Definition, Clustering Overview, Evaluation of Clustering Algorithms, Partitioning Clustering -K-Means Algorithm, K-Means Additional issues, PAM Algorithm; Hierarchical Clustering – Agglomerative Methods and divisive methods, Basic

Agglomerative Hierarchical Clustering, Strengths and Weakness; Outlier Detection.

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. Data Mining – Concepts and Techniques – Jiawei Han, Michelinen Kamber, Morgan Kaufmann Publishers, Elsevier.
2. Introduction to Data Mining, Pang – Ning Tan, Vipin Kumar, Michael Steinbanch, Pearson Education.
3. Data Mining Techniques, Arun K Pujari, Universities Press.
4. Data Warehouse Fundamentals, Pualraj Ponnaiah, Wiley Student Edition.
5. Data Mining, Vikaram Pudi, P Radha Krishna, Oxford University Press

Course Outcomes (COs):

At the end of this course students will be able to

1. Identify components of data warehouse and data mining.
2. Understand functionality of various components.
3. Use association rules, classification, clustering to mine the data.
4. Analyze parameters of various techniques and its applicability to dataset.
5. Compare different methodologies of data warehousing and data mining.
6. Judge the efficacy of various data warehousing strategy and data mining techniques.

Course Code: 18CP3156						**Course Name: Principles of Programming Language**		
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

Learning Objective

- To Understand various programming paradigms.
- To provide conceptual understanding of High level language design and implementation.
- To analyze the power of scripting languages.

UNIT I (9 L)

Preliminary Concepts: Reasons for studying, concepts of programming languages, Programming domains, Language Evaluation Criteria, influences on Language design, Language categories, Programming Paradigms – Imperative, Object Oriented, functional Programming , Logic Programming. Programming Language Implementation – Compilation and Virtual Machines, programming environments. Syntax and Semantics: general Problem of describing Syntax and Semantics, formal methods of describing syntax - BNF, EBNF for common programming languages features, parse trees, ambiguous grammars, attribute grammars, denotational semantics and axiomatic semantics for common programming language features.

UNIT II (10 L)

Data types: Introduction, primitive, character, user defined, array, associative, record, union, pointer and reference types, design and implementation uses related to these types. Names, Variable, concept of binding, type checking, strong typing, type compatibility, named constants, variable initialization. Expressions and Statements: Arithmetic relational and Boolean expressions, Short circuit evaluation mixed mode assignment, Assignment Statements, Control Structures – Statement Level, Compound Statements, Selection, Iteration, Unconditional Statements, guarded commands.

Subprograms and Blocks: Fundamentals of sub-programs, Scope of life time of variables, static and dynamic scope, design issues of subprograms and operations, local referencing environments, parameter passing methods, overloaded sub-programs, generic sub-programs, parameters that are sub-program names, design issues for functions user defined overloaded operators, co routines.

Unit III (10 L)

Abstract Data types: Abstractions and encapsulation, introductions to data abstraction, design issues, language examples, C++ parameterized ADT, object oriented programming in small talk, C++, Java, C#, Ada 95

Concurrency: Subprogram level concurrency, semaphores, monitors, massage passing, Java threads, C# threads.

Exception handling: Exceptions, exception Propagation, Exception handler in Ada, C++ and Java.

Logic Programming Language: Introduction and overview of logic programming, basic elements of prolog, application of logic programming.

Unit IV(10 L)

Functional Programming Languages: Introduction, fundamentals of FPL, LISP, ML, Haskell, application of Functional Programming Languages and comparison of functional and imperative Languages.

Scripting Language: Pragmatics, Key Concepts, Case Study: Python- Values and Types, Variables, Storage and Control, Bindings and Scope, Procedural Abstraction, Separate Compilation, Module Library.

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. Concepts of Programming Languages Robert .W. Sebesta 8/e, Pearson Education.
2. Programming Language Design Concepts, D. A. Watt, Wiley dreamtech, rp.
3. Programming Languages, A. B. Tucker, R. E. Noonan, TMH.
4. Programming Languages, K. C. Louden, Thomson.
5. LISP Patric Henry Winston and Paul Horn Pearson Education.
6. Programming in Prolog, W. F. Clocksin & C. S. Mellish, Springer.
7. Programming Python, M. Lutz, O'reilly, SPD, rp.
8. Core Python Programming, Chun, Pearson Education.
9. Guide to Programming with Python, Michel Dawson, Thomson

Course Outcomes (COs):

At the end of this course students will be able to

1. Remember syntax and semantics in formal notation.
2. Apply suitable programming paradigm for the application.
3. Compare distinct features of programming languages.
4. Analyze the cases of error handling.
5. Optimize the codes.
6. Judge the application oriented programming language.

Course Code: 18CP3157					**Course Name: Data Communication and Coding**			
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: Basic Electronics, Digital Logic Design

Learning objectives:

- To get familiarize with information and signals.
- To facilitate the understanding of the data communication systems.
- To facilitate the understanding of different information coding schemes.

Unit wise allocation of course content

UNIT 1 (08 L)

Signals, Spectra and Manipulations:

Information sources, Signals, Classification of signals: Periodic and aperiodic, Analog and digital, Time and frequency domain representations: Fourier series and transform, Concept of bandwidth, Unit impulse signal and concept of linear time invariant systems in terms of unit impulse response, Low pass and band pass signals, Concept of liner filtering in terms of convolution and its frequency domain interpretation, Modulation property of Fourier transform, Concept of modulation and demodulation, Baseband and carrier Modulation, Sampling and reconstruction, Sampling theorem (only statement), Aliasing and its remedy, Quantization, Pulse code modulation (PCM), Concept of analog to digital and digital to analog conversion (ADC and DAC).

UNIT 2 (10 L)

Fundamentals of Communication Systems:

Elements of communication systems, Amplitude modulation and demodulation; double-sideband and single sideband schemes, Modulation index, Power and bandwidth of modulated signals; Basic receiver block diagram, Frequency division multiplexing (FDM), Frequency and phase modulation and demodulation principles, , Digital baseband (Data) communication: Line codes –NRZ; RZ; Bipolar, Manchester, AMI, M-ary coding, ISI and its solutions, Time division multiplexing (TDM), Parallel and serial Transmission, Synchronous and asynchronous transmission, Simplex and duplex communication, Digital carrier modulation and demodulation schemes (ASK, FSK, PSK, QAM and QPSK): Block diagram. Working and types of modems, Transmission media, Guided and unguided: Twisted pair cable, fiber optic cable, wireless channels, Basic of radio wave propagation, Fading, Linear and non-linear distortions and channel effects, Noise effects.

UNIT 3 (07 L)

Information Theory and Source Coding:

Measure of information, Entropy, Source coding, Huffman and Shanon Fano coding, Uniqueness property, Error free communication Over a noisy channel, Shannon's theorem and Channel capacity, Practical communication system In light of Shannon's equation.

UNIT 4 (14 L)

Error Control Coding (Channel Coding):

Introduction, Linear block codes, Matrix description, Syndrome decoding, Error probability, Hamming distance, Perfect codes and Hamming codes, Low density parity check (LDPC) codes, Cyclic codes, Polynomial representation, Generation and decoding of Cyclic codes, Burst error correcting and detecting code, Golaycodes, Cyclic redundancy check (CRC) codes, Circuit implementation of cyclic codes, Introduction to BCH codes, Reed-Solomon codes, Convolution codes and Turbo codes, Encoding and decoding algorithms in brief, Comparison of coded and un-coded system.

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. Frenzel. "Communication electronics, principles and applications", Tata Mc-Graw Hill
2. Analog and Digital Communication, P. Chakrabarti, Dhanpat Rai & Co.
3. Modern Digital and Analog Communication Systems, B.P.Lathi, Zhi Ding, Oxford University Press.
4. Behrouz A Forouzan, "Data Communications and Networking", Tata Mc-Graw Hill
5. William Stallings, "Data and Computer Communications", Pearson Education
6. Information Theory, Coding and Cryptography, Ranjan Bose, PHI.
7. Digital Communication, Amitabh Bhattacharya, TMH.

Course Outcomes (COs):

At the end of this course students will be able to

1. Understand concepts related to bandwidth and spectrum.
2. Analyze the working of electronic communication systems.
3. Understand the use of communication systems for data transmission and reception.
4. Interpret the information as a quantity and its measurement.
5. Handling error detection and correction mechanism in modern data communication systems.
6. Design error resilience data communication system using coding techniques.

Course Code: 18CP3158					Course Name: Wireless Sensor Networks			
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

Learning Objective

- To understand the concepts of sensor networks.
- To understand the MAC and transport protocols for ADHOC networks.
- To understand the various routing protocols in sensor networks.
- To understand the security of sensor networks.
- To understand the applications of Adhoc and sensor networks.
- To critique protocol designs in terms of their energy-efficiency.

UNIT I (9 L)

OVERVIEW OF WIRELESS SENSOR NETWORKS: Key definitions of sensor networks, Advantages of sensor Networks, Unique constraints an challenges, Driving Applications, Enabling Technologies for Wireless Sensor Networks.

ARCHITECTURES: Single-Node Architecture – Hardware Components, Energy Consumption of Sensor Nodes, Operating Systems and Execution Environments, Network Architecture -Sensor Network Scenarios, Optimization Goals and Figures of Merit, Gateway Concepts.

UNIT II (10 L)

MAC Protocols for Wireless Sensor Networks: Issues in Designing a MAC protocol for Ad Hoc Wireless Networks, Design goals of a MAC Protocol for Ad Hoc Wireless Networks, Classifications of MAC Protocols, Contention – Based Protocols, Contention – Based Protocols with reservation Mechanisms, Contention – Based MAC Protocols with Scheduling Mechanisms, MAC Protocols that use Directional Antennas, Other MAC Protocols.

UNIT III (10 L)

ROUTING PROTOCOLS: Introduction, Issues in Designing a Routing Protocol for Ad Hoc Wireless Networks, Classification of Routing Protocols, Table –Driven Routing Protocols, On – Demand Routing Protocols, Hybrid Routing Protocols, Routing Protocols with Efficient Flooding Mechanisms, Hierarchical Routing Protocols, Power – Aware Routing Protocols, Proactive Routing.

UNIT IV (10 L)

TRANSPORT LAYER AND SECURITY PROTOCOLS: Introduction, Issues in Designing a Transport Layer Protocol for Ad Hoc Wireless Networks, Design Goals of a Transport Layer Protocol for Ad Hoc Wireless Networks, Classification of Transport Layer Solutions, TCP Over Ad Hoc Wireless Networks, Other Transport Layer Protocol for Ad Hoc Wireless Networks.

SECURITY IN WSNs: Security in Ad Hoc Wireless Networks, Network Security Requirements, Issues and Challenges in Security Provisioning, Network Security Attacks, Key Management, Secure Routing in Ad Hoc Wireless Networks.

SENSOR NETWORK PLATFORMS AND TOOLS: Sensor Node Hardware – Berkeley Motes, Programming Challenges, Node- level software platforms, Node-level Simulators, State-centric programming.

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. Ad-Hoc Wireless Networks: Architectures and Protocols – C. Siva Ram Murthy and B.S.Manoj, PHI
2. Wireless Ad- hoc and Sensor Networks: Protocols, Performance and Control – Jagannathan Sarangapani, CRC Press
3. Holger Karl & Andreas Willig, "Protocols And Architectures for Wireless Sensor Networks", John Wiley.
4. Kazem Sohraby, Daniel Minoli, & Taieb Znati, "Wireless Sensor Networks- Technology, Protocols, and Applications", John Wiley.
5. Feng Zhao & Leonidas J. Guibas, "Wireless Sensor Networks- An Information Processing Approach", Elsevier.
6. Ad-Hoc Mobile Wireless Networks: Protocols & Systems, C.K. Toh, Pearson Education.
7. Wireless Sensor Networks – C. S. Raghavendra, Krishna M. Sivalingam, Springer.
8. Wireless Sensor Networks – S Anandamurugan , Lakshmi Publications

Course Outcomes (COs):

At the end of this course students will be able to

1. Understand and explain the concept of wireless sensor networks and their applications.
2. Understand typical node and network architectures.
3. Critique protocol design in terms of their energy-efficiency.
4. Design and implement sensor network protocol in different environment.
5. Setup and evaluate measurements of protocol performance in wireless sensor networks.
6. Understand security issues in wireless sensor networks.

Course Code: 18CP3159					Course Name: Cryptography and Network Security			
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

Learning objectives:

- To introduce the concepts, technologies, practices, and challenges associated with Information and Network Security.
- Introduce the principles and practice of elementary number theory.
- To understand the advance level cryptographic algorithms.
- Introduce the importance and principal of network security.
- To understand the different types of threats and protection against those.

Unit wise allocation of course content

UNIT 1 (10 L)

Introduction to Elementary Number Theory, Finite Fields, Arithmetic and algebraic algorithms, Prime Numbers, Fermat’s and Euler’s Theorems, Primality Testing Algorithms – Deterministic and Probabilistic, Chinese Remainder Theorem, Quadratic Congruence, Discrete Logarithm, Factorization Methods, Side Channel Attacks, Shannon Theory, Perfect Secrecy, One Time Pad, Security proof of OTP, Semantic Security

UNIT 2 (10 L)

ElGamal Cryptosystem, Knapsack Algorithms, Elliptic Curve Arithmetic, Elliptic Curve Cryptography, Pseudorandom Number Generation, Need of Block Cipher, Luby-rackoff Construction and its security proof, Modes of Operation, Merkle-Damgard family of Hash Functions, Zero-Knowledge Protocols, Multi Party Protocols – Secret Sharing, Verifiable Secret Sharing, Introduction to Digital Signature, Digital Signature Schemes. Identity Based Cryptography, Attribute Based Cryptography.

UNIT 3 (10 L)

Security at Application Layer – Email, PGP, S/MIME, Security at Transport Layer – SSL, TLS, HTTPs, Security at Network Layer – IPSec, Wireless Network Security – IEEE 802.11

UNIT 4 (09 L)

Protection against Threats, intruders, Viruses and Worms, Malicious Software, Distributed Denial of Service Attacks, Security issues in Operating Systems, Intrusion Detection System Overview, Malware Detection and Prevention, Firewalls.

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. Behrouz A. Forouzan, “Cryptography and Network Security”, McGraw-Hill Education
4. Charlie Kaufman, Radia Perlman, Mike Speciner, “Network Security: Private Communication in a Public World”, Prentice Hall
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

Course Outcomes (COs):

At the end of this course students will be able to

1. Define the concepts related to the basics of network security and cryptography.
2. Demonstrate basic understanding of various network security applications, network security issues, authentication, and authorization mechanisms to protect against the threats in the networks.
3. Develop the understanding of software vulnerabilities exploited for attacks.
4. Classify the type of attack and type of vulnerability from given application
5. Choose appropriate mechanisms for protecting the network.
6. Design a security solution for a given application, system with respect to security of the system.