Bachelor of Engineering

(Four Year Full Time Degree Program)

SYLLABUS

(B. Tech. Third Year)

School of Computer Science and Engineering

Shri Mata Vaishno Devi University Katra

(April 2018)

	ABBREVIATIONS / CODES / NOMENCLATURE
Course Code Convention	
SCT – LSAY Example ALL 9101 ALP 9102 ALS 9110	Course Code for various Courses / Subjects SC: School Code T: Course Type Code (Lecture/Studio/Practical/Project etc.) L: Course Level (1, 2, 3, 4 & 5 for First, Second years …) SA: Study Area / Sub Area Y: Semester Wise Course Number
CSE	School Code (SoCSE)
L	Lecture
P	Practical
E	Elective
C	Colloquium
D	Project Based
T	Training
S	Self Study
N	Non Credit
V	Special Lecture Topic
Teaching Scheme Convention	
L	Lecture
T	Tutorial
P	Practical
C	Course Credit
Evaluation Scheme Convention	
Minor	(Mid Term Exams / Tests) I & II
Major	Semester End Examination (ESE)
FFCS	Fully Flexible Credit System
CBCS	Choice Based Credit System

Teaching & Examination Scheme

			B-Tech. Semester-V (Fall), Third Year											
			Teaching & Credits Scheme						Evaluation & Examination Scheme					
S. No	**Subject Code**	**Title of the Subject**	**L**	**T**	**P**	**S**	**Total Periods /week**	**C**	**Minor E Duration (Hours)**	**Major E Duration (Hours)**	**Internal Marks**	**Minor Marks (I+II)**	**Major Marks**	**Total Marks**
1		*Open Elective III	2	0	0			2	1.5	3	10	40	50	100
2	ECL 3061	Microprocessor & Interfacing	2	0	0			2	1.5	3	10	40	50	100
3	CSL 4053	Compiler Design	3	1	0			4	1.5	3	10	40	50	100
4	CSL 3023	Java Programming	3	0	2			5	1.5	3	10	40	50	100
5	CSL 3101	Artificial Intelligence	3	1	0			4	1.5	3	10	40	50	100
6	CSL 3032	Design & analysis of Algorithm	3	1	2			5	1.5	3	10	40	50	100
7	CSP 3100	Adobe Experience Manager Lab	0	0	2			1	1.5	3	10	40	50	100
8		Mini Project I						2						

Teaching & Examination Scheme

			B-Tech. Semester- VI (winter), Third Year											
			Teaching & Credits Scheme						Evaluation & Examination Scheme					
S. No	**Subject Code**	**Title of the Subject**	**L**	**T**	**P**	**S**	**Total Periods /week**	**C**	**Minor E Duration (Hours)**	**Major E Duration (Hours)**	**Internal Marks**	**Minor Marks (I+II)**	**Major Marks**	**Total Marks**
1	CSEXXXX	School Elective (IOT)	3	0	2			4	1.5	3	10	40	50	100
2	CSEXXXX	School Elective (Python)	3	0	2			4	1.5	3	10	40	50	100
3	CSL 3091	Software Engineering	3	0	2			4	1.5	3	10	40	50	100
4	CSL 3034	Soft Computing	3	0	2			4	1.5	3	10	40	50	100
5	CSL 4107	Machine Learning	3	1	2			5	1.5	3	10	40	50	100
6	CSL 4121	Computer Network Security	3	0	0			3	1.5	3	10	40	50	100
7		Mini Project II						2						

ECL 3061			**Microprocessors & Interfacing**				Pre Requisites			
Version R-01							Co-requisites			
L	T	P	C	Minor Duration	Major Duration	Internal Marks	Minor-I Marks	Minor-II Marks	Major Marks	Total Marks
3	0	2	4	1.5 Hours	3.0 Hours	10	20	20	50	100

COURSE OBJECTIVES

- To introduce 8085 architecture and programming in assembly language.
- To introduce basic concepts of interfacing memory and peripheral devices to a microprocessor.
- To introduce serial and parallel bus standards.
- To introduce 8051 microcontroller.
- To introduce various advanced processor architectures such as 80X86, Pentium and Multicore processors.

COURSE CONTENTS

Unit-I

8085 Architecture: Introduction to microprocessors and microcontrollers, 8085 Processor Architecture, Internal operations, Instructions and timings, Programming the 8085 – Introduction to 8085 instructions, addressing modes and Programming techniques with Additional instruction.

Unit-II

Stacks and subroutines, interfacing peripherals - Basic interfacing concepts, interfacing output displays, interfacing input keyboards. Interrupts - 8085 Interrupts, Programmable Interrupt Controller (8259A). Direct Memory Access (DMA) – DMA Controller (Intel 8257), Interfacing 8085 with Digital to Analog and Analog to Digital converters.

Unit-III

Programmable peripheral interface (Intel 8255A), Programmable communication interface (Intel 8251), Programmable. Interval timer (Intel 8253 and 8254), Programmable Keyboard / Display controller (Intel 8279). Serial and parallel bus standards RS 232 C, IEEE 488.

Unit-IV

Introduction to Microcontrollers, 8051 – Architecture – Instruction set, Addressing modes and Programming Techniques. Comparison of various families of 8-bit micro controllers. System Design Techniques Interfacing of LCD, ADC, Sensors, Stepper motor, keyboard and DAC using microcontrollers Communication standards – serial RS232 and USB

Unit-V

Microprocessor Applications and trends in microprocessor Technology – 8-bit, 16-bit and 32-bit microprocessors. Advanced Processor Architecture – Register structure, Instruction set, Addressing modes of 8086.Features of advanced microprocessors.80386, 80486, Pentium and Multi-Core Processors.

SUGGESTED BOOKS

1. "Microprocessor Architecture, Programming, and Applications with the 8085" by R Gaonkar
2. "The 8051 Microcontroller and Embedded Systems : Using Assembly and C" by Muhammad Ali Mazidi
3. "Introduction to Microprocessors and Microcontrollers" by Crisp John Crisp
4. "Microprocessors And Microcontrollers" by A Nagoor Kani
   Barry B. Brey, The Intel Microprocessor, 8086/8088,8018/80188, 80286, 80386, 80486, Pentium and Pentium pro-processors – architecture, Programming and interfacing, 4 Edition, Prentice Hall 1993.

COURSE OUTCOMES

After learning the course the students should be able to:

- List and specify the various features of microprocessor, memory and I/O devices including concepts of system bus.
- Identify the various elements of 8085 microprocessor architecture, its bus organization including control signals.
- List the pin functions of the 8085 microprocessor.
- Describe the 8085 processor addressing modes, instruction classification and function of each instruction and write the assembly language programs using 8085 instructions.
- Explain the concepts of memory and I/O interfacing with 8085 processor with Programmable devices.
- List and describe the features of advance microprocessors.

CSL 4053			Compiler Design				Pre Requisites			
Version R-01							Co-requisites			
L	T	P	C	Minor Duration	Major Duration	Internal Marks	Minor-I Marks	Minor-II Marks	Major Marks	Total Marks
3	1	0	4	1.5 Hours	3.0 Hours	10	20	20	50	100

COURSE OBJECTIVES

- To understand the theory and practice of compiler implementation.
- To learn finite state machines and lexical scanning.
- To learn context free grammars, compiler parsing techniques, construction of abstract syntax trees, symbol tables, intermediate machine representations and actual code generation

COURSE CONTENTS

Unit I: Introduction

Issues related to programming Language Design, Issues related to Finite-State Machines, Phases of Compiler Design, Lexical Analysis, Error Detection and Recovery.

Unit II: Basic Parsing Techniques

Parsers, Shift-Reduced Parsers, Operator-Precedence Parsing, Predictive Parsers.

Unit III: Top-Down Parsing, Bottom-up Parsing

LL(1) Grammars, Recursive Descent Parsers, LR Grammars – Concepts and Terminology, LR(O) Parsers, SLR(1) Parsers, Canonical LR(1) Parsers, LALR(1) Parsers, using ambiguous grammar. Attributed Translation Grammar, L-Attributed Translation Grammar.

Unit IV: Syntax-Directed Translation (SDT)

SDT Schemes, Implementation of SDTs, Intermediate Code, Parse Trees and Syntax Trees. Three Address Code, Quadruples and Triples. Translation schemes for Declarations, Assignment statements, Boolean Expressions, Flow of control statements, Array references in Arithmetic Expressions, Procedure Calls, Case Statements, and |Structures.

Unit V: Semantic Analysis & Type Checking

Introduction, Implicit-Stacking in Recursive Descent Compilation, Semantic Stacks in Bottom-up Compilation, Action-Symbols in Top-Down Compilation, Type Expressions, Overloaded Functions, Polymorphic Functions.

Unit VI: Symbol Table Handling Techniques

When to construct and Interact with the symbol Table, Symbol-Table Contents, Operations on symbol Table. Symbol Table organizations for Block – Structured Languages.

Unit VII: Run-Time Storage Organization and Management

Static Storage Allocation, Dynamic Storage Allocation, Heap Storage Allocation, Garbage Collection and Compaction.

Unit VIII: Code Optimization

Principal sources of Optimization, Loop Optimization, Loop-Invariant Computation, Induction variable elimination, Other Loop Optimizations, The DAG representation of Basic Blocks. Global Data – Flow Analysis.

Unit IX: Code Generation

Object programs, Problems Code Generation, A simple Code Generator. Register Allocation and Optimization, Code Generation from DAG, PEECPhole optimization.

Recommended Books:

1. Principles of Compiler Design; A. V. Aho & J. D. Ullman Narosa
2. The Theory and Practice of Compiler Writing, J Tremblay and Paul G. S.

COURSE OUTCOMES

After learning the course the students should be able to:

- To apply the knowledge of Lex tool & Yacc tool to develop a scanner & parser.
- To design & conduct experiments for Intermediate Code Generation in compiler.
- To design & implement a software system for backend of the compiler.
- To deal with different translators.
- To develop program to solve complex problems in compiler
- To learn the new code optimization techniques to improve the performance of a program in terms of speed & space.
- To acquire the knowledge of modern compiler & its features.
- To learn & use the new tools and technologies used for designing a compiler
- To use the knowledge of patterns, tokens & regular expressions for solving a problem in the field of data mining.

CSL 3023			Java Programming (PL – 5)				Pre Requisites			
Version R-01							Co-requisites			
L	T	P	C	Minor Duration	Major Duration	Internal Marks	Minor-I Marks	Minor-II Marks	Major Marks	Total Marks
3	0	2	4	1.5 Hours	3.0 Hours	10	20	20	50	100

COURSE OBJECTIVES

This Java course will provide you with

- A strong understanding of basic Java programming elements and data abstraction using problem representation and the object-oriented framework.
- As the saying goes, "A picture is worth a thousand words." This course will use sample objects such as photos or images to illustrate some important concepts to enhance understanding and retention.
- Student will learn to write procedural programs using variables, arrays, control statements, loops, recursion, data abstraction and objects in an integrated development environment.

COURSE CONTENTS

Unit I: Principles of Object Oriented Programming with Introduction to JAVA

The Traditional approach, drawback of procedure oriented languages, the three basic constructs of OOPS including abstraction and encapsulation, inheritance and polymorphism, comparison of various object oriented languages, Need of java, The creation of java, Basic differences of java and c++, byte code, difference between JDK, JRE, JVM, java applets and applications, java buzzword, three basic constructs of oops applicable to java.

Unit II: Data types, variables, and Arrays

Classification of various data types used in java(including Integer, float, characters, Boolean), closer look at the literals used in java, defining and initialization of variables, type conversion and casting, automatic type promotions in expressions, arrays(one dimensional and multidimensional).

Unit III: Operators and control statement

Arithmetic operators, bitwise operators, relational operators, Boolean logical operators, assignment operator, ? Operator, operator precedence, java's selection statement (if, switch statement), iteration statement(while, do-while, for, nested loops) Jump statement (break, continue).

Unit IV: Classes and Methods

Class fundamentals, declaring objects, assigning object to reference variables, constructors (default and parameterized), this keyword, garbage collection, finalize keyword, method introduction and returning a value from a method, overloading method, overloading constructor, object as parameter, returning objects, recursion, understanding static keyword, final keyword, introduction to inner and nested classes, exploring String class, using command line argument.

Unit V: Inheritance and package introduction and Exception Handling

Inheritance basic, use of super, method overriding, abstract class, Object class, defining a package, access protection, importing a package, introduction to interface, defining a interface, applying a interface, variables in interface, extension of interface, fundamentals of Exception handling, types of exception, use

of try and catch, nested try block, throw, throws, finally keywords, java's built in exception, creating your own exception.

Unit VI: Multithreaded Programming and I/O

The java thread model, thread priorities, synchronization, crating a thread, creating multiple thread, using is Alive() and join(),Synchronization in multiple thread, I/O basics, streams(byte and character), reading and writing console input and output, Reading and writing files.

Unit VII: Applet, Event Handling, and AWT

Applet fundamentals, applet architecture, Applet skeleton, passing parameter to applet, Delegation event model, Exploring all Event Classes, Event Listener interface, Adapter class, Anonymous inner class, Window fundamentals, working with frame window, working with graphics, working with color, fonts, layout managers, using of buttons, checkbox, choice lists, lists, scroll bar, text fields, text area, menu bars and menus, and handling the corresponding events generated by above components.

Recommended Books:

1. The complete Reference Java2(Fifth Edition),Herbert Schildt: TMH

2. Java how to program(Sixth Edition), Deitel and Deitel : PHI Publication

3. Programming with Java, E. Balaguruswamy : TMH publication

COURSE OUTCOMES

After learning the course the students should be able to:

- Write, compile, and execute Java programs using J2SE or other Integrated Development Environments (IDEs) such as Eclipse, NetBeans, and JDeveloper.
- Write, compile and execute Java programs using object oriented class structures with parameters, constructors, and utility and calculations methods, including inheritance, test classes and exception handling.
- Write, compile, and execute Java programs using arrays and recursion.
- Write, compile, and execute Java programs manipulating Strings and text documents.
- Write, compile, execute Java programs that include GUIs and event driven programming

CSL 3101			Artificial Intelligence				Pre Requisites			
Version R-01							Co-requisites			
L	T	P	C	Minor Duration	Major Duration	Internal Marks	Minor-I Marks	Minor-II Marks	Major Marks	Total Marks
3	1	0	4	1.5 Hours	3.0 Hours	10	20	20	50	100

COURSE OBJECTIVES

The objective of the course is to –

- Present an overview of artificial intelligence (AI) principles and approaches.
- Develop a basic understanding of the building blocks of AI as presented in terms of intelligent agents: Search, Knowledge representation, inference, logic, and learning.
- Students will implement a small AI system in a team environment. The knowledge of artificial intelligence plays a considerable role in some applications students develop for courses in the program.

COURSE CONTENTS

The course aims to provide an understanding of the diverse branches of AI through a discussion of its theoretical foundations. At the end of the course the students shall have in-depth understanding of different knowledge representation formalisms and various techniques used for "reasoning" and "theorem proving".

Unit I: Introduction

AI History and applications. Overview of AI application areas: game playing, automated reasoning and theorem proving, expert systems, natural language understanding, planning and robotics, machine learning and Alan Turing Test.

Unit II: The Propositional and Predicate Logic

Symbol and sentences, the semantics of the Propositional Calculus & Predicate Calculus. Inference Rules and Theorem Proving. Axioms, Literals, Horn clause & Clausal forms.

Unit III: Reasoning

Inductive, Deductive, Abductive and Default reasoning. More examples on Resolution proof.

Unit IV: Problem Solving as Search

Structures and strategies for state space search. Algorithms for Heuristic search, Heuristic evaluation functions.

Unit V: Knowledge Representation

Knowledge representation Techniques; conceptual graphs; structured representations; frames, scripts; issues in knowledge representation: hierarchies, inheritance, exceptions.

Unit VI: Knowledge Elicitation and Knowledge Acquisition

An overview of the induction methods, types and tools. Stages in Knowledge acquisition with examples. Analyzing, coding, documenting and diagramming. Scope of knowledge.

Unit VII: Expert Systems

Overview of expert system technology; rule-based expert systems; Construction of ES. Components of an ES, The explanation facility, Rule-based formation and forward and backward chaining techniques for problem solving.

Unit VIII: Natural Language Processing

Introduction. Vocabulary and issues, How NLP programs work, Natural Language application, NL Interfaces.

Recommended Books:

1. Artificial Intelligence - A New Synthesis by Nils J. Nilsson, Morgan Kaufmann Publishers.

2. Artificial Intelligence: Strategies and techniques for complex problems solving by George Luger, Addison-Wesley, 2003.

3. Artificial Intelligence - A Modern Approach by Stuart Russell & Peter Norvig, Prentice Hall.

COURSE OUTCOMES

Upon successful completion of this course student will:

- To understand the basics of Artificial Intelligence , Intelligent Agents and its structure
- To understand the problem solving by various searching techniques
- To understand the concept of informed search and Exploration
- To understand the concept of constraint satisfaction Problems and Adversarial Search
- To Understand what is Reasoning and Knowledge Representation
- To understand the concept of Reasoning with Uncertainty & Probabilistic Reasoning
- To Understand the basic forms of Machine Learning, decision trees and statistical Learning setting.

CSL 3032			Design And Analysis Of Algorithm				Pre Requisites			
Version R-01							Co-requisites			
L	T	P	C	Minor Duration	Major Duration	Internal Marks	Minor-I Marks	Minor-II Marks	Major Marks	Total Marks
3	1	2	5	1.5 Hours	3.0 Hours	10	20	20	50	100

COURSE OBJECTIVES

Upon completion of this course, students will be able to do the following:

1. Analyze the asymptotic performance of algorithms.
2. Write rigorous correctness proofs for algorithms.
3. Demonstrate a familiarity with major algorithms and data structures.
4. Apply important algorithmic design paradigms and methods of analysis.
5. Synthesize efficient algorithms in common engineering design situations.

Unit I: Introduction

Notion of an Algorithm – Fundamentals of Algorithmic Problem Solving – Important Problem Types – Fundamentals of the Analysis of Algorithm Efficiency – Analysis Framework – Asymptotic Notations and its properties – Mathematical analysis for Recursive and Non-recursive algorithms.

Unit II: Brute Force And Divide-And-Conquer

Brute Force - Closest-Pair and Convex-Hull Problems-Exhaustive Search - Traveling Salesman Problem - Knapsack Problem - Assignment problem. Divide and conquer methodology – Merge sort – Quick sort – Binary search –

Unit III: Dynamic Programming And Greedy Technique

Computing a Binomial Coefficient – Warshall's and Floyd' algorithm – Optimal Binary Search Trees – Knapsack Problem and Memory functions. Greedy Technique– Prim's algorithm- Kruskal's Algorithm- Dijkstra's Algorithm-Huffman Trees.

Unit IV: Iterative Improvement

The Simplex Method-The Maximum-Flow Problem – Maximum Matching in Bipartite Graphs- The Stable marriage Problem

Unit V: Coping With The Limitations Of Algorithm Power

Limitations of Algorithm Power-Lower-Bound Arguments-Decision Trees-P, NP and NP-Complete Problems--Coping with the Limitations - Backtracking – n-Queens problem – Hamiltonian Circuit Problem – Subset Sum Problem-Branch and Bound – Assignment problem – Knapsack Problem – Traveling Salesman Problem- Approximation Algorithms for NP – Hard Problems – Traveling Salesman problem – Knapsack problem..

Text Books:

1. A. Aho, J. Hopcroft and J. Ullman “The Design and Analysis of algorithms”

2. D. E. Knuth “The Art of Computer Programming”, Vol. I & Vol. 2

3. Horowitz Ellis, Sahani Sartaz, R. Sanguthevar " Fundamentals of Computer Algorithms".

4. Goodman: Introduction to Design and Analysis Of Algorithms, TMH

COURSE OUTCOMES

Students who complete the course will have demonstrated the ability to do the following:

- Analysis of Algorithms: computational models, order notation, time and space complexities, worst-case and expected complexities, lower and upper bounds, Amortized cost
- Techniques for designing efficient algorithms: recursion, divide-and-conquer, dynamic programming, balancing and backtracking
- Problems on sets and sequences: merging, sorting, searching, and selection (including external memory)
- Graph/Network algorithms
- String matching algorithms
- Optimization algorithms LP, IP, SDP
- Tractable and intractable problems: The classes of P, NP and NP-Complete problems
- Approximation algorithms
- Spectral algorithms

CSP 3100			**Adobe Experience Manager Lab**				Pre Requisites		**HTML, CSS, Javascript**	
Version R-01							Co-requisites			
L	T	S/P	C	Minor Duration	Major Duration	Internal Marks	Minor-I Marks	Minor-II Marks	Major Marks	Total Marks
0	0	2	1	-	2.0 Hours	40	-	-	60	100

COURSE CONTENTS

Unit- 1 Architecture Overview

What is Adobe Experience Manager (AEM)?, Infrastructure-Level Functions, Basics of the Architecture Stack, Introduction to the Granite Platform, OSGi Framework, Introduction to the Java Content Repository (JCR), Introduction to Apache Sling, The Functional Building Blocks of AEM, Application-Level Functions, AEM Installation, AEM author instance, AEM Publish instance.

Unit-2 Authoring Skills

Create and edit a page: create responsive pages and subpages, edit the responsive page; Page properties: edit property of a page, add a common property to multiple pages; Organizing and managing assets: create and edit assets, edit properties of asset, add a content fragment and asset to a page; Reorganizing site structure; Visioning and Tagging a page: create page version, compare page version by using timewrap, add tags to a page and search the tag and save result; Annotation on a page; Locking and unlocking a page; Publishing and Unpublishing pages, implementing business processes: automating the page publishing process by using a workflow.

Unit-3 Advanced Authoring Skills

Creating and Managing Templates: Create and edit a template, edit the template structure, define initial content and layout of the template; Creating and Managing content fragments: create and edit content fragment, create and edit experience fragment; Creating and Managing workflows, Launches, Projects, Tags

Unit -4 Personalization and Content Targeting

Personalization Console, ContextHub Console, Create an activity and map audience with user experiences, create and simulate targeted content

Unit-5 Developer Tools

Create, build and download a package, install a package; Content Rendering: create a project structure, create a page-rendering component, and create content to be rendered; Sling resource resolution and hunt for a rendering script

Unit-6 Advanced Developer Tools

Create a Template, test the contentpage template, restrict template use, Add content structure to the template, create the pages for the site using template; Introduction to HTL: Render basic page content, Modularize the contentpage component, Inheritance: Investigate the contentpage sling: resourceType property, investigate Foundation page component, delete the component.html script.

Suggested Books

1. Adobe Experience Manager: A practioners guide – V ilango, Vishwanath C R, Vandana, Shiva Balan, 2017
2. Adobe Experience Manager: Classroom in a Book – Ryan D Lunka, 2014, Adobe Press.
3. Adobe Experience Manager: Quick Reference Guide, Shane closer, 2014, Adobe Press.
4. Digital Marketing Handbook, Shivani karwal, 2015, Create Space Independent Publishing Platform

COURSE OUTCOMES

After successful completion of this course, students shall be able to;

- Create online brand building initiatives using AEM.
- Integrate digital solutions and strategic marketing plans using AEM.
- Implement online solution for business and marketing problems.
- Get exposure in developing websites using AEM environment.
- Implement solutions for online marketing in a global scenario.

CSE XXXX			**Internet of Things**				Pre Requisites		Computer Networks	
Version R-01							Co-requisites			
L	T	P	C	Minor Duration	Major Duration	Internal Marks	Minor-I Marks	Minor-II Marks	Major Marks	Total Marks
3	0	2	4	1.5 Hours	3.0 Hours	10	20	20	50	100

COURSE OBJECTIVES

1. To understand what Internet of Things is.
2. To get basic knowledge of RFID Technology, Sensor Technology and Satellite Technology.
3. To make students aware of resource management and security issues in Internet of Things.

COURSE CONTENTS

Unit I: Introduction

What is the Internet of Things? : History of IoT, About IoT, Overview and Motivations, Examples of Applications, Internet of Things Definitions and Frameworks : IoT Definitions, IoT Architecture, General Observations, ITU-T Views, Working Definition, IoT Frameworks, Basic Nodal Capabilities

Unit II: Fundamental IOT Mechanisms and Key Technologies

Identification of IoT Objects and Services, Structural Aspects of the IoT, Environment Characteristics, Traffic Characteristics, Scalability, Interoperability, Security and Privacy, Open Architecture, Key IoT Technologies, Device Intelligence, Communication Capabilities, Mobility Support, Device Power, Sensor Technology, RFID Technology, Satellite Technology,

Unit III: Radio Frequency Identification Technology

RFID: Introduction, Principle of RFID, Components of an RFID system, Issues

EPCGlobal Architecture Framework: EPCIS & ONS, Design issues, Technological challenges, Security challenges, IP for IoT, Web of Things.

Wireless Sensor Networks: History and context, WSN Architecture, the node, Connecting nodes, Networking Nodes, Securing Communication

WSN specific IoT applications, challenges: Security, QoS, Configuration, Various integration approaches, Data link layer protocols, routing protocols and infrastructure establishment.

Unit IV: Resource Management in The Internet Of Things

Clustering, Software Agents, Clustering Principles in an Internet of ThingsArchitecture, Design Guidelines, and Software Agents for Object Representation, Data Synchronization.

Identity portrayal, Identity management, various identity management models: Local, Network, Federated and global web identity, user-centric identity management, device centric identity management and hybrid-identity management, Identity and trust.

Unit V: Internet of Things Privacy, Security and Governance

Vulnerabilities of IoT, Security requirements, Threat analysis, Use cases and misuse cases, IoT security tomography and layered attacker model, Identity establishment, Access control, Message integrity, Non repudiation and availability, Security model for IoT.

Unit VI: Business Models For The Internet Of Things

Business Models and Business Model Innovation, Value Creation in the Internet of Things, Business Model Scenarios for the Internet of Things.

Unit VII: Internet of Things Application

Smart Metering Advanced Metering Infrastructure, e-Health Body Area Networks, City Automation, Automotive Applications, Home Automation, Smart Cards,

Suggested Books

1. Daniel Minoli, "Building the Internet of Things with IPv6 and MIPv6: The Evolving World of
M2M Communications", ISBN: 978-1-118-47347-4, Willy Publications
2. Bernd Scholz-Reiter, Florian Michahelles, "Architecting the Internet of Things", ISBN 978-3-642-19156-5 e-ISBN 978-3-642-19157-2, Springer
3. Parikshit N. Mahalle& Poonam N. Railkar, "Identity Management for Internet of Things", River Publishers, ISBN: 978-87-93102-90-3 (Hard Copy), 978-87-93102-91-0 (ebook).

COURSE OUTCOMES

After learning the course the students should be able to:

- Explain what Internet of Things is.
- Describe key technologies in Internet of Things.
- Understand wireless sensor network architecture and its framework along with WSN applications.
- Explain resource management in the Internet of Things.
- Understand business models for the Internet of Things.

CSL 3029			Programming in Python (PL-2)				Pre Requisites			
Version R-01							Co-requisites			
L	T	P	C	Minor Duration	Major Duration	Internal Marks	Minor-I Marks	Minor-II Marks	Major Marks	Total Marks
3	0	2	4	1.5 Hours	3.0 Hours	10	20	20	50	100

COURSE OBJECTIVES

The course is designed to provide Basic knowledge of Python. Python programming is intended for software engineers, system analysts, program managers and user support personnel who wish to learn the Python programming language.

COURSE CONTENTS

Unit -1

Statements, expressions, variables

Understand the structure of this class, explore Python as a calculator.

Unit 2

Functions, logic, conditionals

Learn the basic constructs of Python programming, create a program that plays a variant of Rock –Paper-Scissors.

Unit 3

Event –Driven programming

Learn the basics of event driven programming, understand difference Local and global variables, create an interactive progam that plays a simple guessing game.

Unit 4

Canvas, drawing, timers

Create a canvas in Python ,learn how to draw on the canvas ,create a digital stopwatch.

Unit 5

Lists, keyboard input, the basics of modeling motion

Learn the basics of lists in python ,model moving objects in python ,recreate the classic arcade game “Pong”.

Text Books:

1. Think Python,by Allen B. Downey ,second edition ,O'Reilly,Sebastopol,California.
2. Online Version
   www.greenteapress.com/thinkpython2.pdf.

3. How to think like a computer Scientist,by Brad Miller and David Ranum.
   Online Version
   www.interactivepython.org/runstone/static/thinkscpy/index.html.

COURSE OUTCOMES

After learning the course the students should be able to:

- Implement a given algorithm as a computer program (in Python).
- Adapt and combine standard algorithms to solve a given problem (includes numerical as well as non-numerical algorithms).
- Adequately use standard programming constructs: repetition, selection, functions, composition, modules, aggregated data (arrays, lists, etc.).
- Explain what a given program (in Python) does.
- Identify and repair coding errors in a program.
- Understand and use object based software concepts (constructing OO software will be dealt with in the course Software Engineering).
- Use library software for (e.g.) building a graphical user interface, web application, or mathematical software.

CSL 3091			Software Engineering				Pre Requisites			
Version R-01							Co-requisites			
L	T	P	C	Minor Duration	Major Duration	Internal Marks	Minor-I Marks	Minor-II Marks	Major Marks	Total Marks
3	0	2	4	1.5 Hours	3.0 Hours	10	20	20	50	100

COURSE OBJECTIVES

- Graduates are effective team members, aware of cultural diversity, who conduct themselves ethically and professionally.
- Use effective communication skills and technical skills to assure production of quality software, on time and within budget.
- Build upon and adapt knowledge of science, mathematics, and engineering to take on more expansive tasks that require an increased level of self-reliance, technical expertise, and leadership.

COURSE CONTENTS

Unit I: Introduction

The role of Software, Software Characteristics, Industrial strength software, Classification of software products, Legacy Software, Software Engineering Challenges, Software Development Life Cycle.

Unit II: Software Process

Software Development Process Models: Waterfall, Prototyping, Iterative, Spiral. Comparison of Models, Project Management Process, Inspection Process, Software Configuration management Process, Requirements Change management Process, Agile Process.

Unit III: Feasibility Study, Requirements Engineering & Analysis Modeling

Feasibility study: Technical, Economic & Behavioral; Data Gathering: Sources of Data, Observation, Interviewing, Questioners, On-site Observation, Software Process & Characteristics, Software Requirements, Problem Analysis: Data Flow Modeling, Object Oriented Modeling, Prototyping, Cost Benefit Analysis, SRS, Developing Use Cases. Validation & Metrics

Unit IV: Planning Software Projects

Effort Estimation: Constructive Cost Model (COCOMO), Project Scheduling, SCM planning, Quality Planning, Risk Management, Project Monitoring Planning

Unit V: Design Engineering

Design Concepts & Principles, Cohesion, Coupling, Design Methodology, Introduction to Unified Modeling Language (UML), Verification, Metrics

Unit VI: Coding & Testing

Programming principles, Coding Conventions, Coding process, Refactoring, Verification, Coding Metrics, Test Cases, Test Plan, White box & Black box testing, Unit Testing, Integration Testing, Validation Testing: Alpha & Beta Testing, System Testing, Debugging, Testing Metrics

Unit VII: Reliability, Quality & Maintenance

Software Reliability & Metrics, ISO 9000 Standard, Capability Maturity Model, CASE Tools, User Training, Software Maintenance

Recommended Books:

1. Software Engineering: A practitioner's Approach, Pressman, 6th Ed., McGraw Hill
2. System Analysis & Design, Elias M Awad
3. Fundamentals of Software Engineering, Ghezzi, C ,PHI
4. Managing the Software Process, W S Humphrey Addison–Wesley
5. Ed. Encyclopedia of Software Engineering, Vols 1&2 , J J Marciniak, John Wiley
6. Software Engineering, 5th Edition, Sommerville Ian Addison Wesley.
7. Software Engineering., Manmdrioli, Dino
8. Software Engineering: A programming Approach,3rd Edition, Bell, Douglas
9. An integrated Approach to Software Engineering. ,Jalote, P ,Narosa Pub House

COURSE OUTCOMES

The following are the Software Engineering Student Outcomes describing what students are expected to know or be able to do by the time of graduation from the program:

- To broaden your knowledge of Software Process Models.
- To increase your proficiency in Software Project Management.
- To gain practical experience in Requirements Engineering.
- To gain practical experience in UML tools.
- To understand and be able to explain Software Metrics and Software Reliability.
- To learn concepts associated with Software Construction.
- To learn about Software Verification.
- To develop, implement, and demonstrate the learning through a project that meet stated specifications.
- To learn User Interface Design. Understand Software Cost Estimation and web engineering

CSL 3034			Soft Computing				Pre Requisites			
Version R-01							Co-requisites			
L	T	P	C	Minor Duration	Major Duration	Internal Marks	Minor-I Marks	Minor-II Marks	Major Marks	Total Marks
3	0	2	4	1.5 Hours	3.0 Hours	10	20	20	50	100

COURSE OBJECTIVES

After Successful Completion of this Course, students shall be able to;

- To understand the fundamental theory and concepts of neural networks, neuro-modelling, several neural network paradigms and its applications.
- To understand the concepts of fuzzy sets, knowledge representation using fuzzy rules, approximate reasoning, fuzzy inference systems, and fuzzy logic control and other machine intelligence applications of fuzzy logic.
- To understand the basics of an evolutionary computing paradigm known as genetic algorithms and its application to engineering optimization problems.

COURSE CONTENTS

Unit I
Artificial Neural Network
Introduction – Fundamental concept – Evolution of Neural Networks – Basic Models of Artificial Neural Networks – Important Terminologies of ANNs – McCulloch-Pitts Neuron – Linear Separability – Hebb Network. Supervised Learning Network: Perceptron Networks – Adaline – Multiple Adaptive Linear Neurons – Back-Propagation Network – Radial Basis Function Network.

Unit II
Artificial Neural Network Associative Memory Networks: Training Algorithms for Pattern Association – Autoassociative Memory Network – Heteroassociative Memory Network – Bidirectional Associative Memory – Hopfield Networks – Iterative Autoassociative Memory Networks – Temporal Associative Memory Network. Unsupervised Learning Networks: Fixed weight Competitive Nets – Kohonen Self-Organizing Feature Maps – Learning Vector Quantization – Counter propagation Networks – Adaptive Resonance Theory Networks – Special Networks.

Unit III
Fuzzy Set Theory Introduction to Classical Sets and Fuzzy sets
Classical Relations and Fuzzy Relations – Tolerance and Equivalence Relations – Noninteractive Fuzzy sets – Membership Functions: Fuzzification – Methods of Membership Value Assignments – Defuzzification – Lambda-Cuts for Fuzzy sets and Fuzzy Relations – Defuzzification Methods.

Unit IV
Fuzzy Set Theory
Fuzzy Arithmetic and Fuzzy Measures: Fuzzy Rule Base and Approximate Reasoning: Truth values and Tables in Fuzzy logic – Fuzzy Propositions – Formation of Rules – Decomposition and Aggregation of rules – Fuzzy Reasoning – Fuzzy Inference Systems (FIS) – Fuzzy Decision Making – Fuzzy Logic Control Systems.

Unit V
(Genetic Algorithm) Introduction – Basic Operators and Terminologies in GAs – Traditional Algorithm vs. Genetic Algorithm – Simple GA – General Genetic Algorithm – The Scheme Theorem – Classification of Genetic Algorithm – Holland Classifier Systems – Genetic Programming. Applications of Soft Computing: A Fusion Approach of Multispectral Images with SAR Image for Flood Area Analysis – Optimization of Travelling Salesman Problem using Genetic Algorithm Approach – Genetic Algorithm based Internet Search Technique – Soft Computing based Hybrid Fuzzy Controllers – Soft Computing based Rocket Engine – Control.

Text Book:

S.N. Sivanandan and S.N. Deepa, Principles of Soft Computing, Wiley India, 2007. ISBN: 10: 81-265-1075-7.

Reference Books:

S. Rajasekaran and G.A.V.Pai, Neural Networks, Fuzzy Logic and Genetic Algorithms, PHI, 2003.

Timothy J.Ross, Fuzzy Logic with Engineering Applications, McGraw-Hill, 1997. 3. J.S.R.Jang, C. T.Sun and E.Mizutani, Neuro-Fuzzy and Soft Computing, PHI, 2004, Pearson Education.

COURSE OUTCOMES

- Ability to apply the knowledge of mathematics, science, engineering, computing in practice.
- An ability to identify, critically analyze, formulate and solve computational problems
- An ability to select appropriate engineering tools and techniques and use them comprehensively
- Ability to design and conduct research based experiments, perform analysis and interpretation of data and provide valid conclusions.
- Ability to use current techniques, skills, and IT tools necessary for computing practice.
- Understanding of ethical, legal, health, security, multi-cultural and social issues, and the consequent responsibilities relevant to the professional engineering practice.
- Understanding the impact of environment and propose the technological solutions for sustainable development.
- Practice ethical principles and oblige to professional ethics and responsibilities and norms of the engineering practice.
- Function effectively as an individual, and as a team member or team leader in diverse environments.
- Ability to communicate effectively, using different modes of communication and write technical reports and publications effectively.
- Ability to have adaptation towards learning new technologies, and develop an approach for their effective usage throughout the life.
- Understanding different principles of management and develop an ability to apply in software industry.

CSL 4107			Introduction to Machine Learning				Pre Requisites			
Version R-01							Co-requisites			
L	T	P	C	Minor Duration	Major Duration	Internal Marks	Minor-I Marks	Minor-II Marks	Major Marks	Total Marks
3	1	2	5	1.5 Hours	3.0 Hours	10	20	20	50	100

COURSE OBJECTIVES

1. Students have understanding of issues and challenges of Machine Learning.
2. Should be able to select data, model selection, model complexity etc.
3. Understanding of the strengths and weaknesses of many popular machine learning approaches.

COURSE CONTENTS

Unit 1

Brief Introduction to Machine Learning
Supervised Learning
Unsupervised Learning
Reinforcement Learning

Unit 2

Probability Basics
Linear Algebra
Statistical Decision Theory – Regression & Classification
Bias – Variance
Linear Regression
Multivariate Regression

Unit 3

Dimensionality Reduction
Subset Selection, Shrinkage Methods, Principle Components Regression
Linear Classification, Logistic Regression, Linear Discriminant Analysis
Optimization, Classification-Separating Hyperplanes Classification

Unit 4

Artificial Neural Networks (Early models, Back Propagation, Initialization, Training & Validation)
Parameter Estimation (Maximum Likelihood Estimation, Bayesian Parameter Estimation)
Decision Trees
Evaluation Measures, Hypothesis Testing
Ensemble Methods, Graphical Models

Unit 5

Clustering, Gaussian Mixture Models, Spectral Clustering
Ensemble Methods
Learning Theory, Reinforcement Learning

Suggested Books

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

COURSE OUTCOMES

- Identify the characteristics of datasets and compare the trivial data and big data for various applications.
- Understand machine learning techniques and computing environment that are suitable for the applications under consideration .
- Solve problems associated with batch learning and online learning, and the big data characteristics such as high dimensionality, dynamically growing data and in particular scalability issues.
- Develop scaling up machine learning techniques and associated computing techniques and technologies for various applications.
- Implement various ways of selecting suitable model parameters for different machine learning techniques.
- Integrate machine learning libraries, and mathematical and statistical tools with modern technologies like hadoop distributed file system and mapreduce programming model

CSL 4121			**Computer Network Security**				Pre Requisites			
Version R-01							Co-requisites			
L	T	P	C	Minor Duration	Major Duration	Internal Marks	Minor-I Marks	Minor-II Marks	Major Marks	Total Marks
3	0	0	3	1.5 Hours	3.0 Hours	10	20	20	50	100

COURSE OBJECTIVES

- Extensive, detailed and critical understanding of the concepts, issues, principles and theories of computer network security
- Detailed and practical understanding of formalisms for specifying security related properties and validating them using model checking
- Critical theoretical and detailed practical knowledge of a range of computer network security technologies as well as network security tools and services
- Practical experience of analysing, designing, implementing and validating solutions to computer network security challenges using common network security tools and formal methods.

COURSE CONTENTS:

Unit I

Attacks on computers and computer security, need for security, approaches , principles, types of attacks ,operational model of network security, Cryptography concepts and techniques, substitution transposition, encryption and decryption, symmetric ,Asymmetric key cryptography, key range size, possible type of attacks

Unit II

Mathematics of cryptography and DES Block ciphers modes, feistel ciphers DES. working of DES ,cracking des ,problems on des., 2DES, 3DES, des design ,Side channel attacks, Differential cryptanalysis.

Unit III

Symmetric-Key Cryptography**:** Glosis field theory, AES , overview of Rijndael comparison with others. Symmetric ciphers ,Blowfish in practice ,RC4, RC5,RC6,IDEA, RSA

Unit IV

Asymmetric-Key Cryptography**:** RSA, Elliptic curve cryptography ECC,Digital certificates and PKI.

Unit V

Cryptographic Hash Functios: Hashing schemes SHA- family, MAC, Digital Signature RSA El Gomel , DSS DSA, Authentication Protocols , applications Kerberos, X.509 Directory services

Unit VI

Network Security: Internet security protocols , SSL,TLS TSP WAP security, SET Hashing Authentication & Signature Schemes E-mail security, Email architecture SSL, PGP, MIME, S/MIME

Internet Protocol Security (IPSec) IPSec architecture, IPSec verses other layers security Mobile IPSec, VPN, Web security SSL, TLS, SET etc

Unit VII

System Security**:** Intruders , types of attacks , protecting against Intruders honeypots, scanning and analysis tools, Viruses and worms, types of viruses, protection, Firewall architecture implementing firewalls , xml firewalls , Trusted systems , trusted system applications , multi level security, trusted products.Security implementation, wireless security, securities in Adhoc-networks.

Suggested Books:

1.Cryptography and Network Security

Behrouz A. Forouzan, TMH

2. Cryptography And Network Security

Principles and Practices William Stallings, Prentice Hall

3. Cryptography and Network Security

Atul Kahate, Tata McGraw-Hill.

COURSE OUTCOMES

Upon completion of the Course:

- Understand the basic security services e.g. Authentication, Access Control, Confidentiality, Integrity, and Non repudiation).
- Understand the concepts of risk, threats, vulnerabilities and attack.
- Know the important ethical and legal issues to consider in computer security.
- Know the goals of end-to-end data security. [Familiarity]
- Understand the role of random numbers and prime numbers in security.
- Learn standard symmetric encryption algorithms[Assessment]
- Learn the architecture for public and private key cryptography and how public key infrastructure (PKI) supports network security.
- Learn the methods of digital signature and encryption.
- Learn key management and how key exchange protocols work.
- Learn security protocols at different layers of Network layer hierarchy.
- Learn futuristic cryptographic techniques like Eliptic Curve and quantum cryptography.
- Learn the concept of trusted computing.
- Learn the Web security Protocol.