ADVANCED DATABASE MANAGEMENT SYSTEM			
Course Code:	**WCS521**	**Course Credits:**	**3**
Course Category:	**CC1**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 Knowledge of database design
2 A general understanding of database, design and dependency
3 Understanding of different types of databases
4 Knowledge of databases on the internet
5 Application on enhanced database

COURSE OUTCOMES
At the end of the course the students should be able to:
1 Basic knowledge and understanding of ER diagram and UML class diagram.
2 Ability to apply functionality and Normalization in relational databases.
3 Recognize and fetch data from object oriented, parallel and distributed databases.
4 Use XML and understand unstructured data
5 Implement concept and deduction of enhanced database on different applications

UNIT I INTRODUCTION TO DATABASE DESIGN

Entities, Attributes, Entity Sets, Relationships, Key Constraints, Participation Constraints, Weak Entities, UML Class Diagrams, Subclasses, Superclasses, Inheritance,Specialization, Generalization, Constraints and Characteristics of Specialization and Generalization Hierarchies, Modeling of UNION Types Using Categories, Representing Specialization and Generalization In UML Class Diagrams, Data Abstraction, Knowledge Representation and Ontology Concepts.

UNIT II DATABASES DESIGN THEORY

Problems Caused by Redundancy, Decompositions, Problems Related to Decomposition, Reasoning About FD's, FIRST, SECOND, THIRD Normal Form, BCNF, Fourth Normal Form, Lossless Join Decomposition, DependencyPreserving Decomposition, Schema Refinement in DataBase Design, Multi Valued Dependencies.

UNIT III OBJECT- ORIENTED, PARALLEL AND DISTRIBUTED DATABASES

Overview of Object-Oriented Concepts, Object Identity, Object Structure, Type Constructor, Encapsulation of Operations, Methods and Persistence; Architectures for Parallel Databases, Parallel Query Evaluation, Parallelizing Individual Operations, Sorting Joins, Distributed Database Concepts, Data Fragmentation, Replication and Allocation Techniques for Distributed Database Design, Query Processing in Distributed Databases, Concurrency Control and Recovery in Distributed Databases.

UNIT IV DATABASES ON THE WEB AND SEMI-STRUCTURED DATA

Web interface, XML, structure of XML data, querying XML data, storage of XML data, XML applications, semi-structured data model, indexes for text data.

UNIT V ENHANCED DATA MODELS FOR ADVANCED APPLICATIONS

Active database concepts, temporal database concepts, spatial databases: concept and architecture, deductivedatabases and query processing, mobile databases, Geographic Information Systems (GIS).

Textbooks:

1. Elmasri and Navathe, Fundamentals of Database Systems,
2. Ramakrishnan and Gehrke, Database Management Systems,

References Books:

3. Korth, Silberschatz, Sudarshan, Database System Concepts,
4. Rob and Coronel, Database Systems: Design, Implementation and Management,
5. Date and Longman, Introduction to Database Systems.

ADVANCED OPERATING SYSTEM			
Course Code:	**WCS523**	**Course Credits:**	**3**
Course Category:	**CC1**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Understanding how Operating System is Important for Computer System.
2. To make aware of different types of Operating System and their services.
3. To learn different process scheduling algorithms and synchronization techniques to achieve better performance of a computer system
4. To know virtual memory concepts and secondary memory management
5. Understanding of Security & protection in Operating System

COURSE OUTCOMES
At the end of the course the students should be able to:
1.Understands the different services provided by Operating System at different level
2. They learn real life applications of Operating System in every field.
3. Understands the use of different process scheduling algorithm and synchronization techniques to avoid deadlock.
4. They will learn different memory management techniques like paging, segmentation and demand paging etc.
5. Perform implementation of protection mechanisms in operating system

UNIT I INTRODUCTION TO OPERATING SYSTEM

Importance of operating systems, basic concepts and terminology about operating system, memory management functions, processor management functions, device management functions, information management functions**.**

UNIT II PROCESS MANAGEMENT

Elementary concept of process, job scheduler, process scheduling, operation on process, threads, overview, scheduling criteria, scheduling algorithms, algorithm evaluation process synchronization, synchronization hardware, semaphores, monitors and atomic transaction deadlocks: system model, deadlock characterization, deadlocks prevention, deadlocks avoidance, deadlocks detection, recovery from deadlock.

UNIT III MEMORY MANAGEMENT

Memory management, logical versus physical address space, swapping, contiguous allocation, paging, segmentation, demand paging, page replacement, page replacement algorithms, allocation of frames, thrashing, demand segmentation.

UNIT IV STORAGE MANAGEMENT

File concept, directory structure, protection, file-system structure, allocation method, free-space management, directory implementation.

UNIT V I/O SYSTEMS

I/O hardware, Application of I/O interface, Overview of Kernel I/O subsystem, three types of I/O systems, memory based I/O, I/O based I/O, peripheral based I/O**.**

Text Books:

1. Galvin, Wiley, Operating Systems Concepts, 8th edition, 2009.
2. James L Peterson, Operating Systems Concept, John Wiley & Sons Inc, the 6Rev edition, 2007.
3. Deitel H. M., An Introduction to Operating Systems, Addison-Wesley, 1990.
4. Stallings William, Operating Systems, PHI, New Delhi, 1997.
5. Madnick and Donavon, Operating Systems, McGraw Hill, International edition, 1978.
6. S. Tanenbaum Modern Operating Systems, , Pearson Education, 3rd edition, 2007..

ADVANCED DATA STRUCTURE AND ALGORITHM			
Course Code:	**WCS525**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
Recognizing the appropriate data structures, ADT libraries, and using it to designalgorithms for a specific problem.
2. Be capable of solving problems using abstraction techniques.
3. Be able to choose appropriate algorithms for a specific problem.
4. Be able to analyse algorithms in terms of their efficiency and correctness.
5. To accept the recent developments in the area of algorithm design.
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Design and analyse programming problem statements.
2. Choose appropriate data structures and algorithms for a specific problem.
3. Gain the necessary mathematical abstraction to solve problems.
4. Come up with an analysis of efficiency and proof of correctness.
5. Comprehend and select algorithm design approaches in a problem specific manner.

UNIT I INTRODUCTION

Review of Basic Concepts: Abstract data types, Data structures, Algorithms, Big-Oh, Small-Oh, Omega, Small-Omega and Theta Notations, finding time complexity of programs, **Recurrence Relations:** Solving Recurrence Relations, Substitution Method, Master Theorem.

UNIT II

Hashing: Review of Hashing, Hash Function, Collision Resolution Techniques in Hashing, Separate Chaining, Open Addressing, Linear Probing, Quadratic Probing, Double Hashing, Rehashing, Extendible Hashing, Recent Trends in Hashing.

UNIT III TREES & GRAPH

Trees: Binary Search Trees, AVL Trees, Red Black Trees, 2-3 Trees, B-Trees, Splay Trees, Minimum Spanning Tree (MST), Kruskal's Algorithm and Prim's Algorithm, Applications to MST.

Graph: Graph, Breadth First Search, Depth First Search, Shortest path in edge-weighted case (Dijkstra's), Bellman Ford Algorithms, Topological Sorting.

UNIT IV SELECTED TOPICS

Strassen's Matrix Multiplication, Greedy method VS Dynamic Programming, Job sequencing with deadlines, Fractional Knapsack Problem, 0/1 Knapsack Problem, Travelling Salesman Problem, Huffman coding, Pre order, Post order, Inorder traversal, Postfix to infix notation, Infix to Postfix notation.

UNIT V

Linear Programming: Geometry of the feasibility region and Simplex algorithm
NP-completeness: Examples, proof of NP-hardness and NP-completeness.
Recent Trends: Recent Trends in problem solving paradigms using recent searching and sortingtechniques by applying recently proposed data structures.

Text Books:

1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, andClifford Stein.
2. Algorithms Unlocked: Thomas H. Cormen.
3. The Algorithm Design Manual, Steven S. Skiena.

References Books:

1. Algorithms: Robert Sedgewick and Kevin Wayne.
2. Advanced Data Structures: Peter Brass.

RESEARCH TECHNIQUES IN ICT			
Course Code:	**WCS527**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
No. of Lectures + Tutorials (Hrs./Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. To know what research is.
2. To make the students aware and apply ICT in Research.
3. To make the students recognize Scientific Research Methodology.
To equip the students to examine the philosophical and socio-cultural context ofresearch and relate it to the contemporary paradigm shift.
5. To explore different traditions such as empiricism, rationalism, and constructivism etc.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Discuss issues of research and knowledge creation
2. Apply appropriate research methods to research problems
3. Critique research methodologies
4. Able to write research papers so as to develop a great career in the field of research.
5. Design a research program.

UNIT I INTRODUCTION TO RESEARCH TECHNIQUES

Meaning of research, objectives of research, motivation in research, types of research, characteristics and prerequisites of research, significance of research, research process, sources of research problem, criteria of identifying the problem, necessity of defining the problem, errors in selecting research problem, technique involved in defining the problem, report, and paper writing.

UNIT II DATA ANALYSIS ANDSTATISTICAL TECHNIQUES

Data and their analyses, quantitative methods and techniques, Measure of central tendency, measuresof variation, frequency distribution, analysis of variance, methods, Correlation analysis, regression analysis, time series and forecasting, introduction to discriminant analysis, factor analysis, cluster analysis, conjoint analysis, probability distribution, binomial distribution, poisson distribution, uniform distribution, exponential distribution, and normal distribution, sampling methods, test of hypothesis.

UNIT III MATHEMATICAL MODELING

Steps of modelling, operations research models like queuing theory, stochastic processes, application of models, conceptual framework development and validation techniques, optimization techniques.

UNIT IV ALGORITHMIC RESEARCH

Algorithmic research problems, types of algorithmic research, types of solution procedure, steps of algorithm development, steps of algorithmic research, design of experiments.

UNIT V SIMULATION AND SOFT COMPUTING TECHNIQUES

Introduction to soft computing, artificial neural network, genetic algorithm, fuzzy logic and theirapplications, tools of soft computing, need for simulation, types of simulation, simulation language, fitting the problem to simulation study, simulation models, output analysis, data simulation packages like MATLAB, NS2, ANSYS, Cadence.

Text books:

1. Research Methodology: Methods and Techniques, C.R. Kothari

Reference Books:

1. Research Methodologies, R. Panneerselvam, Prentice Hall, 2007.
2. Research in Education, Best John V. and James V Kahn, Wiley eastern, 2005.
3. Elements of Educational Research, Sukhia, S.P., P.V. Mehrotra, and R.N. Mehrotra, PHI publication,2003.
4. Methodology of Research Education, K. Setia, IEEE publication, 2004.
5. Research methodology, Methods and Techniques, Kothari, C.R., 2000.

JAVA PROGRAMMING			
Course Code:	**WCS529**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 Knowledge of basic Object-Oriented paradigm, practices and application.
2 A general understanding of class, object and methods.
3 Understanding of multithreading and applets.
4 Basic knowledge of swings and Beans with implementation.
5 Understanding of Servlet programming.
COURSE OUTCOMES
At the end of the course the students should be able to:
1 Basic knowledge and understanding of object-oriented programming.
2 Ability to apply OOPs concept in real life problems.
3 Ability to design, develop, maintain and evaluate large-scale software systems.
4 To produce efficient, reliable, robust and cost-effective software solutions using Java.

5 Ability to perform independent research and analysis.

UNIT I OBJECT-ORIENTED PROGRAMMING

Concept of object-oriented programming (OOP), benefits of OOP, application of OOP, Java history, Java features, Data types, Java key words, identifiers, constants, variables, declaration and scope of the variable, symbolicconstant, type casting, arrays, strings, vectors, wrappers classes, operator, expressions, program control statements: decision making and branching: if, if…. else, else…. if, else if ladder, switch, decision making and looping while, do…. while, for.

UNIT II CLASSES, OBJECTS AND METHODS

Java class libraries, class fundamentals, object, methods, adding variables, add methods, creating objects, accessing class members, constructors, methods overloading, static members, nesting of methods, inheritance: extending a class, overriding methods, final variables and methods, final classes, finalizer methods, abstract methods and classes, visibility control, exception handling fundamental, Interface and Packages.

UNIT III MULTITHREADING AND APPLET PROGRAMMING

Multithreading programming: creating threads, thread class and runnable interface extending the thread class, stopping and blocking a thread, life cycle of a thread, thread methods, thread exceptions, thread priority, synchronization, thread communication using notify(), wait(), and notify all(), applet programming : applet basic, applets architecture, a complete applet skeleton, building applets code, applets life cycle, creating a executable applet, designing a web page, applets tag, passing parameters to applets, applets and HTML.

UNIT IV SWING AND BEANS

Introduction to Swing, Differences between AWT Controls & Swing Controls, JApplet, Swing Button: JButton, JToggleButton, CheckBoxes, Radio Button, JComboBox, Text Boxes etc., Icons, Labels, JTabbed Pains, JScroll Pains, JList, JTrees, JTables Java Beans: Introduction to Java Beans, Advantages of Java Beans, BDK Introspection, Developing a Home page using Applet & Swing.

UNIT V SERVLET PROGRAMMING

Introduction to Servlets: Lifecycle of a Servlet, The Servlet API, The javax. Servlet Package, Reading Servelet parameters, Reading Initialization parameters; The javax.servlet HTTP package, Handling Http Request & Responses, Security Issues Introduction to JSP, Problem with Servlet. The Anatomy of a JSP Page, JSP Processing. JSP Application Design with MVC Setting Up and JSP Environment: Installing the Java Software DevelopmentKit, Tomcat Server & Testing Tomcat.

REFERENCE BOOKS:

1. Programming with JAVA, E. Balagurusawamy, Tata McGraw Hill, 1998.
2. JAVA Beginner's guide, Herbert Schildt, Tata McGraw Hill, 2007.
3. Java How to Program, Deitel & Deitel, Prentice-Hall, 1999.
4. The Complete Reference JAVA 2, Herbert Schildt, 5th Edition, Tata McGraw Hill, 2002.
5. The Complete Reference JAVA 2, Herbert Schildt, 7th Edition, Tata McGraw Hill, 2009.
6. The Java Programming Language, Ken Arnold, James Gosling, Addison-Wesley, 1996.
7. How to Program Java, Peter Coffee, Ziff-Davis Press, 1996.

GRAPH THEORY			
Course Code:	**WCS531**	**Course Credits:**	**3**
Course Category:	**E2/DSE**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1.To understand and apply the fundamental concepts in graph theory.
2.To apply graph theory based tools in solving practical problems.
3.To improve the proof writing skills.
4.To introduce the idea of coloring in graphs
5.To have an idea of automorphism groups of graphs

COURSE OUTCOMES
At the end of the course the students should be able to:
1.Apply principles and concepts of graph theory in practical situation.
2.Identify induced subgraphs, cliques , matching, covers in graphs.
3.Determine whether graphs are Hamiltonian and/or Eulerian.
4.Solve problems involving vertex and edge coloring.
5.Fix the problems involving vertex and edge connectivity, planarity and crossing numbers.

UNIT I

Graphs, Sub graphs, some basic properties, various example of graphs & their sub graphs, walks, path& circuits, connected graphs, disconnected graphs and component, euler graphs, various operation on graphs, Hamiltonian paths and circuits, the traveling sales man problem.

UNIT II

Trees and fundamental circuits, distance diameters, radius and pendent vertices, rooted and binary trees, on counting trees, spanning trees, fundamental circuits, finding all spanning trees of a graph and a weighted graph, algorithms of primes, Kruskal and Dijkstra Algorithms.

UNIT III

Cuts sets and cut vertices, some properties, all cut sets in a graph, fundamental circuits and cut sets ,connectivity and separability, network flows Planer graphs, combinatorial and geometric dual: Kuratowski graphs, detection of planarity, geometric dual, Discussion on criterion of planarity, thickness and crossings.

UNIT IV

Vector space of a graph and vectors, basis vector, cut set vector, circuit vector, circuit and cut set subspaces, Matrix representation of graph – Basic concepts; Incidence matrix, Circuit matrix, Path matrix, Cut-set matrix and Adjacency matrix.

UNIT V

Coloring, covering and partitioning of a graph, chromatic number, chromatic partitioning, chromatic polynomials, matching, covering, four color problem Discussion of Graph theoretic algorithm wherever required.

Text Books:

1. Deo, N, Graph theory with applications to Engineering and Computer Science, PHI
2. Gary Chartrand aend Ping Zhang, Introduction to Graph Theory, TMH
3. Robin J. Wilson, Introduction to Graph Theory, Pearson Education

4. Harary, F, Graph Theory, Narosa
5. Bondy and Murthy: Graph theory and application. Addison Wesley.

ADVANCED DATABASE MANAGEMENT SYSTEM LAB			
Course Code:	**WCS581**	**Course Credits:**	**2**
Course Category:	**CC**	**Course (U / P)**	**U / P**
Course Year (U / P):	**4U / 1P**	**Course Semester (U / P):**	**8U / 2P**
No. of Lab (Hrs/Week) / Total No. of Lab	**03 / 10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. To explore the features of a Database Management Systems
2. To interface a database with front end tools
3. To comprehend the internals of a database system
4. To provide a strong foundation in advanced database concepts from an industry perspective.
o learn query processing and transaction management concepts for object-relational databaseand distributed database
COURSE OUTCOME
At the end of the course the students should be able to:
1. Develop and apply critical thinking skills.
2. Design and present Lab as well as project reports
3. Built appropriate methods for the analysis of raw data
4. Perform logical troubleshooting as and when required.
5. Verify and implement the concepts and theory learnt in class.

1. Introdution to MySQL, Postgre Sql, Microsoft Sql softwares.
2. An exercise of data types in PostGresql & Data Definition Language Commands
3. Exercise on Data Manipulation Language and Transaction Control Commands using PostgreSql.
4. Exercise on Types of Data Constraints using PostgreSql.
5. Exercise on JOINS (Single-Table) Using Normalization
6. Exercise on JOINS (Multiple-Table) Using Normalization
7. Exercise on GROUP BY/ORDER BY Clause and Date Arithmetic using PostgreSql.
8. Exercise on different Functions (Aggregate, Math and String)
9. Exercise on different types of sub queries

10. Procedures, View and Triggers.

JAVA PROGRAMMING LAB			
Course Code:	**WCS583**	**Course Credits:**	**2**
Course Category:	**CC-L2**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
No. of Labs (Hrs./Week):	**01 (3 Hr)**		
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 Knowledge of basic Object-Oriented paradigm, practices, and application.
2 A general understanding of class, object, and methods.
3 Understanding of multithreading and applet.
4 Basic knowledge of Swings and Beans with implementation.
5 Understanding of Servlet programming.

COURSE OUTCOMES
At the end of the course the students should be able to:
1 Basic knowledge and understanding of object-oriented programming.
2 Ability to apply OOPs concept in real life problem.
3 Ability to design, develop, maintain, and evaluate large-scale software systems.
4 To produce efficient, reliable, robust, and cost-effective software solutions using Java.
5 Ability to perform independent research and analysis.

1. Write a separate Java Code to implement each of the following:
   Class, Command Line Argument, how to enter value through keyboard

2. Write a separate Java Code to implement each of the following data types:
   Variable, Constant, Arrays, Strings, Vectors, Wrappers Classes, Type Casting, Operators,Decision statement, Loops statement and Branch statements and Exception handling

3. Write a separate Java Code to implement each of the following OOP's concepts:
   Abstraction, Encapsulation, Inheritance, Polymorphism, Method Overloading and MethodOverriding

4. Write a separate Java Code to implement each of the following:
   Class, Object, Constructors, Method, and Visibility Controls: Private, Public and Protected

5. Write a separate Java Code to implement each of the following:
   Final variable, final class, final method, abstract class, abstract method, and concrete method

6. Write a separate Java Code to implement each of the following:Interface, extending and implementing interface

7. Write a separate Java Code to implement each of the following:
   Multithreading: Create thread with thread class and runnable interface, thread priorities, synchronization

8. Write a separate Java Code to implement each of the following:Swing and Beans

9. Write a separate Java Code to implement each of the following:
   Swing Button: JButton, JToggleButton, CheckBoxes, Radio Button, JComboBox, Text Boxes

10. Write a separate Java Code to implement each of the following:Servlet and JSP

PYTHON PROGRAMMING			
Course Code:	**WCS522**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**2P**
No. of Lectures + Tutorials(Hrs./Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. To learn and understand Python programming basics and paradigm
2. To learn and understand python looping, control statements and string manipulations.
3. Students should be made familiar with the concepts of GUI controls and designing GUI applications.
4. To learn and know the concepts of file handling, exception handling and database connectivity.
5. To learn and understand database connectivity in python programming language.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. To read and write simple Python programs.
2. To develop Python programs with conditionals and loops.
3. To define Python functions and to use Python data structures –- lists, tuples, dictionaries
4. To do input/output with files in Python
5. To do searching, sorting and merging in Python

UNIT I INTRODUCTION

The Programming Cycle for Python, Python IDE, Interacting with Python Programs, Elements of Python, Type Conversion. Basics: Expressions, Assignment Statement, Arithmetic Operators, Operator Precedence, Boolean Expression.

UNIT II CONDITIONALS

Conditional statement in Python (if-else statement, its working and execution), Nested-if statement and Elif statement in Python, Expression Evaluation & Float Representation. Loops: Purpose and working of loops, while loop including its working, For Loop, Nested Loops, Break and Continue.

UNIT III FUNCTION

Parts of A Function, Execution of A Function, Keyword and Default Arguments, Scope Rules. Strings: Lengthof the string and perform Concatenation and Repeat operations in it. Indexing and Slicing of Strings. Python Data Structure: Tuples, Unpacking Sequences, Lists, Mutable Sequences, List Comprehension, Sets, DictionariesHigher Order Functions: Treat functions as first-class Objects, Lambda Expressions

UNIT IV FILE I/O

File input and output operations in Python Programming Exceptions and Assertions Modules: Introduction, Importing Modules, Abstract Data Types: Abstract data types and ADT interface in Python Programming. Classes: Class definition and other operations in the classes, Special Methods (such as _init_, _str_, comparison methods and Arithmetic methods etc.), Class Example, Inheritance, Inheritance and OOP.

UNIT V ITERATORS & RECURSION:

Recursive Fibonacci, Tower of Hanoi Search: Simple Search and Estimating Search Time, Binary Search andEstimating Binary Search Time Sorting & Merging: Selection Sort, Merge List, Merge Sort, Higher Order Sort.

Textbooks:

1. Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist ‘‘, 2nd edition, Updated forPython 3, Shroff/O ‘Reilly Publishers, 2016 (http://greenteapress.com/wp/thinkpython/)
2. Guido van Rossum and Fred L. Drake Jr, —An Introduction to Python – Revised and updated for Python3.2, Network Theory Ltd., 2011.
3. John V Guttag, —Introduction to Computation and Programming Using Python ‘‘, Revised and expandedEdition, MIT Press, 2013

ADVANCED SOFTWARE ENGINEERING			
Course Code:	**WCS524**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**2P**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 Knowledge of basic to advanced software engineering methods.
2 A General understanding of object-orientated software engineering.
3 Understanding of component-based software engineering.
4 Understanding of aspect-oriented software engineering.
5 Understanding of software re-engineering and reverse engineering.

COURSE OUTCOME
At the end of the course the students should be able to:
1 Basic to advanced knowledge and understanding of software engineering methods.
2 Ability to apply software engineering principles and techniques.
3 Ability to design, develop, maintain and evaluate object-orientated software.
4 To produce efficient, reliable, robust and cost-effective solutions for component-based software.
5 Ability to design a good software.

UNIT I SOFTWARE ENGINEERING

Introduction to software engineering, Software Development Life Cycle, software process models, requirement analysis and design, software design process, coding, software testing, implementation and maintenance, software metrics.

UNIT II OBJECT-ORIENTED SOFTWARE ENGINEERING

Object-Oriented Software Engineering, object-orientated paradigm, object modeling languages, object-oriented analysis, object-oriented design, object-oriented programming, object-oriented metrics, object-oriented case tools, object-oriented software testing.

UNIT III COMPONENT-BASED SOFTWARE ENGINEERING

Component-Based Software Engineering (CBSE), CBSE and software reuse, CBSE vs. object-oriented software engineering, CBSE processes, domain engineering, component engineering, component-based software development life cycle, component vs. object, component-oriented programming, component-oriented programming vs. object-oriented programming, component-based technology, component-based software testing, component-oriented metrics.

UNIT IV ASPECT-ORIENTED SOFTWARE ENGINEERING

Software engineering with aspects, aspects, aspect vs. object, aspect vs. component, join points and pointcuts, separation of concerns, crosscutting concerns, scattering and tangling, aspect-oriented programming, aspect-oriented software testing.

UNIT V SOFTWARE RE-ENGINEERING AND REVERSE ENGINEERING

Re-engineering concept and approaches, redevelopment vs. reengineering, reengineering process, software re-engineering techniques, reverse engineering, levels of reverse engineering: re-documentation, design recovery, specification recovery, conditions for reverse engineering, forward engineering, restructuring, re-engineering, benefits of reverse engineering.

TEXT BOOKS:

- Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa Publishing House,New Delhi 1997.
- Ian Sommerville, Software Engineering, Pearson Education, 2009.

REFERENCE BOOKS:

- Pressman Roger S., Software Engineering: Practitioner's Approach, McGraw-Hill Inc., 2004.
- N. S. Gill, Software Engineering: Software Reliability, Testing and Quality Assurance,Khanna Book Publishing Co (P) Ltd., New Delhi, 2002.
- J. Rumbaugh, M. Blaha, W. Premerlani, Object-Oriented Modeling and Design, PHI, 1991.
- George T. Heineman, William T. Councill, Component-Based Software Engineering: Puttingthe Pieces Together, Addision Wesley, 2001.
- Robert E. Filman, Tzilla Elrad, Siobhán Clarke, Mehmet Aksit, Aspect-Oriented Software Development Addison-Wesley Professional, 2004.

ADVANCED COMPUTER ARCHITECTURE			
Course Code:	**WCS-526**	**Course Credits:**	**3**
Course Category:	**CC9**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

UNIT I INTRODUCTION TO PARALLEL PROCESSING

Parallelism in uniprocessor system, basic uniprocessor architecture, parallel processing mechanism, balancing of sub system bandwidth, multiprogramming and time sharing, parallel computer structures, pipeline computers, array computers, multiprocessor systems, dataflow computer concept, architectural classification scheme: multiplicity of instruction-data streams, serial versus parallel processing, parallelism versus pipelining, parallel processing applications, productive modeling simulation,engineering design and automation.

UNIT II PRINCIPLES OF PIPELINING AND VECTOR PROCESSING

Pipelining- an overlapped parallelism, principles of linear pipelining, clock period, efficiency,

throughput, classification of pipeline processors, general pipeline and reservation tables.

UNIT III PRINCIPLES OF DESIGNING PIPELINE PROCESSORS

Effect of branching , data buffering and bussing structures, internal forwarding and register tagging, hazard detection and resolution, job sequencing and collision prevention, reservation and latency analysis, collision free scheduling, state diagram, greedy cycle, pipeline schedule optimization, pipeline throughput, pipeline efficiency.

UNIT IV STRUCTURE ANDALGORITHM FOR ARRAY PROCESSORS

SIMD array processor, SIMD computer organization, inter –PE communication, SIMD interconnection network, static versus dynamic networks, cube interconnection network, shuffle-exchange omega networks, parallel algorithms and SIMD matrix multiplication.

UNIT V MULTIPROCESSOR ARCHITECTURE AND SCHEDULING

Functional structure, loosely coupled and tightly coupled multiprocessor, deterministic scheduling strategy, deterministic scheduling model, control flow versus data flow computer, data flow graphs and languages.

Reference Books

1. Kai Hwang, “Advanced Computer Architecture”, Tata McGrawHill Edition
2. Kai Hwang and Faye A. Briggs, “Computer Architecture and Parallel Processing”, McGraw-Hill International Edition
3. Richard Y. Kain, “Advanced computer architecture: a systems design”, Prentice Hall.
4. James M. Feldman, Charles T. Retter, “Computer architecture: a designer's text based on ageneric RISC”, McGraw-Hill
5. Jurij Silc, Borut Robic, Theo Ungerer, “Processor Architecture: From Dataflow to Superscalar and Beyond”, Springer.
6. Hennessy and Patterson, “Computer Architecture: A Quantitative Approach”, Elsevier.
7. Dezso and Sima, “Advanced Computer Architecture”, Pearson.
8. Quinn, “Parallel Computing: Theory & Practice”, TMH.
9. Quinn, “Parallel Programming in C with MPI and Open MP”, TMH

PROBLEM SOLVING USING ARTIFICIAL INTELLIGENCE			
Course Code:	**WCS528**	**Course Credits:**	**3**
Course Category:	**CC10**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1. To provide a strong foundation of fundamental concepts in Artificial Intelligence
2. To enable the student to apply these techniques in applications which involve perception, reasoning and learning
3. To provide a basic exposition to the goals and methods of Artificial Intelligence
4. Explain the role of agents and how it is related to environment and the way of evaluating it and how agents can act by establishing goals.
5. Learn the different machine learning techniques to design AI machine and enveloping applications for real world problems.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Understand the various searching techniques, constraint satisfaction problem and example problems- game playing techniques.
2. Apply these techniques in applications which involve perception, reasoning and learning
3. Acquire the knowledge of real world Knowledge representation
4. Analyze and design a real world problem for implementation and understand the dynamic behavior of a system.
5. To enable the student to apply these techniques in applications which involve perception, reasoning and learning

UNIT I INTRODUCTION TO ARTIFICIAL INTELLIGENCE

Basic concept of artificial intelligence (AI), history of AI, AI and consciousness, weak and strong AI, physical symbol system hypothesis, comparison of computer and human skills, practical systems based on AI, development of logic, components of AI.

UNIT II PROBLEM SOLVING THROUGH AI

Defining problem as state space search, analysing the problem, representing the problems from AI viewpoint, production system, developing production rules, characteristics of production system, algorithm for problem solving using AI technique.

UNIT III SEARCH TECHNIQUES

Use of search in AI problem solution, blind search techniques, heuristic search techniques, concept of heuristic knowledge, designing of the heuristic function, types of heuristic search techniques: generate and test, best first search, problem reduction using AND-OR graph, local search technique, branch and bound search, memory bounded search technique, local beam search, properties ofheuristic search techniques, overestimation and underestimation of heuristic function hill climbing search, simulated annealing search, constraint satisfaction means ends analysis.

UNIT IV INTRODUCTION TO LOGIC

Introduction, proposition calculus, syntax o propositional calculus, semantics of propositional calculus, well-formed formula, properties of statements, inferencing of propositional logic, predicate logic, syntax of predicate logic, semantics of predicate logic, concept of resolution, resolution algorithm, skolemization, types of resolution unit resolution, binary resolution.

UNIT V PROLOG AND LISP

Basic concept of programming languages related to artificial intelligence problems, concept of programming in Logic, basic prolog constructs, atoms, defining the rules, writing small programs in prolog, concept of list processing, basic LISP constructs, writing functions in LISP, some simple programs of LISP.

Text Books:

1. Artificial Intelligence, Elanie Reich: Tata mcgraw Hill publishing house, 2008.

2. Artificial Intelligence, Peterson, TataMcGraw Hill, 2008.
3. Artificial Intelligence, Russel and Norvig, Pearson Printice Hall Publication, 2006.
4. Artificial Intelligence, Winston, PHI publication, 2006

PYTHON PROGRAMMING LAB			
Course Code:	**CS582**	**Course Credits:**	**2**
Course Category:	**CC**	**Course (U / P)**	**U / P**
Course Year (U / P):	**4U / 1P**	**Course Semester (U / P):**	**8U / 2P**
No. of Lab (Hrs/Week) / Total No. of Lab	**03 / 10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
nterpret the use of procedural statements like assignments, conditional statements, loopsand function calls.
2. Infer the supported data structures like lists, dictionaries and tuples in Python
3. Illustrate the application of matrices and regular expressions in building the Python programs.
4. Discover the use of external modules in creating excel files and navigating the file systems.
5. Describe the need for Object-oriented programming concepts in Python.

COURSE OUTCOME
At the end of the course the students should be able to:
1. To write, test, and debug simple Python programs.
2. To implement Python programs with conditionals and loops.
3. Use functions for structuring Python programs.
4. Represent compound data using Python lists, tuples, dictionaries
5. Read and write data from/to files in Python.

1. Write a python program find the maximum of a list of numbers.
2. Write a python program to perform Matrix Multiplication.
3. Write a python program first n prime numbers
4. Write a python program selection sort.
5. write a python program to compute the GCD of two numbers.
6. Write a python program to find the most frequent words in a text file.
7. Write a Python program to create a scientific calculator
8. Write a Python program to print all the Disarium numbers between 1 and 100.
9. Write a Python program to encrypt the text using Caesar Cipher technique. Display the encrypted text.Prompt the user for input and the shift pattern.
10. Write a Python program to construct a linked list. Prompt the user for input. Remove any duplicate numbersfrom the linked list.
11. Perform the following file operations using Python

OPEN SOURCE SOFTWARE SYSTEMS LAB			
Course Code: WCS584	**CS**	**Course Credits:**	**3**
Course Category:	**CC-P**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**2P**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	
Total No. of Lectures (L + T):	**10 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1.To expose students to free open-source software environment and introduce them to useopen-source packages.

2.Install Red Hat and Debian based Linux distributions

3.Maintain operating system updates

4. Interoperate between Linux and Windows, Install and configure useful applicationsoftware (LAMP stack apps, SAMBA, and others);

5.Understanding of each of the following: ◦ Basics of Linux security; Major issues involved in Open-Source licensing; Basics of Linux scripting and be able to write a simple script; Pros & cons of Linux & be able to decide where it makes sense to use it ornot.

COURSE OUTCOMES

At the end of the course the students should be able to:

1.Implement various applications using build systems.

2.Understand the installation of various packages in open-source operating systems

3.Create simple GUI applications using Gambas

4.Understand various version control systems

5. Understand the kernel configuration and virtual environment

LAB PRACTICALS

1. Compiling from source: Learn about the various build systems used like the cmake /make / ant etc. instead of just running the commands.
2. Introduction to package management system: Given set of RPM or DEB, how to buildand maintain, serve packages over http or ftp. And also, how do you configure client systems to access the package repository in any Open-Source Software Systems.
3. Install various software packages: (A)Install Linux and share files to windows (B)Install Common Unix Printing System (CUPS)
4. Linux commands for operations such as redirection, pipes, filters, job control, changing ownership/permissions of files/links & directories.
5. Advanced LINUX Commands:curl, wget, ftp, ssh and grep.
6. Text Processing with Perl : Simple programs , connecting with database e.g., MYSQL
7. Running PHP: Simple applications like login forms after setting up a LAMP stack 8.Running Python : Set up the complete network interface using ifconfig command likesetting gateway, DNS, IP tables, etc.
8. Kernel Configuration, Compilation and installation : Download / Access the latest kernelsource code from kernel.org, compile the kernel and install it in the local system. Try to view the source code of the kernel.
9. Linux in the cloud, system & network management demos, configuration management,Linux on non-PC platforms, recompiling the kernel
10. Virtualisation environment (e.g., xen, kqemu, virtualbox or lguest) to test applications,new kernels and isolate applications.

SEMESTER III

COMPUTER NETWORKS			
Course Code:	**WCS621**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1.Understand how computer networks are organized with the concept of layered approach.
2.Implement a simple LAN with hubs, bridges and switches.
3.Analyze the contents in a given Data Link layer packet, based on the layer concept.
4.Figure out what classless addressing scheme is.
5.Describe how routing protocols work.

COURSE OUTCOMES
At the end of the course the students should be able to:
1.Analyse the requirements for a given organizational structure and select the most appropriate networking architecture and technologies.
2.Have a basic knowledge of the use of cryptography and network security.
3.Specify and identify deficiencies in existing protocols, and then go onto formulate new and better protocols.

4.Analyse, specify and design the topological and routing strategies for an IP based networking infrastructure
5.Have a working knowledge of datagram and internet socket programming

UNIT I INTRODUCTION AND PHYSICAL LAYER

Key concepts of computer network, transmission media, network devices, network topology, topologydesign issues, types of network: LAN, MAN, WAN, PAN, ISDN systems and ATM network, OSI-referencemodel, open system standards, characteristics of network, TCP/IP model, protocols and standards, encoding technique.

UNIT II SWITCHING AND DATA LINK LAYER

Circuit switching, packet switching, message switching, hybrid switching, and ATM switching, multiplexing techniques: TDMA, FDMA, WDMA, CDMA, data link layer: LLC &MAC level protocols anddesign issues, issues IEEE 802 LAN Standards, framing, CRC, error control, flow control, HDLC, ALOHA

and performance issues. Frames relay networks and performance parameters.

UNIT III NETWORK LAYER

Network layer design issues, overview of IPv4 and IPv6, addressing: class full and classless, static and dynamic, subnet and super net, auto configuration through DHCP, routing protocols: RIP, DVR,LSR, OSFP, BGP, congestion control algorithm, subnet concept, virtual LAN, ICMP, multicasting, mobile IP. **UNIT IV TRANSPORT LAYER**

Port addressing schemes, connectionless and connection oriented services: TCP and UDP, wireless TCP, Congestion control, queue management, NAT, PAT, socket format at transport level, socket interface and programming.

UNIT V APPLICATION LAYER

Client server architecture, domain name services, application services: HTTP, TELNET, RLOGIN, FTP, CBR, NFS, SMTP, POP, IMAP, MIME, voice and video over IP, social issues- privacy, freedom of speech,copy right.

Text Books:

1. S. Tanenbaum, Computer Networks, 4th edition, Prentice Hall, 2008
2. Forouzan, B.A., Data Communication and Networking, Tata McGraw-Hill.
3. W. Stallings, Data and Computer Communications, 8th edition, Prentice Hall, 2007
4. Douglus E. ComerTCP/IP Principles, Protocols and Architecture, Pearson Education

DATA ANALYTICS				
Course Code:	**WCS-623**	**Course Credits:**	**3**	
Course Category:	**CC12**	**Course (U / P)**	**P**	
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**	
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**	
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**	

COURSE OBJECTIVES
1. Understand the Big Data Platform and its Use cases
2. Provide an overview of Apache Hadoop
3. Provide HDFS Concepts and Interfacing with HDFS
4. Understand Map Reduce Jobs
5. Apply analytics on Structured, Unstructured Data. Exposure to Data Analytics with R.
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Identify Big Data and its Business Implications Access and Process Data on Distributed File System
2. List the components of Hadoop and Hadoop Eco-System
3. Manage Job Execution in Hadoop Environment
4. Develop Big Data Solutions using Hadoop Eco System
5. Analyze Infosphere Big Insights Big Data Recommendations

UNIT I INTRODUCTION

Introduction to big data : Introduction to Big Data Platform, Challenges of Conventional Systems, Intelligent data analysis, Nature of Data, Analytic Processes and Tools, Analysis vs Reporting.

UNIT II DATA STREAMS

Mining data streams : Introduction To Streams Concepts, Stream Data Model and Architecture, Stream Computing, Sampling Data in a Stream, Filtering Streams, Counting Distinct Elements in a Stream , Estimating Moments, Counting Oneness in a Window, Decaying Window, Real time Analytics Platform(RTAP) Applications - Case Studies - Real Time Sentiment Analysis- Stock Market Predictions.

UNIT III HADOOP

Hadoop: History of Hadoop, the Hadoop Distributed File System, Components of Hadoop Analysing the Data with Hadoop- Scaling Out- Hadoop Streaming- Design of HDFS-Java interfacesto HDFS Basics, Developing a Map Reduce Application-How Map Reduce Works, Anatomy of a Map Reduce Job run, Failures, Job Scheduling-Shuffle and Sort – Task execution, Map Reduce Types and Formats-Map Reduce Features,Hadoop environment.

UNIT IV DATA PROCESSING

Frameworks: Applications on Big Data Using Pig and Hive, Data processing operators in Pig, Hive services, HiveQL, Querying Data in Hive, fundamentals of HBase and ZooKeeper, IBM InfoSphere BigInsights and Streams.

UNIT V DATA ANALYTICS TECHNIQUE

Predictive Analytics- Simple linear regression, Multiple linear regression., Interpretation 5 of regression coefficients. Visualizations, Visual data analysis techniques, interaction techniques, Systems and applications.

Text Books:

1. Michael Berthold, David J. Hand, “Intelligent Data Analysis”, Springer, 2007.
2. Tom White “Hadoop: The Definitive Guide” Third Edition, O’reilly Media, 2012.
3. Chris Eaton, Dirk DeRoos, Tom Deutsch, George Lapis, Paul Zikopoulos, “Understanding Big Data:
4. Analytics for Enterprise Class Hadoop and Streaming Data”, McGrawHill Publishing, 2012.

COMPUTER NETWORKS LAB			
Course Code:	**WCS681**	**Course Credits:**	**2**
Course Category:	**CC-P**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Labs (Hrs/Week):	**02(3 hrs)**		
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1.Practical knowledge of working principles of various communication protocols.
2.Analyze structure and formats of TCP/IP layer protocols.
3.Understanding of networking fundamentals.
4.Considrate the learning process of Internet of Things applications planning.
5.Understanding of configuration of various end devices, server, routers and switches.

COURSE OUTCOMES
At the end of the course the students should be able to:
1.Understand the practical approach to network communication protocols.
2.Find Out the network layers, structure/format and role of each network layer.
3.Able to design and implement various network application such as data transmission between client and server, file transfer, real-time multimedia transmission.
4.Figure out the various Routing Protocols/Algorithms and Internetworking.
5.Learn to configure server.

List of Experiments:

1.Introduction to transmission media(CAT5, OFC, COAXIAL CABLE Wireless)
2.Introduces network interfaces(Wired and Wireless)
3.Configure and installing a Ethernet(10/100)
4.Performance evaluation of Ethernet(10/100)
5.Topology design(Ring, Bus)
6. Generation of data packet and measurement(CBR, VBR, Poison)

7.Implement the following:

a) Router configuration

b) Switch configuration

c) Server configuration

8. Congestion control of network and QoS of network

9. Protocols and the configuration

10. Security (WEP, WPA) and Qualnet.

ELECTIVES

Computer Vision			
Course Code: WCS534	**DS407**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**7U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1 Understand Camera model , Image formation in CV			
2 To understand different filter and features			
3 Understanding clustering , grouping and model fitting			
4 To understand register different objects			
5 Understand computer vision using machine learning			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 How the camera model works and different color shading			
2 Apply different filter and features			
3 To do application of clustering , grouping and model fitting			
4 Apply registering rigid and deformable objects			
5 Apply Computer Vision in Machine learning			

UNIT I Image Formation

Geometric Camera Model : Image formation - pinhole perspective ,Weak Perspective , Cameras with Lenses , Human eye , Intrinsic and Extrinsic Parameters , Geometric Camera calibration , Light and shading : Modelling pixel brightness , interference from shading , modelling Interreflection , Color : Human color perception , representing color , model of image color , inference from color

UNIT II Filters and Features

Linear filters and convolution , spatial frequency and fourier transformation , sampling and alias , Technique : Scale and Image pyramids , Computing the Image Gradient , Derivative of Gaussian Filters ,Representing the Image Gradient , Describing Neighborhoods with SIFT and HOG Features , Local Texture Representations Using Filters , Pooled Texture Representations by Discovering Textons

UNIT III Clustering , grouping and model fitting

Important application of clustering , Image segmentation by clustering pixels , segmentation clustering and graphs , The Hough Transform , Fitting lines and Planes , Robustness , Fitting using probabilistic models

UNIT IV Registration and Range Data

Registering rigid objects , registering deformable objects , active range sensors , range data segmentation , object recognition

UNIT V Classify using Machine Learning

Building good image feature using ML , Image classification datasets using ML , Face detection , detecting human , detecting boundaries using ML , Visual Hulls in python

Text Book

1. Forsyth , Ponce ," Computer Vision using A Modern Approch"

Reference Book

1. George C. Stockman , " Computer Vision"
2. Simon J. D. Prince, " Computer Vision: Models, Learning, and Inference"

DATA SCIENCE			
Course Code: WCS540	**DS314**	**Course Credits:**	**3**
Course Category:CC	**CC**	**Course (U / P)**	**U**
Course Year (U / P):U	**3U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03+ 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):30	**45+ 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1. Understand the data science basics			
2. Understand the basic mathematics for data science			
3. Understand the basic statistics for data science			
4. Understand the basic data processing techniques			
5. Understand the data visualization			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Understand what is data science			
2. Apply basic mathematics for data science			
3. Apply the basic statistics for data science			
4. Apply the basic data processing techniques			
5. Create and apply data visualization			

UNIT I Introduction to Data Science

What is data science?, venn diagram, flavorus of data, quantitative versus qualitative data, levels of data, obtain, explore, model and visualize the data

UNIT II Basic mathematics of data science

Vectors and matrices, graphs, logarithmic and exponents, basic linear algebra, probability, normal and poisson distribution, bayes theorem, random variables

UNIT III Basic statistics for data science

Obtain and sample data, Measures of center, variation, and relative standings, the empirical rule, point estimates, samplings distributions, hypothesis test,

UNIT IV Data Processing Techniques

Understanding the data processing, numpy operations, data cleaning, slicing, indexing, manipulating, and cleaning dataframes, pandas and csv, pandas and json, python relational database

UNIT V Data Visualization

Data visualiztion using matplotlib, style plots, line chart, bar plot, box plot, scatter plot, heatmap, 3-d plotting, time series plot or line plot with pandas, python geospatial data

Text Books:

[1]. Ozdemir Sinan ,Principles of Data Science

Reference Books:

[2]. Sanjiv Ranjan Das, Data Science: Theories, Models, Algorithms, and Analytics

SOFTWARE QUALITY ASSURANCE			
Course Code:	WCS544	**Course Credits:**	**3**
Course Category:	**E2**	**Course (U / P)**	**P**
Course Year (U / P):	**2P**	**Course Semester (U / P):**	**3P**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 consists of a set of reporting and auditing functions.
2It ensures management of data which is important for product quality.
3It also ensures that software which is developed, does it meet and compiles with standard quality assurance.
4It ensures that end result or product meets and satisfies user and business requirements.
5It simply finds or identify defects or bugs, and reduces effect of these defects.

COURSE OUTCOMES
At the end of the course the students should be able to understand:
1what SQA is and why SQA is not SQC(testing).
2 Reasons for SQA failures and factors critical to success of SQA in IS development.
3 Proactive methods for more effectively reviewing requirements and designs.
4 To produce efficient, reliable, robust and cost-effective software solutions.
5 How to measure system quality and SQA/Testing .

UNIT I SOFTWARE QUALITY AND ENGINEERING

Quality concepts and productivity relationship, software quality factors, software quality costs, Total Quality Management (TQM), continuous improvement cycle: Plan, Do, Check and Act (PDCA), quality policy, cost of quality, quality engineering, quality planning: goal setting and strategy formation, assessment and improvement.

UNIT II SOFTWARE QUALITY ASSURANCE (SQA)

Components of SQA, classification, defect detection, defect prevention, defect reduction, defect containment, QA activities in software processes, verification and validation, software review, inspection, formal verification, statistical software quality approach.

UNIT III COMPONENTS MEASUREMENT WITH REFERENCE TO SQA

Metrics, product quality metrics, process quality metrics, metrics for software maintenance, quality tools for quality control, test management and organizational structures, Capability Maturity Model(CMM), Capability Maturity Model Integration (CMMI), ISO 9000, quality and quality management metrics, Deming's Principle, SQA team formation

UNIT IV QUALITY MANAGEMENT MODEL

Integrating quality activities in project life cycle, reviews, software testing, strategies and implementation, Computer-Aided Software Engineering (CASE) tools, The Rayleigh model framework, code integration pattern, Problem Tracking Report (PTR), reliability growth model,Service Quality, Kano Model, Customer retention, continuous process improvement, Juran's Trilogy,TQM principles, Kaizen Technique, Statistical Quality Assurance, Mc call quality factors

UNIT V SOFTWARE QUALITY ASSURANCE BEYOND TESTING

Defect prevention and process improvement, root cause analysis for defect prevention, software inspection, inspection related activities, fault tolerance and failure containment, comparing quality assurance techniques and activities.

REFERENCE BOOKS:

1. Metrics and Models in Software Quality Engineering, Stephan H. Kan, Pearson Education,2007.
2. An Integrated Approach to Software Engineering, Pankej Jalote, Narosa PublishingHouse, New Delhi 1997.
3. Making Sense of Software Quality Assurance, Raghav J. Nandyal, Tata McGRAW Hill,2007.
4. Software Quality Assurance: A Practitioner Approach, Kaman Malik, Praveen Chaudhary,Tata McGRAW Hill, 2008.

Embedded System Design			
Course Code:	WCS625	**Course Credits:**	**3**
Course Category:	**E**	**Course (U / P)**	**U / P**
Course Year (U / P):	**2P**	**Course Semester (U / P):**	**3P**
No. of Lab (Hrs/Week) / Total No. of Lab	**03 / 10**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
6. Building Blocks of Embedded System			
7. Various Embedded Development Strategies			
8. Bus Communication in processors, Input/output interfacing.			
9. Various processor scheduling algorithms.			
10.Basics of Real time operating system and example tutorials to discuss on one real time operating system too.			
COURSE OUTCOME			
At the end of the course the students should be able to:			
1. understand and analyse Embedded systems.			
2. study about the bus Communication in processors.			
3. operate various Embedded Development Strategies			
4. acquire knowledge on various processor scheduling algorithms.			
5. suggest an embedded system for a given application			

UNIT I INTRODUCTION TO EMBEDDED SYSTEMS

Introduction to Embedded Systems –Structural units in Embedded processor, selection of

processor & memory devices- DMA – Memory management methods- Timer and Counting devices, Watchdog Timer, Real Time Clock, In circuit emulator, Target Hardware Debugging**.**

UNIT II EMBEDDED NETWORKING

Embedded Networking: Introduction, I/O Device Ports & Buses– Serial Bus communication protocols RS232 standard – RS422 – RS 485 - CAN Bus -Serial Peripheral Interface (SPI) – Inter Integrated Circuits (I2C) –need for device drivers.

UNIT III EMBEDDED FIRMWARE DEVELOPMENT ENVIRONMENT

Embedded Product Development Life Cycle- objectives, different phases of EDLC, Modelling of EDLC; issues in Hardware-software Co-design, Data Flow Graph, state machine model, STUDENTSFOCUS.COM Sequential Program Model, concurrent Model, object-oriented Model.

UNIT IV RTOS BASED EMBEDDED SYSTEM DESIGN

Introduction to basic concepts of RTOS- Task, process & threads, interrupt routines in RTOS, Multiprocessing and Multitasking, Pre-emptive and non-pre-emptive scheduling, Task communication shared memory, message passing-, Inter process Communication – synchronization between processes-semaphores, Mailbox, pipes, priority inversion, priority inheritance.

UNIT V EMBEDDED SYSTEM APPLICATION AND DEVELOPMENT

Case Study of Washing Machine- Automotive Application- Smart card System Application- ATM machine –Digital camera

TEXTBOOKS:

1. Peckol, "Embedded system Design", John Wiley & Sons,2010
2. Lyla B Das," Embedded Systems-An Integrated Approach", Pearson, 2013
3. Shibu. K.V, "Introduction to Embedded Systems", 2e, Mc graw Hill, 2017.

REFERENCES

1. Raj Kamal, 'Embedded System-Architecture, Programming, Design', Mc Graw Hill, 2013.
2. C.R.Sarma, "Embedded Systems Engineering", University Press (India) Pvt. Ltd, 2013.
3. Tammy Noergaard, "Embedded Systems Architecture", Elsevier, 2006.
4. Han-Way Huang, "Embedded system Design Using C8051", Cengage Learning, 2009.
5. Rajib Mall "Real-Time systems Theory and Practice" Pearson Education, 2007.

SOFT COMPUTING TECHNIQUES		
Course Code:	**CS403**	**Course Credits:**
Course Category:	**CC**	**Course (U / P)**
Course Year (U / P):	**4U**	**Course Semester (U / P):**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**

COURSE OBJECTIVES
1.Primary objective of this course is to provide an introduction to the basic principles, techniques, and applications of soft computing.
2. Understanding of the basic areas of Soft Computing including Artificial Neural Networks, Fuzzy Logic and Genetic Algorithms.
3.Provide the mathematical background for carrying out the optimization associated with neural network learning.
4.Aim of this course is to develop some familiarity with current research problems and research methods in Soft Computing by working on a research or design project.
5. Genetic algorithms, its applications and advances.
COURSE OUTCOMES
At the end of the course the students should be able to:
1.Apply basics of Fuzzy logic and neural networks..
2.Discuss the ideas of fuzzy sets, fuzzy logic and use of heuristics based on human
3. Describe with genetic algorithms and other random search procedures useful while seeking global optimum in self-learning situations
4. Develop some familiarity with current research problems and research methods in Soft Computing Techniques

5. experience Relate with neural networks that can learn from available examples and generalize to form appropriate rules for inference systems

UNIT I INTRODUCTION

Introduction to Soft Computing; Definition, requirement, necessity and adequacy; various dialects of soft computing – Evolutionary Algorithms, Fuzzy Sets and Fuzzy Logic, Artificial Neural Networks - theirsuitability in Searching, optimization, decision matching and pattern related problems; potential areasof applications.

UNIT II FUZZY SETS AND FUZZY LOGIC

Introduction to fuzzy sets and fuzzy logic; difference between classical and fuzzy sets; chance vs fuzziness; limitations of fuzzy systems; typical shapes of membership functions and their usage; operations on fuzzy sets: compliment, intersection, union; combinations on operations, aggregationoperation.

UNIT III FUZZY RELATIONS AND FUZZY SYSTEMS

Cartesian Product; Classical Relations and Fuzzy Relations; Cardinality, operations and properties of crisp and fuzzy relations; Composition of operations, Fuzzy cartesian product; The linguistic variables, Reasoning in fuzzy logic, Fuzzification and defuzzification; Mamdani and Sugano Fuzzy Inference Systems.

UNIT IV NEURAL NETWORK

Overview of biological neurons; McCulloch-Pitts model, Rosenblatt's Perceptron model, difference, capabilities and limitations; Model of generic computational neuron; Basic activation functions; Basic Learning laws of neurons; Single layer and multilayer architectures; Feedforward and feedback networks.

UNIT V LEARNING FUNDAMENTALS

Learning paradigms, supervised and unsupervised learning, reinforced learning; back propagation algorithm; Radial basis neurons, Generalized Regression Neural network, Probabilistic Neural Networks; Competitive learning; Self Organizing Features Map, Hopfield networks, associative memories, applications of artificial neural networks. Elasticity vs plasticity dilemma, preprocessing, post processing, early stopping.

UNIT VI EVOLUTIONARY ALGORITHMS

Problems suitable and not suitable for applying evolutionary algorithms; Various dialects of evolutionary Algorithms; Terminology of Genetic Algorithms; Canonical Genetic Algorithm; Common representations and related reproduction operators; premature convergence, schema theorem, minimal deceptive problem and Royal Road function; fitness function, Roulette wheel selection, Rank selection, Tournament Selection; termination criteria, survivor selection, population models; parallelimplementations.

Text Books:

1. Artificial Neural Networks: An introduction to ANN Theory and Practice, Peteus J.Braspenning,PHI publication, 2005.
2. Fuzzy Logic: A spectrum of Theoretical and Practical issues, Paul P. Wang, pearson publication 2004.
3. An Introduction to Genetic Algorithms, Milanie Mitchell, MIT Press 1998.
4. A Genetic Algorithm Tutorial, Darrell Whitley.
5. Fuzzy Sets, Fuzzy logic, and Fuzzy Systems: Selected Papers- Lotfi Asker Zadeh, George J.Kilr, Bo yuan, 2005.
6. Foundations of Fuzzy logic and Soft Computing: 12th International Fuzzy conferenceproceeding, 2005.
7. Neural Networks Theory, Particia Melin, Oxford University press, 2003
8. Neural Networks Theory and Application, Oscar Castillo, Wiley Eastern publication
9. Genetic Algorithms in Search, Optimization and Machine Learning, David E Goldberg,Eddison-Wesley, 1988.

APPLIED MACHINE LEARNING			
Course Code:	WCS627	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**7U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Explain Machine Learning concepts, classifications of Machine Learning and write simple programs using python.
2.Describe Supervised Learning concepts.
3.Describe unsupervised learning concepts and dimensionality reduction techniques
4.Discuss simple Machine Learning applications in a range of real-world applications using Python programming
5.To develop skills of using recent machine learning software for solving practical problems.

COURSE OUTCOMES
At the end of the course the students should be able to:
1.Recognize the characteristics of machine learning that make it useful to real-world problems.
2.Characterize machine learning algorithms as supervised, semi-supervised, and unsupervised.
3.Effectively use machine learning toolboxes.
4.Understand the concept behind neural networks for learning non-linear functions.
5.Figure out the algorithms for learning Bayesian networks

Unit 1: Introduction – Well defined learning problems, Designing a Learning System, Issues in Machine Learning; THE CONCEPT LEARNING TASK - General-to-specific ordering of
hypotheses, Find-S, List then eliminate algorithm, Candidate elimination algorithm, Inductive bias
Unit 2: Decision Tree Learning - Decision tree learning algorithm-Inductive bias- Issues in Decision tree learning; ARTIFICIAL NEURAL NETWORKS – Perceptrons, Gradient descent and the Delta rule, Adaline,Multilayer networks, Derivation of backpropagation rule Backpropagation Algorithm Convergence, Generalization

Unit 3: Evaluating Hypotheses: Estimating Hypotheses Accuracy, Basics of sampling Theory,Comparing Learning Algorithms;
Bayesian Learning: Bayes theorem, Concept learning, Bayes Optimal Classifier, Naïve Bayes classifier,Bayesian belief networks, EM algorithm;
Unit 4: Computational Learning Theory: Sample Complexity for Finite Hypothesis spaces, SampleComplexity for Infinite Hypothesis spaces, The Mistake Bound Model of Learning;
INSTANCE-BASED LEARNING – k-Nearest Neighbour Learning, Locally Weighted Regression,Radial basis function networks, Case-based learning
Unit 5: Genetic Algorithms: an illustrative example, Hypothesis space search, Genetic Programming, Models of Evolution and Learning; Learning first order rules-sequential covering algorithms- General to specific beam search-FOIL; REINFORCEMENT LEARNING - The Learning Task, Q
Learning.

Text Books:

1. Tom M. Mitchell, ―Machine Learning, McGraw-Hill Education (India) Private

Limited, 2013.
2. Ethem Alpaydin, ―Introduction to Machine Learni

Machine Learning), The MITPress 2004.
3. Stephen Marsland, ―Machine Learning: An Algorithmic Perspective, CRC Press, 2009.Bishop, C., Pattern Recognition and Machine Learning. Berlin: Springer-Verlag.

AI ENABLED CYBER SECURITY			
Course Code: WCS633		**Course Credits:**	**3**
Course Category:		**Course (U / P)**	**P**
Course Year (U / P):	**2P**	**Course Semester (U / P):**	**3P**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1. To prepare students with the technical knowledge and skills needed to protect and defend computer systems and networks.			
To develop graduates that can plan, implement, and monitor cyber security mechanisms through AI to ensure the protection of information technology assets.			
3. Analyze and resolve security issues in networks and computer systems using AI and Machine Learning models.			
Evaluate and communicate the human role in security systems with an emphasis on ethics, social engineering vulnerabilities and training			
Infuse AI capabilities when building smart defensive mechanisms			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Follow a structured model in Security Systems using the concepts in AI and ML			
2. Plan, implement and audit security in a networked, multi-platform and cross platform environment			
Protect data and respond to threats through various AI and Machine learning models.			
4. Ability to apply AI and machine learning models in cyber security issues.			
. Detect attack methodology and combat hackers from intrusion or other suspicious attempts at connection to gain unauthorized access to a computer and its resources			

UNIT-I Introduction to AI for Cyber security:
Applying AI in cyber security, The evolution from expert systems to AI, The different forms of automated learning, characteristics of algorithm training and optimization, AI in Cyber Security and Security Framework: Artificial Intelligence in Cyber Security, Challenges and Promises, Security Threats of Artificial Intelligence

UNIT-II Role of AI in Cyber Security:
Arsenal Classification, Regression, Dimensionality reduction, Clustering, Speech recognition, Video anomaly detection, Natural language processing, Large-scale image processing, Social media analysis

UNIT-III Detecting Cyber security Threats with AI:
Detect spam with Perceptrons, Image spam detection with support vector machines (SVMs), Phishing detection with logistic regression and decision trees, Spam detection with Naive Bayes, Spam detection adopting NLP.

UNIT-IV Protecting Sensitive Information and Assets:
Authentication abuse prevention, Account reputation scoring, User authentication with

keystroke recognition, Biometric authentication with facial recognition.

UNIT-V Fraud Prevention with AI Solutions:

AI and ML algorithms for fraud detection, How bagging and boosting techniques can improve an algorithm's effectiveness, Model Stealing & Watermarking, Network Traffic Analysis, Malware Analysis.

Reference Books:-

1. Leslie F. Sikos, AI in Cybersecurity, Springer Cham, Springer Nature Switzerland AG 2019
2. Tom Mitchell. Machine Learning. McGraw Hill, 1997.
3. Gupta, Brij B., and Quan Z. Sheng, eds. Machine learning for computer and cyber security: principle, algorithms, and practices. CRC Press, 2019.
4. Artificial Intelligence and Data Mining Approaches in Security Frameworks Editor(s): Neeraj Bhargava, Ritu Bhargava, Pramod Singh Rathore, Rashmi Agrawal, 2021.
5. Tsai, Jeffrey JP, and S. Yu Philip, eds. Machine learning in cyber trust: security, privacy, and reliability. Springer Science & Business Media, 2009.
6. Machine Learning: A Probabilistic Perspective, Kevin P Murphy, MIT Press. 6. Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer 2006
7. Dr. Nilakshi Jain, Artificial Intelligence, As per AICTE: Making a System Intelligent, Wiley Publication (2019).
8. Alessandro Parisi, Hands-On Artificial Intelligence for Cybersecurity: Implement smart AI systems for preventing cyber attacks and detecting threats and network anomalies, Packt Publication (2019).

INTERNET OF THINGS			
CourseCode: WCS637	**CS 401**	**CourseCredits:**	**3**
CourseCategory:CC	**CC**	**Course(U /P)**	**4U**
No.ofLectures+Tutorials(Hrs/Week):	**03 + 00**	**MidSem. ExamHours:**	**1**
TotalNo. ofLectures(L+T):30	**45**	**EndSem.ExamHours:**	**3**
COURSE OBJECTIVES			
1. Explore to the interconnection and integration of the physical world in IoT.			
2.Learning of networking concepts in IoT environment.			
3.Understanding of various wireless network, topologies, IoT protocols.			
4.Understad the importance of security issues in IoT.			
5.Implementation of IoT in real life with learning of tools like MATLAB.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Figure out about all concepts of Internet of Things.			
2 understand building blocks of Internet of Things and its characteristics.			
3 learn application protocols for IoT.			
4 Able to understand the application areas of IoT.			
5 Realize the revolution of Internet in Mobile Devices, Cloud & Sensor Networks.			

UNIT I INTRODUCTION TO IOT

Genesis of IoT, IoT and Digitization, IoT Impact, Convergence of IT and OT, IoT Challenges, Drivers Behind New Network Architectures: Scale, Security, Constrained Devices and Networks, Data, Legacy Device Support .

UNIT II IOT NETWORK ARCHITECTURE AND DESIGN

Comparing IoT Architectures : The one M2M IoT Standardized Architecture, The IoT World Forum (IoTWF) Standardized Architecture, Additional IoT Reference Models, A Simplified IoT Architecture, The Core IoT Functional Stack- Layer 1: Things: Sensors and Actuators Layer, Layer 2: Communications Network Layer, Layer 3: Applications and Analytics Layer, IoT Data Management and Compute Stack :Fog Computing , Edge Computing, The Hierarchy of Edge, Fog, and Cloud.

UNIT III NETWORK AND APPLICATION PROTOCOLS FOR IOT

Wireless Communication Technologies: ZigBee, ESP8266, Introduction to sensors and modules - concept, layout, working, applications, Introduction of IoT Development Boards-Node MCU, Arduino, IoT Access Technologies 107IEEE 802.15.4, IEEE 802.15.4g and 802.15.4e, IEEE 1901.2a, IEEE 802.11ah, LoRaWAN, Constrained Devices, Constrained-Node Networks, Optimizing IP for IoT :From 6LoWPAN to 6Lo, Header Compression, Fragmentation, Mesh Addressing, Mesh-Under Versus Mesh-Over Routing, Authentication and Encryption on Constrained Nodes , Application Protocols for IoT: CoAP, Message Queuing Telemetry Transport (MQTT) .

UNIT IV DATA ANALYTICS AND SECURITY OF IOT

An Introduction to Data Analytics for IoT, Structured Versus Unstructured Data, Data in Motion Versus Data at Rest, IoT Data Analytics Overview, IoT Data Analytics Challenges, Machine Learning : Machine Learning Overview Supervised Learning, Unsupervised Learning, Neural Networks, Securing IoT : Common Challenges in IoT Security, Device Insecurity, Network Characteristics Impacting Security, Security Priorities: Integrity, Availability, and Confidentiality, Formal Risk Analysis Structures: IAS OCTAVE, Top Vulnerabilities of Iot.

UNIT V. IMPLEMENTING IoT IN REAL LIFE

Interfacing sensors with development boards, communication modules with sensors, communication modules with development boards, MATLAB and Arduino Interfacing, Hands-on in IoT - various real life projects involving different boards, sensors, modules and communication technologies .

Text Books :

1. IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things by Rob Barton, Gonzalo Salgueiro, David Hanes
2. Vijay Madisetti and Arshdeep Bahga, “Internet of Things (A Hands-on-Approach)”, 1stEdition, VPT, 2014.
3. Francis daCosta, “Rethinking the Internet of Things: A Scalable Approach to Connecting Everything”, 1st Edition, Apress Publications, 2013

BIG DATA PLATFORMS AND ANALYTICS			
CourseCode:	WCS635	**CourseCredits:**	**3**
CourseCategory:CC	**CC**	**Course(U/P)**	**U**
CourseYear(U/P):U	**3U**	**CourseSemester(U/P):**	**6U**
No.ofLectures+Tutorials(Hrs/Week):	**03 + 00**	**MidSem.ExamHours:**	**1**
TotalNo.ofLectures(L+ T):30	**45 + 00**	**EndSem.ExamHours:**	**3**
COURSEOBJECTIVES			
1. Understand the Big Data Platform and its Use cases.			
2. Provide an overview of Apache Hadoop.			
3. Provide HDFS Concepts and Interfacing with HDFS.			
4. Provide hands on Hodoop Eco System.			
5. Apply analytics on Structured, Unstructured Data.			

COURSEOUTCOMES
At the end of the course the students should be able to:
1. Identify Big Data and its Business Implications.
2. List the components of Hadoop and Hadoop Eco-System
3. Develop Big Data Solutions using Hadoop Eco System
4. Manage Job Execution in Hadoop Environment.
5. Apply Machine Learning Techniques to big data.

UNIT I: INTRODUCTION TO BIG DATA AND HADOOP

Types of Digital Data, Introduction to Big Data, Big Data Analytics, History of Hadoop, Apache

Hadoop, Analyzing Data with Unix tools, Analyzing Data with Hadoop, Hadoop Streaming, Hadoop Echo System, IBM Big Data Strategy, Introduction to Infosphere BigInsights and Big Sheets.

UNIT II: HDFS (Hadoop Distributed File System)

The Design of HDFS, HDFS Concepts, Command Line Interface, Hadoop file system interfaces, Data flow, Data Ingest with Flume and Scoop and Hadoop archives, Hadoop I/O: Compression,

Serialization, Avro and File-Based Data structures.

UNIT III: Machine Learning in Big data platforms

Introduction and Concepts: Ridge Regression; Lasso Regression; and K Nearest Neighbors, Regression and Classification, Supervised Learning with Regression and Classification Techniques, Bias-Variance, Dichotomy, Linear and Quadratic Discriminant Analysis, Classification and Regression, Trees, Ensemble Methods: Random Forest, Neural Networks, Deep Learning.

UNIT IV: Classification and Partitioning Methods:

Decision Trees, Attribute Selection Measures and Tree Pruning, Bayesian and Rule-based Classification, Model Evaluation and Selection, Cross-Validation, Classification Accuracy, Classification by Backpropagation and Support Vector Machine, k-means Hierarchical Methods and Hierarchical Clustering Using Feature Trees, Probabilistic Hierarchical Clustering, Evaluation of Clustering Methods.

UNIT V: Hadoop Eco System

Pig : Introduction to PIG, Execution Modes of Pig, Comparison of Pig with Databases, Grunt, Pig Latin, User Defined Functions, Data Processing operators.

Hive : Hive Shell, Hive Services, Hive Metastore, Comparison with Traditional Databases, HiveQL, Tables, **Querying Data and User Defined Functions.**

Hbase : HBasics, Concepts, Clients, Example, Hbase Versus RDBMS.**Big SQL :** Introduction

Text Books

1. Tom White " Hadoop: The Definitive Guide" Third Edit on, O'reily Media, 2012.
2. Seema Acharya, Subhasini Chellappan, "Big Data Analytics" Wiley 2015.
3. Architecting Modern Data Platforms: A Guide to Enterprise Hadoop at Scale Author:
4. Big Data Analytics, Introduction to Hadoop, Spark, and Machine-Learning, Raj kamal, preeti saxena.

Reference Books

1. Jay Liebowitz, "Big Data and Business Analytics" Auerbach Publications, CRC press

(2013)
2. Tom Plunkett, Mark Hornick, “Using R to Unlock the Value of Big Data: Big Data Analytics with Oracle R Enterprise and Oracle R Connector for Hadoop”, McGraw-Hill/Osborne Media (2013), Oracle press.