# FIFTH SEMESTER CSE 2015-16

**COURSE CODE: CS30108** **REF NO: To be filled by CD office**

## Theory of Computation

**Credits:** 3 **Teaching Scheme: -** Theory 3 Hrs/Week

**Prerequisites**:

1. Computer Programming
2. Discrete mathematics

**Objectives:**

1. To introduce mathematical and computational principles of programming languages.
2. To study and design automata for different class of problems.

**Course Details:**

**Unit 1** **(6 Hrs)**

**Title- Introduction to Automata Theory**

**U1.1**

**Concepts to Automata Theory:** Alphabets, Strings, Languages and Grammar Deterministic finite Automata (DFA) and Nondeterministic finite Automata (NFA), NFA with epsilon transition, Equivalence of NFA and DFA, Minimization of Automata, Conversion of NFA with epsilon to DFA Equivalence, Chomsky Classification.

**U1.2**

**Self Study:** Concepts to Automata with outputs Moore and Mealy Machine.

**Unit II** **(6 Hrs)**

**U2.1**

**Regular Expression and Languages:** Definition, Identities, Arden's theorem, Kleen's Theorem, Regular expression to DFA, DFA to Regular expression, Non Regular Languages, Pumping Lemma for regular Languages, Closure properties of Regular Languages.

**U2.2**

**Self Study:** Conversion from Automata to Grammar and vice versa

**Unit III** **(6 Hrs)**

**Title- Context Free Grammars (CFG) and Push Down Automata (PDA)**

**U3.1**

**Context Free Grammars:** Definition of CFG, Parse trees, Ambiguity in Grammar, Ambiguous and Unambiguous CFG, Inherent ambiguity, Simplification of CFGs, Normal forms for CFGs: CNF and GNF, Closure properties of CFLs, Cook Younger Kasami(CYK)

algorithm and Chomsky Hierarchy. Pumping Lemma for CFLs. Definition and Acceptability of PDA, Language to PDA and CFG to PDA

**U3.2**
**Self Study:** Ogden's lemma and Parikh's theorem, Early's algorithm.

**Unit IV** **(6 Hrs)**
**Title- Turing Machines, Un-decidability & Computable function**

**U4.1**
**Turing Machines:** Definition and representation of TM, Language acceptance by TM. Variants of TM, Universal Turing Machines, Godel numbering, Church-Turing Thesis, Recursive and Recursively Enumerable languages. Halting problem, Post Correspondence Problem, Introduction to countable and uncountable sets, Recursive function, Primitive recursive function, Ackerman's function.

**U4.2**
**Self Study:** Linear Bounded Automata and Contest sensitive language and Modified PCP

**Unit V** **(6 Hrs)**
**Title- Time Complexity**
**U5.1**
Class P, class NP, NP-Completeness and Reducibility
**U5.2**
**Self Study:** Proofs on Class P, NP and NP-C.

**Text Books**
T1. "Introduction to Automata Theory, Languages and Computation", Hopcroft J, Motwani R, Ullman ,Addison-Wesley, ISBN 81-7808-347-7, Second Edition .
T2. "Introduction to Theory of Computation", Michael Sipser, Course Technology, ISBN-10: 053494728X, Third Edition.

**Reference Books**

R1. Introduction to Formal Languages, Automata Theory and Computation: K. Kirthivasan, Rama R, Pearson Education.
R2. Theory of computer Science (Automata Language & computations) K.L. Mishra N. Chandrashekhar, PHI.
R3. Introduction to Languages and the Theory of Computation, J. Martin, Tata McGraw-Hill, ISBN 0-07-049939-x, Third edition, 2003.
R4. Elements of The theory of Computation, H.R.Lewis, C.H.Papadimitriou, Pearson Education, ISBN 81-7808-487-2, Second Edition.
R5. Introduction to Languages and the Theory of Computation: Martin, Tata McGraw Hill, 3rd Edition.

R6. Formal Languages and Automata Theory, C.K.Nagapal, Oxford University Press, First Edition, 2011.

**Course Outcomes:**
**Upon completion of the course, graduates will be able –**

**CO1:** The ability to prove results using proof by induction, proof by contradiction, proof by construction.
**CO2:** Ability to describe various automata theoretic models for recognizing formal languages and transform regular expressions and grammars.
**CO3:** Distinguish different computing languages and classify their respective types.
**CO4:** Able to construct pushdown automata and the equivalent context free grammars and prove the equivalence of the languages described by pushdown automata and context free grammars.
**CO5:** Able to design Turing Machine and prove the equivalence of the languages described by Turing machines and Post machines.
**CO6:** Analyze algorithmic complexity, computability and solvability of problems.

**COURSE CODE: CS30108** **REF NO: To be filled by CD office**

# Theory of Computation Tutorials

**Credits:** 01 **Teaching Scheme:** Tutorial 01 Hrs/Week

**Prerequisites**:

3. Computer Programming
4. Discrete mathematics

**Objectives:**

3. To introduce mathematical and computational principles of programming langauage.
4. To study and design automata for different class of problems.

**List of Tutorials**

| | |
|---|---|
| **Tutorial No. 1:** | Problems on NFA to DFA conversion. |
| **Tutorial No. 2:** | Numerical based on minimization and equivalence of Automata. |
| **Tutorial No. 3:** | Exercises on conversion of Regular expression to DFA and vice versa |
| **Tutorial No. 4:** | Problems on proof of Closure properties of Regular Languages. |
| **Tutorial No. 5:** | Problems on proof of whether a grammar is regular or not by using pumping lemma. |
| **Tutorial No. 6:** | Problems on checking of Ambiguity of Grammar and Simplification of CFGs, |
| **Tutorial No. 7:** | Problems on Normal forms of CFGs: CNF and GNF. |
| **Tutorial No. 8:** | Problems based on CFG to PDA construction. |
| **Tutorial No. 9:** | Problems on Turing machine. |
| **Tutorial No. 10:** | Problems on Ackermann's function, Gödel Numbering, Post Correspondence Problem |

**Text Books**

T1. "Introduction to Automata Theory, Languages and Computation", Hopcroft J, Motwani R, Ullman ,Addison-Wesley, ISBN 81-7808-347-7, Second Edition .
T2. "Introduction to Theory of Computation", Michael Sipser, Course Technology, ISBN-10: 053494728X, Third Edition.

**Reference Books**

R1. Introduction to Formal Languages, Automata Theory and Computation: K. Kirthivasan, Rama R, Pearson Education.
R2. Theory of computer Science (Automata Language & computations) K.L. Mishra N. Chandrashekhar, PHI.
R3. Introduction to Languages and the Theory of Computation, J. Martin, Tata McGraw-Hill, ISBN 0-07-049939-x, Third edition, 2003.
R4. Elements of The theory of Computation, H.R.Lewis, C.H.Papadimitriou, Pearson Education, ISBN 81-7808-487-2, Second Edition.
R5. Introduction to Languages and the Theory of Computation: Martin, Tata McGraw Hill, 3rd Edition.
R6. Formal Languages and Automata Theory, C.K.Nagapal, Oxford University Press, First Edition, 2011.

**Course Outcomes:**
**Upon completion of the course, graduates will be able –**

**CO1:** The ability to prove results using proof by induction, proof by contradiction, proof by construction.
**CO2:** Ability to describe various automata theoretic models for recognizing formal languages and transform regular expressions and grammars.
**CO3:** Distinguish different computing languages and classify their respective types.
**CO4:** Able to construct pushdown automata and the equivalent context free grammars and prove the equivalence of the languages described by pushdown automata and context free grammars.
**CO5:** Able to design Turing Machine and prove the equivalence of the languages described by Turing machines and Post machines.
**CO6:** Analyze algorithmic complexity, computability and solvability of problems.

**COURSE CODE: CS30106** **REF NO: To be filled by CD office**

# Computer Organization

**Credits:**3 **Teaching Scheme: -** Theory 3Hrs/Week

**Prerequisites**:

1. Digital Electronic Circuit.
2. Computer Fundamentals.
3. Computer Programming

**Objectives:**

**1.** To get idea about different hardware used in Computer system and their interconnections.
**2.** To get knowledge on how hardware and software are interlinked to peocess instruction.
**3.** To get knowledge on different types of memory, their utility and mapping.
**4.** To get idea about how control unit control the execution of instructions.
**5.** To realize how arithmetical operations are performed inside ALU using various registers.
**6.** To get idea on storage and I/O transfer.

**Course Details:**

**Unit1: Structure of a Computer System (7 Hrs)**

**U1.1:** Organization & Architecture, Structure & Function, Brief History of computers, Von Neumann Architecture, Bus Structure, Elements of Bus Design, CPU Architecture, Register Organization, Instruction types, Types of operands, Instruction formats, addressing modes and address translation. Instruction cycles, RISC and CISC Processors, Pipelining, Pipelining Hazards, Superscalar Processors. Performance consideration.

**U1.2Self Study:** Single core vs Multicore processor organization, PCI Bus, Basic multiprocessor architecture.

**Unit2: Memory Organization (7 Hrs)**

**U2.1.** Hierarchical memory system, Characteristics, Size, Access time, Read Cycle time and address space. Main Memory Organization, types of memory, memory chip design, Cache memory Organization: Address mapping, Cache Coherence, interleaved memories. Virtual Memory: Paging.

**U2.2. Self Study:** Internal structure of RAM & ROM, Multi Level Cache.

**Unit3: Processor and Control Unit (7 Hrs)**

**U3.1.** Fundamental Concepts: Single Bus CPU organization, Register transfers,Performing an arithmetic/ logic operations, fetching a word from memory, storing a word in memory, Execution of a complete instruction. Micro- operations, Hardwired Control and Micro-programmed Control CU.

**U3.2. Self Study:** Multiple- bus organization. Applications of micro programming.

**Unit4: Computer Arithmetic (7 Hrs)**

**U4.1.** Integer Representation: Fixed point & Signed numbers. Integer Arithmetic: 2's Complement arithmetic, addition, subtraction, multiplication, Booth's Algorithm, Division with restoring algorithm and non restoring algorithm, Floating point representation of number: IEEE754 Standards for Floating point representations (Single Precision Format & Double Precision Format).

**U4.1. Self Study:** ALU Design.

**Unit5: Secondary Storage and I/O (7 Hrs)**

**U5.1.** Magnetic Disk, Read Write mechanism of HDD, Access time, Latency time, Optical memory, CDROM. Basic fundamentals of Input Output organization –Peripheral devices, Asynchronous Data Transfer, Mode of Data Transfer, Direct Memory Access (DMA), Bus Arbitration.

**U5.2. Self Study:** Input Output Processor (IOP), Interrupts, Vectored interrupts, Interrupt handling, USB, PCI.

**Text Books:**

**T1.** "Computer Organization", C. Hamacher, V. Zvonko, S. Zaky, Tata McGraw Hill Publication,ISBN 007-120411-, 5th Edition.
**T2.** " Computer System Architecture", M. Morris Mano, Pearson Education, ISBN-978-81-317-0070-9, 3rd Edition.

**Reference Books:**

**R1.**"Computer Architecture and Parallel Processing", Hwang and Briggs, Tata McGraw Hill Publication , ISBN 13: 9780070315563.
**R2.** "Structured Computer Organization", A. Tanenbaum, Prentice Hall Publication, ISBN 81–203–1553–7, 4th Edition.

**Course Outcomes:**
**Upon completion of the course, graduates will be able –**

**CO1:** Identify and analyze the basic structures of a computer hardware units, connectivity and software.

**CO2:** Design the basic structure of machine instruction and programs, memory location.

**CO3:** Analyze different memory in the hierarchy, their mapping and their performance.

**CO4:** Analyze internal details of a processor, how instructions are executed using different hardware units, and how control unit controls all hardware components.

**CO5:** Study the design of ALU for arithmetic operations and useof registers.

**CO6:** Analyze the organization of secondary storage and how all the I/O devices communicate with CPU and transfer data.

**COURSE CODE: CS30106** **REF NO: To be filled by CD office**

# Computer Organization Tutorial

**Credits: 1** **Teaching Scheme: -** Tutorial 1Hr/Week

**Prerequisites**:

1. Digital Electronic Circuit.
2. Computer Fundamentals.
3. Computer Programming

**Objectives:**

1. To practice assembly language programming and measure the influence of implementing pipeline on a processor.
2. Practice different numerical on memories used in a computer and also to realize how address mapping takes place between different memories.
3. To get idea on micro instructions and CU design.
4. Practice how arithmetical operations are performed inside ALU using various registers.
5. Study on different modern multicore processors.

**Tutorial No.1:** Assembly language programming

**Tutorial No.2:** Numerical on throughput, efficiency, speedup of a pipelined processor

**Tutorial No.3:** Problems on Cache Mapping

**Tutorial No.4**: Problems on VM mapping using paging.

**Tutorial No.5:** Design control steps for instructions.

**Tutorial No.6:** Problems on control unit design.

**Tutorial No.7:** Fast adder and fast multiplier.

**Tutorial No.8:** Floating addition, subtraction, multiplication, division.

**Tutorial No.9:** Numerical on SSD.

**Tutorial No.10:** Case study on multi core processor.

**Course Outcomes:**
**Upon completion of the course, graduates will be able –**

**CO1:** Identify and analyze the basic structures of a computer hardware units, connectivity and software.

**CO2:** Design the basic structure of machine instruction and programs, memory location.

**CO3:** Analyze different memory in the hierarchy, their mapping and their performance.

**CO4:** Analyze internal details of a processor, how instructions are executed using different hardware units, and how control unit controls all hardware components.

**CO5:** Study the design of ALU for arithmetic operations and useof registers.

**CO6:** Analyze the organization of secondary storage and how all the I/O devices communicate with CPU and transfer data.

**COURSE CODE:CS30306** **REF NO: To be filled by CD office**

# Computer Organization Lab

**Credits:**01 **Teaching Scheme: -** Laboratory 02Hrs/Week

**Prerequisites**:

1. Digital Electronic Circuit.
2. Computer Fundamentals.
3. ComputerProgramming .

**Objectives:**

4. Details study of each and every hard ware components used in a computer.
5. Study of Interfacing of different hardwares.
6. Study of Power supply to the different components of thecomputer .
7. Assembling of a PC.
8. Idea on VHDL coding.

**Course Details:**

- Study on mother board and different portd and slots connected to it.
- Study on different storage devices.
- Study on different input and output devices.
- Study on power supply unit and how it provides power supply to different hardware components.
- Fundamental to VHDL coding.

**List ofPracticals:**

Experiment No. 1: To study about the different I/O ports using trainer kit.

Experiment No. 2: To study about chipsets, ports and slots of motherboard using trainer kit.

**Experiment No.3:** To study on the internal architecture of HDD using trainer kit.

**Experiment No.4:** To study on internal architecture and function of keyboard using trainer kit.

**Experiment No.5:** To study on internal architecture and function of mouse using trainer kit.

**Experiment No.6:** To study on internal architecture and function of the printer using the

trainer kit.

**Experiment No.7:** To study on internal architecture and function of optical disk drive using trainer kit.

**Experiment No.8:** To study on internal architecture and function of monitor using trainer kit.

**Experiment No.9:** To study on internal architecture and function of SMPS using the trainer kit.

**Experiment No.10:**To study on dismantling and assembling of PC.

**Experiment No.11:** Experiments on simple fundamental units like half adder, full adder, using VHDL code.

**Experiment No.12:** Multiplexer , De-multiplexer using VHDL code.

**Text Books:**

T1. Patterson, D.A., and Hennessy, J.L. , "Computer Organization and Design: The Hardware/Software Interface", Morgan Kaufmann Publishers, 4th Edition, Inc.2005

T2. Michael Meyers, Lloyd Jeffries, " PC Hardware", McGraw Hill Professional, Inc. 2004

T2. VHDL Programming by Perry

**Reference Books:**

R1. Kauffmann,Computer System Architecture by M. Mano, 2001, Prentice-Hall.
R2. Computer Architecture- Nicholas Carter, 2002, T.M.H.

**Course Outcome:**

**CO1:** Get idea on different components on a motherboard.
**CO2:** Get idea on read and write mechanism of HDD and also idea on formatting and partitioning of it.
**CO3:** Get idea on all the different I/O devices used in a computer.
**CO4:**Get idea on power supplay to a computer.
**CO5:**Get information about assembling a personal computer.
**CO6:**idea on VHDL coding to realize different digital circuits.

**COURSE CODE: IT30104** **REF NO: _____**

# Concepts of Data Modelling

**Credits:** 3 **Teaching Scheme: -** Theory 3 Hrs/Week

**Prerequisites**:
1. Data Structures
2. Database Engineering

**Objectives:**

1. To get a clear understanding of computerize database concepts and their applications .
2. Be able to explain the concepts of relational database.
3. Be able to solve queries using SQL.
4. Be able to build relational database according to organization requirement.
5. To improve the database design knowledge to reduce redundancy and different anomalies
6. To get a clear understanding of different database system.

**Course Details:**

**Unit 1**
**Title – : Data Modelling (6 Hrs)**

**U1.1.**
Physical, Logical and Conceptual Data models, Entity-Relationship model

**U1.2. Self Study:** Types of Database systems, 3-schema architecture

**Unit 2**
**Title – : Relational Model (6 Hrs)**

**U2.1** Mapping E-R model to Relational model, Query Language: SQL, QBE, Datalog
**U2.2. Self Study**: Database Privacy, Integrity and Security

**Unit 3**
**Title – Relational Database Design (8 Hrs)**

**U3.1**
Functional dependency and Decomposition, Dependency Preservation & lossless Design,
Normalization, Normal forms:1NF, 2NF,3NF,and BCNF, Multi-valued Dependencies, .

**U3.2. Self Study:** 4NF & 5NF
Structured, Unstructured data and management policies

**Unit 4 (6 Hrs)**
**Title: Object Oriented Database Systems**
**U4.1:** Object relational database systems: Extensibility features and object orientation in relational database systems, ODBC, JDBC,

**U4.2. Self Study:** Object orientation in relational systems

**Unit 5**
**Title- Web Databases and advance topics**

**U5.1.** **(10 Hrs)**

XML/Web databases: semi-structured data, querying, Data mining, Data warehousing, online analytical processing, and information retrieval. Database system architectures (2-Tierand 3-Tier), Client_server architecture. Parallel and distributed database architectures, Performance issues.

**U5.2. Self Study:** Structured,Unstructured data and management policies

**Note:** Five assignments to be given to the students on self study, comprising of one assignment from each unit.

**Text Books:**

1 Database System Concepts, Silberschatz, Korth, Sudarshan, McGraw Hill International Edition, ISBN- 0-07-228363-7, 4th Edition
2. "Fundamentals of Database Systems", Elmasri and Navathe, Pearson Education, ISBN 81-297-0228-2, 4th Edition.

**References Books:**

1. An introduction to Database System – Bipin Desai, Galgotia Publications
2. Database System: concept, Design & Application by S.K.Singh (Pearson Education)
3. Database Modeling and Design: Logical Design by Toby J. Teorey, Sam S.Lightstone, and Tom Nadeau, "", 4th Edition, 2005, Elsevier India Publications, New Delhi
4. Fundamentals of Database Management System – Gillenson, Wiley India

**Course Outcomes:**

**Upon completion of the course, graduates will be able to –**

1. Apply the basic concepts of DBMS to Maintain the database and protect it.
2. Use different Data model concepts to design the appropriate database according to the requirement.
3. Use different design techniques to design the database(Relational).
4. Apply the DDLC and guidelines to avoid redundancy and anomalies.
5. Use the tools to connect frontend and backend.

**COURSE CODE: CS30103** **REF NO: _____**

# Big Data Analysis

**Credits:** 3 **Teaching Scheme: -** Theory 3 Hrs/Week

**Prerequisites**:

1. Computer Programming.
2. Linux OS
3. Basic of algorithms design

**Objectives:**

1. Master the concepts of HDFS and MapReduce framework
2. Understand Hadoop 2.x Architecture
3. Setup Hadoop Cluster and write Complex MapReduce programs
4. Perform data analytics using Pig, Hive and others
5. Implement HBase and MapReduce integration
6. Implement best practices for Hadoop development
7. Learn how to work with PIG

**Course Details:**

**Unit 1**

**Title- Introduction: (12 Hrs)**

**U1.1.**

Big Data Introduction, Hadoop Introduction, Different types of Components in Hadoop HDFS, MapReduce, PIG, Hive.

**Deep Drive in HDFS (for Storing the Data):** Introduction of HDFS, HDFS Design, HDFS role in Hadoop, Features of HDFS, Daemons of Hadoop and its functionality, Name Node,Secondary Name Node, Job Tracker, Data Node, Task Tracker, Anatomy of File Wright, Anatomy of File Read, Network Topology, Nodes, Racks, Data Center, Parallel Copying using DistCp, Basic Configuration for HDFS, Data Organization, Blocks and Replication, Rack Awareness.

**U1.2. Self Study:**

SQOOP, HBASE, OOZIE, Flume, Zookeeper.

**Unit 2**

**Title – Processing the Data with MapReduce: (06 Hrs)**

**U2.1.** The introduction of MapReduce, MapReduce Architecture, Data flow in MapReduce Splits, Mapper, Portioning, Sort and shuffle, Combiner, Reducer, Basic Configuration of MapReduce, MapReduce life cycle, Driver Code, Mapperand Reducer, How MapReduce Works.

**U2.3. Self Study:** Types of Counters, Task Counters, Job Counters, User Defined Counters, Propagation of Counters

**Unit 3**
**Title – Map Reduce Programming: (06 Hrs)**

**U3.1**
Submission & Initialization of MapReduce Job, File Input/output Formats in MapReduce Jobs, Text Input Format, Key Value Input Format, Sequence File Input Format, NLine Input Format, Joins, Map-side Joins, Reducer-side Joins

**U3.2. Self Study**: Job Scheduling, Understand Difference Between Block and InputSplit, Role of RecordReader

**Unit 4**
**Title – PIG: (06 Hrs)**

**U4.1**
Introduction to Apache PIG, Introduction to PIG Data Flow Engine, MapReduce vs PIG in detail,When should PIG use, Data Types in PIG,Basic PIG programming, Modes of Execution in PIG

**U4.2. Self Study:** Side Data Distribution

**Unit 5**
**Title – Cluster Setup: (06 Hrs)**

**U5.1**
Local Mode and MapReduce Mode, Execution Mechanisms, Grunt Shell, Script Embedded, Operators/Transformations in PIG, PIG UDF's with Program, The difference between the MapReduce and PIG

**U5.2. Self Study:** Heartbeat Signal

**Note: (1)** Five assignments to be given to the students on self study, comprising of one assignment from each unit.
(2) This course is equivalent to the "Hadoop Ecosystem" course of CDAC PG Diploma

**Text Books:**
1. Hadoop: The Definitive Guide, 4th Edition, Storage and Analysis at Internet Scale By:Tom White.
2. Hadoop Operations, By: Eric Sammer, Publisher: O'Reilly Media, Print ISBN: 978-1-4493-2705-7 | ISBN 10: 1-4493-2705-2

**Reference Books:**

1. Instant MapReduce Patterns - Hadoop Essentials How-to, By: Srinath Perera, Publisher: Packt Publishing Limited, Language: English, ISBN-10: 1782167706
2. Hadoop in Practice, By: Alex Holmes, 2ND Edition

**Course Outcomes:**

**Upon completion of the course, graduates will be able –**

**CO1**: Analyze Big Data and Hadoop ecosystem
**CO2**: Use SQOOP and Zookeeper
**CO3**: Apply Hadoop Distributed File System (HDFS)
**CO4**: Develop Map Reduce programs and implementing HBase
**CO5**: Develop Hive and Pig scripts

**COURSE CODE: CS30303** **REF NO: To be filled by CD office**

# Big Data Analysis Lab

**Credits:** 01 **Teaching Scheme: -** Laboratory 02 Hrs/Week

**Prerequisites**:

1) Computer Programming.
2) Object Oriented Programming Concepts.
3) Linux OS.
4) Computer Network.

**Objectives:**

**Experiment No. 1.** Master the concepts of HDFS and MapReduce framework
**Experiment No. 2.** Setup Hadoop Cluster and write Complex MapReduce programs
**Experiment No. 3.** Perform data analytics using Pig, Hive and others
**Experiment No. 4.** Implement HBase and MapReduce integration
**Experiment No. 5.** Implement best practices for Hadoop development
**Experiment No. 6.** Learn how to work with PIG

**Course Details:**

1. CLI commands (Introduction of Basic UNIX commands)
2. shell scripts
3. Counters (with Program)
4. Writing and Executing the Basic MapReduce Program using Java

**List of Practicals:**

**Experiment No. 1:**
1. Linux commands and Hadoop commands
2. Installing Java latest version
3. Installing Hadoop
4. Creating Cluster

**Experiment No. 2:**
1. Increasing Decreasing the Cluster size, Monitoring the Cluster Health
2. Starting and Stopping the Nodes
3. Hadoop Versioning and Configuration

**Experiment No. 3:**
1. Hadoop HDFS Commands
2. Storing Data into HDFS, How to Read the Data from HDFS, Accessing HDFS

**Experiment No. 4:**
1. Writing and Executing the Basic MapReduce Program

1. Word Count Example, Partition MapReduce Program.
2. Counters Program with Map-Reduce

**Experiment No. 5:**

1. Hive Installation
2. Commands in Hive
3. Exploring Internal and External Table

**Experiment No. 6:**

4. PIG installation
5. Word Count Example in PIG
6. Distributed Cache with Program

**Experiment No. 7:**

1. Sqoop Installations
2. Importing Data from Oracle to HDFS
3. Exporting Data from HDFS to Oracle

**Experiment No. 8:**

1. Hbase Installation
2. Exploring HBase Shell
3. Hive HBase Integration

**Experiment No. 9:**

1. Installing Oozie
2. Running Map-Reduce with Oozie

**Experiment No. 10:**

1. Running Pig and Sqoop with Oozie
2. Other Open source tools

**Note:** (1) This course is equivalent to the "Hadoop Ecosystem" course of CDAC PG Diploma

**Text Books:**

1. Hadoop: The Definitive Guide, 4th Edition, Storage and Analysis at Internet Scale By:Tom White

2. Hadoop Operations, By: Eric Sammer, Publisher: O'Reilly Media, Print ISBN: 978-1-4493-2705-7 | ISBN 10: 1-4493-2705-2

**Reference Books:**

1. Instant MapReduce Patterns - Hadoop Essentials How-to, By: Srinath Perera, Publisher: Packt Publishing Limited, Language: English, ISBN-10: 1782167706
2. Hadoop in Practice, By: Alex Holmes, 2ND Edition

**Course Outcomes:**
**Upon completion of the course, graduates will be able –**

**CO1**: Analyze Big Data and Hadoop ecosystem
**CO2**: Use SQOOP and Zookeeper
**CO3**: Apply Hadoop Distributed File System (HDFS)
**CO4**: Develop Map Reduce programs and implementing HBase
**CO5**: Develop Hive and Pig scripts

**COURSE CODE: IT30108** **REF NO: To be filled by CD office**

# Data Mining

**Credits:** 03 **Teaching Scheme:** Theory 03 Hrs/Week

**Prerequisites**:

- Database Engineering
- Computer Programming

**Objective**:

- To understand difference between data, information and knowledge
- To gain knowledge on the principles and techniques of data mining and knowledge discovery
- To get familiar to different data mining and web mining techniques

**Course Details:**

**Unit I: Data Mining and Pre-processing (8Hrs)**

**U1.1 Introduction:**

Need of Data Mining, Knowledge Discovery in Database (KDD), Architecture of Data Mining System; Data Objects and Attribute Types, Statistical Description of Data, Data Visualization

**U1.2. Data Preprocessing**:

Introduction to Data mining, Data mining Functionalities, Data preprocessing (data summarization, data cleaning, data integration and transformation, data reduction, data discretization)

**U1.3. Self Study**

Integration of Data Mining with a Database or Data Warehouse System, Issues in Data Mining

**Unit 2: Mining Frequent Patterns, Association and Correlations (8Hrs)**

**U2.1 Frequent Itemset Mining**:

Interesting Item Set Mining, Market Basket Analysis, Generating Association Rules, Apriori Algorithm, A pattern growth approach for mining frequent item set, Mining frequent item-sets using vertical data, Evaluation of Association Patterns, From Association Analysis to Correlation Analysis

**U2.2. Self Study**

Sequential Pattern Mining Algorithms, Pattern mining in multi-level, multi-dimensional space Data Integration: different types of digital data and their sources, ETL (extract transform and load) Tools

**Unit3: Classification and Prediction (8Hrs)**

**U3.1 Classification:**

Decision Tree Classifier, Rule Based Classification, Bayesian Classification, Neural Network Classification: Back Propagation Algorithm, Lazy Learner: KNN Classifier, Support Vector Machine Classifier Accuracy Measures, Techniques for Evaluating Classifier Accuracy, Ensemble Methods, Multiclass Problem.

**U3.2 Prediction:**

Linear, Non-Linear Regression.

**U3.3 Self Study:**

Case-Based Reasoning, Associative Classification, Other Classification Techniques: Genetic Algorithm, Fuzzy Set Approach, Rough Set, Constraints Based Association Mining

**Unit 4: Clustering and Outlier Detection (6 Hrs)**

**U4.1: Cluster Analysis:**

Categories of Clustering methods, Different Types of Clusters, Partitioning methods: k-Means, k-Medoids; Hierarchical Clustering Methods: BIRCH, Chameleon; Grid Based Methods: STING; Density based Clustering: DBScan, Cluster Evaluation

**U4.2: Outlier Analysis:**

Types of outlier, Proximity based approach: distance based, Density based approach

**U4.3 Self Study**:

Grid Based Methods: CLIQUE, Density based Clustering: OPTICS, Deviation based outlier detection approach: grid based

**Unit 5: Advanced Topics in Data Mining (6 Hrs)**

**U5.1 Web Mining**: Introduction, Web Mining, Web Content Mining, Web Structure Mining, Web Usage Mining, Text Mining, Unstructured Text, Episode Rule Discovery for Texts, Hierarchy of Categories, Text Clustering.

**U5.2 Temporal and Spatial Data Mining:** Introduction, What is Temporal Data Mining? , Temporal Association Rules, Sequence Mining, The GPS Algorithm, SPADE, SPRITE, WUM, Episode Discovery, Event Prediction Problem, Time-Series Analysis, Spatial Mining, Spatial Mining Tasks, Spatial Clustering, Spatial Trends, Conclusion.

**U5.3 Self Study:**

Graph Mining, Mining Time – Series Data, Multi-relational Data Mining, Data Mining Applications

**Text Books**


1. "Data Mining: Concepts and Techniques", Jiawei Han and Micheline Kamber, Morgan Kaufman, ISBN 978-81-312-0535-8, 2nd Edition.
2. "Data Mining Techniques", Arun K Pujari, 4st Edition,University Press, 2016.


**Reference Books**


1. "Data Mining and Analysis: Fundamental Concepts and Algorithms", Mohammed J. Zaki, Wagner Meira Jr., Cambridge University Press.
2. "Mastering Data Mining: The art and science of customer relationship management", M Berry and G. Linoff, John Wiley, ISBN 9971-51-369-2, 2001 Edition.
3. "Data Mining : Theory and Practice" , Soman K P, Diwakar Shyam, Ajay V, New Delhi, Prentice Hall Of India, ISBN 81-203-2897-3, 2006 Edition.
4. "Introduction to Data Mining", Pang-Ning Tan, Vipin Kumar, Michael Steinbach, Pearson 2014.
5. "Data Mining Introductory and advanced topics" , Margaret H Dunham, 6th Edition, Pearson Education, 2009.


**Course Outcomes:**

**Upon completion of the course the graduate students will be able to**

1. Identify data mining architecture and different pre-processing techniques required for analysis of given dataset
2. Analyze frequent patterns, determine associations and correlations
3. Apply different classification and prediction to data mining applications
4. Use different clustering mechanisms for data mining
5. Apply data mining for textual, temporal and unstructured data on the Web

**COURSE CODE: IT30308** **REF NO: To be filled by CD office**

# Data Mining Lab

**Credits:** 01 **Teaching Scheme:** Lab 02 Hrs/Week

**Prerequisites**:

- Database Engineering
- Computer Programming

**Objective**:

- To understand difference between data, information and knowledge
- To gain knowledge on the principles and techniques of data mining and knowledge discovery
- To get familiar to different data mining and web mining techniques

**Course Details:**

- Basic programming using data mining software such as R (Rattle), WEKA, Rapid Miner, & KNIME (Any One)
- Program for statistical processing of data
- Programs on data pre processing, cleaning and transformations
- Programs on
    - Pattern mining
    - Prediction
    - Cluster analysis
- Report writing using open source documentation software such as Latex
    - Creation of text with header, footer, formatting
    - Creation of Tables
    - Creation of Graphs
    - Mathematical Equation

**Course Outcomes:**
**Upon completion of the course the graduate students will be able to**

1. Gain expertise on the use of data mining software
2. Develop programs for Data pre-processing
3. Analyze data mining requirements and develop programs using suitable data mining technique
4. Create reports using suitable software

**COURSE CODE: IT34352** **REF NO: To be filled by CD office**

# J2EE Enterprise Java Lab

**Credits:** 01 **Teaching Scheme: -** Laboratory 02 Hrs/Week

**Prerequisites**:

1) Basic Object Oriented Programming concepts.
2) Basic knowledge of Java.
3) Basic knowledge of HTML, XML.

**Objectives:**

1. Understand multi-tiered enterprise applications.
2. Understand J2EE framework for developing enterprise applications.
3. Understand various components of J2EE like JSP, Servlets, and effectively use them.
4. Understand Application Server and its configurations.
5. Learn and deploy web based applications in application server.

**Course Details:**

1. Remote Method Invocation
2. Servlets
3. Java Server Pages
4. Enterprise Java Beans
5. Naming Services, Java Mail and Java Messaging Services
6. Introduction to Struts Framework
7. Introduction to hibernate and HQL
8. Introduction to Spring Framework,
9. Web services

**List of Practicals:**

**Experiment No. 1:** Create a RMI Program showing marshalling and un-marshalling processes.

**Experiment No. 2:** Log in application using servlet and JDBC.

**Experiment No. 3:** Implement session tracking techniques in servlets.

**Experiment No. 4:** Log in application using JSP and JDBC.

.

**Experiment No. 6:** Implement session tracking techniques in JSP.

**Experiment No. 7:** Implementation of session, message-driven and entity bean using EJB.

**Experiment No. 8:** Programs on

- **i.** Implementation of JMS to send mail
- **ii.** Implementation of Java Message Service to send SMS.

**Experiment No. 9:** Programs on

- **i.** Creating registration form using Struts
- **ii.** Implementing Dependency injection and inversion of control

**Experiment No. 10:** Develop a web application using hibernate to maintain student data.

**Experiment No. 11:** Create a web application for ticket booking using spring.

**Experiment No. 12:** Creation of web services with JAX-WS.

**Text Books:**


1. Kongent S., "Java Server Programming (JEE 6) Black Book, Platinum Edition", 2008, Dreamtech / Wiley India Pvt. Ltd.
2. Eric Jendrock, D. Carson, I. Evans, D. Gollapudi, K. Haase, C. Srivastha, "The Java EE6 Tutorial", Volume-1, Fourth Edition, 2010, Pearson India, New Delhi.


**Reference Books:**


1. Douglas E. Comer, "Internetworking with TCP/IP, Volume 1: Principles, Protocols and Architecture", Fifth Edition, 2006, PHI Learning Pvt. Ltd., New Delhi.
2. SANTOSH KUMAR K , "Jdbc, Servlets, And Jsp Black Book" Black Book, New Edition, 2008


**Course Outcomes:**

**Upon completion of the course, graduates will be able to-**

1. Use RMI for invoking remote methods for user benefit.
2. Implement different applications using Servlets and JSPs.
3. Use MVC architecture through EJB
4. Develop applications through Struts & Spring frameworks

# SIXTH SEMESTER CSE 2015-16

**COURSE CODE: CS30109** **REF NO: To be filled by CD office**

# Operating System

**Credits:** 3 **Teaching Scheme: -** Theory 3 Hrs/Week

**Prerequisites**:

1. Computer Organization
2. Computer Programming

**Objectives:**

1. To understand main components of OS and their working
2. To study the operations performed by OS as a resource manager
3. To understand the different scheduling policies of OS
4. To understand the different memory management techniques
5. To understand process concurrency and synchronization
6. To understand the concepts of input/ output, storage and file management
7. To study different OS and compare their features.

**Course Details:**
**Unit 1**
**Title- Introduction to OS** **(6hrs)**

**U1.1.**
Operating Systems Objectives and functions, Components of OS, OS Structure, Evolution of Operating Systems - Simple Batch, Multiprogramming, Time shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, Operating System services, System Calls.
**U1.2. Self Study:** System Programs, System structure, Virtual Machines, Dual Mode Operation.

**Unit 2**
**Title – Study of Process Management:** **(6Hrs)**

**U2.1.**
**Process Management:**
Process and CPU Scheduling - Process concepts – Process and Process States, Process Control Block, Cooperating Processes, Inter-process Communication, Process Scheduling - Scheduling Queues, Schedulers, Context Switch, Non Pre-emptive and Pre-emptive Scheduling, Dispatcher, Schedulability Criteria, Scheduling algorithms.

Process Coordination: Process Synchronization, The Critical section Problem, Peterson's solution, Synchronization Hardware, Semaphores, and Classical Problems of Synchronization, Monitors. Threads and its type.
**U2.2. Self Study:** Process creation mechanism and scheduling algorithms used in Linux and Windows.

**Unit 3**
**Title – Deadlock: (6 Hrs)**

**U3.1**
Deadlocks: System model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock avoidance, Deadlock Detection, recovery from Deadlock.
**U3.2. Self Study**: Thread creation and Thread scheduling in Linux and Windows.

**Unit 4**
**Title – Memory Management: (6 Hrs)**

**U4.1**
Memory Management strategies, Background, Logical versus Physical Address space, MMU, Address Translation, Swapping, Contiguous Allocation, Paging, Segmentation.
Virtual Memory: Background, Demand paging, Page Replacement concepts, Page Replacement Algorithms. Allocation of frames, Thrashing, Segmentation with Paging,
**U4.2. Self Study:** Multilevel Paging, Inverted Page Table, Demand Segmentation, Case study using Linux and Windows.

**Unit 5**
**Title- File System & Storage Management (8 Hrs)**

**U5.1.**
File System Interface - The Concept of a File, Access methods, Directory Structure, File System Mounting, File Sharing, Protection, File System Structure, File System Implementation, Allocation methods.
I/O Systems- Overview of Mass Storage Structure, Device Drivers, Disk Structure, Disk Scheduling, Disk Management, and Swap space Management, Free-space Management, Directory Implementation, RAID Structure.
**U5.2. Self Study:** Disk Attachment, Stable Storage Implementation, Case studies on File system: LINUX and Windows.

**Note:** Five assignments to be given to the students on self study, comprising of one assignment from each unit.

**Text Books:**
1. Operating System Principles, Abraham Silberchatz, Peter B. Galvin, Greg Gagne 9th Edition, Wiley Student Edition.
2. Operating systems - Internals and Design Principles, W. Stallings, 6th Edition, Pearson.

**Reference Books:**

1. Modern Operating Systems, Andrew S Tanenbaum 3rd Edition PHI.
2. Operating Systems A concept - based Approach, 2nd Edition, D. M. Dhamdhere, TMH.
3. Principles of Operating Systems, B. L. Stuart, Cengage learning, India Edition.

4. Operating Systems, A. S. Godbole, 2nd Edition, TMH
5. An Introduction to Operating Systems, P.C.P. Bhatt, PHI.
6. Operating Systems, S, Haldar and A. A. Arvind, Pearson Education.
7. Operating Systems, R. Elmasri, A. G. Carrick and D. Levine, Mc Graw Hill.
8. Operating Systems in depth, T. W. Doeppner, Wiley.

**Course Outcome**

1. Compare and contrast different types of operating system and analyze the fundamentals of operating system concepts, services, and components.
2. Analyze and understand operating system process scheduling criteria, scheduling algorithms, threading issues, thread scheduling and inter-process communication mechanism.
3. Analyze process synchronization mechanism by studying and using classical problems of synchronization.
4. Characterize methods and mechanisms to handle, prevent, avoid and detect deadlock in a system and inspect main memory and virtual memory management strategies.
5. Analyze file and directory structure management along with I/O management issues.
6. Carry out case studies in different contemporary operating systems.

**COURSE CODE: CS30309** **REF NO: To be filled by CD office**

# Operating System Laboratory

**Credits:** 1 **Teaching Scheme: -** Laboratory **2 Hrs/Week**

**Prerequisites**:

1. Computer Organization and Architecture
2. Computer Programming

**Objectives:**

1. To understand main components of Linux OS and the file system
2. To familiar with shell scripts and the environment
3. To do hands on practice on process creation and synchronization
4. To do hands on practice on system call and different CPU scheduling programs
5. To works for solution to Deadlock situation
6. To have hands on practice on different page replacement algorithm

**Course Details:**

1. Introduction to Linux and its file system
2. Basic UNIX Commands.
3. UNIX Shell Programming.
4. Programs on process creation and synchronization including shared memory, pipes and messages.( classical problems on synchronization)
5. Programs on UNIX System calls.
6. Programs on CPU Scheduling Algorithms.
7. Programs on Banker's Algorithm for Deadlock Avoidance, Prevention
8. Programs on page replacement algorithm.

**List of Practicals:**

**Experiment No. 1:** Introduction to Linux and its File system, File access, mounting.

**Experiment No. 2:** Basic utility commands of LINUX, File & Directory related commands, basic commands on Linux Administration.

**Experiment No. 3:**Introduction to shell programming, Control statements

**Experiment No. 4:** CPU Scheduling Algorithms. (FCFS, RR, SJF, Priority)

**Experiment No. 5:** Program for FIFO, LRU, and OPTIMAL page replacement algorithm

**Experiment No. 6:** Programs on process creation, IPC using Pipelines, IPC using message**.**

**Experiment No. 8:** Thread creation using Pthreads

**Experiment No. 9:** UNIX: Semaphore, Signals**.**

**Experiment No. 10:** Program on Banker's Algorithm for Deadlock Avoidance.

**Text Books:**


1. Unix Concept and Applications 4th Edition Sumitabha Das, The McGraw-Hill Companies
2. UNIX: The Complete Reference, Second Edition, Kenneth Rosen and Douglas Host


**Reference Books:**

**Course Outcome**

At the end of the course, the student should be able to:

1. Differentiate Linux file system from other OS.
2. Comparison of the performance of various CPU scheduling algorithms.
3. Critically analyzing the performance of the various page replacement algorithms.
4. Create processes and implement IPC.
5. Implementation of deadlock avoidance algorithms.

**COURSE CODE: CS30109** **REF NO: To be filled by CD office**

# Operating System Tutorials

**Credits:** 01 **Teaching Scheme:** Tutorial 01 Hrs/Week

**Prerequisites**:

3. Computer Organization
4. Computer Programming

**Objectives:**

8. To understand main components of OS and their working
9. To study the operations performed by OS as a resource manager
10. To understand the different scheduling policies of OS
11. To understand the different memory management techniques
12. To understand process concurrency and synchronization
13. To understand the concepts of input/ output, storage and file management
14. To study different OS and compare their features.

## List of Tutorials

**Tutorial No. 1:** Operating system services and system calls

**Tutorial No. 2:** Numerical on process scheduling (non-preemptive)

**Tutorial No. 3:** Numerical on process scheduling (preemptive)

**Tutorial No. 4:** Classical problems of synchronization

**Tutorial No. 5:** Problems on deadlock prevention and Problems on deadlock avoidance

**Tutorial No. 6:** Virtual to Physical memory translation

**Tutorial No. 7:** Numerical on memory allocation strategies

**Tutorial No. 8:** Numerical on page replacement algorithms

**Tutorial No. 9:** File system structure and its implementation

**Tutorial No. 10:** Problems on disk scheduling

**Text Books:**

1. Operating System Principles, Abraham Silberchatz, Peter B. Galvin, Greg Gagne 9th Edition, Wiley Student Edition.
2. Operating systems - Internals and Design Principles, W. Stallings, 6th Edition, Pearson.

**Reference Books:**

9. Modern Operating Systems, Andrew S Tanenbaum 3rd Edition PHI.
10. Operating Systems A concept - based Approach, 2nd Edition, D. M. Dhamdhere, TMH.
11. Principles of Operating Systems, B. L. Stuart, Cengage learning, India Edition.
12. Operating Systems, A. S. Godbole, 2nd Edition, TMH
13. An Introduction to Operating Systems, P.C.P. Bhatt, PHI.
14. Operating Systems, S, Haldar and A. A. Arvind, Pearson Education.
15. Operating Systems, R. Elmasri, A. G. Carrick and D. Levine, Mc Graw Hill.
16. Operating Systems in depth, T. W. Doeppner, Wiley.

**Course Outcome**

7. Compare and contrast different types of operating system and analyze the fundamentals of operating system concepts, services, and components.
8. Analyze and understand operating system process scheduling criteria, scheduling algorithms, threading issues, thread scheduling and inter-process communication mechanism.
9. Analyze process synchronization mechanism by studying and using classical problems of synchronization.
10. Characterize methods and mechanisms to handle, prevent, avoid and detect deadlock in a system and inspect main memory and virtual memory management strategies.
11. Analyze file and directory structure management along with I/O management issues.
12. Carry out case studies in different contemporary operating systems.

**COURSE CODE: CS30110** **REF NO: ----------**

# Machine Learning

**Credits: 3** **Teaching Scheme: Theory 3 Hrs/Week**

**Prerequisites**:

1. Design & Analysis of Algorithm
2. Discrete Mathematics
3. Probability & Statistics

**Objectives:**

1. To make the students familiarize with fundamental understanding of Machine learning
2. Enable students to use machine learning techniques to solve some real life problems.
3. To evaluate the power and limitation of deep learning in solving computational problems.

**Course Details:**

**Unit-1**
**Title- Introduction** **(7 Hrs)**

**U1.1 Basic concepts**

Basic Mathematical and Statistical concepts: Metric, Matrices, Eigen values and Eigen vectors, mean median, mode, variance, co-variance, correlation, dispersion matrix, binomial distribution, normal distribution, multi-variate normal distribution, basic concepts in probability theory such as Bayes theorem, error risk minimization, laws of large numbers.

**U1.2**

Chebyshev's inequality, central limit theorem, machine learning system.

**Unit-2**
**Title: Supervised learning** **(8 Hrs)**

**U2.1 Supervised learning.**

Supervised learning setup (training, testing). Minimum distance classifier, k-nearest neighbour classifier, density estimation. Instance based learning, linear regression. Logistic regression. Perceptrons (single layer/multi-layer). Model selection, dimensionality reduction, and feature selection. Ensemble methods: Bagging, boosting. Measuring the accuracy of learned hypotheses. Comparing learning algorithms: cross-validation,

**U2.2 <u>Self Study:</u>**

Learning curves and statistical hypothesis testing.

**Unit-3** **(7 Hrs)**

**Title: Unsupervised learning**

**U3.1 Unsupervised learning**

Clustering. Similarity measures, K-means algorithm, Hierarchical clustering, Density based clustering, anomaly detection, cluster validation, Expectation Maximization. PCA (Principal components analysis). ICA (Independent components analysis).

**U3.2 Self Study**

Mixture of Gaussian, Factor analysis.

**Unit-4**

**Title: Learning theory. (8hrs)**

**U4.1**

Inductive and deductive learning. Models of learnability: learning in the limit; probably approximately correct (PAC) learning, Generative learning algorithms. Gaussian discriminant analysis. Bayesian Classification. VC dimension, Maximum Margin Classifiers, Support vector machines. Bias/variance trade-off, no free lunch theorem, Union and Chernoff/Hoeffding bounds. Worst case (online) learning.

**U4.2 Self Study**

Machine Learning System Design, Handling Skewed Data, Using Large Data Sets

**Unit-5**

**Title: Recent techniques (6 hrs)**

**U5.1**

Decision trees, Random forests, Semi-supervised and active learning; reinforcement learning, kernel functions, one class classifier

**U5.2 Self Study**

Case studies using Ensemble learning: bagging, boosting.

**Text Books:**


1. Machine Learning. Tom Mitchell. First Edition, McGraw- Hill, 1997.
2. Introduction to Machine Learning Edition 2, by Ethem Alpaydin


**Reference Books:**

**Course Outcomes:**

**Upon completion of the course, the students will be able to:**

1. Use mathematical concepts required for machine learning
2. Identify and differentiate different types of supervised learning
3. Identify and differentiate different types of unsupervised learning
4. Apply learning mechanisms like Bayesian Classifier, SVM etc.
5. Explore advanced methods of machine learning

**COURSE CODE: CS30110** **REF NO:----------**

# Machine Learning: Tutorial

**Credits: 1** **Teaching Scheme: Theory** 1 Hr/Week

**Prerequisites**:

4. Design & Analysis of algorithm
5. Discrete Mathematics
6. Probability & Statistics

**Objectives:**

4. To make the students familiarize with fundamental understanding of Machine learning
5. Enable students to use machine learning techniques to solve some real life problems.
6. To evaluate the power and limitation of deep learning in solving computational problems.

**Tutorial Details:**

1. Discussion on statistical concepts and prability theory used for Machine Learning.
2. Discussion and problem solving related to supervised machine learning methods:
    - k-nearest neighbour classifier, density estimation
    - Instance based learning, linear regression. Logistic regression.
    - Perceptrons (single layer/multi-layer)
    - Model selection, dimensionality reduction, and feature selection.
    - Ensemble methods: Bagging, boosting
3. Discussion on unsupervised machine learning methods:
    - Similarity measures, K-means algorithm
    - Hierarchical clustering, Density based clustering, anomaly detection, cluster validation
    - PCA (Principal components analysis)
    - ICA (Independent components analysis)
4. Discussion and application of learning theory:
    - Probably approximately correct (PAC) learning
    - Generative learning algorithms
    - Gaussian discriminant analysis. Bayesian Classification

- VC dimension, Maximum Margin Classifiers, Support vector machines
- Union and Chernoff/Hoeffding bounds

5. Discussion of problems on advanced topics:
    - Semi-supervised and active learning
    - Reinforcement learning
    - Kernel functions

**Text Books:**

3. Machine Learning. Tom Mitchell. First Edition, McGraw- Hill, 1997.
4. Introduction to Machine Learning Edition 2, by EthemAlpaydin

**Course Outcomes:**

**Upon completion of the course, the students will be able to:**

1. Use mathematical concepts required for machine learning
2. Identify and differentiate different types of supervised learning
3. Identify and differentiate different types of unsupervised learning
4. Apply learning mechanisms like Bayesian Classifier, SVM etc.
5. Explore advanced methods of machine learning

**COURSE CODE: IT30109** **REF NO: To be filled by CD office**

# IT Security

**Credits:** 03 **Teaching Scheme:** Theory 3hrs/Week

**Prerequisites**:

1. Data Communication and Computer Networks.
2. Computer Programming.

**Objectives:**

1. Provide a comprehensive introduction to security fundamentals.
2. Get familiarize the student about Intrusion detection, prevention and recovery Schemes.
3. Provide detail study about Virtual private network deployment and management along with web application security risks.
4. Gain knowledge about OWASP application security risks.

**Course Details:**

**UNIT 1: Security fundamentals and Firewalls (6hrs)**

**U1.1** Introduction to Security fundamentals: Security Goals, Security Services, Types of security attacks. Firewalls: Types of Firewalls, Limitations of firewall, Cryptography Network security.

**U1.2** **Self Study**: Case study on different Crypto Systems

**UNIT 2: Intrusion detection and prevention (8hrs)**

**U2.1** Intrusion Detection and Prevention, Intrusion risks, Security policy, Monitoring and reporting of traffics, Traffic shaping, Investigating and verifying detected intrusions, recovering from, reporting and documenting intrusions, Define the Types of intrusion Prevention Systems

**U2.2** **Self Study:** Intrusion prevention system basics, Limitations of Intrusion Prevention System.

**UNIT 3: Packet signature and analysis (6hrs)**

**U3.1** Spoof Prevention, Dos, Qos Policy, Packet Signature and Analysis.

**U3.2** **Self Study:** Web Application Firewall.

**UNIT 4: Virtual private network (8hrs)**

**U4.1** Virtual Private Networks, Deploy and managing VPN, VPN Performance tuning and error handling, VPN routing, DMZ and virtual host, Reverse proxy, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), Web application Security Risks.

**U4.2** **Self Study:** SQL Injection.

**UNIT 5: OWASP** **(8hrs)**

**U5.1** Identifying the Application Security Risks, Open Web Application Security Project (OWASP) Top 10 Concepts Invalidated Redirects and Forwards

**U5.2** **Self Study:** Threat Risk Modelling.

**Text Books:**


1. William Stallings ,"Cryptography and Network Security-Principles and Practices" , Pearson Education, 2006, ISBN 81-7758-774-9, 4th Edition.
2. B. A. Forouzan & D Mukhopadhyay,"Cryptography and Network Security.", McGraw Hill, 2nd ed.2010
3. B. Menezes,"Network Security and Cryptography", Cengage Learning, 1st ed.2010


**Reference Books:**


1. Matt Bishop ,"Computer Security: Art and Science", Pearson Education, 2002, ISBN 0201440997, 1st Edition.
2. Charlie Kaufman, Radia Perlman and mike speciner, "Network security, private communication in a public world", Prentice Hall, 2002, ISBN 9780130460196, 2nd Edition.


**Course Outcomes:**

**Upon completion of the course, the students will be able to:**

1 Distinguish among different type of security attack on a given system.
2 Analyze Intrusion risks, investigate, verify and recover intrusion.
3 Estimate future needs of security for a system by researching current environment on a continuous basis for the benefit of society.
4 Justify various methods to undertake security projects for application of technologies to various sections of industry and society.

**COURSE CODE: CS30112** **REF NO: To be filled by CD office**

# Compiler Design

**Credits:** 3 **Teaching Scheme: -** Theory 3 Hrs/Week

**Prerequisites**:

1. Theory of Computation

**Objective:**

1. To apply the fundamental of formal attributed grammars for writing the syntax and semantics of programming languages
2. To introduce the concepts underlying the design and implementation of language processors.

**Course Details:**

**Unit 1**

**Title- Introduction to Compiler, Lexical analysis** **(5 Hrs)**

**U1.1**

**Introduction:** Overview of Compiler, Phases and Passes, Bootstrapping, Cross Compiler.
**Lexical Analysis:** Role of a Lexical Analyzer, Specification and Recognition of Tokens, Design of a lexical analyzer as a DFA, lexical analyzer generator, Converting regular expression directly to a DFA, Regular expressions and regular languages, Non-deterministic and deterministic finite automata (NFA & DFA).

**U1.2**

**Self Study:** Lexical analysis using LEX tools.

**Unit II**

**Title- Syntax Analysis** **(9 Hrs)**

**U2.1**

**Syntax Analysis:** Role of a parser, Top Down Parsing: LL (1) grammars, predictive parsing. Bottom Up Parsing: Handle pruning and shift reduce parsing, Operator precedence parsing, SLR parsers and construction of SLR parsing tables, LR(1) parsers and construction of LR(1) parsing tables, LALR parsers and construction of efficient LALR parsing tables, parsing using ambiguous grammars, Context free grammars (CFG) and Context free languages.

**U2.2**

**Self Study:** Parsing using YACC tools.

**Unit III**

**Title- Syntax Directed Translation, Error recovery and Intermediate code generation** **(9 Hrs)**

**U3.1**

**Syntax Directed Translation:** Syntax-Directed Definitions (SDD), Semantic Rules, Evaluation

of SDD using syntax tree.

**Error Detection & Recovery:** Lexical Phase errors, syntactic phase errors, semantic errors.

**Intermediate Code Generation:** Three address codes - quadruples and triples, types and declarations, translation of expressions, array references, translation of Boolean expressions and control flow statements, Back patching, intermediate code generation for procedures.

**U3.2**
**Self Study:** Structure and features of symbol tables, symbol attributes and scopes, type checking, type conversion.

**Unit IV**
**Title- Code Generation** **(5 Hrs)**

**U4.1**
**Code Generation:** Issues in Code Generation, Basic Blocks and Flow Graphs, DAG representation of Basic Blocks, Generating code from DAG.
**U4.2**
**Self Study:** Storage Organization, Storage Allocation strategies, handlings of activation records for calling sequences.

**Unit V**
**Title- Code Optimization** **(5 Hrs)**

**U5.1**
**Code Optimization:** Introduction, Principal Sources of Optimization, Optimization of basic Blocks, Introduction to Global Data Flow Analysis, Peephole optimization.
**U5.2**
**Self Study:** Runtime Environments.

**Text Books**

1. "Compilers: Principles, Techniques and Tools", A. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman, Pearson Education, ISBN 978-81317-2101-8, Second Edition, 2007.
2. "Engineering a Compiler", K. Cooper, L. Torozon, Morgan Kaufmann, ISBN 1-55860-698-X, First Edition, 2003.

**Reference Books**

1. "Lex & Yacc", J. R. Levine, T. Mason, D. Brown, "Lex & Yacc", J. R. Levine, T. Mason, D. Brown, O"Reilly, ISBN 1-56592-000-7, Second Edition, 1992.
2. "Compiler Construction: Principles and Practice", K. Louden, Course Technology, ISBN 0-534-93972-4, First Edition, 1997.

**Course Outcomes:**

1. Review different phases and passes of compiler and be able to use the compiler tools like LEX, YACC, etc.
2. Able to explain lexical analysis phase and its underlying formal models such as finite state automata, and their connection to language definition through regular expressions and grammars.
3. Differentiate various parser construction techniques.
4. Able to use formal attributed grammars for specifying the syntax and semantics of programming languages.
5. Able to generate the code for the target machine.
6. Use code optimization techniques to improve the performance of a program in terms of speed & space.

**EC30108/ EC31108:** **REF NO: To be filled by CD office**

# MICROPROCESSOR AND MICROCONTROLLER

**Credits:**3 **Teaching Scheme: -** Theory 03Hrs/Week

**Prerequisites**: Digital Electronics Circuit

**Co requisites:** Nil

**Post Requisites:** Computer Organization and Architecture

**Objectives:**

To provide a theoretical & practical introduction to microprocessors and microcontrollers, assembly language programming techniques, design of hardware interfacing circuit, microcontroller and microprocessor system design considerations and preamble of some of the advanced processors

**Course Details:**

**Unit1**

**Introduction to Microprocessor and its organization (08Hrs)**

**U1.1.** General concept of microprocessor and architecture, Bus organization, Memory concepts (T1: CH2.1,2.2,2.3), 8085 microprocessor, Pins and Signals, Programming model, Instruction execution and timings, Instruction Set, Interfacing I/O devices.(T1: CH2.5, 2.6, 2.7,2.8 )

**U1.2.** Addressing modes, 8085 interrupts(T1: CH2.4, 2.9).

**Unit2**

**Intel 8086 Microprocessor (06Hrs)**

**U2.1.** Bus Interface unit, Execution Unit, Register Organization, Memory Segmentation, Minimum and Maximum mode system configuration, Physical Memory Organization (T2: CH1.1, 1.2, 1.3, 1.4), 8086 Interrupts (T2: CH4.4, 4.5, 4.6).

**U2.2.** 8255Programmable Peripheral Interface (T2: CH5.4, 5.5), 8257 DMA controller (T2: CH7.1, 7.2)

**Unit3**

**8051 MICROCONTROLLER AND FEATURES (08Hrs)**

**U3.1.** Introduction to Microcontroller, Embedded versus External Memory Devices, 8-bit and 16-bit Microcontrollers, CISC and RISC Processors (T3: CH1.1, 1.2). MCS-51 Architecture, Registers in MCS-51, 8051 Pin Description, Memory Organization, 8051 Addressing

Modes (T3: CH2, 5.1, 5.2, 8.1), MCS-51 Instruction Set, 8051 Instructions and Simple Programs(T3: CH3.1, 3.2, 6.1, 6.2, 6.3, 6.4), Interrupts in MCS-51(T3: CH11.1).

**U3.2.** Special function Registers, Assembly language programming(T3:CH7, 11).

**Unit4**

**MICROCONTROLLER AND ITS APPLICATIONS (08Hrs)**

**U4.1.** 8051 Timers and Counters (T3: CH9.1), Serial Communication (T3: CH10.1), I/O Interfacing using 8255, Light Emitting Diodes (LEDs), Push Buttons, Relays and Latch Connections, Interfacing 7-Segment Displays, ADC and DAC Interfacing (T3: CH13, 15, 17.1).

**U4.2.** Traffic light controller, stepper motor control (T3: CH17.2).

**Unit5**

**ADVANCED MICROPROCESSOR AND MICROCONTROLLER (06Hrs)**

**U5.1.** Introduction to processor design, ARM architecture (R3: CH3.1), Pentium Processors (T2: CH11.1, 11.2, 11.3), MSP 430 (TI Systems Manual).

**U5.2.** Advanced Programming using ARM, MSP 430 (R3: CH10).

## Text Books:


T1. "Microprocessors and Microcomputer based System Design", M. Rafiquzzaman,UBS, 2nd Edition, 2001.

T2. "Advanced Microprocessors and Perpherals", K. M. Bhurchandi, A. K. Ray,McGraw Hill Education, 3rd Edition, 2012.

T3. "The 8051 Microcontroller and Embedded Systems: Using Assembly and C", Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. Mckinlay,Pearson, 2nd Edition, 2011.

T4. "ARM system-on-chip Architecture", Steve Furber, Pearson, 2nd Edition, 2009


## Reference Books


R1. "Microprocessor Architecture, Programming, and Applications with the 8085", Ramesh Gaonkar,CBS Publishers, 5th Edition, 2011.

R2. "The 8086 Microprocessor: Programming & Interfacing the PC", Kenneth Ayala,Delmar Cengage Learning, 1st Edition, 2007.

R3. ""ARM Assembly Language : An Introduction", J. R. Gibson, Cengage Learning, 1st Edition, 2010.

**Course Outcomes:**

CO1. Apply the knowledge of digital systems to understand architectural differences, memory organization and functionality of microprocessor and microcontrollers.

CO2. Analyze the programming model and interfacing techniques of microprocessor and microcontroller.

CO3. Design and demonstrate the advanced microprocessor and microcontroller systems for real time applications.

CO4. Read voluntarily to enhance the knowledge in processing, controlling systems in computing domain.

**EC30308/EC31308:** **REF NO: To be filled by CD office**

# Microprocessors & Microcontrollers Laboratory

**Credits:**01 **Teaching Scheme: -** Laboratory 02Hrs/Week

**Prerequisites**: Digital Electronics Circuit

**Co requisites:** Nil

**Post Requisites:** Microcontroller for Embedded System Laboratory (PD)

**Objectives:**

a. Familiarize the architecture of 8085, 8086 processor, assembling language programming and interfacing with various modules.

b. The student can also understand of 8051 Microcontroller concepts, architecture, programming and application of Microcontrollers.

c. The Student able to do any type of VLSI, embedded systems, industrial and real time applications by knowing the concepts of Microprocessor and Microcontrollers

**Course Details:**

**List ofPracticals: (Any 10)**

**Experiment No.1:** Addition, Subtraction, Multiplication, division and 2s complement of 8-bit and 16-bit numbers using 8085 microprocessor.

**Experiment No.2:** Transfer a group of data from one set of memory to another set of memory locations using 8085 microprocessor.

**Experiment No.3:** To write an assembly language program to obtain a rolling display of a particular data using 8085 microprocessor.

**Experiment No.4:** Speed control of stepper motor using 8085 microprocessor.

**Experiment No.5:** Study of traffic light controller using 8255 PPI with 8085 microprocessor.

**Experiment No.6:** To find the largest and smallest data among a group of data using 8086 microprocessor.

**Experiment No.7:** Generation of different type of analog signals using DAC interfaced to 8086 microprocessor.

**Experiment No.8:** Interfacing of 8255 with 8086 microprocessor and square wave, PWM generation.

**Experiment No.9:** Study of interrupt control applications using 8086 microprocessor.

**Experiment No.10:** Study of addressing modes of 8051 microcontroller and arithmetic operation using 8051 microcontroller.

**Experiment No.11:** Code converters using 8051 microcontroller.

**Experiment No.12:** To initialize 8279 and display character in the first digit of the display using 8051 microcontroller.

## Text Books:


T1. "Fundamentals of Microprocessors and Microcontrollers", B. Ram,Dhanpat Rai Publications, 2nd Edition, 2010.

T2. "Microprocessors and Microcontrollers: Architecture, Programming and Interfacing Using 8085, 8086 and 8051", Soumitra K. Mandal, TMH, 1st Edition, 2011.

T3. "The 8051 Microcontroller and Embedded Systems: Using Assembly and C", Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. Mckinlay,Pearson, 2nd Edition, 2011.


## Reference Books


R1. "Microprocessors and Microcontrollers: Architecture, Programming and System Design 8085, 8086, 8051", Krishna Kant,PHI Learning, 7th Edition, 2010.

R2. "The Intel Microprocessors: Architecture, Programming and Interfacing", Barry B. Brey,Pearson, 8th Edition, 2008.

R3. "The 8051 Microcontroller", Kenneth Ayala,Cengage Learning, 3rd Edition, 2007.


## Course Outcomes:

CO1. Ability to understand the assembly language programming concept by the programming model of different microprocessors

CO2. Ability to interface various I/O devices with the microprocessor and microcontroller for different controlling applications

CO3. Analyze the generation of different analog signals using various interfacing circuits.

CO4. Practice the programming skills to solve the problems of microprocessor and microcontroller system.

**COURSE CODE: CS30313** **REF NO: To be filled by CD office**

# Object Oriented Programming with C++ Lab

**Credits:** 01 **Teaching Scheme:** Theory 02 Hrs / Week

**Prerequisites**:
1. Knowledge of any programming language.
2. Knowledge of various control structures
3. Knowledge of functions
4. Knowledge of basic I/O mechanisms.
5. Ability to apply logic.

**Objectives:**
1. To get a clear understanding of object oriented programming and C++ concept.
2. To be able to explain the difference between OOP and POP.
3. To be able to program using various C++ features such as operator overloading, dynamic memory allocation, inheritance, polymorphism, exception handling and templates.
4. To be able to build C++ classes using appropriate encapsulation and design principles.
5. To improve the problem solving skills by applying OOP techniques to solve bigger computing problems.

**Course Details:**

## Unit 1: Introduction (06 Hrs)

**U1.1.** Introduction to object oriented programming, Object Oriented Programming paradigm. Basic concepts of object oriented programming, i.e. Object, Class, polymorphism, encapsulation, data abstraction, inheritance, data hiding and message passing.

Getting started with C++ syntax, Input and Output in C++, C++ tokens: Keywords, Identifiers, Constants, Operators. Data-types: user-defined & derived data-types, Reference variables, Dynamic initialization of variables, Special operators in C++ (i.e. scope resolution, new, delete and other operators), Pointers. Functions: Call by reference, Default parameter values in functions, Inline functions.

**U1.2.** Self Study: Control structures, arrays, functions returning reference, Manipulators, Operator precedence.

## Unit 2: Class & Object: (06Hrs)

**U2.1.** Abstraction mechanism: Difference between structure and class, Specifying a class, access specifiers, data members, member functions, array of objects, static members, friend functions, constructors, destructors.

**U2.2.** Self Study: Making an outside function inline, Constructor with default arguments, constant member functions.

**Unit 3: Inheritance: (06 Hrs)**

**U3.1.** Inheritance: Class hierarchy, derived classes, single inheritance, multiple, multilevel, hierarchical, hybrid inheritance, role of virtual base class, constructor and destructor in inheritance.

**U3.2.** Self Study: Initialization list in constructors, Delegation, Nested classes.

**Unit 4: Polymorphism: (06 Hrs)**

**U4.1.** Polymorphism: Binding, Static binding, Dynamic binding. Function Overloading, Ambiguity in function overloading. Operator Overloading: Operator function, member and non member operator function, type conversion.

Dynamic polymorphism: Base class pointer, object slicing, late binding, method overriding with virtual functions, pure virtual functions, abstract classes.

**U4.2.** Self Study: this pointer, applications of this pointer, overloading I/O operators, virtual destructors, typeid operator.

**Unit 5: Exception Handling, Template and Namespace (06 Hrs)**

**U5.1.** Exception handling Mechanism: use of try, throw and catch. Generic catch, rethrowing an exception, Specifying exceptions for a function.

Template: Class templates, Function templates, Overloading a function template.

Namespaces: user defined namespaces, namespaces provided by library.

**U5.2.** Self Study: Exception in inheritance, Non-type Template parameter, Standard Template Library.

**Note:** Five assignments to be given to the students on self study, comprising of one assignment from
each unit.

**Text Books:**
1. Object Oriented Programming with C++ by E. Balagurusamy, McGraw-Hill Education (India).
2. ANSI and Turbo C++ by Ashoke N. Kamthane, Pearson Education

**Reference Books:**
1. Big C++ - Wiley India
2. C++: The Complete Reference- Schildt, McGraw-Hill Education (India)
3. C++ and Object Oriented Programming – Jana, PHI Learning.

4. Object Oriented Programming with C++ - Rajiv Sahay, Oxford
5. Mastering C++ - Venugopal, McGraw-Hill Education (India)

**Course Outcomes:**

**Upon completion of the course, graduates will be able to –**

1. Identify the principles and practices of object oriented analysis.

2. Apply various object oriented features like inheritance, encapsulation and polymorphism to solve various computing problems using C++ language.

3. Apply concepts of operator overloading, constructors and destructors.

4. Use dynamic memory management, object copying, assignment operator etc to formulate and solve complex problems.

5. Apply exception handling and templates to write application program