SIXTH SEMESTER IT 2018-19 (PATTERN B-16)

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, Security Mechanism, Relationship between Security Mechanism and Services Types of security attacks. Firewalls: Types of Firewalls, Limitations of firewall.

U1.2 Self Study: Computer related privacy problems

UNIT 2: Cryptography and Its Application (8hrs)

U2.1 Principles of Security, Cryptographic Techniques: Plain Text and Cipher Text, Substitution Techniques, Transposition Techniques, Encryption and Decryption, Symmetric and Asymmetric Key Cryptography, Steganography, Key Range and Key Size, Possible Types of Attacks. An overview of Symmetric Key Cryptography DES**,** An overview of Asymmetric Key Cryptography, The RSA Algorithm.

U2.2 Self Study: Case study on different Crypto Systems

UNIT 3: Intrusion detection and prevention (8hrs)

U3.1 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. Spoof Prevention, Dos, Qos Policy.

U3.2 Self Study: Intrusion prevention system basics, Limitations of Intrusion Prevention System.

UNIT 4: Privacy Concepts (6hrs)

U4.1 Definition of privacy, Data Privacy/Information Privacy: Internet Privacy, Risks to internet privacy (HTTP cookies, Flash Cookies, Evercookies, Device fingerprinting), other potential Internet privacy risks (Malware, Spyware, Phishing etc.), Types of privacy, Privacy Paradox. Global privacy policies, Data protection regulation with some case studies.

U4.2 Self Study: case study on data protection regulation (pick any country)

UNIT 5: Virtual private network (8hrs)

U5.1 Virtual Private Networks, Deploy and managing VPN, VPN Performance tuning and error handling, VPN routing, DMZ and virtual host, Reverse proxy, Web application Security Risks, Identifying the Application Security Risks, Open Web Application Security Project (OWASP).

U5.2 Self Study: Top 10 Concepts Invalidated Redirects and Forwards, 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. Network Security Bible, Eric cole, Wiley India. ISBN-9788126523313

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.
3. B. Menezes,"Network Security and Cryptography", Cengage Learning, 1st ed.2010.
4. Other WEB resources.

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.

SIXTH SEMESTER IT 2018-19 (PATTERN B-16)

COURSE CODE: CS30110 **REF NO:**

Machine Learning

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

Prerequisites:

1. Knowledge on Computer programming
2. Algorithm design
3. basics of probability & statistics

Objectives:

1. To let the students understand the basic concepts of machine learning.

Course Details:

Unit 1

Title- Introduction to Machine Learning: (06 Hrs)

U1.1 Introduction: Introduction to Machine learning system, machine learning Basic definitions, types of learning, Examples of machine learning applications: Learning Associations, Classification, Regression, hypothesis space and inductive bias, Evaluation.

U1.2 Basic Mathematical and Statistical concepts: Metric, Matrices, Eigen values and Eigen vectors, mean, median, mode, variance, co-variance, correlation, dispersion matrix, Basic concepts in probability theory such as Bayes theorem, Error risk minimization.

U1.3. Self Study: Study of Binomial distribution and normal distribution

Unit 2

Title – Supervised learning (08Hrs)

U2.1 Supervised learning setup (Training, Testing). Minimum distance classifier, k-nearest neighbour classifier, density estimation. Linear regression. Logistic regression. Perceptrons (single layer / multi-layer). Model selection, dimensionality reduction, and feature selection.
Measuring the accuracy of learned hypotheses. Comparing learning algorithms: cross-validation, Learning curves and statistical hypothesis testing.

U2.2. Self Study: a case study on the use of supervised learning algorithm for a classification problem

Unit 3

Title – Unsupervised learning (08 Hrs)

U3.1 Clustering. Similarity measures, K-means algorithm, Hierarchical clustering, Density based clustering, anomaly detection, cluster validation Expectation Maximization. Mixture of Gaussians. Factor analysis. PCA (Principal components analysis). ICA (Independent components analysis).

U3.2. Self Study: A case study on k nearest neighborhood based clustering problem

Unit 4

Title – Learning theory (08 Hrs)

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. Support vector machines.

U4.2. Self Study: Study of different kernel functions

Unit 5

Title- Recent techniques (08 Hrs)

U5.1 Deep Learning, Recurrent NNs, Decision trees, Random forests, Semi-supervised and active learning; Reinforcement learning, kernel functions, one class classifier, ensemble learning: bagging, boosting.

U5.2. Self Study: Exploring and using some open source deep learning libraries for designing a simple classifier

Text Books:

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

Reference Books:

1. "Machine Learning in Python, Essential Techniques in Predictive Analytics", Michel Bowles, Wiley

Course Outcomes:

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

1. Identify different components of machine learning system and analyze the mathematical theories and concepts behind machine learning.
2. Understand the design and modeling of supervised machine learning techniques and compare different learning algorithm validation methods.
3. Learn and implement unsupervised machine learning models and their applications.
4. Gain idea about inductive and deductive learning theories and support vector machine.
5. Explore different recent machine techniques.

SIXTH SEMESTER IT 2018-19 (PATTERN B-16)

COURSE CODE: CS30310 **REF NO:**

Machine Learning Lab

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

Prerequisites:

1. Knowledge on Computer programming
2. Algorithm design
3. basics of probability & statistics

Objectives:

1. To let the students solve problems using machine learning methods by developing programs in a suitable programming language. (Python, R. WEKA, MATLAB)

List of Experiments:

1. Downloading, Installing and getting started with Python (Exp #1)
2. Getting familiarity with Python syntax.
3. Introduction to python open source libraries for machine learning (scipy, numpy, matplotlib, pandas, sklearn) (Exp #2)
4. Loading and understanding data. (Exp #3)
5. Finding out statistical summary of the data. (Exp #4)
6. Visualize data with univariate and multivariate plots. (Exp #5)
7. Building the models for the following algorithms: Logistic Regression (LR)v model, Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), Classification and Regression Trees (CART), Gaussian Naive Bayes (NB), Support Vector Machines (SVM). (Exp #6, 7)
8. Evaluating the build models. (Exp #8,9)
9. Compare all algorithms and models. (Exp #10)

Note: The above experiments can be done using **Python** or alternate software such as **R, WEKA, MATLAB.** Item numbers 1 to 3 will be modified accordingly.

Text Books:

1. "Machine Learning Mastery with Python", Jason Brownlee, 2016/2017. Copyright Jason Brownlee. All Rights Reserved, Edition: v1.2, V1.4, http://MachineLearningMastery.com
2. "Mastering Machine Learning with scikit-learn", Gavin Hackeling, PACKT publishing, ISBN 978-1-78398-836-5, Open Source, www.it-ebooks.info
3. Python Machine Learning, Sebastian Raschka, PACKT publishing, ISBN 978-1-78355-513-0, Open Source, www.it-ebooks.info

Reference Books:

1. "Introduction to computing & problem solving with Python" Dr. Jeeva Jose , Khanna Publishers
2. *"Taming Python by Programming*", Dr.Jeeva Jose, byKhanna Publishers.

Course Outcomes:

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

1. Install and set-up machine learning language environment.
2. Gain familiarity on the features of the language including data transfer, display and graphics, etc.
3. Computing statistical estimates of data.
4. Solving problems using ML techniques: Logistic Regression (LR)v model, Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), Classification and Regression Trees (CART), Gaussian Naive Bayes (NB), Support Vector Machines (SVM).
5. Evaluation and Comparison of different models on data.

COURSE CODE: CS30103 **REF NO: To be filled by CD office**

Big Data Analysis

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

Prerequisites:

1. Computer Programming.
2. Linux OS
3. Design and Analysis of Algorithms

Objectives:

1. Master the concepts of HDFS and Map Reduce framework
2. Understand Hadoop 2.x Architecture
3. Setup Hadoop Cluster and write Complex Map Reduce programs
4. Perform data analytics using Pig, Hive and others
5. Implement HBase and Map Reduce 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 to Big Data Hadoop: Introduction, Different types of Components in Hadoop

HDFS, Map Reduce, PIG, and 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.

U2.2. 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 Map Reduce Job, File Input/output Formats in Map Reduce 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 Input Split, Role of Record Reader

Unit 4
Title – PIG: (06 Hrs)

U4.1
Introduction to Apache PIG, Introduction to PIG Data Flow Engine, Map Reduce VS PIG in detail, When should PIG be used, 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 Map Reduce Mode, Execution Mechanisms, Grunt Shell, Script Embedded, Operators/ Transformations in PIG, PIG UDF's with Program, The difference between the Map Reduce 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

SIXTH SEMESTER IT 2018-19 (PATTERN B-16)

Course Code: CS30103 **Ref No.: To be filled by office**

Big Data Analysis - Tutorial

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

Prerequisites:

1. Computer Programming
2. Linux OS
3. OOPS Concepts

Objectives:

1. Understanding the basics of Big Data Analytics.
2. Understanding the concepts of Hadoop Ecosystem.
3. Understanding the concepts of HDFS and Map Reduce Framework.
4. Perform Read/Write Operations using Map Reduce Programming.
5. Writing Pig Scripts.

Course Details:

List of Contents

Problems and exercises on the following topics:

Tutorial No.1: Characteristics of Big Data and its real life applications
Tutorial No. 2: Hadoop Ecosystem.
Tutorial No. 3: HDFS and its role in Hadoop.
Tutorial No.4: Read/Write Operations in HDFS.
Tutorial No.5: Map Reduce Fundamentals: Basics, Architecture, Data flow, Configuration
Tutorial No.6: Map Reduce Program: Word Count Example.
Tutorial No.7: File Input/Output Formats: Text Input Format, Key Value Input Format, Sequence File Input Format, NLine Input Format
Tutorial No.8: Map Reduce Joins: Mapper Join, Reducer Join
Tutorial No. 9: Pig: Features, Map Reduce vs Pig, Data Types, Basic Pig Programming
Tutorial No.10: Modes of Execution in Pig, UDF's in Pig.

Text Books:

T1. Hadoop: The Definitive Guide, 4th Edition, Storage and Analysis at Internet Scale By:Tom White.
T2. Hadoop Operations, By: Eric Sammer, Publisher: O'Reilly Media, Print ISBN: 978-1-4493-2705-7 | ISBN 10: 1-4493-2705-2

Reference Books:

R1. Instant MapReduce Patterns - Hadoop Essentials How-to, By: Srinath Perera, Publisher: Packt Publishing Limited, Language: English, ISBN-10: 1782167706
R2. 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: Apply Hadoop Distributed File System (HDFS)
CO3: Develop Map Reduce programs
CO4: Develop Pig scripts

SIXTH SEMESTER IT 2018-19 (PATTERN B-16)

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:

1. Master the concepts of HDFS and MapReduce framework
2. Setup Hadoop Cluster and write Complex MapReduce programs
3. Perform data analytics using Pig, Hive and others
4. Implement HBase and MapReduce integration
5. Implement best practices for Hadoop development
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 Experiments:

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. Use of 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

SIXTH SEMESTER IT 2018-19 (PATTERN B-16)

COURSE CODE: CS30114 **REF NO: To be filled by CD office**

Advanced Computer Architecture

Credits:3 **Teaching Scheme: - Theory**

Prerequisites:

1. Digital Electronic Circuit.
2. Computer Organization.

Objectives:

1. Ability to describe the operation of modern and high performance computers.
2. Analyze the techniques to design high performance computer.
3. Explain the concepts of different memory technologies.

Course Details:

Unit1: Fundamentals of Computer Design (7 Hrs)

U1.1: Review of basic computer architecture, Quantitative principles in computer design, measuring and reporting performance consideration, Amdahl's Law, Instruction set architecture, Classifying Instruction set Architecture, CISC and RISC processors.

U1.2Self Study: Performance of a computer, Von Neumann Machine Architecture.

Unit2: Pipelining (7 Hrs)

U2.1. Pipelining fundamentals, Arithmetic and instruction pipelining, major hurdles in pipelining: data hazards, control hazards, and structural hazards, Performance of pipeline with stalls, Techniques for overcoming or reducing the effects of various hazards.

U2.2. Self Study: Case study of MIPS architecture

Unit3: Memory Organization (7 Hrs)

U3.1 Inclusion, Coherence and locality properties, Cache memory organizations, Multi-level caches, Data and Instruction caches, Techniques for Cache optimization, Virtual memory organization, mapping and management techniques, memory replacement policies .

U3.2 Self Study: Types of storage devices: RAID – Reliability, availability and dependability

Unit4: Instruction-level parallelism: (7 Hrs)

U4.1. Concepts of instruction-level parallelism (ILP), Techniques for increasing ILP, Dynamic Scheduling, Superscalar, super-pipelined and VLIW processor architectures; Array and vector processors.

U4.2. Self Study: Case Study: SPARC and ARM processors.

Unit5: Multiprocessor Architecture (7 Hrs)

U5.1. Taxonomy of parallel architectures; Flynn's Classification, Centralized shared-

memory architecture: synchronization, memory consistency, interconnection networks. Distributed shared-memory architecture. Cluster computers, Massively Parallel Processors: Graphics Processors, GPU, GP-GPU.

U5.2. Self Study: Hyper threading, CUDA

Text Books

T1. "Computer Organization and Design", David A. Patterson and John L. Hennessy, Elsevier, Fourth Edition.
T2. "Computer Architecture, A Quantitative Approach", John L Hennessy and David A. Patterson, Elsevier, Fourth Edition.
T3. "Programming Massively Parallel Processors A Hands on Approach", David B. Kirk and Wen-mei W. Hwu , Morgan Keifmann, Elsevier

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.
R3. "Using MPI: Portable Parallel Programming with the Message-Passing Interface", William Gropp, Ewing Lusk, Anthony Skjellum, 3rd Edition, MIT Press
R4. "Introduction to Parallel Computing", Second Edition, Ananth Gram, Anshul Gupta, George Karypis, Vipin Kumar, Pearson Education.
R5. "Parallel computing Theory and Practice", Second Edition, Michael J. Quinn, TMH.

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

CO1: Analyze the basic principles of a computer design and the various performance measures.
CO2: Understand pipelining, its speed advantage and techniques to reduce hazard.
CO3: Evaluate the components and operation of a memory hierarchy and the performance issues
influencing its design.
CO4: Understand the various techniques to enhance a processors ability to exploit Instruction-level parallelism (ILP), and its challenges.
CO5**:** Analyze the organization and operation of current generation parallel computer systems, including multiprocessor and cluster computers.

COURSE CODE: CS30114 **REF NO: To be filled by CD office**

Advanced Computer Architecture Tutorial

Credits: 1 **Teaching Scheme: - Tutorial**

Prerequisites:

1. Digital Electronic Circuit.
2. Computer Organization.

Objectives:

1. Ability to describe the operation of modern and high performance computers.
2. Analyze the techniques to design high performance computer.
3. Explain the concepts of different memory technologies.

Course Details:

Tutorial No.1: Problems on Measuring performance, Amdahl's Law.

Tutorial No.2: Instruction set architecture

Tutorial No.3: Arithmetic and instruction pipelining

Tutorial No.4: Pipeline Hazards.

Tutorial No.5: Problems on Multi level cache memory .

Tutorial No.6: Problems on cache memory optimization.

Tutorial No.7: Instruction-level parallelism.

Tutorial No.8: Array and vector processors.

Tutorial No.9: Centralized shared- memory architecture: synchronization, memory consistency.

Tutorial No.10: Problems on interconnection networks.

Text Books

T1. "Computer Organization and Design", David A. Patterson and John L. Hennessy, Elsevier, Fourth Edition.
T2. "Computer Architecture, A Quantitative Approach", John L Hennessy and David A. Patterson, Elsevier, Fourth Edition.
T3. "Programming Massively Parallel Processors A Hands on Approach", David B. Kirk and Wen-mei W. Hwu , Morgan Keifmann, Elsevier

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.
R3. "Using MPI: Portable Parallel Programming with the Message-Passing Interface", William Gropp, Ewing Lusk, Anthony Skjellum, 3rd Edition, MIT Press
R4. "Introduction to Parallel Computing", Second Edition, Ananth Gram, Anshul Gupta, George Karypis, Vipin Kumar, Pearson Education.
R5. "Parallel computing Theory and Practice", Second Edition, Michael J. Quinn, TMH.

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

CO1: Analyze the basic principles of a computer design and the various performance measures.

CO2: Understand pipelining, its speed advantage and techniques to reduce hazard.

CO3: Evaluate the components and operation of a memory hierarchy and the performance issues influencing its design.

CO4: Understand the various techniques to enhance a processors ability to exploit Instruction-level parallelism (ILP), and its challenges.

CO5**:** Analyze the organization and operation of current generation parallel computer systems, including multiprocessor and cluster computers.

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 (7 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 (6 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

T1. "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.
T2. "Engineering a Compiler", K. Cooper, L. Torozon, Morgan Kaufmann, ISBN 1-55860-698-X, First Edition, 2003.

Reference Books
R1. "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.
R2. "Compiler Construction: Principles and Practice", K. Louden, Course Technology, ISBN 0-534-93972-4, First Edition, 1997.

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

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

SIXTH SEMESTER IT 2018-19 (PATTERN B-16)

COURSE CODE: IT30307 **REF NO: To be filled by CD office**

Parallel Computing Lab

Credits: 1 **Teaching Scheme: Laboratory (2Hrs /Week)**

Prerequisites:

1. Computer Organization
2. Data Communications and Computer Networks
3. Operating Systems
4. Computer programming

Objectives:

1. To provide students with contemporary knowledge in parallel computing.
2. To equip students with skills to design and analyze parallel algorithms in different applications.
3. To understand, appreciate and apply parallel algorithms in problem solving.
4. To gain knowledge on how to improve performance metrics using parallel programs.

Course Details / List of Practicals:

1. Parallel GPU implementation of addition up to 'n' numbers, vector-vector operations, vector-Matrix operations

2. Parallel computation of binomial coefficient matrix, Matrix-Matrix operations

3. Assignment focusing on optimization of data transfer between CPU and

   GPU: using page locked host memory and to avoid the data transfer

4. Assignment focusing on memory optimization: use of GPU shared, constant and texture memory.

5. Parallel GPU implementation involving kernel looping.

6. Parallel computation of set of multi-indices on GPU.

7. Parallel optimization of algorithms on OpenAcc.

8. Parallel implementation using MPI, OpenMP

9. Exercise using parallel compilers / Open CL.

Text books:

1. "CUDA by Example: An Introduction to General-Purpose GPU Programming", by Jason Sanders, Edward Kandrot
2. "Programming Massively Parallel Processors", by Kirk & Hwu,2nd edition,ISBN:9780124159921
3. "Introduction to Parallel Computing", by Ananth Grama, George Karypis , Vipin Kumar, Anshul Gupta , second edition, Addison Wesley, 2003, ISBN: 0201648652

Reference Books:

1. " Introduction to Parallel Processing: Algorithms and Architectures", by Behrooz Parhami
2. " Computer Architecture and Parallel Processing", by Kai Hwang , Faye A. Briggs

Course Outcomes:

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

1. Design efficient parallel algorithms and applications.
2. Write parallel programs for large scale parallel systems, with distributed and shared address space, and GP-GPU platforms.
3. Analyze performance of parallel programs.
4. Implement parallel computing algorithms using alternate methods (parallel compilers, Open CL etc)

COURSE CODE: IT34354 **REF NO: To be filled by CD office**

Advanced Programming Lab

Credit: 01 **Teaching Scheme:** Laboratory 02 Hrs / Week

Prerequisites:

1. Computer Programming.
2. Object Oriented Programming Concepts

Objectives:

1. To make the student familiar with various advanced topics in C programming.
2. To teach the student to write programs in C++ to solve the problems.

Course Details:

1. Study of Structures & Unions
2. Study of Pointer – Dereferencing, Pointers & Arrays, Functions & Pointers
3. Study of Dynamic Memory Allocations - malloc, calloc, realloc, free
4. Study of Preprocessor - Conditional Compilation, Nested Macros
5. Study of class and objects in C++.
6. Study of friend function and friend class.
7. Study of various types of inheritance.
8. Study of static polymorphism
9. Study of dynamic polymorphism with virtual functions.
10. Study of exception handling mechanism.
11. Study of generic programming using templates.
12. Study of Standard Template Library.

List of Practicals:

Experiment No. 1: Program to understand Structures & Unions

a. Program to create structures
b. Structure within structure
c. Use of unions

Experiment No. 2: Program to understand Pointer

a. Program to use pointer with primitive variables
b. Pointer with array
c. Pointer with function- call by address & function pointer
d. Pointer with structure.

Experiment No. 3: Program to understand Dynamic Memory Allocations

a. Program to use malloc()
b. Program to use calloc()
c. Program to use realloc()

d. Program to use free()

Experiment No. 4: Program to understand Preprocessor

a. Program to use #define statements
b. Program to use Conditional Compilation
c. Program to use Nested Macros
d. Program to use Multiline Macros

Experiment No. 5: Program to understand class and object.

a. Program to differentiate between structure and class
b. Defining member functions inside and outside the class.
c. Program to implement array of objects.
d. Implementation of static data members and member functions.

Experiment No. 6: Programs to Understand Friend Function & Friend Class.

a. Friend Function
b. Friend class

Experiment No. 7: Program to implement constructors and destructors.

Experiment No. 8: Programs to Implement Inheritance

a. Single Inheritance (private and public mode derivation)
b. Multiple inheritances
c. Hierarchical inheritance
d. Multilevel Inheritance
e. Multipath Inheritance

Experiment No. 9: Understanding static Polymorphism

a. Program to implement function overloading and its ambiguity.
b. overloading unary operator as member and non member function.
c. Programs to Overload Binary Operators as member and non member function.

Experiment No. 10: Program to implement type conversion techniques.

a. basic to class type
b. class to basic type
c. one class type to another class type

Experiment No. 11: Program to implement dynamic polymorphism

a. Implementation of function overriding with virtual function b. Use of this pointer

Experiment No. 12: a. Program to implement exception handling mechanism
b. Programs on Class Templates and Function Templates.

Text Books:

1. C in Depth by Deepali Srivastava, S. K. Srivastava, BPB Publications
2. Object Oriented Programming with C++ by E. Balagurusamy, McGraw-Hill Education (India)
3. ANSI and Turbo C++ by Ashoke N. Kamthane, Pearson Education

Reference Books:

1. Exploring C by Yashavant Kanetkar, BPB Publications
2. Object Oriented Programming with C++ - Rajiv Sahay, Oxford
3. Mastering C++, Venugopal, McGraw-Hill Education (India)
4. C++: The Complete Reference- Schildt, McGraw-Hill Education (India)

Course Outcomes:

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

1. Identify the usage of various feature such as Structure, Pointer of C programming language
2. Identify classes, methods, and libraries of object-oriented programming using C++.
3. Design and implement a class based on attributes and behavior of objects.
4. Design solutions for practical problems based on Inheritance and Polymorphic behavior of Objects.
5. Analyze and design C++ programs using advanced features such as exception handling, Template and STL.