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.

Pattern A15 **SIXTH SEMESTER IT 2017-18** 04 Nov 2017

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.

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 soft ware 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: **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, Multiprogrammed, 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.

Pattern A15 **SIXTH SEMESTER IT 2017-18** 04 Nov 2017

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

Operating System Tutorial

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

Advanced Computer Architecture

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

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** 1 Hrs/Week

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.

Pattern A15 **SIXTH SEMESTER IT 2017-18** 04 Nov 2017

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.

COURSE CODE: IT37398 **REF NO: To be filled by CD office**

Major Project Stage I

Credit: 02 **Teaching Scheme:** Sesional 04 Hrs / Week

Prerequisites:

- Completion of departmental courses till 5th semester

Objectives:

3. To make the student design and implement IT solution to real life like problem.
4. To make students learn new technology and tools as required implementing the solution.

Course Details:

Self study and guidance from faculty member guide.

Course Outcomes:

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

COURSE CODE: IT37402 **REF NO: To be filled by CD office**

Comprehensive Viva Voce

Credit: 01 **Teaching Scheme:** Sessional

Prerequisites:

- Completion of departmental courses till 5th semester

Objectives:

5. To make the fundamental concepts strong on all departmental subjects taught
6. To make students familiar with answering technical questions

Course Details:

Self study on the departmental courses taught during the semester.

Course Outcomes:

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

Pattern A15 **SIXTH SEMESTER IT 2017-18** 04 Nov 2017

COURSE CODE: IN33301 **REF NO: To be filled by CD office**

Professional Training

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

Pre-requisite: English Communication Skills, Fundamentals of computer programming and language, basic knowledge on life skills

COs:

CO1: Acquire knowledge on Communication and Interpersonal skills

CO2: Develop analytical and technical skills and leadership quality

Two of the following modules shall be covered during the semester for all branches

Module 1: Exercises on *Verbal Ability with presentation*: 15 hours

Verbal simulation practices, Reading comprehension, Articles, Prepositions & Its functionalities & Placement Exercises', Subject Verb Agreement and Parts of Speech – Functionalities', Grammar and Tenses –Functionalities' , Errors and Common Mistakes –Usages

Module 2: Exercises on *Quantitative Aptitude:* 15 hours

Number theory, Divisibility Rules, Number Properties, Averages, Problems on ages, Time and Work, Pipes and Cisterns , Man Days' Concept, Ratio Proportion, Allegation, Mixture, Percentages, Unit digit problems, SI & CI ,Profit, Loss, Discount, Successive Discount Techniques, Time , Speed, Distance, Train Problems, Boats and Streams

Module 3: Exercises on *Logical and Verbal Reasoning:* 15 hours

Syllogisms, Cubes, Number Series, Directions, Blood Relations, Seating Arrangements (Linear and Circular), Venn Diagrams, Data Sufficiency & Data Interpretation -Pie, Tabular and Bar Models, Clocks, Calendar, Coding, Decoding, Alphabet Series, Crypto Arithmetic Patterns and Strategic Approaches

Module 4: Exercises on *Technical Training/Programming Skills:* 15 hours

Basic coding practices using programming languages C, C++ and JAVA, Practices on One database (Oracle), Practices on one operating system

Module 5: Exercises on *Group Discussion and Personal Interview:* 15 hours

Group Discussion: Many-on-many discussion on current topics with special focus on team work, reasoning ability, leadership, initiative, listening and awareness, assertiveness. Tips on Body Language, Eye Contact, Positive Gestures during group discussion; Resume and CV Preparation; Mock Personal Interview- Real time Interview with focus on both common interview questions and frequently asked technical questions. Real time experience on HR and MR Mock sessions with emphasis on do's and don'ts of interview, frequently asked HR sessions, Tips to handle HR and MR round, Strategies to break managerial trivia

Assessment Scheme

Parameters of evaluation	Marks
Formative/Continuous Evaluation in Each Unit	**70**
Summative Assessment	**30**
Total	**100**

Pass Marks in this subject is 50% percentage point.