

Course Structure for SE Computer Engineering
2012 Course (w.e.f. June 2013)

Subject Code	Subject	Teaching Scheme Hrs/Week			Examination Scheme					Mark
		Lect.	Tutorials	Pract	Paper	Tw	Pr	OR	Online	Total
SEM – I										
210241	Discrete Structures	4	—	—	50	—	—	—	50	100
210242	Data Structures and Problem Solving	4	—	4	50	—	50	—	50	150
210243	Digital Electronics and Logic Design	3	—	2	50	25	—	—	50	125
210244	Operating System and Administration	3	—	2	50	25	50	—	50	175
210245	Microprocessor Architecture	3	—	2	50	25	—	50	50	175
210246	Soft Skills	1	—	2	—	25	—	—	—	025
	Total of Semester – I	18	—	12	250	100	100	50	250	750
SEM – II										
Subject Code	Subject	Teaching Scheme Hrs/Week			Examination Scheme					Mark
		Lect.	Tutorials	Pract	Paper	Tw	Pr	Or	Online	Total
207003	Engineering Maths - III	4	1	—	50	25	—	—	50	125
210247	Object Oriented and multi-core Programming	4	—	4	50	25	50	—	50	175
210248	Microprocessors and Interfacing Techniques	3	—	4	50	—	50	—	50	150
210249	Computer Graphics and Gaming	3	—	—	50	—	—	—	50	100
210250	Computer Organization	3	—	—	50	—	—	—	50	100
210251	Programming Laboratory	—	—	4	—	50	—	50	—	100
	Total of Semester – II	17	1	12	250	100	100	50	250	750

210241 DISCRETE STRUCTURES

Teaching Scheme
Lectures: 4 Hrs/week

Examination Scheme
Theory: 50 Marks
Online: 50 Marks

Discrete mathematics- The mathematics of integers and of collections of object underlies the operation of digital computer, and is used widely in all fields of computer science for reasoning about data structures algorithms and complexity. The primary objective of subject is to prepare students mathematically for the study of computer engineering. Topics covered in the course include proof techniques, logic and sets, functions, relations, counting techniques, probability and recurrences.

Prerequisite: Basic Mathematics

Program Educational Objectives: ... *the student will be able to*

- *Use appropriate set, function, or relation models to analyze practical examples, interpret the associated operations and terminology in context.*
- *Determine number of logical possibilities and probability of events*
- *Learn logic and proof techniques to expand mathematical maturity*
- *Formulate problems precisely, solve the problems, apply formal proof techniques, and explain their reasoning clearly.*

Program Educational Outcomes:

By the end of the course, students should be able to formulate problems precisely, solve the problems, apply formal proof techniques, and explain their reasoning clearly. Illustrate by example, basic terminology and model problems in computer engineering using graphs and trees

Unit I	Sets and Propositions	(8 Hrs)
	Sets, Combination of sets, Finite and Infinite sets, Un-count-ably infinite sets, Principle of inclusion and exclusion, multi-sets. Propositions, Conditional Propositions, Logical Connectivity, Prepositional calculus, Universal and Existential Quantifiers, Normal forms, methods of proofs, Mathematical Induction.	
Unit II	Relations and Functions	(8 Hrs)
	Properties of Binary Relations, Closure of relations, Warshall's algorithm, Equivalence relations and partitions, Partial ordering relations and lattices, Chains and Anti chains. Functions, Composition of functions, Invertible functions, Pigeonhole Principle, Discrete numeric functions and Generating functions, Job scheduling Problem. Recurrence Relation, Linear Recurrence Relations With constant Coefficients, Homogeneous Solutions.	
Unit III	Groups and Rings	(8 Hrs)
	Algebraic Systems, Groups, Semi Groups, Monoids, Subgroups, Permutation Groups, Codes and Group codes, Isomorphism and Automorphisms, Homomorphism and Normal Subgroups, Ring, Integral Domain, Field, Ring Homomorphism, Polynomial Rings and Cyclic Codes	
Unit IV	Graph Theory	(8 Hrs)
	Basic terminology, representation of a graph in computer memory, multi-graphs and weighted graphs, Subgraphs, Isomorphic graphs, Complete, regular and bipartite graphs, operations on graph, paths and circuits, Hamiltonian and Euler paths and circuits, shortest path in weighted graphs (Dijkstra's algorithm), factors of a graph, planer graph and Traveling salesman problem, Graph Coloring.	
Unit V	Trees	(8 Hrs)
	Basic terminology and characterization of trees, Prefix codes and optimal prefix codes, binary search trees, Tree traversal, Spanning trees, Fundamental Trees and cut sets, Minimal Spanning trees, Kruskal's and Prim's algorithms for minimal spanning trees, The Max flow-Min Cut Theorem (Transport network).	
Unit VI	Permutations, Combinations and Discrete Probability	(8 Hrs)

	Permutations and Combinations: rule of sum and product, Permutations, Combinations, Algorithms for generation of Permutations and Combinations. Discrete Probability, Conditional Probability, Bayes' Theorem, Information and Mutual Information	
--	--	--

Text Books	
1.	C. L. Liu and D. P. Mohapatra, "Elements of Discrete Mathematics", SiE Edition, TataMcGraw-Hill, 2008, ISBN 10:0-07-066913-9
2.	R. Johnsonbaugh, "Discrete Mathematics", 5 th Edition, Pearson Education, 2001 ISBN 81 – 7808 – 279 - 9 (Recommended for Unit I and Unit II)
Reference Books	
1.	N. Biggs, "Discrete Mathematics", 3 rd Edition, Oxford University Press, ISBN 0 –19 –850717 – 8
2.	Kenneth H. Rosen, "Discrete Mathematics and its Applications", 6 th edition, McGraw-Hill, 2007. ISBN 978-0-07-288008-3
3.	E. Goodaire and M. Parmenter, "Discrete Mathematics with Graph Theory", 2 nd edition, Pearson Education, 2003 ISBN 81 – 7808 – 827 – 4
4.	Semyour Lipschutz & Marc Lipson, " Discrete Mathematics", McGraw-Hill, 3 rd Special Indian Edition, ISBN-13 : 978-0-07-060174-1
5.	B. Kolman, R. Busby and S. Ross, "Discrete Mathematical Structures", 4 th Edition, Pearson Education, 2002, ISBN 81-7808-556-9
6.	N. Deo, "Graph Theory with application to Engineering and Computer Science", Prentice Hall of India, 1990, 0 – 87692 – 145 – 4

210242 DATA STRUCTURES & PROBLEM SOLVING

Teaching Scheme
Lectures: 4 Hrs/week
Practical: 4 Hrs/week

Examination Scheme
Theory: 50 Marks
OnLine: 50 Marks
Practical: 50 Marks

Prerequisites: FPL –I and FPL-II

Course Objectives:

- To develop ability to understand problem and select relevant data structures
- To develop abilities to understand algorithmic requirements
- To develop ability to understand data structures using OOP concepts
- To develop ability to use data structures effectively in concurrent algorithms
- To develop ability to use different programming environments

Course Outcomes:

- Effective selection and use of data structures while problem solving and programming
- Effective use of algorithmic foundations while problem solving and programming
- Effective use of OOP in data structures and files in programming
- Effective use of multi-core programming architecture in programming
- Effective using of latest programming tools

Unit I	General Problem Solving Concepts	(8 Hrs)
	Types of problems, problems solving with computers, difficulties with problem solving, Problem Solving Aspects, Problem Solving Concepts for computer-constants and variables, data types, functions, operators, expressions and equations, Programming Concepts – communicating with computers, organizing the problem, using the tools, testing the solution, coding the program, Top down design, Analysis of algorithm, time & space complexity calculation, Sorting, stability in sorting, internal & External Sorting methods(to be implemented in practical session Quick sort, merge sort, shell sort, radix sort)	
Unit II	Programming using Trees and Graphs	(8 Hrs)
	Concept of stack & Queue data structure, Types of Queue, Programmer’s perspective of a tree, binary tree and its properties, representation using sequential and linked organization, Full and complete binary trees, Algorithm for converting tree to binary tree, Binary tree traversal: BFS, DFS (non-recursive and recursive), infix, prefix, postfix, Huffman’s codes. Binary search trees and operations: BST as ADT, Threaded Binary Tree, Insertion and deletion of nodes in a threaded binary tree, pre-order, in-order and post-order traversal of threaded binary tree, Algorithmic applications of binary trees: Gaming, Expression and decision trees, Analysis of algorithms	
Unit III	Graphs	(6 Hrs)
	Programmers perspective of graphs, Graph operations, storage structure, Traversal: Breadth first, depth first graph algorithms, Graph as an ADT, Applications of graph: GIS based, Analysis of algorithms, spanning Trees, Connected Components.	
Unit IV	Tables	(6 Hrs)
	Symbol Tables: Static and dynamic tree tables, AVL trees, AVL Tree implementation, Algorithms and analysis of AVL Tree Hash Tables: Basic Concepts, Hash Function, Hashing methods, Collision resolution, Bucket hashing, Dynamic Hashing.	
Unit V	Heaps and Multi-way Trees	(6 Hrs)
	Heaps: Basic Concepts, algorithmic implementation, Heap as ADT, Heap Sort, Heap	

	Applications Multi-way Trees: B Tree Implementation, Introduction to B+, B*, External Storage devices. Files: Definition and Concepts, Sequential files.	
Unit VI	Data Structures in parallel algorithms	(6 Hrs)
	Computational Model, Basic techniques and Algorithms: Complete Binary Tree, Pointer doubling, Prefix Computation, Selection, Merging, Case study of ODD-Even Parallel Merge Sorting	

Text Books	
1.	“Problem Solving and Programming Concepts”, Maureen Spankle, ISBN: 81-317-0711-3
2.	E. Horowitz S. Sahani, D. Mehata, “Fundamentals of data structures in C++”, Galgotia Book Source, New Delhi, 1995, ISBN: 1678298
Reference Books	
1.	A. Tharp, “File organization and processing”, 2008, Willey India, ISBN: 9788126518685
2.	E. Horowitz S. Sahani, S. Rajashekharan, “Fundamentals of Computer Algorithms”, Galgotia Book Source, 2008, ISBN: 817515-257-5
3.	A Michael Berman, “Data structures via C++”, Oxford University Press, 2002, ISBN 0-19-510843-4
4.	Y. Langsam, M. Augustin and A. Tannenbaum, “Data structures using C and C++”, 2 nd Edition, Prentice Hall of India, 2002, ISBN 81-203-1177-9
5.	R. Gilberg, B. Forouzan, “Data Structures: A pseudo code approach with C++”, Cengage Learning, ISBN: 9788131504925
6.	A. Drozdek, “Data structures in C++”, 2 nd Edition, Thomos Books/ COLE Books, 2002, ISBN 981-240-079-6
7.	J. Tremblay, P. Sorenson, “An Introduction to data structures with applications”, 2 nd Edition, Tata McGraw-Hill International Edition, 1984, ISBN 0-07-462471-7
8.	M. Folk, B. Zoellick, G. Riccardi, “File Structure an Object Oriented Approach with C++”, Pearson Education, 2002, ISBN 81-7808-131-8 ”
9.	M. Weiss, Data Structures and Algorithm Analysis in C++”, 2 nd Edition, Pearson Education, 2002, ISBN 81-7808-670-0
10.	R. Gilberg, B. Forouzan, “Data Structures: a pseudo code approach with C”, Cengage Learning, ISBN: 9788131503140
11.	“How to solve it by Computer”, R.G. Domey ISBN: 978-81-317-0562-9

List of Practical Assignments:

Tools		
1.	Operating Systems (64-Bit)	64-BIT Fedora 17 or latest 64-BIT Update of Equivalent Open source OS or latest 64-BIT Version and update of Microsoft Windows 7 Operating System onwards
2.	Programming Tools (64-Bit)	Latest Open source update of Eclipse Programming frame work, Microsoft Visual Studio, TC++, QT/Java/Python
3.	Assignment Allocation	All assignments from groups A,B,C given below must be covered in a batch of 15 students or AICTE recommended Student : Teacher ratio. Submission journal will have assignments from group A, atleast 6 six assignments from group B and 1 assignment from group C covering all assignments in groups A,B,C per batch.
4.	Write up	Aim, Index terms, use of discrete mathematics to re-write/describe the problem definition. Use of set theory, Probability Theory or other relevant theory, Discrete Structures to give the problem solution, Data-independence / Dataflow Architecture for the development of Turing machine/state Diagram. LATEX Soft Copy CD and Handwritten Hard Copy Journal to be submitted as Term-work.

	Group A (Mandatory)
1.	A Vegetable and Fruit Mall wants to organize its vegetables and fruit products in a combination of purchase pattern of customers. Solve the problem by suggesting appropriate data structures. Design necessary class.
2.	A Dictionary stores keywords & its meanings. Provide facility for adding new keywords, deleting keywords, & updating values of any entry. Also provide facility to display whole data sorted in ascending/ Descending order, Also find how many maximum comparisons may require for finding any keyword. Make use of appropriate data structures.
3.	A news paper delivery boy every day drops news paper in a society having many lanes & each lane have many houses. Design a program to provide different paths that he could follow & also suggest the path which will make him to finish his task with less effort. Solve the problem by suggesting appropriate data structures. Design necessary class.
4.	Extending to problem 2. Consider dictionary data is stored in a file in random order. Thus, to search any the word & its meanings from given data, program should create reasonably balanced tree.
	Group B (Atleast Six)
1.	Write a program using object oriented programming features to implement Doubly circular linked list with different manipulation facilities in C++.
2.	Write a modular program using object oriented programming features to implement different sorting methods(quick, merge, radix, shell, heap sort) using Python
3.	Write a modular program using object oriented programming features to implement primitive operations on of Queues using Java Frame/Applet.
4.	Write a program using object oriented programming using C++ to create a binary tree if inorder& preorder or inorder & postorder any two traversals are given.
5.	Write a C++ program to implement traversals on Threaded Binary Tree using object oriented programming features. Design necessary class.
6.	Write a Java program to implement topological sorting on graph using object oriented programming features Design necessary class.
7.	Write a program to find shortest path for given source & destination of a given graph using C.
8.	Write a C program to generate optimal Binary Search Tree for given data.
9.	Write a modular program to implement primitive operations on Min/Max Heap using object oriented programming features Design necessary classes using Python.
10.	Write a C program to perform insert node, delete node and display operations on B tree.
11.	Tic-Tac- Toe gaming application using C++ Programming
12.	Binary tree traversals BFS & DFS using Python classes
13.	Write a Java program on Android/Win Platform that uses a hashing algorithm to create a list of inventory parts and their qualities sold in the past month. After creating the hashed list write a simple menu driven user interface that allows the user to select from the following options: <ul style="list-style-type: none"> a) Search for an inventory item and report its quantity sold b) Print inventory parts and their quantities sold c) Analyze efficiency of algorithm
14.	Write a program in Python – to implement following operations on text file :- create , Read, calculate the frequency of each vowel, Count the words, characters, lines, white space & special characters, Write all the results into another text file
15.	Write a program in C++ to implement hash table and handle the collision using linear probing and chaining. (with or without replacement) for employee information

	Using the above implement direct access file mechanism.
16	Write a program to create a binary tree if preorder & post-order traversals are given in JAVA .
	Group C: Advanced Data Structure Assignments (At least One)
1.	Write bubble Sort program using Python programming
2.	Write binary search program using multithreading in Java programming
3.	Write Tree Traversal Program using Java programming

Note: Examination will be based on the assignments performed.

210243 Digital Electronics & Logic Design

Teaching Scheme
Lectures: 3 Hrs/week
Practical: 2Hrs/Week

Examination Scheme
Theory: 50 Marks
OnLine: 50 Marks
Term Work: 25 Marks

Prerequisites: Basic Electronics Engineering

Course Objectives

- To learn and understand basic digital design techniques.
- To learn and understand design and construction of combinational and sequential circuits.
- To introduce to Digital logic design Software

Course Outcomes

- Solve K-MAPs and Boolean Algebra Experiments
- Use necessary A.C, D.C. and Loading characteristics and functioning while designing with digital gates
- Identify the Digital Circuits, Input/Outputs to replace by FPGA

Unit I	Number System & Logic Design Minimization Techniques:	8 Hrs.(TB-1: Ch- 2, RB-1)
	<p>Introduction: Binary, Hexadecimal numbers, octal numbers and number conversion.</p> <p>Signed Binary number representation: Signed Magnitude, 1's complement and 2's complement representation.</p> <p>Binary, Octal, Hexadecimal Arithmetic: 2's complement arithmetic.</p> <p>Algebra for logic circuits: Logic variables, Logic functions -NOT, AND, NOR, XOR, OR, XNOR, NAND</p> <p>Boolean algebra: Truth tables and Boolean algebra. Idealized logic gates and symbols. DeMorgan's rules Axiomatic definition of Boolean algebra, Basic theorems and properties of Boolean algebra</p> <p>Logic minimization: Representation of truth-table, SOP form, POS form, Simplification of logical functions, Minimization of SOP and POS forms, Don't care conditions</p> <p>Reduction techniques: K-Maps up to 4 variables and Quine-McClusky technique</p>	
Unit II	Logic Families:	6 Hrs. (TB-1: Ch-3, 4, RB- 2)
	<p>TTL: Standard TTL characteristics- Speed, power dissipation, fan-in, fan-out, current and voltage parameters, noise margin, operating temperature etc. Operation of TTL NAND gate. TTL Configurations- Active pull-up, Wired AND, totem pole, open collector.</p> <p>CMOS: CMOS Inverter, CMOS characteristics, CMOS configurations- Wired Logic, Open drain outputs</p> <p>Interfacing: TTL to CMOS and CMOS to TTL</p>	
Unit III	Combinational Logic:	
	<p>Codes:- BCD, Excess-3, Gray code , Binary Code and their conversion</p> <p>Arithmetic Operations: - Binary Addition, Subtraction, Multiplication, Division, BCD Addition</p>	9 Hrs.(TB-1: Ch-5,6, RB- 1, 3)

	<p>Circuits: - Half- Adder, Full Adder, Half Subtract or, Full Sub tractor, BCD adder using and subtract using 7483, look ahead and carry, parity generator and checker using 74180, magnitude comparator using 7485.</p> <p>Multiplexers (MUX):- Working of MUX, Implementation of expression using MUX (IC 74153, 74151).</p>	
	<p>Demultiplexers (DEMUX):- Implementation of expression using DEMUX, Decoder. (IC 74138).</p>	
Unit IV	Sequential Logic:	
	<p>Introduction: Sequential Circuits. Difference between combinational circuits and sequential circuits</p> <p>Flip- flop: SR, JK, D, T; Preset & Clear, Master and Slave Flip Flops their truth tables and excitation tables, Conversion from one type to another type of Flip Flop.</p> <p>Application of Flip-flops: Bounce Elimination Switch, registers, counters.</p> <p>Registers: Buffer register; shift register;</p> <p>Counters: Asynchronous counter. Synchronous counter, ring counters, BCD Counter, Johnson Counter, Modulus of the counter (IC 7490), Pseudo Random Binary Sequence Generator, Sequence generator and detector</p>	<p>8 Hrs.(TB-1: Ch-12, TB2: Ch-13, RB-1)</p>
Unit V	ASM & VHDL :	
	<p>Algorithmic State Machines: ASM charts, notations, design of simple controller, multiplexer controller method</p> <p>Examples: Sequence Generator, Types of Counter</p> <p>VHDL: Introduction to HDL, VHDL- Library, Entity, Architecture, Modeling Styles, Concurrent and Sequential Statements, Data Objects & Data Types, Attributes.</p> <p>Design Examples: VHDL for Combinational Circuits-Adder, MUX. VHDL for Sequential Circuits-Synchronous and Asynchronous Counter</p>	<p>9 Hrs.(TB-2: Ch-6,7, Appendix-A, RB-4)</p>
Unit VI	PLDs :	
	<p>PLD: PLA- Input, Output Buffers, AND, OR, Invert/ Non-Invert Matrix.</p> <p>Design Example: Any 4 Variables SOP function using PLDs, Study of basic Architecture of FPGA.</p>	<p>8 Hrs.(TB-1: Ch-13 (13.1 to 13.9))</p>

Text Books	
1.	R. Jain, “Modern Digital Electronics”, 3rd Edition, Tata McGraw-Hill, 2003, ISBN0 – 07 – 049492 – 4
2.	Stephen Brown, Zvonko Vranesic “ Fundamentals of Digital Logic with VHDL Design” Mcgraw-Hill
Reference Books	
1.	John Yarbrough, “Digital Logic applications and Design” Thomson
2.	Flyod “Digital Principles”, Pearson Education
3.	Malvino, D.Leach “ Digital Principles and Applications”, 5 th edition, Tata Mc- Graw Hill
4.	J.Bhaskar “VHDL Primer” 3 rd Edition, Pearson Edition

Digital Electronics Laboratory

Suggested List of Assignments:

A. Combinational Logic Design

1. T.T.L Characteristics (Study and Write up only).
2. Code converters e.g. Excess-3 to BCD and vice versa
3. Multiplexers: Application like Realization of Boolean expression using Multiplexer.
4. Demultiplexers: Applications like Realization of ROM using Demultiplexer.
5. BCD adder/Subtractor using 4 bit binary adder 7483.
6. Parity generator / detector.

B. Sequential Circuit Design

1. Flip flops, Registers and Counters (Study and Write up only).
2. 4-bit Multiplier / Divider (Study and Write up only).
3. Ripple counter using flip-flops.
4. Sequence generator using JK flip-flop.
5. Sequence detector using JK flip-flop.
6. Up-down counter using JK flip-flop.
7. Modulo N counter using 7490 & 74190 ($N > 10$).
8. Pseudo random number generator.
9. Design of a barrel shifter.

C. Study /Implement of VHDL and examples of Combinational and sequential circuits

1. Combinational Circuits: Adder, MUX
2. Sequential Circuits: Asynchronous or Synchronous Counter

D. ASM, PALS and FPGA

1. Simple ASM using multiplexer controller method.
2. Implementation of combinational logic using PLAs
3. Study of FPGA devices (Study and Write up only).

- Instructor will frame assignments based on the suggested assignments as given above. Students will submit the term work in the form of journal consisting of minimum of 16 assignments of which assignment of Group C and 2 assignments from Group D are compulsory.
- Term work assessment be done progressively and questions will be asked to judge the understanding of assignments performed.

210244 Operating System and Administration

Teaching Scheme
Lectures: 3 Hrs/week
Practical: 2Hrs/week

Examination Scheme
Theory: 50 Marks
OnLine: 50 Marks
Practical: 50 Marks
Term Work: 25 Marks

Prerequisites: FPL-I and FPL-II

Course Objectives

- To learn and understand basics of Operating Systems including Boot process.
- To learn and understand Shells, Scripts and File System.
- To introduce to administrative features of Operating Systems

Course Outcomes

- Basic knowledge of Unix/Linux operating system
- Writing Basic shell script commands and Admin commands
- Knowledge of files and storage systems

Unit I	Introduction to Operating Systems	6 Hrs
	General Overview: History of Unix, System Structure, User perspective, Operating system Services, Assumptions about Hardware	
Unit II	Introduction to the Kernel	6 Hrs
	Architecture of Unix operating system, Introduction to the system concepts, Kernel data structure, System Administration	
Unit III	Introduction to the File System	6 Hrs
	Pathnames, File system Mounting and unmounting, The organization of the File Tree, File Types, File Attributes, Access Control lists	
Unit IV	Bootting and Shut Down, Scripting and Shell	8 Hrs
	Bootstrapping, Booting PCs, GRUB, Booting with single user mode, Rebooting and Shutting down. Shell Basics, bash scripting, Regular Expression, Perl Programming, Python scripting, Scripting Best Practices, Working with Startup Scripts.	
Unit V	Access Control, Rootly Powers and Controlling Processes	6 Hrs
	Traditional UNIX access control, Modern Access Control, Real-world Access Control, Pseudo-users other than root. Components of a process, the lifecycle of a process, Signals, Kill, Process states, nice and renice, ps, Dynamic monitoring with top, prstat and topas, the /proc file system, strace, truss and tusc, runaway processes.	
Unit VI	Adding New Users and Storage	8 Hrs
	The /etc/passwd file, The /etc/shadow and /etc/security/passwd files, /etc/group file, Adding users, Adding users with useradd Storage: Adding a hard Disk, Storage Hardware, Storage hardware Interfaces, Software aspects of storage, Formatting, Disk Partitioning, RAID, LVM, Linux File System: The ext family, file system terminology, file system polymorphism, mkfs, fsck, file system mounting, setup for automatic mounting, USB drive mounting, Enabling swapping.	

Text Books

1	[Paperback] The Design of the Unix Operating System, Maurice J. Bach, Pearson Education, ISBN: 81-7758-770-6
2	Evi Nemeth, Garth Snyder, Tren Hein, Ben Whaley, Unix and Linux System Administration

Laboratory Assignments

List of tools and methods:

Tools		
1.	Operating Systems (64-Bit)	64-BIT Fedora 17 or latest 64-BIT Update of Equivalent Open source OS or latest 64-BIT Version and update of Microsoft Windows 7 Operating System onwards
2.	Programming Tools (64-Bit)	Latest Open source update of Eclipse Programming frame work, Microsoft Visual Studio, TC++, QT-cmake,-mingw.
3.	Assignment Allocation	All assignments from groups A,B,C given below must be covered in a batch of 15 students or AICTE recommended Student : Teacher ratio. Submission journal will have assignments from group A, atleast 6 six assignments from group B and 1 assignment from group C covering all assignments in groups A,B,C per batch.
4.	Write up	Aim, Index terms, use of discrete mathematics to re-write/describe the problem definition. Use of set theory, Probability Theory or other relevant theory, Discrete Structures to give the problem solution, Data-independence / Dataflow Architecture for the development of turing machine/state Diagram, multiplexer logic to identify opportunity of morphism and overloading and data flow architecture and optimal/effective use of multicore of the CPU. LA-TEX Soft Copy CD and Handwritten Hard Copy Journal to be submitted as Term-work.

A	Group A Assignments (Mandatory)
1.	Implementation of Create/ rename/ delete a file using Unix/Linux commands
2.	Write a function to display the list of devices connected to your system including the physical names and its instance number. Write a function using mount and unmount command to mount device and un-mount it.
3.	Adding users and access rights
4.	Implement the commands for creation and deletion of directory. Write a program to change current working directory and display the node details for each file in the new directory.
5.	Process related commands list the processes for the current shell, Display information about processes, Display the global priority of a process, change the priority of a process with default arguments.
6.	Use Operating system Commands to obtain the following results 1. To print the name of operating system 2. To print the login name 3. To print the host name
B	Group B Assignments (At least 6)
1	Write a C/C++ script to display all logged in users.
2	C/C++ Program to display the list of devices connected to your system including the physical names and its instance number.
3	C/C++ Program to Parent creating the child process by use of fork.
4	C/C++ Program to assign nice values to processes and dynamically monitor them.
5	C/C++ program to identify the available memory in the system.
6	C/C++ Scripts required to start, stop and perform daily cleanup tasks.

7	Write Java script to display all logged in users. Count the number of logged-in users. Write a program to create a foreground and background process for th selected user and display its status.
8	Java Program to Identify the available memory in the system..
9	Java Program to display the list of devices connected to your system including the physical names and its instance number.
10	Java Program to List the processes for the current shell, Display information about processes, Display the global priority of a process, Change the priority of a process with default arguments.
11	Java Working with startup scripts. (Init script)
12	Java Scripts required to start, stop and perform daily cleanup tasks.
13	Write python script to display all logged in users.
14	Python Program to demonstrate debugging of script.
15	Write a shell program to convert all lowercase letter in a file to uppercase letter.
16	Python Program to assign nice values to processes and dynamically monitor them.
17.	Write a Perl program that reads a file and counts the number of lines, characters, and words it contains and also prints count of the number of times the word appeared
18.	Write a daemon process that monitors a directory and reports the changes in directory contents.(e.g. addition of new files, directory access times etc)
	Group C (At least One)
1	Write program to find number of CPU cores and CPU Manufacturer
2	RAID installation and configuration in Linux derivative.
3	EXT4 file system installation and administration of Linux derivative.

Examination will be conducted on experiments performed

210245 Microprocessor Architecture

Teaching Scheme

Lectures: 3 Hrs/week

Practical: 2 Hrs/week

Examination Scheme

Theory: 50 Marks

OnLine: 50 Marks

Oral: 50 Marks

Term Work: 25 Marks

Course Objectives

- To study segmented, pipelined architecture in microprocessors
- To study memory management in microprocessors
- To study multitasking functioning
- To study microprocessor assembly language
- To take overview of multicore Architectures

Course Outcomes

- Use of segment descriptors, privileges, TSS in the programming
- Use of Memory management unit, page descriptor in the programming
- Understanding of multiprogramming and microprocessor architecture
- Write assembly language programs using 32/64 bit registers
- Knowledge of Multicore architecture

Unit I	80386DX Architecture	8 Hrs
	History of 8086 microprocessor, Concept of segmentation in 8086, 8086 Register block diagram, 80386DX functional Block Diagram, PIN Description, Register set, Flags, Physical address space, Data types	
Unit II	Memory Management Unit and Segment Description and Paging	10 Hrs
	80386Dx descriptor Tables GDT, LDT, IDT, descriptor cache, Code, data and stack descriptors, system descriptors, privilege levels, Segmentation in 80386DX, comparison of segmentation with 8086, paging, TSS, Nested Tasks, Operating in Real Mode, Protected Mode, Virtual 86 mode, Virtual addressing	
Unit III	Pipelined Architecture	6 Hrs
	Non-pipelined machine cycle, pipelined machine cycle.	
Unit IV	Assembly Language Programming	8 Hrs
	80386DX instruction set, setting protected mode, setting v86 mode, Real mode programming	
Unit V	New Architectures	4 Hrs
	What is multicore?, Multicore Architectures, The Software Developer's Viewpoint, The Bus Connections, Single Core to Multicore.	
Unit VI	Multicore Designs	12 Hrs
	Intel 64bit Architecture: Block Diagram, Basic Execution Environment, Data Types, Specific advances: Instruction set, Intel Microarchitecture code name Nehalem, SIMD Instructions, Hyper threading Technology, Virtualization Technology (Refer TB3) Systems Programming, Multiple Processor Management (Refer RB1)	

Text Books	
1.	(Paperback) 80386 Microprocessor Handbook, Chris H. Pappas, William H. Murray
2.	Professional Multicore Programming: Design and implementation for C++ Developers, Wiley India Edition, ISBN: 978-81-265-1875-3.
3	Intel® 64 and 32 bit Architectures Software developer's Manual, Volume –I, Intel, (Digital Content PDF, 253665.pdf)
4.	Introduction to 64 bit Intel Assembly Language Programming for Linux, 2 nd Edition, Ray Seyfarth, ISBN10: 1478119209, ISBN-13: 9781478119203, 2012

References	
1.	Intel 64 and IA-32 bit architectures Software Developer's Manual, Volume 3A, Intel, (Digital Content PDF: 253668.pdf)
2.	Ensuring Development Success By Understanding and Analyzing Assembly Language, Intel 2009 (Digital Content PDF: 321059.pdf)
3.	Assembly Language Step-by-step: Programming with Linux, 3 rd Edition, Jeff Duntemann, Wiley ISBN:- 10 0470497025, ISBN-13: 978-0470497029 , 2009

Laboratory Assignments

List of Practical Assignments:

Tools		
1.	Operating Systems (64-Bit)	Latest 64-BIT Version and update of Microsoft Windows 7/ Windows 8 Operating System onwards or 64-bit Open source Linux or its derivative
2.	Programming Tools (64-Bit)	Preferably using Linux equivalent or MASM64x or equivalent, Microsoft Visual Studio x64 Intrinsic (Refer your MSDN copy or http://msdn.Microsoft.com , C++
3.	Assignment Allocation	All assignments from groups A,B,C given below must be covered in a batch of 15 students or AICTE recommended Student : Teacher ratio. Submission journal will have assignments from group A, atleast 6 six assignments from group B and 1 assignment from group C covering all assignments in groups A,B,C per batch.
4.	Write up	Aim, Index terms, use of discrete mathematics to re-write/describe the problem definition. Use of set theory, Probability Theory or other relevant theory, Discrete Structures to give the problem solution, Data-independence / Dataflow Architecture for the development of turing machine/state Diagram, multiplexer logic to identify opportunity of morphism and overloading and data flow architecture and optimal/effective use of multicore of the CPU. LA-TEX Soft Copy CD and Handwritten Hard Copy Journal to be submitted as Term-work.

Group A (Mandatory)	
1	Write ALP to print "Hallo World!" Program using 16, 32 and 64-bit model and segmentation.
2	Write an ALP to accept ten 32-bit and 64 bit Hexadecimal numbers from user and store then in data segment table and display then numbers.
3	Write an ALP to accept a string and to display it's length.
4	Write an ALP to perform arithmetic and logical operations using 'n', 32-bit and 64-bit numbers stored in an array using 64 bit register operations.
5	Write an ALP to perform memory segment and register load/store operations using different addressing modes.
6	Write an ALP to program to use GDTR, LDTR and IDTR in Real Mode.
Group B (at least 6)	
1	Write an ALP to find the largest of given byte/Word/Dword/64-bit numbers
2.	Write C program using ALP instructions to convert Binary to decimal number.
3.	Write ALP using to read and display the table content pointed by GDTR/LDTR and IDTR
4	Write ALP to demonstrate the use of paging tables
5.	Write C program using an ALP instructions to set to use ETP (Extended page tables) with Core i3/i5/i7 Nehalem processors
6.	Write a switch case driven ALP to perform 64-bit hexadecimal arithmetic operations (+, -, *, /) using suitable macros. Define procedure for each operation.

7	Write an ALP to read command line arguments passed to a program.
8	Write an ALP to count no. of positive and negative numbers from the array.
9	Write ALP to identify CPU type and FPU type.
10	Write ALP to find average of n numbers stored in memory
11	Write program to read & display contents of file
12	Write a program to calculate area of circle using (co-processor)
13	Write a program to switch between real mode and protected mode
14	Write an ALP password program that operates as follows: a. Do not display what is actually typed instead display asterisk ("*"). If the password is correct display, "access is granted" else display "Access not Granted"
15	Write program to read & display ASCII contents of a file
16	Write an ALP to convert 64-Bit Big-Endian Number to Little-Endian Number
	Group C (at least One)
1.	Write a ALP to define two tasks and execute them on different CPU cores.
2.	Write a password program that operates as follows: b. Do not display what is actually typed instead display asterisk ("*"). c. If the password is correct display, "access is granted" else "sound the alarm".
3	Write ALP to switch from real mode to protected mode and display the values of GDTR, LDTR, IDTR, TR and MSW Registers.
4	Write a program for cache memory simulation. Consider 16kb of main memory, 128 byte of two way set associative caches. Assume 4 byte cache line. Find hit or miss in a cache line. If miss occurs line should be replaced in the cache using LRU technique.
5	Write a program to implement the multitasking application

Examination will be conducted on experiments performed

210246 SOFT SKILLS

Teaching Scheme
Lectures: 1 Hrs/week
Practical: 2 Hrs/week

Examination Scheme
Term Work: 25 Marks

UNIT I:

(04 hours)

Self Awareness & self Development –

a) Self Assessment , Self Appraisal, SWOT, Goal setting - Personal & career - Self-Assessment, Self-Awareness, Perceptions and Attitudes, Positive Attitude, Values and Belief Systems, Self-Esteem, Self appraisal, Personal Goal setting,

b) Career Planning, Personal success factors, Handling failure, Depression and Habit, relating SWOT analysis & goal setting, prioritization.

UNIT II: Communication Skill

(06 hours)

a) Importance of communication, types, barriers of communication, effective communication

b) Speaking Skills – Public Speaking, Presentation skills, Group discussion- Importance of speaking effectively, speech process, message, audience, speech style, feedback, conversation and oral skills, fluency and self expression, body language phonetics and spoken English, speaking techniques, word stress, correct stress patterns, voice quality, correct tone, types of tones, positive image projection techniques.

c) Listening Skills: Law of nature- you have 2 ears and 1 tongue so listen twice and speak once is the best policy, Empathic listening, Avoid selective listening-

d) Group Discussion - characteristics, subject knowledge, oral and leadership skills, team management, strategies and individual contribution and consistency.

e) Presentation skills - planning, preparation, organization, delivery.

f) Written Skills – Formal & Informal letter writing, Report writing, Resume writing - Sentence structure, sentence coherence, emphasis. Paragraph writing. letter writing skills - form and structure, style and tone. Inquiry letters, Instruction letters, complaint letters, Routine business letters, Sales Letters etc.

UNIT III: Corporate / Business Etiquettes

(02 hours)

Corporate grooming & dressing, Email & telephone etiquettes, etiquettes in social & office setting-

Understand the importance of professional behaviour at the work place, Understand and Implement etiquettes in workplace, presenting oneself with finesse and making others comfortable in a business setting. Importance of first impression, Grooming, Wardrobe, Body language, Meeting etiquettes (targeted at young professionals who are just entering business environment) , Introduction to Ethics in engineering and ethical reasoning, rights and responsibilities,

UNIT IV: Interpersonal relationship

(04 hours)

a) Team work, Team effectiveness, Group discussion, Decision making - Team Communication. Team, Conflict Resolution, Team Goal Setting, Team Motivation Understanding Team Development, Team Problem Solving, Building the team dynamics. Multicultural team activity

b) Group Discussion- Preparation for a GD, Introduction and definitions of a GD, Purpose of a GD, Types of GD, Strategies in a GD , Conflict management, Do's and Don'ts in GD

UNIT V: Leadership skills

(02 hours) Leaders'

role, responsibilities and skill required - Understanding good Leadership behaviours, Learning the difference between Leadership and Management, Gaining insight into your Patterns, Beliefs and Rules, Defining Qualities and Strengths of leadership, Determining how well you perceive what's going on around you, interpersonal Skills and Communication Skills, Learning about Commitment and How to Move Things Forward, Making Key Decisions, Handling Your and Other People's Stress, Empowering, Motivating and Inspiring Others, Leading by example, effective feedback

UNIT VI: Other skills

(02 hours)

a) Time management- The Time management matrix, apply the Pareto Principle (80/20 Rule) to time management issues, to prioritise using decision matrices, to beat the most common time wasters, how to plan ahead, how to handle interruptions , to maximise your personal effectiveness, how to say “no” to time wasters, develop your own individualised plan of action

b) Stress management- understanding the stress & its impact, techniques of handling stress

c) Problem solving skill, Confidence building Problem solving skill, Confidence building

Term Work/Assignments

Term work will consist the record of any 8 assignments of following exercises

1. SWOT analysis
2. Personal & Career Goal setting – Short term & Long term
3. Presentation Skill
4. Letter/Application writing
5. Report writing
6. Listening skills
7. Group discussion
8. Resume writing
9. Public Speaking
10. Stress management
11. Team Activity-- Use of Language laboratory

*** Perform any 8 exercises out of above 11 with exercise no. 11 as compulsory.**

Teaching Methodology

Each class should be divided into three batches of 20-25 students each. The sessions should be activity based and should give students adequate opportunity to participate actively in each activity. Teachers and students must communicate only in English during the session. Specific details about the teaching methodology have been explained in every activity given below.

Practical Assignments (Term work)

Minimum 8 assignments are compulsory and teachers must complete them during the practical sessions within the semester. The teacher should explain the topics mentioned in the syllabus during the practical sessions followed by the actual demonstration of the exercises. . Students will submit report of their exercise (minimum 8) assignments as their term work at the end of the semester but it should be noted that the teacher should assess their assignment as soon as an activity is conducted. The continual assessment process should be followed.

1. **SWOT analysis**

The students should be made aware of their goals, strengths and weaknesses, attitude, moral values, self confidence, etiquettes, non-verbal skills, achievements etc. through this activity. The teacher should explain to them on how to set goals, SWOT Analysis, Confidence improvement, values, positive attitude, positive thinking and self esteem. The teacher should prepare a questionnaire which evaluate students in all the above areas and make them aware about these aspects.

2. **Personal & Career Goal setting – Short term & Long term**

3 **Presentation Skills**

Students should make a presentation on any informative topic of their choice. The topic may be technical or non-technical. The teacher should guide them on effective presentation skills. Each student should make a presentation for at least 10 minutes.

4. **Letter/Application writing**

Each student will write one formal letter, and one application. The teacher should teach the students how to write the letter and application. The teacher should give proper format and layouts.

5. **Report writing**

The teacher should teach the students how to write report .. The teacher should give proper format and layouts. Each student will write one report based on visit / project / business proposal etc.

6. **Listening skills**

The batch can be divided into pairs. Each pair will be given an article (any topic) by the teacher. Each pair would come on the stage and read aloud the article one by one. After reading by each pair, the other students will be asked questions on the article by the readers. Students will get marks for correct answers and also for their reading skills. This will evaluate their reading and listening skills. The teacher should give them guidelines on improving their reading and listening skills. The teacher should also give passages on various topics to students for evaluating their reading comprehension.

7. **Group discussion**

Each batch is divided into two groups of 12 to 14 students each. Two rounds of a GD for each group should be conducted and teacher should give them feedback.

8. **Resume writing**

Each student will write one formal letter, and one application. The teacher should teach the students how to write the letter and application. The teacher should give proper format and layouts.

9. **Public Speaking**

Any one of the following activities may be conducted :

2. **Prepared speech** (topics are given in advance, students get 10 minutes to prepare the speech and 5 minutes to deliver.
3. **Extempore speech** (students deliver speeches spontaneously for 5 minutes each on a given topic)
4. **Story telling** (Each student narrates a fictional or real life story for 5 minutes each)
5. **Oral review** (Each student orally presents a review on a story or a book read by them)
- 10.. **Team Activity**-- Use of Language laboratory

Text Books:

- 1 Communication Skills : Sanjay Kumar and Pushpa Lata , Oxford University Press
- 2 Developing Communication Skill : Krishna Mohan, Meera Banerji,- McMillan India Ltd.
- 3 English for Business Communication : Simon Sweeney , Cambridge University Press

Books for references:

- 1.NASSCOM-Global Business Foudation Skills: Accenture,Convergys,Dell et.al.
Foundation Books : Cambridge University Press
2. Basic Managerial Skills for all E. H. McGrath, Eastern Economy Edition, Prentice hall India.
3. Personality Development and Group Discussions,Barun K. Mitra, Oxford University Press
- 4 Group Dissussions and Interview Skills : Priyadarshi Patnaik : Foundation Books :
Cambridge University Press
- 5.Thinks and Grow Rich: Napoleon Hill, Ebury Publishing, ISBN 9781407029252
6. Awaken the Giant Within: Tony Robbins HarperCollins Publishers,
ISBN-139780743409384
7. Change Your Thoughts, Change Your Life: Wayne Dyer, Hay House India,
ISBN-139788189988050
- 8 Habits of Highly Effective People: Stephen Covey Pocket Books, ISBN-13
9781416502494
- 9The Power of Your Subconscious Mind: Dr Joseph Murphy Maanu Graphics ,
ISBN-13 9789381529560
- 10- The new Leaders: Daniel Coleman Sphere Books Ltd , ISBN-139780751533811
- 11 The 80/20 Principal: by Richard Koch, Nicholas Brealey Publishings ,
ISBN-13 9781857883992
- 12 Time management from inside out: Julie Morgenstern, Owl Books (NY),
ISBN-13 9780805075908
- 13.Wonderland of Indian Manageress: Sharu Ranganekar, Vikas Publishing Houses,
ISBN-13 9788125942603
14. You can win: Shiv Khera, Macmillan, ISBN-139789350591932
15. **The Ace of Soft Skills: Attitude, Communication and Etiquette for Success:**
[Gopalswamy Ramesh](#), [Mahadevan Ramesh](#)

UNIVERSITY OF PUNE
For Computer Engineering & Information Technology (Sem II)
207003 ENGINEERING MATHEMATICS – III (2012 Course)

Teaching Scheme:

Lectures – 4 Hrs./Week

Tutorials – 1 Hr./Week

Examination Scheme:

Paper – 50 Marks (2 Hrs.)

Online – 50 Marks

Term work: 25 Marks

Section I

Unit I: Linear Differential Equations (LDE) and Applications (09 Hours)

LDE of n^{th} order with constant coefficients, Method of variation of parameters, Cauchy's & Legendre's DE, Simultaneous & Symmetric simultaneous DE. Modeling of Electrical circuits.

Unit II: Transforms (09 Hours)

Fourier Transform (FT): Complex Exponential form of Fourier series, Fourier integral theorem, Sine & Cosine integrals, Fourier Transform, Fourier Sine and Cosine Transform and their inverses.

Z - Transform (ZT): Introduction, Definition, Standard properties, ZT of standard sequences and their inverses. Solution of difference equations.

Unit III: Statistics and Probability (09 Hours)

Measures of central tendency, Standard deviation, Coefficient of variation, Moments, Skewness and Kurtosis, Correlation and Regression, Reliability of Regression Estimates.

Probability, Probability density function, Probability distributions: Binomial, Poisson, Normal and Hypergeometric. Test of Hypothesis: Chi-Square test.

Section II

Unit IV: Vector Differential Calculus (09 Hours)

Physical interpretation of Vector differentiation, Vector differential operator, Gradient, Divergence and Curl, Directional derivative, Solenoidal, Irrotational and Conservative fields, Scalar potential, Vector identities.

Unit V: Vector Integral Calculus and Applications (09 Hours)

Line, Surface and Volume integrals, Work-done, Green's Lemma, Gauss's Divergence theorem, Stoke's theorem. Applications to problems in Electro-magnetic fields.

Unit VI: Complex Variables (09 Hours)

Functions of Complex variables, Analytic functions, Cauchy-Riemann equations, Conformal mapping, Bilinear transformation, Cauchy's integral theorem, Cauchy's integral formula, Laurent's series, Residue theorem.

Text Books:

1. Advanced Engineering Mathematics, 9e, by Erwin Kreyszig (Wiley India).
2. Advanced Engineering Mathematics, 7e, by Peter V. O'Neil (Cengage Learning).

Reference Books:

1. Advanced Engineering Mathematics, 2e, by M. D. Greenberg (Pearson Education).
2. Advanced Engineering Mathematics, Wylie C.R. & Barrett L.C. (McGraw-Hill, Inc.)
3. Higher Engineering Mathematics by B. S. Grewal (Khanna Publication, Delhi).
4. Applied Mathematics (Volumes I and II) by P. N. Wartikar & J. N. Wartikar (Pune Vidyarthi Griha Prakashan, Pune).
5. Higher Engineering Mathematics by B.V. Ramana (Tata McGraw-Hill).
6. Advanced Engineering Mathematics with MATLAB, 2e, by Thomas L. Harman, James Dabney and Norman Richert (Brooks/Cole, Thomson Learning).

Tutorial and Term Work:

- i) Tutorial for the subject shall be engaged in minimum of four batches (batch size of 20 students maximum) per division.
- ii) Term work shall consist of six assignments (one per each unit) based on performance and continuous internal assessment.

#Mujumdar

[Handwritten signature]

210247 OBJECT ORIENTED AND MULTICORE PROGRAMMING

Teaching Scheme
Lectures: 3 Hrs/week
Practical: 2Hrs/Week

Examination Scheme
Theory: 50 Marks
On-Line: 50 Marks
Practical: 50 Marks
Term Work: 25 Marks

Prerequisite: Data Structures and problem Solving (Subject Code: 210244)

Course Objectives

1. To study the representation, implementation and applications of data structures
2. To study implementation of data structures using OOP concepts
3. To compare the benefits of static and dynamic data structures
4. To choose the appropriate data structure for modeling a given problem

Course Outcomes

1. Ability to use overloading and morphism in programming
2. Ability to understand basics of multicore-programming
3. Ability to perform programming using threads

Unit I	Foundations of Object Oriented Programming	(9 Hrs)
	Introduction to procedural, modular, object-oriented and generic programming techniques, Limitations of procedural programming, Need of object-oriented programming, fundamentals of object-oriented programming: objects, classes, data members, methods, messages, data encapsulation, data abstraction and information hiding, inheritance, polymorphism,	3 Hrs
	++: Extensions to C: Variable declarations, global scope, 'const', reference variables, comments, default parameters, function prototypes, function overloading, inline functions, default and constant arguments, 'cin', 'cout', formatting and I/O manipulators, new and delete operators	2 Hrs
	Defining a class, data members and methods, public, private and protected members, inline member functions, static data members, static member functions, 'this' pointer, constructors, destructors, friend function, dynamic memory allocation, array of objects, pointers and classes, class as ADTs and code reuse	4 Hrs
Unit II	Overloading and Inheritance	(7 Hrs)
	Introduction, Need of operator overloading, overloading the assignment, binary and unary operators, overloading using friends, rules for operator overloading, type conversions	(3Hrs)
	Concept and need, single inheritance, base and derived classes, friend classes, types of inheritance, hybrid inheritance, member access control, static class, multiple inheritance, ambiguity, virtual base class, polymorphism, virtual functions, pure virtual functions, abstract base class, virtual destructors, early and late binding, container classes	(4 Hrs)
Unit III	Templates and Exception Handling	(7 Hrs)
	Introduction, Templates: Function template and class template, function overloading vs. function templates, member function templates and template arguments, Introduction to Generic Programming: Introduction to Standard Template Library (STL), containers, iterators and algorithms, study of container template classes for vectors and stacks and related algorithms, Namespaces: Introduction, Rules of namespaces	(5 Hrs)
	Exception Handling: Introduction, syntax for exception handling code: try-catch-	(2Hrs)

	throw, Multiple Exceptions, Exceptions with arguments, Introduction to RTTI Managing Console I/O Operations: Introduction, C++ streams, stream classes, unformatted I/O, formatted I/O and I/O manipulators	
Unit IV	Challenges in Multi-core Programming	(4 Hrs)
	Sequential Models, Concurrency, Challenges for software development, New libraries for C++ developers, Processor Architecture Challenges, Operating systems (OS) roll in concurrent development: Consistent interfaces, Resource Management, OS interaction, Core OS services, Application Program Interfaces, Decomposition and Operating systems Roll, Hiding the Operating systems Roll: Abstraction and Encapsulation, Interface classes for POSIX API	
	Processes, Interface classes and predicates	(8 Hrs)
	Multicore and multiprocessors, Processes and threads, Parent-child relations, Process control block, Anatomy of a process, Process States, Process Scheduling, ps utility, process priorities, Context switch, Activities in process creation, Process environment variables, using system() to Spawn Processes, Killing a process, Process Resources, Asynchronous and Synchronous Processes, the wait() function call, Predicates, Processes and Interface classes	
Unit V	Multithreading	(7 Hrs)
	Thread, User and Kernel level threads, Thread Context, Hardware Threads and software threads, Comparing threads to processes, setting thread attributes, The architecture of a Thread, Compiling and linking threaded programs, Creating threads, Managing threads, Thread Interface classes	
Unit VI	Communication and synchronization of Concurrent tasks	(6 Hrs)
	Communication and synchronization, Synchronizing concurrency, Thread Strategy approaches, Decomposition and Encapsulation of Work Case studies of concurrency models	

Text Books(TB):

1. R. Gilberg, B. Forouzan, "Data Structures: A pseudo code approach with C", Cengage Learning, ISBN 9788131503140.
2. A. Michael Berman, "Data structures via C++", Oxford University Press, 2002, ISBN-0-19-510843-4.
3. Professional Multicore Programming: Design and implementation for C++ Developers, Wiley India Edition, ISBN: 978-81-265-1875-3,

Reference Books(RB):

1. E. Horowitz, S. Sahni, D. Mehta "Fundamentals of Data Structures in C++", Galgotia Book Source, New Delhi, 1995, ISBN 16782928.
2. Y. Langsam, M. Augenstin and A. Tannenbaum, "Data Structures using C and C++", 2nd Edition, Prentice Hall of India, 2002, ISBN-81-203-1177-9.
3. R. Gilberg, B. Forouzan, "Data Structures: A pseudo code approach with C++", Cengage Learning, ISBN 9788131504925.
4. A. Tharp, "File organisation and processing", 2008, Willey India edition, 9788126518685
5. A. Drozdek, "Data Structures in C++", 2nd Edition, Thomson Brookes /COLE Books, 2002, ISBN 981 – 240 – 079 – 6.
6. J. Tremblay, P. Soresan, "An introduction to data structures with Applications", 2nd edition, Tata McGraw-Hill International Editions, 1984, ISBN-0-07-462471-7.
7. M. Folk, B. Zoellick, G. Riccardi, "File Structure An Object oriented approach with C++", Pearson Education, 2002, ISBN 81 – 7808 – 131 – 8.
8. M. Weiss, "Data Structures and Algorithm Analysis in C++", 2nd edition, Pearson Education, 2002, ISBN-81-7808-670-0

Laboratory Assignments

List of Practical Assignments:

Tools	
1.	Operating Systems (64-Bit) Latest 64-BIT Version and update of Microsoft Windows 7/ Windows 8 Operating System onwards or 64-bit Open source Linux or its derivative
2.	Programming Tools (64-Bit) Any 64 bit C++ Programming tool like TC++/VC++/BC++/G++/GCC, JAVA, Python
3.	Assignment Allocation All assignments from groups A,B,C given below must be covered in a batch of 15 students or AICTE specified Student : Teacher ratio. Submission journal will have assignments from group A, atleast 3 assignments from group B and atleast 1 assignment from group C covering all assignments of the groups A, B, C per batch.
4.	Write up Aim, Index terms, use of discrete mathematics to re-write/describe the problem definition. Use of set theory, Probability Theory or other relevant theory, Discrete Structures to give the problem solution, Data-independence / Dataflow Architecture for the development of turing machine/state Diagram, multiplexer logic to identify opportunity of morphism and overloading and data flow architecture and optimal/effective use of multicore of the CPU. LA-TEX Soft Copy CD and Handwritten Hard Copy Journal to be submitted as Term-work.

Laboratory Experiments

Group A (Mandatory Assignments)

(Use of Constructor, destructor, Static member functions, friend class, this pointer, inline code and dynamic memory allocation is expected)

1	<p>1. Create a class named weather report that holds a daily weather report with data members day_of_month, hightemp, lowtemp, amount_rain and amount_snow. The constructor initializes the fields with default values: 99 for day_of_month, 999 for hightemp, -999 for low emp and 0 for amount_rain and amount_snow. Include a function that prompts the user and sets values for each field so that you can override the default values. Write a C++/Java/Python program that creates a monthly report.</p> <p>d) Menu driven program with options to Enter data and Display report e) Report Format</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Day</th> <th>Amt_Rain</th> <th>Amt_snow</th> <th>High_temp</th> <th>Low_temp</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td>Avg</td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>	Day	Amt_Rain	Amt_snow	High_temp	Low_temp											Avg									
Day	Amt_Rain	Amt_snow	High_temp	Low_temp																						
Avg																										
2	<p>A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and requests for the number of copies required. If the requested copies book details and requests for the number of copies required. If the requested copies are available, the total cost of the requested copies is displayed; otherwise the message "Required copies not in stock" is displayed.</p>																									

	Design a system using a class called books with suitable member functions and Constructors. Use new operator in constructors to allocate memory space required. Implement C++ program for the system.
3	Develop an object oriented program in C++ to create a database of the personnel information system containing the following information: Name, Date of Birth, Blood group, Height, Weight, Insurance Policy, number, Contact address, telephone number, driving license no. etc Construct the database with suitable member functions for initializing and destroying the data viz constructor, default constructor, copy, constructor, destructor, static member functions, friend class, this pointer, inline code and dynamic memory allocation operators-new and delete.
4	Design a C++ Class 'Complex ' with data members for real and imaginary part. Provide default and parameterized constructors. Write a program to perform arithmetic operations of two complex numbers using operator overloading (using either member functions or friend functions).
5	Write a C++ program to perform String operations i. = Equality ii. == String Copy iii. + Concatenation iv. << To display a string v. >> To reverse a string vi. Function to determine whether a string is a palindrome To find occurrence of a sub-string. Use Operator Overloading
6	Develop an object oriented program in C++ to create a database of the personnel information system containing the following information: Name, Date of Birth, Blood group, Height, Weight, Insurance Policy number, Contact address, telephone number, driving licence no. etc Construct the database with suitable member functions for initializing and destroying the data viz constructor, default constructor, copy constructor, destructor, static member functions , friend class, this pointer, inline code and dynamic memory allocation operators-new and delete.
7	Write a program in C++ using function template to read two matrices of different data types such as integers and floating point values and perform simple arithmetic operations on these matrices separately and display it.
8	Design a C++ base class consisting of the data members such as name of the student, roll number and subject. The derived class consists of the data members subject code, internal assessment and university examination marks. Construct a virtual base class for the item name of the student and roll number. The program should have the facilities. i) Build a master table ii) List a table iii) Insert a new entry iv) Delete old entry v) Edit an entry vi) Search for a record
9	Create a C++ class named Television that has data members to hold the model number and the screen size in inches, and the price. Member functions include overloaded insertion and extraction operators. If more than four digits are entered for the model, if the screen size is smaller than 12 or greater than 70 inches, or if the price is negative or over \$5000 then throw an integer. Write a main() function that instantiates a television object, allows user to enter data and displays the data members .If an exception is caught, replace all the data member values with zero values.
Group B (any Two)	
1	A 'C' program function having one IF-THEN-ELSE returns the truth-ness value (TRUE/FALSE) is to be replaced by overloading while porting it to C++. Use appropriate overloading to replace IF-THEN-ELSE. Demonstrate the functioning by

	using it in a class.
2	A 'C' program uses a structure to implement a circular linked list for maintaining the numbers in ascending order. New arrival of number increases the size of circular linked list. This program is to be ported to C++ using appropriate C++ Data structures and programming. (In C++ avoid use of structure and IF-Then-Else or while/do-while etc.)
3	Implement C++/Java/Python program to create a base class called shape . Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called function get_data() to initialize base class data members and another member function display_area() to compute and display the area of figures. Make classes to suit their requirements. Using these three classes, design a program that will accept dimension of a triangle or a rectangle interactively, and display the area. Remember the two values given as input will be treated as lengths of two sides in the case of rectangles, and as base and height in the case of triangles, and used as follows: Area of rectangle= x*y Area of triangle =1/2*x*y
4	Implement C++/Java/Python program to implement a base class consisting of the data members such as name of the student, roll number and subject. The derived class consists of the data members subject code ,internal assessment and university examination marks. The program should have the facilities. i) Build a master table ii) List a table iii) Insert a new entry iv) Delete old entry v) Edit an entry vi) Search for a record. Use virtual functions.
5	Implement C++/Java/Python program to write a class template to represent a generic vector. Include following member functions: To create the vector. To modify the value of a given element To multiply by a scalar value To display the vector in the form (10,20,30,...)
6	Implement C++/Java/Python program for bubble sort using function template
7	Refer the standard template library to use list container and using C++/Java implement following member functions of list class: empty, insert, merge, reverse, sort
8	Write a C++/Java program for the following: 1) A function to read two double type numbers from keyboard 2) A function to calculate the division of these two numbers 3) A try block to throw an exception when a wrong type of data is keyed in 4) A try block to detect and throw an exception if the condition "divide-by-zero" occurs 5) Appropriate catch block to handle the exceptions thrown
9	Write a C++/Java program for the following: Create a class named Television that has data members to hold the model number and the screen size in inches, and the price. Member functions include overloaded insertion and extraction operators. If more than four digits are entered for the model, if the screen size is smaller than 12 or greater than 70 inches, or if the price is negative or over \$5000 then throw an exception. Write a main() function that instantiates a television object, allows user to enter data and displays the data members .If an exception is caught ,replace all the data member values with zero

	values.
10	Create employee bio-data using following classes i) Personal record ii) Professional record iii) Academic record Assume appropriate data members and member function to accept required data & print bio-data. Create bio-data using multiple inheritance using C++/Java/Python.
11	Using multi-core programming implement POSIX-spawn() function to create a process
12	Implement a simple interface class for a POSIX Process using multi-core programming
13	Using multi-core programming implement a predicate class
14	Implement POSIX queue class that encapsulates the basic function such as open, send, receive, remove, close. Use multi-core programming
15	Implement POSIX semaphore using multi-core programming
16	Using multi-core programming, implement Mutex semaphore for : Initialization Request ownership Release ownership Try ownership Destruction
17	Using multi-core programming implement a thread interface class
18	Write a Object Oriented Program using C++/Java for 6. passing command line arguments to the thread function 7. using the command line argument to determine the number of threads Use multi-core programming
Group C (Advanced Assignments) Any One	
1.	Write a concurrent program to implement the Odd-Even Merge Sort. Effective use of Multicore Architecture Core 1 and Core 2 effectively is expected.
2.	Write a concurrent program to implement the Dining philosophers problem. Effective use of Multicore Architecture is expected.
3.	Write a concurrent program for Matrix Multiplication. Effective use of Multicore Architecture is expected.

Note: Examination will be based on the assignments performed.

210248 Microprocessor and Interfacing Techniques

Teaching Scheme
Lectures: 3 Hrs/week
Practical: 2Hrs/Week

Examination Scheme
Theory: 50 Marks
OnLine: 50 Marks
Term Work: 25 Marks

Prerequisites: Microprocessor Architecture

Course Objectives:

1. To learn the Peripheral architecture and programming Of Microprocessor.
2. To learn peripherals and their interfacing with 8086 Microprocessor.
3. To study the DOS Internals.
4. To Study NDP and Design of Microprocessor based System.

Course Outcomes:

- 1.Ability to handle, interface and program using legacy peripherals
- 2.Ability to understand I/O Hub functions

Unit I	Introduction to Interfacing	
	Introduction to 64/32 bit microprocessors, Differences in Architecture and Pin diagram of 8086, 80386DX, i7 microprocessor, Differences in Programmers model (Register SET) and Segmentation of 8086, 80386DX and i7 microprocessor, logical to physical address translation, BUS Sizing, Memory Read/write (8/16/32/64-bit) cycle timing diagrams, I/O Read /Write cycles timing diagrams, Address mapping and decoding. Addressing modes, Instruction set of 80386 in detail, Instruction Formats, Stacks, Assembly Language Programming, Assembler, Linker, Debugger (Turbo debugger), Directives, Procedures (Near & Far), Macros, Loop constructs, 80386 Programming examples.	(8 Hrs)
Unit II	Legacy DOS Internals and Interrupt Controller	
	Interrupt Structure, Interrupt Vector Table (IVT), ISR, Hardware and software Interrupts Internals of DOS, DOS loading, DOS memory map, Internal and external commands of DOS, BIOS & DOS Interrupts. Concepts of PSP, .EXE & .COM files, Concepts of TSR, 8259 (Programmable Interrupt Controller): Features, Block Diagram, Control & status registers, Interfacing & Programming. Computer Memory and Memory Mapping in 64-bit mode	(8 Hrs)
Unit III	Legacy Serial and parallel Communication	
	(Programmable Peripheral Interface), Serial Communication- Synchronous & Asynchronous, 8251 (USART): Features, Block Diagram, Control & status registers, Operating modes, Interfacing & Programming (8255 and 8251) Concept of ADC -Successive Approximation & Interfacing, Concept of DAC R-2R (ladder) & Interfacing, Introduction to Sensors & Transducers, Keyboard Display & Centronics Printer Parallel Interfacing, USB Printers	(9 Hrs)
Unit IV	Legacy Timer and DMA Controller	
	Keyboard and Display Controller, 8253 (Programmable Interval Timer): Features, Block Diagram, Control & status registers, Operating modes, Interfacing & Programming. Concept of DMA, 8237 DMA Controller: Features, Block Diagram	(7 Hrs)
Unit V	Legacy Math-Controller	
	Minimum & Maximum mode of 8086, Support chips 8282,8284,8286,8288 8087(NDP) - Features, Block Diagram, Control & status registers, typical Instruction Set & Programming Detail Design of 8086 based minimum system with EPROM,	(8 Hrs)

	SRAM & Peripherals such as 8255,8253,8251,8279 with keyboard & seven segments Display.	
Unit VI	Advanced I/O Controllers	
	Introduction to Intel 58X chipset, Intel® 82801IJR I/O Controller Hub, Study of Intel i5 Motherboard Block Diagram	(8 Hrs)

Text Books:

1. Douglas Hall, "Microprocessors & Interfacing", McGraw Hill, Revised 2nd Edition, 2006 ISBN 0-07-100462-9
2. John Uffenbeck, "The 8086/88 Family: Design, Programming & Interfacing", PHI,
3. A.Ray, K.Bhurchandi, "Advanced Microprocessors and peripherals: Arch, Programming & Interfacing", Tata McGraw Hill, 2004 ISBN 0-07-463841-6
4. Introduction to 64 bit Intel Assembly Language Programming for Linux, 2nd Edition, Ray Seyfarth, ISBN10: 1478119209, ISBN-13: 9781478119203, 2012

References Books:

1. Liu, Gibson, "Microcomputer Systems: The 8086/88 Family", 2nd Edition, PHI, 2005
2. Kenneth Ayala, "The 8086 Microprocessor: Programming & Interfacing the PC", Cengage Learning, Indian Edition, 2008
- Ray Dunkon, "Advanced MSDOS Programming", 2nd Edition, BPB Publication.
3. Kip Irvine, "Assembly language for IBM PC", PHI, 2nd Edition, 1993
4. Peter Abel, "Assembly language programming", Pearson Edu, 5th Edition, 2002
5. Intel Microprocessor and peripheral Handbook: Volume 1
6. Yashwant Kanitkar, "TSR through C", BPB Publication, 1995, ISBN 81- 7029-520-3.

MICROPROCESSOR INTERFACING LABORATORY

Suggested List of Assignments

Group A

1. Write X86/64 Assembly language program (ALP) to add array of N hexadecimal numbers stored in the memory. Accept input from the user.
2. Write X86/64 ALP to perform non-overlapped and overlapped block transfer (with and without string specific instructions). Block containing data can be defined in the data segment.
3. Write 64 bit ALP to convert 4-digit Hex number into its equivalent BCD number and 5-digit BCD number into its equivalent HEX number. Make your program user friendly to accept the choice from user for: (a) HEX to BCD b) BCD to HEX (c) EXIT.
Display proper strings to prompt the user while accepting the input and displaying the result. (use of 64-bit registers is expected)
4. Write X86/64 ALP for the following operations on the string entered by the user. (use of 64-bit registers is expected)
 - a) Calculate Length of the string
 - b) Reverse the string
 - c) Check whether the string is palindrome

OR

Make your program user friendly by providing MENU like:

- (a) Enter the string
 - b) Calculate length of string
 - c) Reverse string
 - d) Check palindrome
 - e) Exit
- Display appropriate messages to prompt the user while accepting the input and displaying the result.
5. Write 8086 ALP to perform string manipulation. The strings to be accepted from the user is to be stored in data segment of program_1 and write FAR PROCEDURES in code segment program_2 for following operations on the string:

(a) Concatenation of two strings (b) Number of occurrences of a sub-string in the given string Use PUBLIC and EXTERN directive. Create .OBJ files of both the modules and link them to create an EXE file.
6. Write X86/64 ALP to perform multiplication of two 8-bit hexadecimal numbers. Use successive addition and add and shift method. Accept input from the user. (use of 64-bit registers is expected)

7. Write 8087ALP to obtain:

i) Mean ii) Variance iii) Standard Deviation

For a given set of data elements defined in data segment. Also display result.

Group B

1. 8255

(a) Write 8086 ALP to convert an analog signal in the range of 0V to 5V to its corresponding digital signal using successive approximation ADC and dual slope ADC. Find resolution used in both the ADC's and compare the results.

(b) Write 8086 ALP to interface DAC and generate following waveforms on oscilloscope,

(i) Square wave - Variable Duty Cycle and Frequency.

(ii) Ramp wave - Variable direction, (iii) Trapezoidal wave (iv) Stair case wave

(c) Write 8086 ALP to rotate a stepper motor for given number of steps at a given angle and in the given direction of rotation based on the user choice such as

(i) If 'C' key is pressed - clockwise rotation, (ii) If 'A' key is pressed -

anticlockwise rotation. (iii) If 'B' is pressed - 1/2 clockwise and Vz

Anti-clock wise rotation, (iv) If 'S' key is pressed - stop rotation. Also write routines to accelerate and de-accelerate the motor.

(d) Write 8086 ALP to print a text message on printer using Centronix parallel printer interface.

NOTE: Select any two from 8255 assignments

2. 8253

Write 8086 ALP to program 8253 in Mode 0, modify the program for hardware retriggerable Mono shot mode. Generate a square wave with a pulse of 1 ms. Comment on the difference between Hardware Triggered and software triggered strobe mode. Observe the waveform at GATE & out pin of 1C 8254 on CRO

3. 8279

Write 8086 ALP to initialize 8279 and to display characters in right entry mode.

Provide also the facility to display

- Character in left entry mode.
- Rolling display.
- Flashing display

4. 8251

Perform an experiment to establish communication between two 8251 systems A and B. Program 8251 system A in asynchronous transmitter mode and 8251 system B in asynchronous receiver mode. Write an ALP to transmit the data from system A and receive the data at system B. The requirements are as follows:

Transmission:

- message is stored as ASCII characters in the memory.
- message specifies the number of characters to be transmitted as the first byte.

Reception:

- Message is retrieved and stored in the memory.
- Successful reception should be indicated.

5. 8259

Write 8086 APL to interface 8259 in cascade mode (M/S) and demonstrate execution of ISR in following manner:

Main program will display two digits up counter. When slave IRQ interrupt occurs, it clears the counter and starts up counting again. When Master IR1 interrupt occurs, it resets the counter to FFH and starts down counting.

6. TSR Program

Write a TSR program in 8086 ALP to implement Real Time Clock (RTC). Read the Real Time from CMOS chip by suitable INT and FUNCTION and display the RTC at the bottom right corner on the screen. Access the video RAM directly in your routine.

7. TSR Program

Write a TSR program in 8086 ALP to implement Screen Saver. Screen Saver should get activated if the keyboard is idle for 7 seconds. Access the video RAM directly in your routine.

Group C

1. Study of Intel i5 Motherboard Block Diagram, Peripheral Connectors Pin Diagrams and functioning of I/O Hub, DDR-3 memory BUS

Student will submit the term work in the form of Journal consisting of minimum of 13 experiments with all seven experiments from group A and any 5 assignments from group B and group C assignments. Practical examination will be based on the term work and questions will be asked to judge the understanding of assignments performed at the time of examination.

210249 Computer Graphics and Gaming

Teaching Scheme
Lectures: 3 Hrs/week
Practical: 2Hrs/Week

Examination Scheme
Theory: 50 Marks
OnLine: 50 Marks
Term Work: 25 Marks

Prerequisite: Knowledge of C/C++ Programming and Basic Data Structures & Mathematics

Course Objectives

1. To understand basics of computer graphics
2. To give more emphasis on implementation aspect of Computer Graphics Algorithm.
3. To prepare the student for advance courses like multimedia / Computer Vision.

Course Outcome

3. Become knowledgeable about computer graphics hardware and peripherals
4. To generate 2D line and polygonal diagram and apply transformations.

Teaching aid

Faculties should use LCD to demonstrate the concept of Graphics.

Unit I	Basic Concepts and devices	(6 Hrs)
	Introduction to computer graphics, state of art applications of computer graphics, pixel, frame buffer, resolution, aspect ratio. Video display devices: CRT (Raster scan and random scan displays), flat panel displays. Interactive devices: joysticks, touch panels, light pens. Data generating devices: scanners and digitizers. graphics Files: TIFF. Introduction to GTK+.	
Unit II	Scan Conversions	(6 Hrs)
	Line and line segments, line and circle drawing algorithms: DDA and Bresenham , Line styles: thick, dotted and dashed. Antialiasing and antialiasing techniques. Character generating methods: stroke and bitmap method, Multilingual character standards. Concurrent line drawing using midpoint sub-division algorithm.	
Unit III	Clipping and Windowing	
	Polygon and Clipping Introduction to polygon, types: convex, concave and complex. Representation of polygon, Inside test, polygon filling algorithms – seed fill, scan line fill and filling with patterns. Windowing and clipping: viewing transformations, 2-D clipping: Cohen – Sutherland algorithm. Polygon clipping: Sutherland Hodgeman algorithm, generalized clipping	(6 Hrs)
Unit IV	Geometric Transformations	
	2-D transformations: introduction, matrices, Translation, scaling, rotation, homogeneous coordinates and matrix representation, translation, coordinate transformation, rotation about an arbitrary point, inverse and shear transformation. 3-D transformations: introduction, 3-D geometry, primitives, 3-D transformations and matrix representation, rotation about an arbitrary axis, concept of parallel and perspective projections, viewing parameters, 3-D clipping, 3-D viewing transformations.	(6 Hrs)
Unit V	Curves, Fractals, Hidden Surfaces, Light and Color Models	

	Hidden surfaces: introduction, back-face removal algorithm: Painter’s algorithm, binary space partition, Warnock algorithm, Z –buffer. Light and Color: Introduction, Diffused illumination, point source illumination, Shading Algorithms, reflections, shadows, ray tracing, Transparency, Color models and tables: RGB, HIS, Introduction to curve generation, interpolation, B-splines, Bezier curve, Blending function, fractals, Fractal lines and surfaces, Irregular Curves: Joints and Knots, Connectivity.	(6 Hrs)
Unit VI	Getting Started with Animation and Gaming	
	Segments: Introduction, segment table, segment creation, deletion, renaming. Image transformations, raster techniques. Animation: Conventional and computer based animation, Methods of Controlling Animations, Basic guidelines of animation, Gaming Technologies, animation languages: Introduction to OpenGL ES, Gaming platforms: NVIDIA workstation (Block Diagram only), i860(Block Diagram only), Graphics Memory Pipeline (Block diagram only). Introduction to Interactive Graphics & usage of at least two tools of computer graphics (3D studio, Maya, or Similar open source tools)	(6 Hrs)

Text Books:

1. S. Harrington, “Computer Graphics”, 2nd Edition, McGraw-Hill Publications, 1987, ISBN 0 – 07 – 100472 – 6.
2. D. Rogers, “Procedural Elements for Computer Graphics”, 2nd Edition, Tata McGraw-Hill Publication, 2001, ISBN 0 – 07 – 047371 – 4.

Reference Books:

1. J. Foley, V. Dam, S. Feiner, J. Hughes, “Computer Graphics Principles and Practice”, 2nd Edition, Pearson Education, 2003, ISBN 81 – 7808 – 038 – 9.
2. D. Hearn, M. Baker, “Computer Graphics – C Version”, 2nd Edition, Pearson Education, 2002, ISBN 81 – 7808 – 794 – 4.
3. D. Rogers, J. Adams, “Mathematical Elements for Computer Graphics”, 2nd Edition, Tata McGraw-Hill Publication, 2002, ISBN 0 – 07 – 048677 – 8.
4. Beginning Android 4 Games Development, Mario Zechner, Robert Green, Apress, ISBN: 978-81-322-0575-3

210250 COMPUTER ORGANIZATIONS

Teaching Scheme
Lectures: 3 Hrs/week
Practical: 2Hrs/Week

Examination Scheme
Theory: 50 Marks
OnLine: 50 Marks

Pre requisites: Microprocessor Architecture

Course Objectives

1. To understand the structure, function and characteristics of computer systems
2. To understand the design of the various functional units of digital computers
3. To learn basics of Parallel Computer Architecture.

Course Outcome

1. Able to understand Von Neumann and dataflow Architecture block diagrams
2. Able to understand Internal block diagram and functioning of CPU, ALU and memory organization

UNIT I	The evolution of computers and number operations	8 Hrs
	Mechanical Era, Electronic computer, VLSI – Integrated circuits. SOC Processor architecture performance consideration performance measure speedup techniques. System Architectures – Microprocessor, Micro controller and parallel processing. Designing for Performance, Von-Neumann Architecture, Data flow architecture, Computer Components, Interconnection Structures, Bus Interconnection, Scalar Data Types, Fixed and Floating point numbers, IEEE 488 Number representation, Signed numbers, Integer Arithmetic, 2's Complement method for multiplication, Booths Algorithm,	
UNIT II	Processor Organization	6 Hrs
	Processor Basics: CPU organization, CPU Bus Organization: Central BUS, Buses on periphery, Additional features: RISC & CISC types representative commercial microprocessor of RISC & CISC types, Co-processors Data representation –Integer and floating point representation, Instruction set –Addressing modes formats, Machine Instruction characteristics, types of operands, types of operations, Instruction formats, Processor organization, Register Organization in 8086/88, 80386Dx and i7 microprocessors,	
UNIT III	Data Paths and ALU	6 Hrs
	Data Paths: Fixed point and floating point Arithmetic, ALU, Pipeline processing Case study of Intel Nehalem organization, pipelined and non-pipelined machine cycles	
UNIT IV	Control Design Organization	5 Hrs
	Control Design: Basic concepts Hardwired and micro-programmed control, Pipeline control, Example of ADD Instruction macro/micro design,	
UNIT V	Memory and I/O Organization	6 Hrs
	Memory systems, DDR3 Memory Organization, NUMA and UMA, caches memory mapped I/O and I/O mapped I/O, DMA, buses and standard interfaces –serial parallel buses –PCI, SCSI USB. USB bus organization to interface display and Printer, Case Study: Intel Nehalem Memory Organization.	

UNIT VI	Advanced Computer Organizations	6 Hrs
	Advanced computer Organizations (Block Diagrams only) The AMD Multicore Opteron, The Sun UltraSparc T1, The IBM Cell Broadband Engine (CBE), The Intel IA-64, The IA-64 model: Explicitly Parallel Instruction Computing, Prediction, Speculative Loads. 64-bit architectures i5/ i7 Desktop version and mobile version, NVIDIA GPU architecture (Block diagram only).	

Text Books:

1. Computer Architecture and Organization, Jhon P Hays, 3rd Edition, McGraw-Hill Publication, 2001, ISBN 0071004793
2. Zaky S, Hamacher, "Computer Organization", 5nd Edition, McGraw-Hill Publications, 2001, ISBN 0071122184.

Reference Books:

1. Intel Microprocessor Hand book (or PDF File from www.microsoft.com)
2. Computer Organization by Tanunbum

210251 PROGRAMMING LABORATORY

Teaching Scheme

Lectures: -----

Practical: 2Hrs/Week

Examination Scheme

Term Work: 50 Marks

Oral : 50 Marks

Pre requisites: Microprocessor Architecture

Learning Objectives

4. To understand the structure, function and characteristics of computer systems
5. To understand the design of the various functional units of digital computers
6. To learn basics of Parallel Computer Architecture.

List of Practical Assignments:

Tools		
1.	Operating Systems (64-Bit)	Latest 64-BIT Version and update of Microsoft Windows 7/ Windows 8 Operating System onwards or 64-bit Open source Linux or its derivative, 32/64 bit Android 4 (for mobile) or above.
2.	Programming Tools (64-Bit)	MASM64x or equivalent, Microsoft Visual Studio x64 Intrinsic with IDE (Refer your MSDN copy or http://msdn.microsoft.com OpenGL ES, GTK++, TC++, Cuda C++

A>	Compulsory assignments
1.	Writing a C/C++ Program to emulate CPU Architecture (Central Bus) Develop register, ALU level GUI to display results..
2.	Writing a C++ class for displaying pixel or point on the screen.
3.	Write a C++ class for a Line drawing method using overloading DDA and Bresenham's Algorithms, inheriting the pixel or point.
4.	Write a C++ class for a circle drawing inheriting line class
B>	At least Six (Use of Morphism, Inheritance and associated OO-programming in the implementation is expected)
1	Write a program using python to draw a line with line styles (Thick, Thin, Dotted)
2.	Write a program in C/C++ to draw a line with line style (Thick, Thin, Dotted)
3	Write a program in C/C++ to draw a circle of desired radius.
4	Write a C/C++ program to draw a convex polygons (Square, Rectangle, Triangle)
5.	Write a C/C++ program to draw a Convex polygon with programmable edges.
6.	Write a C/C++ program to fill polygon using scan line algorithm
7.	Write a Java/ Python program to draw a line with line style (Thick, Thin, Dotted)
8	Write a program in to draw a circle of desired radius using VC++ 12 or above. Use of BITBLT command is expected.
9	Write a Java/Python program to draw a simple polygons (Square, Rectangle, Triangle)
10	Write a Java/Python program to draw a simple polygon with programmable edges, a. using mouse click event b. using dialog box to accept to accept edges ordered list and its size in pixels
11	Write a Java/Python program to fill polygon using scan line algorithm
12.	Write a program in C++ to test that given point is inside the polygon
13	Write a program in C++ draw a concave polygon
14	Write a program to scan fill the given concave polygon for Android Mobile Programming. Use Android ADT for Eclipse.
15	Write a class to implement the Booths Multiplier for 8/16/32/64-bit numbers using sign extended multiplication.
16	Write a C++ class for a Line drawing method using overloading DDA and Bresenham's

	Algorithms, inheriting the pixel or point. Use Android ADT for Eclipse.
C>	At least One Advanced Technology Programming
1	Use OpenGL ES to draw a line for Android Mobile
2	Use Microsoft IDE to Draw a line Diagram
3.	Use VRML to draw a line Diagram
4	Use Direct3D/Maya or open source equivalent to draw a Bouncing ball animation
5	Use Parallel programming using Cuda to draw polygoan.

Examination will be conducted on experiments performed