Swami Ramanand Teerth Marathwada University
Dyanteerth, Nanded

Teaching and Examination Scheme for Second Year Computer Science & Engineering(w.e.f. academic year 2009-10)

Semester - III

Sr. No.	SUBJECT	TEACHING SCHEME Hours / Week				EXAMINATION SCHEME			
		Lect	Pract	Total	Paper	Tw	Test	Pract & Oral	Total Marks
01	Mathematics III	4		4	80		20		100
02	Digital Systems and Microprocessor	4		4	80		20		100
03	Discrete Mathematics	4		4	80		20		100
04	Data Structures	4		4	80		20		100
05	Data Communication	4		4	80		20		100
06	Digital Systems and Microprocessor Laboratory		2	2		25		50	75
07	Data Structures Laboratory		4	4		25		50	75
08	Object Oriented Programming Laboratory	2	2	4		25		50	75
09	Data Communication Laboratory		2	2		25		50	75
	Total	22	10	32					800
	Total of Part I	**32 Hrs**				**800**			

Semester - IV

Sr. No.	SUBJECT	TEACHING SCHEME				EXAMINATION SCHEME			
		Theory	Pract	Total	Paper	Tw	Test	Pract & Oral	Total
01	Mathematics IV	4		4	80		20		100
02	Computer Algorithms	4		4	80		20		100
03	Principles of Programming Languages	4		4	80		20		100
04	System Programming	4		4	80		20		100
05	Advanced Microprocessors	4		4	80		20		100
06	Communication Skills	2		2	40		10		50
07	JAVA Programming Laboratory	2	4	6		25		50	75
08	UNIX and System Programming Laboratory		2	2		25		50	75
09	Advanced Microprocessors Laboratory		2	2		25		25	50
10	Communication Skills Laboratory		2			25		25	50
	Total	24	10	34					800
	Total of Part II (B)	**34 Hrs**				**800**			
	Grand Total (A+B)					**1600**			

TW : Term Work **Pract : Practical**

Note: Minimum two test should be conducted for each theory subject in a semester and average of best two tests should be considered.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Structure
(Department: Computer Science & Engineering)
(Semester III)

1.ENGINEERING MATHEMATICS III

Teaching Scheme: L: 4

Evaluation Scheme	Theory	Test	Minimum Passing Marks
	80 Marks	20 Marks	40%

Objective: The objective of this course is to assist the students of Engineering to acquire different techniques of solution of differential equations, Statistical methods and to enhance skills of Mathematical modeling, Simulation in different areas of Engineering.

UNIT - I Linear Differential equations of Higher order – I 8 Hrs

1.1 Introduction to L.D.E. with constant coefficients
1.2 General solution of f(D)y = X, shortcut methods
1.3 Non-Homogeneous linear equations: i) Solution by Method of variation of parameters
ii) Solution by Method of undetermined coefficients
1.4 Equations Reducible to L.D.E. with constant coefficients i.e.
i) Cauchy's Homogeneous linear equation
ii) Legendre' Linear equations

UNIT – II Linear Differential equations of Higher order - II 6 Hrs

2.1 Simultaneous L.D.E. with constant coefficient
2.2 Symmetrical simultaneous equations
$\frac{dx}{P} = \frac{dy}{Q} = \frac{dz}{R}$ by i) Method of Grouping ii) Method of Multipliers.
2.3 Applications of L.D.E. to electrical circuits

UNIT – III Vector Differential Calculus 7 Hrs

3.1 Vector and Scalar functions, fields, derivatives
3.2 Gradient of Scalar field, Directional derivative and Geometrical meaning of gradient (Gradd ϕ)
3.3 Divergence and curl of a vector fields
3.4 Solenoidal and Irrotational vectors
3.5 Second order diff operator and vector identifies

UNIT – IV Vector Integral Calculus 7 Hrs

4.1 Line integral, Line integral independent of path, Line Integral in parametric form
4.2 Circulation of a vector [Work done]
4.3 Green's Theorem [without proof] its verification and applications
4.4 Surface Integral, Stoke's Theorem [without proof] and its applications
4.5 Gauss Divergence Theorem [without proof] and its applications to Engineering problems

UNIT – V Statistics 6 Hrs

5.1 Corelation: Scatter diagram,Types of correlations
5.2 Karl Pearsson's coefficient of correlation
5.3 Regression: Lines of regressions, Lines of regression of Bivariate data
5.4 Curve fitting: Fitting of curves by Least Square Method

UNIT – VI Probability 6 Hrs

6.1 Introduction, Random variable
6.2 Discrete and continuous Probability Distributions
6.3 Bionomical Distribution
6.4 Poisson Distribution
6.5 Normal Distribution

Text Books

1. *Advanced Enggineering Mathematics* by Erwin Kreyszing (8th Edition,Wiley Eastern Ltd.) ISBN-9971-51-283-1

2. *Advanced Enggineering Mathematics* by B. S. Grewal (40th Edition, Oct 2007, Khanna Publication Delhi) ISBN-81-7409-195-5.

3. *Advance Engineering Mathematics* by R. K. Jain and S. R. K. Iyengar (Third Edition, Narosa Publication) ISBN-978-81-7319-730-7.

Reference Books:

1. *Applied Mathematics* (Volume-I ISBN- 81-85825-10-6 & Volume-II ISBN-81-85825-07-06) by P. N. Wartikar and J. N. Wartikar (Pune Vidyarthi Griha Prakashan, Pune)
2. *Higher Engineering Mathematics* by B. V. Ramana (Tata McGraw Hill) ISBN-0-07-063419-X.
3. *Engineering Mathematics* by Thomas and Finney.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Structure
(Department: Computer Science & Engineering)

2.DIGITAL SYSTEMS & MICROPROCESSOR

Teaching Scheme: L: 4

Evaluation Scheme	Theory	Test	Minimum Passing Marks
	80 Marks	20 Marks	40%

Course Objectives

- To make students aware of basics of Digital Systems Design.
- To learn and understand design and construction of combinational and sequential circuits.
- To introduce basic components of microprocessors

UNIT 1 Introduction and Logical Functions Representation (7 Hrs)

Analog, Discrete-time & Digital Signals, Difference between Analog & Digital systems, Universal Gates, Signed Number Representation, Binary Codes, Boolean algebra & its minimization – SOP and POS Form, Don't care combinations, K-Map (up to 5 variables), and Quine McCluskey method.

UNIT 2 Combinational Logic Designs (7 Hrs)

Half/Full Adder/Subtractor, Carry Look Ahead Adder, Carry Save Adder, 4-bit Parallel Adder/Subtractor, BCD Adder/Subtractor, Array Multiplier, Booths Multiplier, Decoders-Encoders, Multiplexers-Demultiplexers, Parity Checker, Comparator, and Barrel Shifter.

UNIT 3 Sequential Logic Designs (6 Hrs)

Review of Latches and Flip-Flops, Master Slave JK and D Flip-Flops, Conversion of Flip-Flops, Counters: 4 bit Ripple & Synchronous, Universal Shift Register, Finite State Machine, State Diagram, Moore & Mealy Machines, Semiconductor

Memories – Memory Organization, RAM & its types, ROM & its types.

UNIT 4 8085 Architecture and Pin Diagram (6 Hrs)

Definition of Microprocessor, Microprocessor based system, Evolution of Microprocessors, Properties of 8085, Detailed Architecture of 8085, Pin Diagram of 8085, and Interrupt Circuitry of 8085, De-multiplexing of address and data bus, 8085 clock & reset circuit, Addressing Modes of IC8085.

UNIT 5 8085 Instruction Set and Programming (8 Hrs)

Instruction set of 8085, 8085 programming model, Assembly language Programming – Simple, Loop, Conditional, Subroutines etc., Memory mapped I/O, I/O mapped I/O, Exhaustive and Partial Decoding, Memory Interfacing with 8085.

UNIT 6 Data Converters (5 Hrs)

Programmable Peripheral Interface IC8255, Introduction to ADC and DAC and their types, ADC IC0808/09 and DAC IC0808/09, Interfacing IC8255 to 8085, Interfacing ADC IC0808/09 and DAC IC0808/09 to 8085.

Outcomes

By the end of the course

- Students will understand fundamentals of Digital Systems, Microprocessors, Memories and Peripherals Interfacing
- Students will be comfortable with Assembly Language Programming and Logical Circuit Designing.

Text Books

1. A. Anand Kumar, *"Fundamentals of Digital Circuits"*,Prentice Hall of India.
2. John F. Wakerly, *"Digital Design: principles and practices"*, Third Edition, Pearson Education.
3. R. P. Jain, *"Modern Digital Electronics"*, Third Edition Tata McGraw-Hill.
4. Ramesh S. Gaonkar *"The Microprocessor: Architecture, Interfacing, Programming and Design,"*, Wiley Eastern.

5. B. Ram, *"Fundamentals of Microprocessors and Microcomputers"*, Dhanpat Rai & Sons.

Reference Books

1. Charles Roth, *"Fundamentals of Logic Design"*, Cenege Learning India Pvt. Ltd.
2. Douglas Hall *"Digital Systems and Microprocessors"*, , Pearson Education Asia.
3. William I. Fletcher *"Engineering Approach to Digital Design"*, , PHI.
4. Samuel C. Lee *"Digital circuits and Logic design"*, , Prentice Hall.
5. John P. Hayes *"Computer Architecture and Organization"*, , McGraw Hill International Editions (Computer Science Series).
6. U.V.Kulkarni and T.R.Sontakke *"The 8085 Basics, Programming and Applications"*, , Sadhusudha Prakashan.
7. *"*, R. Singh, B. P. Singh *"Microprocessors, interfacing and Applications*, New Age International

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Structure
(Department: Computer Science & Engineering)

3.DISCRETE MATHEMATICS

Teaching Scheme: L: 4

Evaluation Scheme	**Theory**	**Test**	**Minimum Passing Marks**
	80 Marks	20 Marks	40%

Course Objectives

- This course is a foundation for the development of more advanced mathematical concepts
- This course presents the foundations for many basic computer related concepts and provide a coherent development and common theme for these ideas..
- Use appropriate set, function, or relation models to analyze practical examples, interpret the associated operations and terminology in context.
- Formulate problems precisely, solve the problems, apply formal proof techniques, and explain their reasoning clearly.

UNIT-1 **(10 Hrs)**

Logic & Proofs Introduction, statements and Notation, Connectives - negation, conjunction, disjunction, Conditional, biconditional, statement formulas and truth tables, well formed formulas, Tautologies, Equivalence of formulas, Duality law, Tautological implications, functionally complete sets of connectives, other connectives, Normal & Principle normal forms. Predicates & Quantifiers, Methods of Proofs, Mathematical Induction.

UNIT-2 **(10 Hrs)**

Sets, Relations & Functions
Basic concepts of set theory Set, Types of operations on sets, Ordered pairs, Cartesian product, Principle of inclusion and exclusion**.**
Relations, Functions Representation of discrete structures, Relations, Properties of binary relations, Partition and Covering of set, Equivalence relation, Composition, Closure of Relation, Warshall's algorithm, POSET and, Functions - Types, Composition of functions, Inverse functions**,** Recurrence relations, Linear recurrence relations, Solution to recurrence relations, Generating functions. (Growth of Functions: Big-*O* notation)

UNIT-3 Graphs (5 Hrs)

Basic terminology, multi graphs and weighted graphs, paths and circuits, shortest path Problems, Euler and Hamiltonian paths and circuits, factors of a graph**.**

UNIT-4 Trees (5 Hrs)

Trees, rooted trees, path length in rooted trees, binary search trees, spanning trees and cut set, theorems on spanning trees, cut sets, circuits, minimal spanning trees**.**

UNIT-5 Lattices (6 Hrs)

Lattice as POSET, definition, examples and Properties, Lattice as algebraic systems, Special lattices, Boolean algebra definition and examples, Boolean functions, representation and minimization of Boolean functions .

UNIT-6 Algebraic Systems (4 Hrs)

Semi groups and monoides, properties and examples. Algebraic Systems, Groups, Semi Groups, Monoid, Subgroups, Codes and Group codes, Isomorphism and Automorphisms, Homomorphism and Normal Subgroups.

Oucomes

By the end of the course,

- Students will understand a number of substantive and diverse topics covered in this course.
- The student is expected to develop an important new skill, the ability to write a mathematical proof, which is an excellent training for writing good computer programs.

Text Books

1.**"***Discrete Mathematical Structures***",** B. Kolman, R. Busby and S. Ross, Pearson Education, 4th Edition, 2002, ISBN 81-7808-556-9**.**

2. "*Discrete Mathematics and Its Applications*", Kenneth H. Rosen, Tata McGraw-Hill, 5th Edition, 2003,ISBN 0-07-053047-5

3. "*Discrete Mathematical structures with applications to computer science*", J. P. Tremblay and R. Manohar, MGH International.

Reference Books

1."*Elements of Discrete Mathematics*", C. L. Liu, Tata McGraw-Hill, 2nd Edition, 2002,ISBN 0-07-043476-X

2."*Theory and Problems in Abstract algebra*", Schaums outline series, MGH International.

3. "*Discrete Mathematics* ", Lipschutz, Lipson,, Tata McGraw-Hill, 2nd Edition, 1999,ISBN 0-07-463710--X

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Structure
(Department: Computer Science & Engineering)

4.DATA STRUCTURES

Teaching Scheme: L: 4

Evaluation Scheme	**Theory**	**Test**	**Minimum Passing Marks**
	80 Marks	20 Marks	40%

Course Objectives

- To understand the fundamentals of data structures and data representations.
- To define high level of abstraction of the needed linear and non linear data structure.
- To study the representation, implementation and applications of linear and non linear data structures

UNIT I **(7 Hrs)**

- Review of 'C': Arrays, pointers, Functions (call by values and call by reference), Recursion, and Files (open, close and edit files in C).
- Introduction to data structures: Information and Meaning, Primitive data structures and Non primitive data structures, Arrays, Structures, Pointers, Abstract Data Types (ADT), Realization of ADT in 'C'.

UNIT II **(7 Hrs)**

- Linear Data structures
- The Stack: Operations (PUSH and POP), Representation in C.
- Applications of stacks: Infix, Prefix and Postfix expressions and their Conversions, Recursion.
- Queues: Operations (insertion and deletion), Representations, Priority queues.

UNIT III **(7 Hrs)**

- Linked Lists: Singly Linked list and its operations, Circular lists and its operation, Doubly linked list and its operations.
- Applications of linked lists: Arithmetic of long integers, Dictionary, Polynomial representations and its arithmetic, Implementation of stack and queue using linked organization.

UNIT IV **(7 Hrs)**

- Non linear Data structures: Trees, Binary trees, Binary tree representations (sequential and linked), Tree traversals (Preorder, Postorder and Inorder)
- Applications of Binary Trees: Evaluating expressions tree, Binary search tree, and Game tree.

UNIT V **(6 Hrs)**

- Graphs: Definition, Types of Graphs, Representations of graph: Adjacency List, Adjancency matrix representations
- Traversal of graph: Breath First Search, Depth First Search.

UNIT VI **(6 Hrs)**

- Indexing and Searching: Basic Indexing and search techniques, Hashing, Tree Indexing, Construction of hash tables and Construction of binary search trees

Outcomes

By the end of course

- Students will learn to choose the appropriate data structure for modeling a given problem.
- Students will be able to understand and implement various data structures along with their application.

Text books

1. *"Data Structures using C and C++"*, Y. Langsam, M.J. Augenstein, A.M. Tenenbaum, Second Edition ,PHI India. ISBN- 81-203-1177-4
2. *"Fundamentals of Data Structures in C"* , E. Horowitz ,S.Sahani, S.Anderson-Freed Universities Press ,2008
3. *"An Introduction to Data Structures with applications"*,J.P.Trembly and P.G.Sorenson, Second Edition, Tata McGraw Hill, 1981

Reference Books

1. *" Data Structrues"* , Richard F. Gilberg, B.A. Forouzen, Thomson's Books/cole publishing.ISBN –981-240-622-0
2. *"Data Structures and Algorithms"*, A. Aho, J. Hopcroft, J. Ulman, Pearson Education, 1998

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Structure
(Department: Computer Science & Engineering)

5.DATA COMMUNICATION

Teaching Scheme: L: 4

Evaluation Scheme	Theory	Test	Minimum Passing Marks
	80 Marks	20 Marks	40%

Course Objectives

- To understand basic concepts of Data or computer communication as a base of Computer Network.
- To learn about basic connecting devices, such as Ethernet, Modems, Cables or other transmission medias.

UNIT I **(7Hrs)**

- Introduction: Data communication, Networks, internets
- Network models: layered tasks, The OSI model and layers, TCP/IP protocol suit, Addressing
- Data and Signals: Analog and Digital, Periodic and analog, digital signals, transmission impairment, data rate limits, Performances.

UNIT II **(6 Hrs)**

- Digital Transmission: Digital to analog and Analog to Digital conversions, Transmission modes.
- Bandwidth Utilization: Multiplexing, Spread Spectrum

UNIT III **(6 Hrs)**

- Transmission media: Guided media, unguided media
- Switching: Circuit Switching Network, Data gram network, virtual- circuit network, and structure of a switch.

UNIT IV **(6 Hrs)**

- Data Transmission: Telephone network, Dial-up modems, Digital subscriber line, cable TV network and Data transfer.

- Error Detection and Correction: Block coding, linear block code, cyclic code, and checksum.

UNIT V **(8 Hrs)**

- Data link control: Framing, flow and error control, protocols, noisy and noiseless channels, HDLC, Point-to-point protocol.
- Multiple Access: Random access, Controlled access, channelization.
- Wired LAN (Ethernet): IEEE standards, Standard Ethernet, changes in the standard, fast Ethernet, Gigabit Ethernet.

UNIT VI **(7 Hrs)**

- Wireless LANs: IEEE 802.11, Blue tooth.
- Connecting LANs: Connecting Devices, backbone network, Virtual LAN.

Outcomes

- Students will understand basics data communication and two lower layers i.e. physical and data link layers.
- Students will understand networking building blocks such as Ethernet, Cables and modems

Text Book

1. *" Data Communications and Networking"*, Beherouz A. Forouzan, 4th Edition, Tata McGraw Hill, 2006. ISBN 13: 978-0-07-063414-5

Reference Books

1. *" Data and computer Communication"* ,William Stalling, 7th Edition, Pearson Education, 2004, ISBN 81-297-0206-1.
2. *"Data Communications and Networks"*, Godbole A., Tata McGraw-Hill Publications
3. *"Communication Networks - Fundamental Concepts and Key Architectures"*, Garcia L., Widjaja I., 2nd edition, Tata McGraw-Hill, 2000,
4. *"Data Communications"*, Gupta P., PHI, 2004

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Structure
(Department: Computer Science & Engineering)

6.DIGITAL SYSTEMS & MICROPROCESSOR LAB

Teaching Scheme: P: 2

Evaluation Scheme	Term Work	Pract	Minimum Passing Marks
	25 Marks	25 Marks	40%

Term Work:

- Instructor will frame experiments based on the suggested experiments as given below. Instructors are expected to incorporate variations in list.
- Students will submit Term Work in the form of a journal that will include at least 12 experiments from all the groups. Four experiments can be chosen from each group. Each experiment will consists of circuit diagram, pseudo-algorithm, program listing with proper documentation and printout of the output.
- Practical Examination will consist of Performance and Viva-voice Examination based on the term work.
- The assessment will be based on the following –

  1. Performance in the practical examination
  2. Record of programs submitted by the candidate.
  3. Setting goals higher than expected from problem statement
  4. Innovation & Creativity.
  5. Team building skills
  6. Technical writing skills

A. Combinational Logic Design

1. Basic Gates designing using Universal gates
2. Designing 4bit Full Adder/Subtractor using IC 7483
3. Designing 4 bit BCD adder/Subtractor using IC 7483.
4. Code converters design (Excess-3 to BCD and vice versa, Gray to BCD and vice versa)
5. Multiplexers (Realization of Boolean expressions)
6. Demultiplexers (Realization of ROM)
7. Parity generator / detector.
8. BCD to 7 segment display converter.

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. 4 bit Ripple counter using JK flip-flops.
4. Sequence generator using JK flip-flop.
5. Up-down counter using JK flip-flop.
6. Modulo N counter using 7490 & 74190 (N>10).
7. Pseudo random number generator.
8. Design of a barrel shifter

C. Microprocessor Programming

1. Multiplication of two 4 and 8 bit numbers using counter approach and Logical Shift & Add approach.
2. Division of 8 bit number by 4 bit number.
3. Addition/Subtraction of N 8 bit numbers and N BCD numbers.
4. Sorting array of N numbers in ascending/descending order.
5. Finding smallest and largest numbers from an array of N numbers.
6. Checking the status of PSW after ALU operation.
7. Finding number of Even/Odd numbers from list of N numbers.
8. Finding number of Positive/Negative numbers from list of N numbers.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Structure
(Department: Computer Science & Engineering)

7.DATA STRUCTURES LAB

Teaching Scheme: P 4

Evaluation Scheme	**Term Work**	**Pract**	**Minimum Passing Marks**
	25 Mark	50 Marks	40%

Term Work

- Instructor will frame assignments based on the suggested assignments as given below. Instructors are expected to incorporate variations in list.
- Students will submit Term Work in the form of a journal that will include at least 16 assignments. Each programming assignment will consists of pseudo-algorithm, program listing with proper documentation and printout of the output.
- Practical Examination will consist of Performance and Viva-voice Examination based on the term work.
- The assessment will be based on the following –

1. Performance in the practical examination
2. Record of programs submitted by the candidate.
3. Setting goals higher than expected from problem statement
4. Innovation & Creativity.
5. Team building skills
6. Technical writing skills

Suggested List of Assignment

1. For any new data type define data structure using structure in C. (e.g. Complex numbers, Polynomials) and write a program in C for its arithmetic.
2. Create new file in write mode & open another file in read mode and copy second file into first file.
3. Implement Tower of Hanoi using recursion.
4. Implement Stack data structure and its operations (PUSH and POP).
5. Implement an application of stack, Conversion of prefix to postfix expression and Conversion of infix to postfix expression.
6. Implement circular queue and double ended queue using arrays.
7. Write a menu driven program to perform following operations on singly linked list: Create, Insert, Delete, Display, Reverse

8. Create two doubly linked lists. Sort them after creation using pointer manipulation. Merge these two lists into one list so that the merged list is in sorted order. (No new node should be created.)
9. Represent polynomial as a circularly linked list and write a menu driven program to perform addition, multiplication and evaluation.
10. Write a program to create generalized linked list and perform following operations: copy, depth, equivalence.
11. Implementation of application of Linked lists: Dictionary of various symbols.
12. Implementation of application of Linked lists: Expressing 40-digit integer using linked list and its arithmetic.
13. Write a program to perform various string operations such as copy, length, reversing, palindrome, concatenation and to find occurrence of a sub-string using and without using library functions.
14. Represent sparse matrix using array and perform simple transpose, fast transpose and matrix addition.
15. Create binary tree and perform recursive and non-recursive traversals
16. Create binary tree. Find height of the tree and print leaf nodes. Find mirror image, print original and mirror image using level-wise printing
17. Implementation of binary tree data structure and perform Pre-order, Post-order and In-order traversals.
18. Create binary in-order threaded tree and perform traversals.
19. Implementation of graph representations with some examples
20. Represent graph using adjacency list and perform Breadth First search and Depth First Search
21. Represent graph using adjacency list or matrix and generate minimum spanning tree using Prim's algorithm.
22. Implement simple index file.
23. Store various files using Hash tables and retrieve files using Hashing.
24. Design a simple game (eg. Tic-tac-toe) using game tree.
25. Implement the Mini Project of Student Database using Linked list for following requirements:
    i. Creation of Student Database in memory containing student ID, Name, Name Initials, Address, Contact No and Date of Birth .
    ii. Insertion, Deletion, Modification of student record for a given student ID.
    iii. Sorting on name initials and searching a particular student record on name initials

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Structure
(Department: Computer Science & Engineering)

8.OBJECT ORIENTED PROGRAMMING LAB

Teaching Scheme: L: 2 **P** 2

Evaluation Scheme	**Term Work**	**Pract**	**Minimum Passing Marks**
	25 Mark	25 Marks	40%

Course Objectives

- To make student aware of the basic concepts of object oriented programming.
- To study C++ language.

UNIT 1
Introduction to Object Oriented Programming

Introduction to procedural, modular, object-oriented and generic programming techniques, Limitations of procedural programming, Need of object-oriented programming **(3 Hrs)**

UNIT 2
Fundamentals

C++ Programming basics, control statements, structures, objects and classes. Data encapsulation, data abstraction and information hiding, inheritance, polymorphism.
Pointers: Memory management - new and delete operators, pointers to objects, pointers to pointers, this pointer **(4 Hrs)**

UNIT 3
Function

Function: Reference arguments, overloaded functions, inline functions, default arguments, returning by reference, friend functions and static functions.
Virtual Functions: Accessing Normal and Virtual member functions, early & late binding, Pure virtual functions, Abstract classes, Virtual base classes

(4 Hrs)

UNIT 4
Operator Overloading & Inheritance

Overloading unary and binary operators, Overloading extraction and insertion operators, data Conversion.

Inheritance: Concept and need, single inheritance, base and derived classes, friend classes, types of inheritance, hybrid inheritance, member access control, static class, multiple inheritance,

(4 Hrs)

UNIT 5

File & Streams

Concept of a file, file operations, streams, opening and closing a file, detecting end-of-file, file modes, file pointer, structures and files, classes and files, sequential file processing, Error handling.

(2 Hrs)

UNIT 6

Advanced C++ features

Templates, Exception handling, Library organisation and containers

NameSpaces: Introduction, Rules of namespaces

(3 Hrs)

Outcomes

- Student will understand the fundamental concepts of object oriented programming.
- Student will be comfortable with the C++ language.

Reference Books:

1. "*Object oriented programming in Turbo C++*", Robert Lafore, Galgotia. ISBN 81-85623-22-8
2. "*C++ programming language*", Bjarne Stroustrup, AT & T.ISBN 81-7808-126-1
3. "Object Oriented Programming with C++", E. Balaguruswamy, Tata McGraw-Hill PublishingCompany Ltd, New Delhi ISBN 0 – 07 – 462038 – X.
4. "*Programming with C++*", D. Ravichandran, TMGH. ISBN 0-07-463349-X

Term Work

List of Practical

1. Write a program based on call by value and call by reference
2. Write a program to demonstrate concept of class. For example: create class matrix, class string, class car, class date, class time, class person etc
3. Write a program to demonstrate following Function concepts
    a. Function overloading
    b. Constructors of all types
    c. Default parameters, returning by reference

4.
    a. Demonstration of friend function
    b. Demonstration of static function

5. Write a program to demonstrate
    a. Operator overloading –for unary as well as binary operation.
    b. Apply above concept on matrix and string classes created above.

6. Write a program to demonstrate C^{++} s capability of all types of inheritance
    a. Single, multiple, multivalued
    b. Virtual function.
    c. Abstract class
    d. Runtime polymorphism

7. Write a program for new and delete operators, pointers to objects.
8. Write a program for pointers to pointers, this pointer.
9. Write a program for Templates, Exception handling.
10. Write a program for Name Spaces

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Structure
(Department: Computer Science & Engineering)

9.DATA COMMUNICATION LABORATORY

Teaching Scheme: P: 2

Evaluation Scheme	**Term Work**	**Pract**	**Minimum Passing Marks**
	25 Marks	25 Marks	40%

Term Work:

- Instructor will frame assignments based on the suggested assignments as given below. Instructors are expected to extend the suggested list.
- Students will submit Term Work in the form of a journal that will include all assignments. Each programming assignment will consists of pseudo-algorithm, program listing with proper documentation and printout of the output.
- In study experiments students are expected to see the devices in the laboratory /institute campus. List the detail specifications and functioning. About the telephone network it is expected to visit telephone exchange/BSNL office and study actual working
- Practical Examination will consist of Performance and Viva-voice Examination based on the term work.
- The assessment will be based on the following –
  1. Performance in the practical examination
  2. Record of programs submitted by the candidate.
  3. Setting goals higher than expected from problem statement
  4. Innovation & Creativity.
  5. Team building skills
  6. Technical writing skills

.

Suggested list of practical assignment.

1. Study of Telephone Network
2. Study of Dial-up Modems, DSL lines, Cable modems
3. Implement in C linear block codes.
4. Implement in C cyclic codes.
5. Implement in C checksum.
6. Study of Ethernet card.
7. Study of IEEE 802.3 and IEEE 802.11 standards.
8. Study of Fast Ethernet (for ex. 1000Base-CX)
9. Study of the gigabit Ethernet. (1000Base-Sx)
10. Study of Guided media,

a. Twisted pair
b. Coaxial cable
c. Optical fiber

11. Study of Switches and hubs in your campus.
12. Study of LAN /WAN of your college/ institute campus.
13. Connect two or three computers using wired media (available in your Lab).
14. Data transfer using stop and wait protocol
15. Implement Go-Back-N sender algorithm
16. Implement Go-Back-N receiver algorithm

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Structure
(Department: Computer Science & Engineering)
(Semester IV)

1.ENGINEERING MATHEMATICS IV

Teaching Scheme: L: 4

Evaluation Scheme	Theory	Test	Minimum Passing Marks
	80 Marks	20 Marks	40%

Objectives: The objective of this course is to assist the students of Engineering to acquire different techniques of integral transforms like L.T., F.T. and Z Transforms, Numerical methods and to enhance skills of these techniques in different areas of Engineering.

UNIT - I Laplace Transforms and Applications 10 Hrs

1.1 Definition, Existence of L.T.
1.2 Properties: Linearity, Change of scale, First shifting, Second shifting, Multiplication by t, Division by t., L.T. of derivative and Integral
1.3 Inverse L.T., Methods of obtaining inverse L.T., Convolution Theorem
1.4 L.T. of special functions
  i) Unit (Heaviside) step function
  ii) Unit Impulse function (Dirac delta function)
  iii) Periodic functions
1.5 Applications of L.T. to initial value problems, simultaneous differential equations

UNIT – II Fourier Series and Fourier Transforms 8 Hrs

2.1 Fourier series: Definition, Dirichlet's conditions, Euler's formulae
2.2 Fourier series over $(0, 2\pi)$, Functions having points of discontinuity
2.3 Change of interval
2.4 Expansions of Even and Odd functions, Half range series
2.5 Fourier Integrals: Definition complex form of Fourier integral
2.6 Fourier Sine and Cosine Integral
2.7 Fourier Transforms, Fourier Sine and Cosine Transforms

UNIT – III Z Transforms 4 Hrs

3.1 Definition: Standard properties, ZT of standard sequences
3.2 Inverse Z Transform
3.3 Applications of Z Transform to simple difference equations

UNIT – IV Complex Analysis – I **6 Hrs**

4.1 Introduction of complex variable, limit, continuity and derivative
4.2 Analytic function, C-R equation in cartesian and polar form
4.3 Harmonic functions, Orthogonal System
4.4 Construction of analytic function $f(z) = u + iv$ if u or v or $u \pm v$ are given

UNIT – V Complex Analysis – II **8 Hrs**

5.1 Complex Integration: Line Integral of complex plane
5.2 Cauchy's Integral Theorem for simply and multiply connected regions
5.3 Cauchy's Integrated formula
5.4 Series of Complex terms: Convergence, Behavior radius of convergence of series
5.5 Taylor's and Laurent's series [without proof]
5.6 Singularities, Residues, Residues Theorem, Evaluation of real definite integrals
5.7 Conformal mappings: Translation, Magnification rotation and Bilinear Transformation

UNIT – VI Numerical Analysis **4 Hrs**

6.1 System of Linear simultaneous equations: 1) Direct Methods of solution- Gauss Elimination method, Gauss Jordon Method.
6.2 2) Iterative Methods of solution: Gauss Seidal Iteration Method.

Text Books

1. *Advanced Enggineering Mathematics* by Erwin Kreyszing (8th Edition,Wiley Eastern Ltd.) ISBN-9971-51-283-1

2. *Advanced Enggineering Mathematics* by B. S. Grewal (40th Edition, Oct 2007, Khanna Publication Delhi) ISBN-81-7409-195-5.

3. *Advance Engineering Mathematics* by R. K. Jain and S. R. K. Iyengar (Third Edition, Narosa Publication) ISBN-978-81-7319-730-7.

Reference Books:

1. Applied Mathematics (Volume-I ISBN- 81-85825-10-6 & Volume-II ISBN-81-85825-07-06) by P. N. Wartikar and J. N. Wartikar (Pune Vidyarthi Griha Prakashan, Pune)

2. Higher Engineering Mathematics by B. V. Ramana (Tata McGraw Hill) ISBN-0-07-063419-X.

3. Engineering Mathematics by Thomas and Finney.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Structure
(Department: Computer Science & Engineering)

2.COMPUTER ALGORITHMS

Teaching Scheme: L: 4

Evaluation Scheme	Theory	Test	Minimum Passing Marks
	80 Marks	20 Marks	40%

Course Objectives

1. To learn how to analyze an algorithm theoretically.
2. To study basic methods of problem solving and algorithms in modern computing systems.

UNIT I **(7 Hrs)**

- **Role of algorithms in Computing:** Algorithms , Algorithms as a technology
- Getting started: Insertion sort, Analyzing algorithms, Designing algorithms.
- Growth of Functions: Asymptotic Notations: o , θ , Ω , Complexities (Time and space).

UNIT II **(7Hrs)**

- **Recurrences**: The substitution method , Tthe recurrence tree method and master's method.
- Sorting algorithms: Heaps, Maintaining a heap property, Building a heap, Heap sort, Priority queues and Quick sort.

UNIT III **(6 Hrs)**

- **Dynamic Programming –** Assembly line scheduling, Matrix chain multiplication, elements of dynamic programming, Longest common subsequences and optimal binary search trees.

UNIT IV **(8 Hrs)**

- **Greedy method** – An activity selection problem, Elements of greedy strategy, Huffman codes.

- Data structures for disjoint sets: Disjoint set operations, Linked list representation of disjoint sets, and Disjoint forest.

UNIT V **(8Hrs)**

- **Graph algorithms**: Minimum cost spanning trees, Single source shortest paths, All pair shortest paths.

UNIT VI **(6Hrs)**

- **NP-Hard and NP-complete problems** - Basic concepts, Reducibility, Cook's theorem (without proof), NP-Hard graph problems.

Outcomes

- Students will be able to analyze any algorithms and able to calculate theoretical complexity of it.

- Students will understand the problem solving methods such as recurrence, greedy method and dynamic programming.

- Students will be aware of Np-Hard and Np-complete concepts.

Text Books

1. *"Introduction to Algorithms"*,T.H.Cormen, C.E. Leiserson, R.L. Rivest , The MIT press, Cambridge, Massachusetts and McGraw Hill, 1990 ISBN-81-203-2141-03
2. *"Fundamentals of Computer Algorithms"*, E. Horowitz and S. Sahni, S Rajasekaran, Third Edition, Golgotha Publications.

Reference books

1. *"The Design and Analysis of Computer Algorithms"*, A.V. Aho, J.E.Hopcroft and J.D.Ullman, Addison Wesley,
2. *"Fundamentals of Algorithms"*,G. Brassad and P. Bratley, PHI India,1996 ISBN-81-203-1131-0

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Structure
(Department: Computer Science & Engineering)

3.PRINCIPLES OF PROGRAMMING LANGUAGES

Teaching Scheme: L: 4

Evaluation Scheme	Theory	Test	Minimum Passing Marks
	80 Marks	20 Marks	40%

Course Objectives

- To study the important and general structural characteristics of high-level languages and associated implementation concepts and strategies.
- To understand the core concepts of different programming paradigms.
- To teach the skills needed for comparing and evaluating the structure and design of programming language
- The intent of the course is not teaching a specific programming language; instead, using several languages to exemplify the main concepts and constructs.

UNIT I **(4 Hrs)**

Introduction, Abstractions in Programming Languages, Computational paradigms, Language Definition, Language Translation, History, Language design principals.

UNIT II **(7 Hrs)**

Syntax, Lexical Structure of Programming Languages, Context-Free Grammars and BNFs, Parse Trees and Abstract Syntax Trees, Ambiguity, Associativity and precedence, EBNFs and Syntax Diagrams, Parsing Techniques and Tools, Lexics Versus Syntax Versus Semantics, Basic Semantics, Attributes, Binding and Semantic Functions, Declarations, Blocks and Scope, The Symbol Table, Name Resolution and Overloading , Allocation, Lifetimes and the Environment, Variables and Constants, Aliases, Dangling References and Garbage.

UNIT III **(6 Hrs)**

Data Types and Type Information, Simple Types, Type Constructors, Type Nomenclature in Sample Languages, Type Equivalence, Type Checking, Type Conversion, Polymophic Type Checking , Explicit Polymorphism, Expressions, Conditional Statements and Guards, Loops and Variation on WHILE, The GOTO Controversy, Exception Handling.

UNIT IV **(9 Hrs)**

Procedures and Environments, Procedure Definition and Activation, Procedure Semantics, Parameter Passing Mechanisms, Procedure Environments, Activations and Allocation, Dynamic Memory Management, Exception Handling and Environments, Abstract Data Types and Modules , The Algebraic Specification of Abstract Data Types , Abstract Data Type Mechanisms and Modules , Separate Compilation, C++ Namespaces, and Java Packages, Ada Packages, Modules in ML, Modules in Earlier Languages, Problems with Abstract Data Type Mechanisms, The Mathematics of Abstract Data Types.

UNIT V **(9 Hrs)**

Object-Oriented Programming, Software Reuse and Independence, Java: Objects, Classes and Methods, Inheritance, Dynamic Binding , C++ , Smalltalk , Design Issues in Object-Oriented Languages, Implementation Issues in Object-Oriented Languages, Functional Programming, Programs as Functions, Functional Programming in an Imperative Language, Scheme: A Dialect of LISP, ML: Functional Programming with Static Typing , Delayed Evaluation, Haskell, A Fully-Curried Lazy Language with Overloading, The Mathematics of Functional Programming I: Recursive Functions, The Mathematics of Functional Programming II: Lambda Calculus

UNIT VI **(5 Hrs)**

Logic Programming, Logic and Logic Programs, Horn Clauses, Resolution and Unification, The Language Prolog, Problems with Logic Programming, Extending Logic Programming: Constraint Logic Programming and Equational Systems.

Outcomes

By the end of the course,

- Students will use a better choice of a programming language to solve a particular problems
- Students will learn to demonstrate facility with BNF for specifying programming language syntax and understand the different approaches to defining programming language syntax and semantics.

- Students will be prepared with the core knowledge in design and implementation of programming languages to pursue advanced courses in programming design and implementation or a course in compilers.

Text Book

1. *"Programming Language Principles and Practices"*, Kenneth C. Louden, Second Edition, Thomson 2003.
2. "*Concepts of Programming Languages"*, Robert W. Sebesta, Eigth Edition, Addison Wesley, 2007. ISBN 978-00321-49362-0.

Reference Books

1. "*Programming Languages: Concepts and Constructs*", 2nd Edition, Ravi Sethi, Pearson Education Asia, ISBN 81 – 7808 – 104 – 0.
2. "*The C Programming Language'*, B. W. Kernighan, D. H. Ritchie, Second Edition, Prentice-Hall, Edition, 1988
3. "*The C++ Programming Language"*, B. Stroustrup, Third Edition, Addison-Wesley, 1997

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Structure
(Department: Computer Science & Engineering)

4.SYSTEM PROGRAMMING

Teaching Scheme: L: 4

Evaluation Scheme	**Theory**	**Test**	**Minimum Passing Marks**
	80 Marks	20 Marks	40%

Course Objectives

- This course is a foundation for the Operating System Courses and Compiler Design Courses.
- This course builds the foundation required for understanding the different components of system softwares.

UNIT I **(5 Hrs)**

Introduction

System Software and Machine architecture, Simplified Instructional Computer, Traditional CISC Machines-VAX architecture, Pentium Pro architecture, RISC Machines-Ultra SPARC architecture, PowerPC architecture, Cray T3E architecture, Instruction formats-Addressing modes.

UNIT II **(8 Hrs)**

Assemblers

Elements of Assembly Language Programming, Basic Assembler Functions, Machine Dependent Assembler Features, Machine Independent Assembler Features, Assembler Design Options-One Pass Design options-One Pass Assemblers, Multi Pass Assemblers, Implementation Examples-MASM Assembler.

UNIT III **(8 Hrs)**

Loaders and Linkers

Basic Loader Functions – Design of an Absolute Loader, A Simple Bootstrap Loader, Machine Dependent Loader Features –Relocation, Program Linking, Algorithm and Data Structures for a Linking Loader, Machine Independent Loader Features – Automatic Library Search, Loader Options, Loader Design Options – Linkage Editors, Dynamic Linking, Bootstrap Loaders, Implementation Examples – MS DOS Linker, SunOS Linkers.

UNIT IV **(8 Hrs)**

Macro Processors

Basic Macro Processor Functions – Macro Definition and Expansion, Macro Processor Algorithm and Data Structures, Machine Independent Macro Processor Features – Concatenation of Macro Parameters, Generation of Unique Labels, Conditional Macro Expansion, Keyword Macro Parameters, Macro Processor Design Options- Recursive Macro Expansion, General Purpose Macro Processor, Macro Processing within Language Translators, Implementation Examples- MASM Macro Processor.

UNIT V **(7 Hrs)**

Compilers and Utilities

Basic Complier Functions, Different phases of a compiler, Machine Dependent Compiler Features, Machine Independent Complier Features, Simple one pass compiler, Implementation Examples- SunOS C Compiler.

Unit VI: **(4 Hrs)**

Software Tools for program development, Steps in Program Development, Editors and its Types, Design of an Editor, Debug Monitors, Programming Environments, User Interfaces – Command Dialogs, Structure of User Interface.

Outcomes

By the end of the course,

- Students will have the understanding of the different components of system softwares such as Assemblers, Loaders, Linkers, Macros and Compilers.

Text Books

1. Leland L.Beck, “System Software, An Introduction to System Programming ", Addison Wesley, 1999.

Reference Books

1. D.M.Dhamdhere, " Systems Programming and Operating Systems ",
Tata McGraw Hill Company, 1999.
2. J.J. Donovan “ System Programming” Mc-Graw Hill.
3. A.V.Aho, Ravi Sethi and J.D.Ullman, " Compilers Principles, Techniques and Tools ",
Addison Wesley, 1988

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Structure
(Department: Computer Science & Engineering)

5.ADVANCED MICROPROCESSORS

Teaching Scheme: L: 4

Evaluation Scheme	Theory	Test	Minimum Passing Marks
	80 Marks	20 Marks	40%

Course Objectives

- To make students aware of basics of 80x86 family and advance concepts of Microprocessor architecture.
- To learn the architecture and assembly language programming 0f 8086 Microprocessor.
- To learn peripherals and their interfacing with 8086 Microprocessor.
- To Study NDP and Design of Microprocessor based System.

UNIT 1 8086/88 Architecture and Instruction Set (8 Hrs)

8086 architecture, Addressing modes, Instruction formats, Data transfer, String, Logical, Arithmetic, Branch, Loop, and Processor control instructions, Logical to Physical address translation, Comparison of 8086 with 8088. (8) Hrs

UNIT 2 8086 Operating Modes, Interrupts and Programming (7 Hrs)

Basic 8086 configurations: Maximum and Minimum mode, Support chips 8282, 8284, 8286, 8288, 8086 stack structure, 8086 interrupts, Assembly Language Programming using DEBUG/MASM.

UNIT 3 Memory Interfacing, RISC/CISC and 8051 (8 Hrs)

Memory interfacing with 8086, Even & Odd Memory banks, Features of RISC-CISC Architecture, RISC vs. CISC, difference between microprocessor & microcontroller, 8051 Architecture, I/O configuration, Interrupts, Port structure

and operation, Timer/ counter functions, Memory organization, Addressing mode and Overview of instruction set of 8051.

UNIT 4 Intel x86 Families (5 Hrs)

80x86 Family i.e.80286, 80386, 80486, Real mode and Protected mode, software model of 80x86 family, Registers, Data Organization, Instruction types, Addressing Modes and Interrupts, Interrupt Vector Table, Interrupt Processing Sequence, Special interrupts, comparison of 80x86 microprocessors.

UNIT 5 Peripherals Interfacing and Advance Programming (9 Hrs)

Programmable Interval Timer IC8253, Programmable Interrupt Controller IC8259, Keyboard and Display controller IC8279, IC8086 interfacing with IC8253, IC8259,and IC8279, Using EXTRN and PUBLIC, Macros, Multitasking, Memory Management, Memory Resident program.

UNIT 6 Pentium and Higher End Processors (8 Hrs)

Pentium's Superscalar Architecture, Pipelining, Branch Prediction, Instruction and Data Caches, Floating Point Unit, Segmentation, Paging, Multitasking, Exceptions and interrupts, Virtual and Protected mode, Intel's Dual Core, Core 2 duo and EPIC Technology: architecture, advantages, future scope.

Outcomes

By the end of the course

- Students will understand architecture, programming models, interrupt structure of Advanced Microprocessors
- Students will be comfortable with application oriented Assembly Language Programming for 80x86.

Text Books

1. Liu & Gibson *"Microcomputer system – The 8086/8088 family"*, , PHI.
2. Ray and Bhurchandi *"Advanced Microprocessors and Peripherals"*,, TMH.

3. James L. Antanakos, *"An Introduction to the Intel Family of Microprocessors"*, Pearson Education Asia.
4. *"Microprocessors, Interfacing & Assembly language programming"*, Douglas Hall, TMH.
5. Kenneth J. Ayala, Delmar Learning *"The 8051 Microcontroller: architecture, programming and applications"*, Second edition.

Reference Books

1. Walter A. Triebel, Avtar Singh *"16-Bit and 32-Bit Microprocessors: Architecture, Software, and Interfacing Techniques"*, , Pepperback edition.
2. Tabak Daniel *"Advanced Microprocessor"*, , TMH.
3. A.P. Mathur, Third Edition *"Introduction Microprocessor*, TMH.
4. Muhammad ali Mazidi and Janice Gillispie Mazidi, *"The 8051 Microcontroller and Embedded System* Pearson Education.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Structure
(Department: Computer Science & Engineering)

6.COMMUNICATION SKILLS

Teaching Scheme: L: 2

Evaluation Scheme	**Theory**	**Test**	**Minimum Passing Marks**
	40 Marks	10 Marks	40%

Course Objectives

- To assist the students of engineering to acquire proficiency, both in spoken (oral) and written language.

UNIT I Basics of Communication (3 Hrs)

Definition; Elements of Communication; Cycle of Communication and Feedback.
Types of Communication: Verbal and Nonverbal (Oral, Written, Graphic Language and Body Language) Upward and Downward, Formal and Informal, Media of communication: verbal non-verbal and audio-visual, Principles of Effective Communication,Barriers of Communication.

UNIT II Listening Skills (3 Hrs)

Active Listening: Basic Principles Listening and Note Making, listening to Conversations from IELTS

UNIT III Reading Skills (2 Hrs)
Active Reading, Types, Skimming, Browsing, etc. Reading and Note Making, Comprehension

UNIT IV Speaking Skills (4 Hrs)

Basics of Presentation,Techniques,.Group Discussions. Interview techniques, Public Speaking and Seminars, Pronunciation Basics.

UNIT V Writing Skills (4 Hrs)

Business Correspondence, Business Letters, Job Application, Resume, Paragraph (Technical, Business or General current issues) Reports.

UNIT VI English Grammar and Vocabulary (4 Hrs)

Tenses, Common Errors in English ,Synonyms ,Antonyms,One Word Substitution.

Outcome

- At the end of the course, the students will be able to develop comprehension, improve Vocabulary, grammatical ability, enhance writing skills, correspond with others and enhance skills in spoken English.

Text Books

1. Mohan, Krishna.Meera Banerji, **"***Developing Communication Skills*", New Delhi Macmillan
2. Anjali Ghanekar "*Communication Skills for Effective Management"*, Everest Publishing House.
3. B.V. Pathak "*Communication Skills"*, , Nirali Publication.
4. Meenakshi Raman and Sangeeta Sharma "*Technical Communication"*,, Oxford University Press.

Reference Books

5. "*Communication Skills for Engineers"*, Sunita Mishra and C. Muralikrishna, Pearson Education.
6. "*Basic Communication Skills*", Rutherford A. Person Education, New Delhi.
7. "*Business Correspondence and Report Writing"*, R.C. Sharma and Krishnamohan, Tata McGraw Hill.
8. "*English in situation"*, R.O, Neill, Oxford University Press.
9. "*Organizational Behavior"*, Fred Luthans, McGraw Hill.
10. "*Spoken English for India"*, R.K, Bansal.
11. "*English Grammar and Composition"*, Pal and Suri, Sultan Chand & Son, Educational Publishers.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Structure
(Department: Computer Science & Engineering)

7.JAVA PROGRAMMING LAB

Teaching Scheme: **L:** 2 **P** 4

Evaluation Scheme	Term Work	Pract	Minimum Passing Marks
	25 Mark	25 Marks	40%

Course Objectives

- To learn programming using Core Java.

UNIT 1
Introduction to Java Applications **Hrs**

Introduction, Java Class Libraries, Typical Java Development Environment, Memory Concepts, Arithmetic. **1**

UNIT 2
Introduction to Classes and Objects

Introduction, Classes, Objects, Methods and Instance Variables, Declaring a Class with a Method and Instantiating an Object of a Class, Declaring a Method, Instance variables, *set* Methods and *get* Methods, Primitive Types vs. Reference type double Types, Initializing Objects with Constructors, floating point numbers. **4**

UNIT 3
Control Statements

Control structures, *if* single-selection statement, *if….else* double-selection statement, *while* repetition statement, *do….while* repetition statement, *switch* multi-selection statement, *break* and *continue* statements, logical operators. **3**

UNIT 4
Methods

Introduction, Program modules in Java, *static* methods, *static* Fields and *Class Math*, declaring methods with multiple parameters, scope of declaration, method overloading and Java API packages. **2**

UNIT 5
Arrays
Arrays, declaring and creating arrays in java, examples using arrays, passing arrays to methods, multidimensional arrays, variable-length argument lists, using command-line arguments. **2**

2

UNIT 6
Inheritance and Polymorphism in Java
Inheritance: Superclasses and Subclasses, protected members, relationship between superclasses and subclasses, constructors in subclasses, object class. **3**
Polymorphism: Abstract classes and methods, final methods and classes, polymorphism examples.
UNIT 7
Exception-handling overview, handling *ArithmeticExceptions* and *InputMismatchExceptions*, when to use exception handling, java exception hierarchy, *finally* block. **3**
Introduction to Java Applets

Text Books

1. Cay, S.Horstmann, Gary Cornell Core Java Fundamentals Vol -I (The Sun Microsystems Press Java Series)
2.) Cay S. Horstmann, Gary Cornell Core Java Vol – II (The Sun Microsystems Press Java Series

Reference Books

1. ", Dietel and Dietel *"Java How to program*, PHI.
2. Herbert Schildt *"Java:The Complete Reference"*, , TMGH.

Outcomes

- Student will understand the fundamental concepts of Core Java.
- Student will build the foundation required for learning Advanced Java.

Term Work

- Instructor will frame assignments based on the suggested assignments as given below. Instructors are expected to incorporate variations in list.
- Students will submit Term Work in the form of a journal that will include at least 16 assignments out of which all assignments of Computer Algorithm are compulsory. Each programming assignment will consists of pseudo-algorithm, program listing with proper documentation and printout of the output.
- Practical Examination will consist of Performance and Viva-voice Examination based on the term work.

- The assessment will be based on the following –

  1. Performance in the practical examination
  2. Record of programs submitted by the candidate.
  3. Setting goals higher than expected from problem statement
  4. Innovation & Creativity.
  5. Team building skills
  6. Technical writing skills

Suggested List of Programs for Core Java

1.Assignment(s) to understand object oriented features in Java such as classes, objects and methods, Vectors, strings, Inheritance, access specifiers, Interfaces and Exception Handling

2.Write a simple Applet to illustrate event handling with interactive radio buttons to control font style of a text field. Also provide a text box wherein the user may enter font size.

3.Multi-threading (Displaying an animated clock showing day, date and time using threads)

4.Write a program to retrieve data from some database table(s) using JDBC

5.Write a simple Echo server using Sockets

6. Write a Java program that prints all real solutions to the quadratic equation ax2 + bx + c = 0. Read in a, b, c and use the quadratic formula. If the discriminant b2 -4ac is negative; display a message stating that there are no real solutions.

7. The Fibonacci sequence is defined by the following rule. The fist two values in the sequence are 1 and 1. Every subsequent value is the run of the two values preceding it. Write a Java program that uses both recursive and non-recursive functions to print the nth value in the Fibonacci sequence.

8. Write a Java program that prompts the user for an integer and then prints out all prime numbers up to that Integer.

9. Write a Java program that checks whether a given string is a palindrome or not. Ex: MADAM is a palindrome.

10. Write a Java program for sorting a given list of names in ascending order.

11. Write a Java program to multiply two given matrices.

12. Write a Java Program that reads a line of integers, and then displays each integer,

and the sum of all the integers (use StringTokenizer class)

13. Write a Java program that illustrates how run time polymorphism is achieved.

14. Write a Java program that illustrates the following
a) Creation of simple package.
b) Accessing a package.
c) Implementing interfaces.

15.Write a Java program that illustrates the following
a) Handling predefined exceptions
b) Handling user defined exceptions

16.Write Java programs that use both recursive and non-recursive functions for implementing the following searching methods:
a) Linear search b) Binary search

17. Write a program that will read the value of x and evaluate the following function.

Y = { 1 for x>0 Using (i) nested if statement
{ 0 for x=0 (ii) else if statement
{ -1 for x<0 (iii) conditional operator

18.Write a program to calculate the sum of the odd integer between 1 and 99.

19.Write a program convert binary number to decimal number.

20.Compute area of a rectangle using calling a method.

21. Write a Program in Java to display

1	1	1	$$$	1
2 2	2 3	0 1	$$	2 2
3 3 3	4 5 6	1 0 1	$	3 3 3
4 4 4 4	7 8 9 10	0 1 0 1		4 4 4 4
5 5 5 5 5		1 0 1 01		

22. Write a program that will contain two array. In the first array Store the following:

Banana
Apple
Mango
Date

And store the following price in the second array

1.30 2.20 3.20 4.15

Write a method that will display the product with their corresponding price.

23.Write a program based on call by value and call by reference.

24. Write a program to demonstrate concept of class. For example: create class matrix, class string, class car, class date, class time, class person etc

25.Write a program to demonstrate following Function concepts

i) Function overloading
ii) Constructors of all types
iii)Default parameters, returning by reference

The following Computer Algorithm Programs should be implemented in Java:

1. Sort the sequence of 2000 to 10,000 numbers using quick sort and insertion sort calculate run time of both algorithms.

2. Represent graph using adjacency list or matrix and generate minimum spanning tree using Prims algorithm.

3. Represent graph using adjacency list or matrix and generate minimum spanning tree using Kruskals algorithm.

4. Building fast priority based data retrieval system using optimal binary search tree.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Structure
(Department: Computer Science & Engineering)

8.UNIX AND SYSTEM PROGRAMMING LAB

Teaching Scheme: P 2

Evaluation Scheme	**Term Work**	**Pract**	**Minimum Passing Marks**
	25 Mark	25 Marks	40%

Course Objectives

To study basic UNIX command and implement programming assignments of the subject System Programming using gcc compiler in UNIX/LINUX environment.

Term Work

- Instructor will frame assignments based on the suggested assignments as given below. Instructors are expected to incorporate variations in list.
- Students will submit Term Work in the form of a journal that will include at least 10-12 assignments from both the groups. Each assignment will consists of pseudo-algorithm, program listing with proper documentation and printout of the output.
- Practical Examination will consist of Performance and Viva-voice Examination based on the term work.
- The assessment will be based on the following –

  1. Performance in the practical examination
  2. Record of programs submitted by the candidate.
  3. Setting goals higher than expected from problem statement
  4. Innovation & Creativity.
  5. Team building skills
  6. Technical writing skills

Suggested List of Assignments

Unix Assignments

The following Practical Assignments should be completed in Unix/Linux Basics:

1. Logging to Unix/Linux environment, changing your password, Correcting Typing Mistakes, Format of UNIX Commands.
2. Creating own directories and assigning/removing permissions to it
3. Creating own text files using Unix Text Editors such as Vi etc., printing those files and giving access permissions to it.
4. Studying UNIX/Linux disks and partitions.
5. Executing these useful Unix Commands : cd , pwd, mkdir, rmdir, ls, mv, cp, rm, cat, file, head, tail, chmod, ln, ps, kill, tar, mount, umount, df, find, grep, ifconfig, man, whatis, which, exit or logout
6. Learning how to Compile a C program using gcc compiler and Java Program on Unix/Linux platform.

System Programming Practicals

The Following programs of System Programming should be implemented in this term work using gcc complier on UNIX/LINUX environment.

1. Implementation of Macros.
2. Implementation of Nested macros.
3. Design and implementation of 1 pass assemblers.
4. Design and implementation of 2 pass assemblers.
5. Symbol table generation for input *.c file.
7. Implementation of Toy-code generator.
8. Simulation of linkers.
9. Simulation of loaders.
10. 3-4 assignments on DLL on Linux shared library.
11. Study of different debugger tools.

Refernces Books

1. http://linuxbasics.org/

2. http://personal.denison.edu/~bressoud/cs372-f04/linux_tutorial.pdf

3. http://www.e-fense.com/helix/Docs/Law.Enforcement.Linux.Intro.2.0.5.pdf

4. http://vic.gedris.org/Manual-ShellIntro/1.2/ShellIntro.pdf

5. http://www.zzee.com/solutions/linux-permissions.shtml

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Structure
(Department: Computer Science & Engineering)

9.ADVANCED MICROPROCESSOR LAB

Teaching Scheme: P 2

Evaluation Scheme	**Term Work**	**Pract**	**Minimum Passing Marks**
	25 Mark	25 Marks	40%

Term Work

- Instructor will frame assignments based on the suggested assignments as given below. Instructors are expected to incorporate variations in list.
- Students will submit Term Work in the form of a journal that will include at least 10-12 assignments from both the groups out of that at least 6 assignments should be from group B. Each assignment will consists of circuit diagram, pseudo-algorithm, program listing with proper documentation and printout of the output.
- Practical Examination will consist of Performance and Viva-voice Examination based on the term work.
- The assessment will be based on the following –

  1. Performance in the practical examination
  2. Record of programs submitted by the candidate.
  3. Setting goals higher than expected from problem statement
  4. Innovation & Creativity.
  5. Team building skills
  6. Technical writing skills

Suggested List of Assignments

A. Assembler Programming:

1. Study of Assembler Directives & Operators.
2. WAP to find out largest/smallest from series of 8/16 bit numbers.
3. WAP for addition of series of 8/16 bit numbers.
4. WAP to find out number of even & odd numbers from series of 16bit numbers.
5. WAP to move strings of data words from one offset to another.

6. WAP for arranging unordered series of 16 bit numbers in ascending and descending orders.
7. WAP to convert 0-9 BCD numbers to their equivalent 7 segment codes using Look Up-Table.
8. WAP for addition of two 3x3 matrices.
9. WAP for matrix multiplication of two 3x3 matrices.
10. WAP to convert given BCD number into its equivalent binary form.
11. WAP to convert given 8 bit binary number into its equivalent gray code.
12. WAP to find square root of 2 digit number.

B. 8086 Interfacing:

1. IC8253 interfacing to 8086 for verifying various modes of 8255.
2. IC8255 interfacing to 8086 for verifying various modes of 8255.
3. Interfacing switches and LED's to 8086 using IC8255.
4. Interfacing 4x4 keyboard using IC8279 to 8086.
5. Interfacing IC8279 to 8086 to display text on LCD screen.
6. ADC 0808 interfacing to 8086 to generate digital equivalent of applied analog voltage signal.
7. DAC 0808 interfacing to 8086 to generate various analog waveforms.
8. Stepper motor interfacing to 8086.
9. Traffic Light Controller interfacing to 8086.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Structure
(Department: Computer Science & Engineering)

10.COMMUNICATION SKILL LABORATORY

Teaching Scheme: P 2

Evaluation Scheme	Term Work	Pract	Minimum Passing Marks
	25 Mark	25 Marks	40%

Term Work

- Instructor will frame assignments based on the suggested assignments as given below. Instructors are expected to incorporate variations in list.
- Students will submit Term Work in the form of a journal that will include at least 8-10 assignments.
- Practical Examination will consist of Performance and Viva-voice Examination based on the term work.
- The assessment will be based on the following –

1. Performance in the practical examination
2. Record of programs submitted by the candidate.
3. Setting goals higher than expected from problem statement
4. Innovation & Creativity.
5. Team building skills
6. Technical writing skills

Suggested List of Practical Assignments

1. Draw a communication cycle showing all the elements.
2. Convert the verbal and numerical data into the suitable nonverbal form.
3. Listen to the presentation by the faculty or student and make running notes.
4. Listen to the pre-recorded conversation and answer the questions based on it. (Ref. IELTS: Book 1: CD: 1 and 2.)
5. Read the given passage and answer the questions following it.(Ref. Books for CAT or IELTS)
6. Introducing Yourself (3 to 5 minutes)

7. Presentation for minimum 5 minutes on the given topic (Current Issues or Technical Topics)

8. Situational English (Dialogues and Role-plays)

9. Group Discussion: Live Session

10. Mock-interview: Demo by expert panel.

11. Drafting: i) Business Letter, ii) Resume.

Note: Use of Language Lab and audio-visual modes of communication is strongly recommended, where necessary.