Swami Ramanand Teerth Marathwada University, Nanded

Teaching and Evaluation Scheme for

Second Year Program in Computer Science & Engineering and Information Technology

Semester-III

Effective from 2016-2017

Course Code	Course Name	Teaching Scheme			Credit Structure		
		L	T	P	T	P	Total
CI201	Engineering Mathematics-III	03	01	--	03	--	03
CI202	Discrete Mathematics	03	01	-	03	-	03
CI203	Data Structures	04	-	04	03	02	05
CI204	Digital Systems	04	-	02	03	01	04
CI205	Engineering Economics	03	-	-	03	-	03
CI206	Programming Lab-I	02	-	02	-	02	02
CI207	Professional Communication Skills	02 AUDIT	-	02	-	-	-
Total		21	02	10	15	05	20

Total Credits : 20
Total Contact Hours/Week: 33

Note : Prefix "CI" Courses will be common for both CSE and IT Dept.

Evaluation Scheme					
Theory Credit Course		Theory Audit Course		Practical / Workshop	
ME	ESE	ME	ESE	CE	ESE
20 M	80 M	10 M	40 M	30 M	70 M
Minimum for Passing in Theory, Audit and Practical / Workshop : 40 % Each, ME – Minor Examination, ESE – End Semester Examination and CE - Continuous Evaluation					

Swami Ramanand Teerth Marathwada University, Nanded

Teaching and Evaluation Scheme for
Second Year Program in Computer Science & Engineering and Information Technology

Semester-IV

Effective from 2016-2017

Course Code	Course Name	Teaching Scheme			Credit Structure		
		L	T	P	T	P	Total
CI208	Microprocessors & Microcontrollers	04	-	02	03	01	04
CI209	Computer Algorithms	04	-	-	03	-	03
CI210	System Programming	04	-	-	03	-	03
CI211	Object Oriented Programming with C++	03	01	02	03	01	04
CI212	Numerical Methods & Scientific Computing	03	01	02	03	01	04
CI213	Programming Lab - II	-	-	02	-	01	01
CI214	Mini Project - I	-	-	02	-	02	02
Total		18	02	10	15	06	21

Total Credits : 21
Total Contact Hours/Week: 30

Note :

- **Prefix "CI" Courses will be common for both CSE and IT Dept.**

Evaluation Scheme					
Theory Credit Course		Theory Audit Course		Practical / Workshop	
ME	ESE	ME	ESE	CE	ESE
20 M	80 M	10 M	40 M	30 M	70 M
Minimum for Passing in Theory, Audit and Practical / Workshop : 40 % each, ME – Minor Examination , ESE – End Semester Examination and CE - Continuous Evaluation					

Note: There should be Minor-I and Minor-II test examinations. Minor-I should be conducted on Unit No. :- 1,2 and Minor-II should be conducted on Unit No. :- 3,4. Finally average of two should be considered.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

SEMESTER-III

CI201 ENGINEERING MATHEMATICS-III

Teaching Scheme	**L:3**	**T:1**	
Evaluation Scheme	**ME**	**ESE**	**Minimum Passing Marks**
	20 Marks	**80 Marks**	**40%**

Course Objective

1. To develop logical understanding of the subject.
2. To create the ability to model, solve and interpret physical and engineering problems.
3. To provide an overview of functions of complex variable which helps in solving many engineering problems.

Course Contents:

Unit-I: Linear differential equation of higher order (8 Hrs)

Introduction to linear differential equation of n^{th} order with constant coefficients. Methods of solving linear differential equation with constant coefficients : Shortcut methods and method of variation of parameters.

Equation reducible to linear differential equation with constant coefficients : Cauchy's equation and Legendre's equation, Application of linear differential equations to electrical circuits.

Unit-II: Laplace transform (8 Hrs)

Definition, Existence of Laplace transform, Laplace transform of standard functions, Properties with proof and examples: Linearity, Change of scale, First shifting, Second shifting, Multiplication by t^n, Division by t, Laplace transform of derivative and integral. Evaluation of real integrals using Laplace transform.

Inverse Laplace transform: Using standard properties, Partial fraction, Convolution theorem.

Unit-III:Laplace transform of special function and applications (7 Hrs)

Unit (Heaviside) step function, Unit impulse (Dirac delta) function, Periodic function. Application of Laplace transform to initial value problem, Partial differential equations.

Unit-IV: Function of a complex variable (7 Hrs)

Introduction to complex numbers: Polar form of complex number, Relation between circular function and hyperbolic functions (only concepts no problems), Limits and continuity of complex functions, Derivative of complex functions, Analytic functions, C-R equations in cartesian and polar form, Harmonic function. Construction of an analytic function by Milne Thomson method if only real or imginary parts are given, Conformal transformations, Translation, Magnification, Rotation and bilinear transformations.

Unit-V: Complex integration (8 Hrs)

Line integral, Cauchy's integral theorem, Extension of Cauchy's integral theorem for multiply connected domain and Cauchy's integral formula, Taylor's and Laurent' series (only problems). Singularities and zeros of complex function, calculation of residue and residue theorem and its applications to integration around unit circle.

Unit –VI: Z-transform (7 Hrs)

Definition, Z- transform of some standard function, Properties of Z- transform, Inverse Z-transform, Applications to difference equation.

Course Outcomes : By the end of the course students will be able to

1. Interpret the mathematical results in physical and other forms.
2. Identify, formulate and solve the linear differential equations.
3. Classify and solve the contour integration of complex functions.

Text Books:

1. "*Higher Engineering Mathematics*", by B.S.Grewal, 43rd edition, Khanna Publication, ISBN - 9788174091955.
2. "*Elementary Differential Equations and Boundary Value Problems*", By William E. Boyce, Richard C.DiPrima ,9th Edition, ISBN – 9780470383346.

Reference Books:

1. "*Advanced Engineering Mathematics*", by Jain and Iyenger, Narosa Pub. House, New Delhi, ISBN- 97881842653128.
2. "*Advanced Engineering Mathematics",*by Erwin Kreyszing , 8th Edition, Wiley Eastern Ltd., ISBN-9971-51-283-1.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

CI202 DISCRETE MATHEMATICS

Teaching Scheme	**L:3 T:1**		
Evaluation Scheme	**ME**	**ESE**	**Minimum Passing Marks**
	20 Marks	**80 Marks**	**40%**

Course Objectives:

1. To understand basic concepts of set theory, arithmetic, logic, proof techniques, binary relation and graphs.
2. To use appropriate set, function or relation models for analysis of practical examples and interpretation of the associated operations and terminology in context.
3. To formulate problems precisely, solve the problems, apply formal proof techniques and explain their reasoning clearly.

Course Contents:

Unit-I: Mathematical logic (8 Hrs)

Introduction, Statements and notations, Connectives, Tautologies and contradiction, Logical equivalence and laws of logic, Converse, Inverse, Contrapositive, Normal forms and principle normal forms, Predicates and quantifiers.

Set theory:Sets, Types of set operations, Venn diagram, Cartesian product, Ordered pairs, Principle of inclusion and exclusion, Cardinality of set.

Unit- II: Relations (8 Hrs)

Relation, Properties of binary relation, Equivalence relation, Composition, Representation of relation, Closure of relation, Warshall’s algorithm, Partial order relation.

Recurrence relation: Recurrence relation**,** Linear recurrence relation, Solutions to recurrence relation.

Unit- III: Functions (6Hrs)

Function**,** Types of function, Composition of function, Inverse function. Mathematical Induction: Methods of proof, Mathematical induction.

Unit-IV: Graph theory (6 Hrs)

Graph terminology, Representation of graph, Graph isomorphism, Multigraphs and weighted graph, Connectivity, Euler's path and circuit, Hamiltonian path and circuit, Planar graph, Graph coloring, Cut sets, Factors of graph.

Unit- V: Groups and rings (6 Hrs)

Groups, Semigroups, Monoids, Subgroups, Codes & group codes, Isomorphism and automorphism, Homomorphism & normal subgroups, Rings, Fields.

Unit –VI: Boolean Algebra (6 Hrs)

POSET, Lattices, Principle of duality, Basic properties of algebraic system, Distributive and complemented lattices, Boolean lattices & boolean algebra.

Outcomes: By the end of the course students will be able to

1. Construct mathematical arguments using logical connectives and quantifiers.
2. Use logical notation to define and reason about fundamental concepts such as sets, relations, functions, and integers.
3. Synthesize induction hypotheses and simple induction proofs.
4. Apply the knowledge and skills obtained to investigate and solve a variety of discrete mathematical problems.

Text Books:

1. "*Discrete Mathematics and its Applications*", by Kenneth Rosen, 6th Edition, Tata McGraw-Hill Education Private Ltd., ISBN-978-0-07-064824-1.
2. "*Elements of Discrete Mathematics*", by C. L. Liu, 2nd Edition, Tata McGraw-Hill Education Private Ltd., ISBN-978-0-07-100544-9.

Reference Books:

1. "*Discrete Mathematical Structures*", by B. Kolman, R. Busby and S. Ross, 6th Edition, Pearson Education Ltd., ISBN-978-81-203-3689-6.
2. *"Discrete Mathematical Structures with Applications to Computer Science",* by J. P. Tremblay, R. Manohar, 35th reprint , Tata McGraw-Hill Education Private Ltd., ISBN-978-0-07-463113-3.
3. "*Discrete Mathematics",* by Semyour Lipschutz, Mark Lipson, 2nd edition, Tata McGraw-Hill publication, ISBN-978-0-07-060174-1.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

CI203 DATA STRUCTURES

Teaching Scheme	**L:4**	**P:4**	
Evaluation Scheme	**ME**	**ESE**	**Minimum Passing Marks**
	20 Marks	**80 Marks**	**40%**

`

Course Objectives:

1. To understand the fundamentals of data structures and data representations.
2. To define high level of abstraction of various linear and nonlinear data structures.
3. To study the representation, implementation and applications of linear and nonlinear data structures.

Course Contents:

Unit-I: Introduction **(6 Hrs)**

Basic terminology-data types, Data structure, Abstract data types (ADT), Representation of information.

Arrays: Array definition, Representation and analysis, Single and multidimensional arrays, Address calculation, Application of arrays.

Unit-II: Stack and queue **(8 Hrs)**

Introduction of stack, Operations on stacks: Push & Pop, Array representation of stack, Linked representation of stack, Application of stack, Conversion of infix to prefix and postfix expressions, Evaluation of the postfix expression using a stack.

Recursion: Problem solving using iteration and recursion with examples such as binary search, Fibonacci numbers and Hanoi towers, Tradeoff between iteration and recursion.

Queue: Array and linked representation, Operations on queue, Types of queue: Circular queues, Dequeues and priority queue, Applications of the queue structure.

Unit-III: Linked lists **(8 Hrs)**
Representation of singly linked lists, Operations on linked list, Types of linked list: Circular linked list, Doubly linked list.

Applications of linked list: Polynomial representation and addition, Arithmetic of long integers, Representation of stack and queue using linked list.

Unit-IV: Trees **(8 Hrs)**
Basic terminology, Types of tree, Array and linked representation of binary trees, Binary tree traversal (Inorder, Preorder and postorder).

Application of binary tree: Binary search tree, Game tree, Huffman algorithm, Expression tree.

Unit-V: Graph **(6Hrs)**
Terminology & representations, Adjacency list and adjacency matrix representation of graph, Types of graph.

Graph traversal techniques: Breadth first traversal & Depth first traversal.

Unit-VI: Indexing and searching **(6 Hrs)**
Basics of indexing and searching techniques, Hashing, Construction of hash table, Tree indexing, File organization.

Outcomes: By the end of the course students will be able to

1. Choose the appropriate data structure for modeling a given problem.
2. Understand and implement various data structures along with their application.

Text Books:

1. "*Data Structures Using C and C++*", by Y. Langsam, M.J. Augenstein, A.M. Tenenbaum, Second Edition, PHI India., ISBN- 81-203-1177-4 .

2. "*Data Structures: A Pseudocode Approach with C*", by Richard Gilberg, Behrouz A. Forouzan, Second Edition, Course Technology Inc., ISBN-10: 0534390803, ISBN 13: 978-0534390808.

Reference Books:

1. "*An Introduction to Data Structures with Applications*", J.P.Trembly and P.G.Sorenson, Second Edition McGraw Hill Education , 1981, ISBN-10: 0074624717, ISBN-13: 978-0074624715.
2. "*Data Structures and Algorithms*", A. Aho, J. Hopcroft, J. Ulman, Pearson Education, 1998, First Edition, Addison Wesley, ISBN-10: 0201000237,ISBN-13: 978-020100023.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

DATA STRUCTURES LAB

Teaching Scheme	**P:4**		
Evaluation Scheme	**CE**	**ESE**	**Minimum Passing Marks**
	30 Marks	**70 Marks**	**40%**

Term Work:

1. Instructor will frame programming assignments based on the suggested list of assignments using C language preferably using gcc compiler.
2. Instructor is expected to incorporate variations in list.
3. Students will submit term work in the form of a journal that will include at least 16 to 20 practical assignments. Each programming assignment will consists of pseudo-algorithm, program listing with proper documentation and printout of the output.
4. 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. Implement stack data structure and its operations (PUSH and POP).
2. Write a program for conversion of infix expression into postfix using stack.
3. Write a program for evaluation of the postfix expression using stack.
4. Write a program to check the imbalance of parenthesis using stack.
5. Implement a simple queue and its operations.

6. Implement circular queue and its operations.
7. Implement a double ended queue and its operations.
8. Write a program for recursive solution of problem of Tower of Hanoi.
9. Write a program for the Fibonacci series using recursion.
10. Write a program to perform a quick sort on a set of entries from a file using recursion.
11. Write a menu driven program to perform following operations on singly linked list: create, insert, delete, display, reverse and search.
12. Write a menu driven program to perform following operations on circular linked list: create, insert, delete, display, reverse and search.
13. Represent polynomial as a circularly linked list and write a menu driven program to perform addition.
14. 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).
15. Implementation of binary tree data structure and perform pre-order, post-order and in-order traversals.
16. Write a program for implementation of graph data structure using adjacency list representation.
17. Write a Program for implementation of graph data structure using adjacency matrix representation.
18. Write a program to perform breadth first search on graph
19. Write a program to perform depth first search on graph.
20. Write a program for implementation of binary search tree.
21. Design a simple game (e.g. Tic-tac-toe) using game tree.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

CI204 DIGITAL SYSTEMS

Teaching Scheme	**L:4**		
Evaluation Scheme	**ME**	**ESE**	**Minimum Passing Marks**
	20 Marks	**80 Marks**	**40%**

Course Objectives:

1. To provide all basic concepts of the design of a digital system.
2. To give the students working knowledge of different methods for logic representation, manipulation and optimization, for both combinational and sequential logic.
3. To understandthe basics of Verilog hardware description language.

Course Contents:

Unit-I: Number systems & binary codes (7 Hrs)
Decimal, Binary, Octal & hexadecimal number systems and their conversions, Signed number representations: 1's & 2's complement, 9's & 10's complement, double precision numbers & floating point numbers, Classification of binary codes, BCD, XS-3, Gray, Error detecting & correcting codes, Alphanumeric codes.

Unit-II: Logic functions representation & minimization (6 Hrs)
Overview of logic gates, Boolean functions and their representation, Minimization techniques: K-maps (up to 5 variables), QuineMcClusky method.

Unit-III: Combinational logic designs (8 Hrs)
Adders, Subtractors, Binary parallel adders, Binary parallel subtractor, Binary adder-Subtractor, Look ahead carry adder, BCD adder, Code converters, Parity bit generator-checker, Comparators, Encoders-Decoders, Multiplexers, De-multiplexers.

Unit-IV: Latches and flip-flops (6 Hrs)
Latches & flip-flops, Flip-flop operating characteristics, Race around condition, Master-slave flip-flops, Excitation tables, Conversion of flip-flops.

Unit V: Introduction to Verilog HDL (6 Hrs)
Design methodologies, Modules, Instances, Components of simulation, Lexical conventions, Data types, Design examples using Verilog HDL.

Unit-VI: Sequential logic designs (7 Hrs)

Shift registers: Types of shift registers, Universal shift register, Counters: Asynchronous counters, Design of asynchronous counters, Effects of propagation delay, Synchronous counters, Shift register counters.

Outcomes: By the end of the course students will be able to

1. Understand several fundamental concepts that can be applied to a wide variety of digital design problems.
2. Apply knowledge of hardware description language in designing.

Text Books:

1. "*Fundamentals of Digital Circuits*", by A. Anand Kumar, Prentice Hall of India, 2/e, 2010, ISBN: 978-81-203-3679-7
2. *"A Verilog HDL Primer",* by J.Bhasker, ISBN:0965627748

Reference Books:

1. "*Modern Digital Electronics*", by R. P. Jain, Third Edition, Tata McGraw-Hill, ISBN: 0070494924, 978-00-704-9492-3.
2. "*Digital circuits and Logic design*", by Samuel C. Lee, Prentice Hall, ISBN:978-81-203-0149-8.
3. "*Engineering Approach to Digital Design*", by William I. Fletcher, PHI, ISBN: 978-81-203-0651-6.
4. "*Digital Design: Principles and Practices*", by John F. Wakerly,Third Edition, Pearson Education, ISBN:978-81-317-1366-2.
5. "*Verilog HDL : A Guide to Digital Design and Synthesis*", by Samir Palnitkar, Second Edition, Pearson Education. ISBN : 978-81-7758-918-4

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

DIGITAL SYSTEMS LAB

Teaching Scheme	**P:2**		
Evaluation Scheme	**CE**	**ESE**	**Minimum Passing Marks**
	30 Marks	**70 Marks**	**40%**

Term Work:

1. Instructor will frame programming assignments based on the suggested list of assignments.
2. Instructor is expected to incorporate variations in list.
3. Students will submit term work in the form of a journal that will include at least 10 to 12 practical assignments from A B and C sections and at least 3 to 4 experiments should be performed using Verilog HDL.
4. Each programming assignment will consists of program listing with proper documentation and printout of the output.
5. 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 and creativity.
5. Team building skills.
6. Technical writing skills.

Suggested List of Experiments:

A. Study of CMOS family.

B. Combinational logic design implementations-

1. Designing basic gates using universal gates.
2. Designing half/full adder/subtractor using logic gates.
3. Designing 4 bit adder-subtractor using IC 7483.
4. Designing 4 bit BCD adder using IC 7483.

5. Designing code converters (BCD to excess-3, Gray to binary and binary to gray).
6. Designing 2-bit comparator.
7. Realization of full adder using multiplexers & demultiplexer

C. Sequential circuit design implementations-

8. Conversions of flip-flops.
9. Asynchronous counters design (3 bit/4 bit/up-down/modulo m counters).
10. Synchronous counters design (3 bit/4 bit/up-down/modulo m counters).
11. Universal shift register.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

CI205   ENGINEERING ECONOMICS

Teaching Scheme	**L:3**		
Evaluation Scheme	**ME**	**ESE**	**Minimum Passing Marks**
	20 Marks	**80 Marks**	**40%**

Course Objectives:

1. To understand various aspects of engineering economics.
2. To evaluate systematically the cost and benefit associated with different projects.
3. To study different methods of depreciation, taxes and cost analysis.

Course Contents:

Unit-I : **(7 Hrs)**

Foundation of engineering economy: Description and role in decision making, Performing an engineering economy study, Professional ethics and economic decisions, Interest rate and rate of return, Terminology and symbols, Cashflows: Estimation and diagramming, Economic equivalence, Simple and compound interest, Minimum attractive rate of return, Introduction to spreadsheet use.

Unit-II : **(7 Hrs)**

Time value for money: Single-amount factors (F_P and P_F), Uniform series present worth factor and capital recovery factor (P_A and A_P), Sinking fund factor and uniform series compound amount factor (A_F and F_A), Factor values for untabulated i or n values, Arithmetic gradient factors (P_G and A_G), Geometric gradient series factors, Determining i or n for known cash flow values. Combining factors and spreadsheet functions calculations for uniform series that are shifted, Calculations involving uniform series and randomly placed single amounts calculations for shifted gradients.

Unit-III : **(6 Hrs)**

Nominal and effective interest rates: Nominal and effective interest rate statements, Effective annual interest rates, Effective interest rates for any time period, Equivalence relations: Payment period and compounding period, Equivalence relations: Single amounts with PP_CP, Equivalence relations: Series with PP _ CP, Equivalence relations: Single amounts and series with PP _ CP, Effective interest rate for continuous compounding, Interest rates that vary over time.

Unit-IV : (6 Hrs)

Present worth analysis: Formulating alternatives, Presentworth analysis of equal-life alternatives, Present worth analysis of different-life alternatives, Future worth analysis, Capitalized cost analysis. Annual worth analysis : Advantages and uses of annual worth analysis , Life-cycle cost analysis.

Unit-V : (7 Hrs)

Effects of inflation: Understanding the impact of inflation, Present worth calculations adjusted for inflation, Future worth calculations adjusted for inflation, Capital recovery calculations adjusted for inflation. Cost estimation and indirect cost allocation: Understanding how cost estimation is accomplished, Unit method, Cost indexes, Cost-estimating relationships: Cost-capacity equations, Cost-estimating relationships: Factor method, Traditional indirect cost rates and allocation, Activity-based costing (ABC) for indirect costs, Making estimates and maintaining ethical practices.

Unit-VI : (7 Hrs)

Depreciation methods: Depreciation terminology, Straight line (SL) depreciation, Declining balance (DB) and double declining balance (DDB) depreciation, Modified accelerated cost recovery system (MACRS) , Determining the MACRS recovery period, Depletion methods. After-tax economic analysis: Income tax terminology and basic relations, Calculation of cash flow after taxes, Effect on taxes of different depreciation methods and recovery periods, Depreciation recapture and capital gains (losses) , After-tax evaluation, After-tax replacement study, After-tax value-added analysis, After-tax analysis for international projects, Value-added tax.

Outcomes: By the end of the course students will be able to

7. Understand various concepts of economics.
8. Economically plan for their own project.
9. Get accustomed to the tax structure prevalent in the Indian economy.

Text Books:

1. "*Engineering Economy*", by Blank L. T., Tarquin A. J., 7th Edition, WCB/McGrawHill,- 2012, ISBN-13: 978-0073376301 ISBN-10: 0073376302.

Reference Books:

1. "*Financial Management Theory and Practice*", P. Chandra, 6'th edition, Tata McGraw Hill, 2007, ISBN:007462086X
2. "*Fundamentals of Financial Management*", V. Sharan, Pearson, 2'nd edition, ISBN 8131723976, 9788131723975

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

CI206 PROGRAMMING LAB - I

Teaching Scheme	**L:2 P:2**		
Evaluation Scheme	**CE**	**ESE**	**Minimum Passing Marks**
	30 Marks	**70 Marks**	**40%**

Course Objectives:

1. To provide a comprehensive study of the C programming language.
2. To identify problems that requires programmed solution.
3. To study, analyze and implement pointers, memory allocation, data handling through files and graphics in 'C'.

Course Contents:

Unit I: Pointers and structures (8 Hrs)
Pointers and addresses, Pointer and functions, Array of pointers, Pointer and strings, Advanced concepts- pointer to pointer, typecast and sizeof() operators, Dynamic memory allocation, DMA functions- malloc(), calloc(), realloc(), free(), Complex declarations involving pointers, structure, passing structures to functions - passing structure members to functions and passing entire structures to functions, Structures containing arrays, Self-referential structures, Nested structures, Advanced concept – memory organization of structures, Structure pointers - declaring a structure pointer and using structure pointers, Union.

Unit II: File management and preprocessor directives (5 Hrs)
File basics, File access functions, Character I/O, Formatted I/O, Advanced concepts – FILE structure, Paging control in display of text files, Direct input/output, File positioning, Preprocessor directives – File inclusion, Macro expansion, Conditional compilation.

Unit III: Graphics in C (7 Hrs)
Preliminaries, Invoking the graphics systems, Setting colors and current position, Drawing graphics entities – drawing points and lines, Drawing rectangles and polygons, Drawing circular and elliptical shapes, Setting line styles and fill styles, Drawing filled rectangles and polygons, Drawing filled circular and elliptical objects, Displaying texts, Interaction with mouse, Animations.

Outcomes: By the end of the course students will be able to

1. Write programs using advance concepts of C language.
2. Understand and apply the pointers, memory allocation techniques and use of files for dealing with variety of problems.
3. Design graphics programs using C.

Text Books:

1. "*Programming with C",* by R S Bichkar, Universities Press India Pvt. Ltd., ISBN-978-81-7371-771-0.
2. "*Programming in ANSI C*", by E.Balagurusamy, Tata McGraw Hill, ISBN-0070534772, 9780070534773.
3. *"Graphics under C",* by Yeshwant Kanetkar, BPB Publications, ISBN-81-7029-993-4.

Reference Books:

1. Herbert Schildt, "*C- The Complete Reference*", Tata McGraw-Hill, ISBN-0070411832, 9780070411838.
2. Kernighan Brian W. & Ritchie Dennis M., "*The C Programming Language*", Prentice Hall, ISBN-10: 0131103628, ISBN-13:9780131103627.
3. Bayron S Gottfried, "*Programming with C*", Tata McGraw-Hill, ISBN – 0070145903, 9780070145900

Term Work

1. Instructor will frame programming assignments based on the suggested list of assignments using C language preferably using gcc compiler.
2. Instructor is expected to incorporate variations in list.
3. Students will submit term work in the form of a journal that will include at least 08 to 10 practical assignments. Each programming assignment will consists of pseudo-algorithm, program listing with proper documentation and printout of the output.
4. 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. Write a program to implement-
   a) Function with pointer.
   b) Array of pointers.
   c) Pointer with string.
   d) Structure with pointer.
   e) Pointer to pointer.
2. Write a program to implement different dynamic memory allocation functions - malloc(), calloc(), realloc(), free(),
3. Write a program to implement passing structures to functions.
4. Write a program to read data from a file using character I/O function-fgetc(), fgets().
5. Write a program to write data into a file using character I/O function-fputc(),fputs().
6. Write a program to read data from a file using formatted I/O function-fscanf().
7. Write a program to write data into a file using formatted I/O function-fprintf().
8. Write a program to read data from a file using directed I/O function-fread().
9. Write a program to write data into a file using directed I/O function.-fwrite().
10. Write a program to implement macros.
11. Write a program to draw different graphics entities-
    a) Points &lines.
    b) Rectangles &polygons.
    c) Circular &elliptical shapes.
    d) Setting line styles &fill styles.
    e) Filled rectangles &polygons.
    f) Filled circular &elliptical objects.
12. Write a program for animation using graphics-
    a) Analog clock.
    b) Moving car.
    c) Bar graph.
    d) Kite flying etc.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

CI207 PROFESSIONAL COMMUNICATION SKILLS

Teaching Scheme	**L: 2 (AUDIT)**		
Evaluation Scheme	**ME**	**ESE**	**Minimum Passing Marks**
	10 Marks	**40 Marks**	**40 %**

Course Objectives:

1. To understand the concept, process and importance of professional communication.
2. To enable students to acquire English speaking and writing skills.
3. To hone presentation skills.

Course Contents:

Unit-I : Fundamentals of professional communication & speaking skills (9 Hrs)

- Definition of communication elements of communication (sender, receiver &media)
- Communication process/ cycle
- Types of communication (verbal- oral &written, Non-verbal- body language, Sign language &paralanguage)
- Patterns of communication in organization (internal, external, upward, downward, horizontal, diagonal, grapevine)
- Barriers of communication (physical, mechanical, language, psychological, linguistic, cultural)
- 7 C's of effective communication
- Public speaking
- Group discussion
- Interview skills

Unit-II : Writing Skills&Phonetics (10 Hrs)

- Elements/ parts of business letters
- Formats: Full block, Semi block
- Job application, Demand letter, Letter of complaint, & Letter of claim
- Resumepreparation
- Comprehension

- E-mail: Nature, Purpose, Advantages, Characteristics of successful E-mail messages & E-mail format
- Reports: Meaning, Significance, Essential features of a good report & types of report
- Study of speechorgans
- List of phonetic alphabets
- Manner of articulation of 44 sounds
- Word transcription

Unit-III : Introduction to behavioural skills & vocabulary (6Hrs)

- Developing positive attitude
- Time management
- Stress management
- Synonyms
- Antonyms
- One word substitution

Note:

- **Use of language laboratory is mandatory**
- **Conduct any eight(08) practical out of twelve (12)**
- **Practical examination is obligatory**

Outcomes: By the end of the course students will be able to

1. Understand the concept, process and importance of professional communication
2. Acquire English speaking and writing skills
3. Develop presentation skills

Text Books:

1. "*Business Communication*", by Sangeeta Magan, Biztantra, New Delhi, ISBN:8177228285.
2. *"Soft Skills for Managers",* by Dr. T. Kalyana Chakravarthi & Dr. T. Latha Chakravarthi, Biztantra, New Delhi, ISBN 10: 8177225685.
3. *"English Grammar and Composition"*, by Rajendra Pal and Prem Lata Suri, Sultan Chanda and Sons Publisher, ISBN: 978-81-8054-868-0.

Reference Books:

1. *"Ace of Soft Skills",* by Gopalswami Ramesh, Mahadevan Ramesh, Pearson Publication, Delhi, ISBN:9788131732854.
2. *"Soft Skills for Everyone"*,by Jeff Butterfield, Cengage, ISBN: 9788131514672.

3. "*Essentials of Business Communication"*, by Rajendra Pal and J. S. Korlhalli, Sultan Chand & Sons, New Delhi, ISBN: 8180547299.
4. *"Essential Communication Skills"*,by Shalini Aggarwal, Ane Books Pvt. Ltd, New Delhi, ISBN: 978-8180522802.
5. *"Spoken English: A Manual of Speech and Phonetics",* by R.K. Bansal & J.B. Harrison, Orient Blackswan Pvt. Ltd, Hyderabad, ISBN: 978-8125050858.
6. *" Effective Technical Communication",* by M. Ashraf Rizvi, McGraw-Hill Education Pvt. Ltd.,Delhi, ISBN:978-00-7059-952-9.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

PROFESSIONAL COMMUNICATION SKILLS LAB

Teaching Scheme	**P:2 (AUDIT)**		
Evaluation Scheme	**CE**	**ESE**	**Minimum Passing**
	15 Marks	**35 Marks**	**40%**

Course Objectives

To understand communication process and hone Phonetics, Speaking skills and writing skills.

Term Work:

1. Instructor will frame programming assignments based on the suggested list of assignments.
2. Instructor is expected to incorporate variations in list.
3. Students will submit term work in the form of a journal that will include at least 08 to 10 practical assignments. Each programming assignment will consists of pseudo-algorithm, program listing with proper documentation and printout of the output.
4. Practical examination will consist of performance and viva-voice examination based on the term work.

List of Practical/Assignments

1. Communication cycle/process
2. Self introduction
3. Extempore
4. Role play
5. Listening phonetic sounds' manner of articulation in language laboratory
6. Group discussion
7. Mock interview
8. Application writing
9. Email writing
10. Resume writing
11. Vocabulary based activity
12. PPT presentation on non-technical issue

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

SEMESTER - IV

CI208	**MICROPROCESSORS AND MICROCONTROLLERS**

Teaching Scheme	**L:4**		
Evaluation Scheme	**ME**	**ESE**	**Minimum Passing Marks**
	20 Marks	**80 Marks**	**40%**

Course Objectives:

1. To develop an in-depth understanding of the operation of microprocessors and microcontrollers.
2. To study the architecture of microprocessors and microcontrollers and To acquire the assembly language programming skills of 8086 and 8051.
3. To learn peripherals and their interfacing with microprocessor and microcontrollers.

Course Contents:

Unit-I : Introduction to microprocessors (8 Hrs)
Evolution of microprocessors, De-multiplexing of address-data bus, Pin diagram of 8085, Architecture of 8085, Pin diagram of 8086, Architecture of 8086, Bus cycles and timing diagram, Addressing modes of 8086, Classification of 8086 instructions, Instruction format of 8086, Instruction set of 8086, Interrupts in 8086.

Unit-II : Intel 80x86 family of processors and assembly language programming of 8086 (6 Hrs)
Introduction to 80386, 80486 & Pentium processor, Levels of programming, Assembly language program development tools, Assembler directives, Interrupt of personal computers, Programming in 8086.

Unit-III : Memory, peripheral devices and interfacing (8 Hrs)
Interfacing static RAM and EPROM, Even and odd banks in 8086, Memory organization of 8086, Memory interface examples peripheral devices and interfaces as 8255, 8237, Serial data communication interface i.e. 8251, Keyboard and display interfacing.

Unit-IV : Introduction to microcontroller 8051 (6 Hrs)
RISC and CISC Architecture, Harvard and Van-Neumann architecture, Microcontrollers and embedded processors, Overview of 8051 family, Inside the 8051, Assembling and running

8051 program, Program counter and ROM space, PSW, 8051 data types, Register bank and stack, Jump, Loop, Call instructions and programming, I/O port programming, Arithmetic and logical instructions programming, Bitwise manipulations and programming.

Unit-V : Programming and peripheral interfacing with 8051 (8 Hrs)
8051 addressing modes, 8051 programming in C, Timer programming, Serial port programming, Interrupt programming, LCD & keyboard interface, ADC, DAC and sensor interface.

Unit-VI : Fundamentals of ARM processor (4 Hrs)
Registers, CPSR, Pipeline, Exceptions, Interrupts, and Vector table, Core extensions, Architecture revisions, ARM processor families.

Outcomes : By the end of the course students will be able to

1. Learn the internal organization of some popular microprocessors/microcontrollers and design of microprocessors/microcontrollers-based systems.
2. Write assembly language and C programs for microprocessors and microcontrollers.
3. Perform hands-on with various interfaces: LCD, Keyboard, ADC, DAC and other peripherals using 8051.

Text Books:

1. *"Microprocessors and Microcontroller",* by A. Nagoor Kani, Second Edition, Tata McGraw Hill Education Private Limited, ISBN-13: 978-0-07-132974-3.
2. *"The 8051 Microcontroller and Embedded Systems-using assembly and C",* by Muhammad Ali Mazidi and Janice Gillispie and Rollin D. McKinlay, Second Edition, Pearson Education, ISBN: 978-81-3171-026-5.

Reference Books:

1. *"Advanced Microprocessors and Peripherals",* by A. K. Ray and K. M. Bhurchandi, Tata McGraw-Hill Education Private Limited, ISBN: 1259051269, 9781259051265.
2. *"Microcomputer Systems: The 8086/8088 Family Architecture, Programming and Design",* by Yu-chengLiu ,Glenn A. Gibson, Second Edition, PHI, ISBN: 978-81-203-0409-3.
3. *"The 8088 and 8086 Microprocessors",* by Walter A. Triebel, Avtar Singh, Fourth Edition, Pearson Education, ISBN: 978-81-7758-481-3.
4. *"Programming and Customizing the 8051 Microcontroller",* by Myke Predko ,TMH , ISBN: 978-00-704-2140-0
5. *"C and 8051",* by Thomas Shultz, Second Edition, PHI, ISBN : 9780137548392

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

MICROPROCESSORS AND MICROCOMTROLLERS LAB

Teaching Scheme **P:2**

Evaluation Scheme	**CE**	**ESE**	**Minimum Passing**
	30 Marks	**70 Marks**	**40%**

Term Work:

1. Instructor will frame programming assignments based on the suggested list of assignments.
2. Instructor is expected to incorporate variations in list.
3. Students will submit term work in the form of a journal that will include at least 08 to 10 practical assignments. Each programming assignment will consists of pseudo-algorithm, program listing with proper documentation and printout of the output.
4. 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. Assembly language programming in 8086 using MASM (Any 4)

1. Write and execute program for addition of two 16 bit numbers.
2. Write and execute program for block transfer of data.
3. Write and execute program for addition of series of N 8/16 bit numbers.
4. Write and execute program for sorting an array.
5. Write and execute program for matrix addition / Multi-byte addition.
6. Write and execute program for multiplication of two 3x3 matrices.
7. Write and execute program to display string on DOS shell as “HELLO WORLD”.

B. Assembly language programming in 8051 using cross compiler and interface boards for peripherals (Any 4).

1. Write and execute program for addition of two 8 bit numbers.
2. Write and execute program for block transfer of data.
3. Write and execute program for conversion of ASCII to BCD / BCD to ASCII.
4. Write and execute program to get data from port and display on other port.
5. Write and execute program to generate delay using timer.
6. Write and execute program to receive data serially, set baud rate, start bit and stop bit.
7. Write and execute program to generate square wave using interrupt.
8. Write and execute program to interface using LCD/ Keyboard/ Stepper motor/ Sensor (Any one).

D. 8051 programming in C using cross compiler and interface boards for peripherals (Any 4)

1. Write and execute program for toggle bits of ports (LED's).
2. Write and execute program to generate delay using timer.
3. Write and execute program to receive data serially, set baud rate, start bit and stop bit.
4. Write and execute program to generate square wave using interrupt.
5. Write and execute program to interface LCD.
6. Write and execute program to interface keyboard.
7. Write and execute program to interface stepper motor.
8. Write and execute program to interface sensor using ADC / DAC.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

CI209 COMPUTER ALGORITHMS

Teaching Scheme	**L:4**		
Evaluation Scheme	**ME**	**ESE**	**Minimum Passing Marks**
	20 Marks	**80 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.

Course Contents:

Unit-I : (8 Hrs)

Algorithms, Algorithms as a technology, Analyzing algorithms, Designing algorithms, Growth of Functions: Asymptotic notations, Time complexity and space complexity.

Unit-II : (9 Hrs)

Divide and conquer (recurrences), Substitution method, Recurrence tree method and master's method.

Sorting algorithms: Merge sort, Quick sort, Heaps: Maintaining a heap property, Building a heap, Heapsort, Priority queues.

Unit-III : (6 Hrs)

Dynamic programming : Elements of dynamic programming, Matrix chain multiplication, Longest common subsequences and optimal binary search trees, 0/1 Knapsak problem.

Unit-IV : (6Hrs)

Greedy method: An activity selection problem, Elements of greedy strategy, Huffman codes, Task scheduling problem.

Unit-V : (7Hrs)

Graph algorithms: Minimum cost spanning trees, Topologocal sort, Single source shortest paths, All pair shortest paths.

Unit-VI : (4Hrs)

NP complete theory : Tractable and intractable problems, Computability, The halting problem, Computability classes: P, NP, NP-complete and NP-hard, Cook's theorem, Standard NPcomplete problems, Reduction techniques.

Outcomes: By the end of the course students will be able to

1. Analyze any algorithms and able to calculate their theoretical complexity.
2. Understand the problem solving methods such as recurrences, dynamic programming and greedy method.
3. Understand 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, Third Edition 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, ISBN-13: 978-0201000290,ISBN-10: 0201000296.
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. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

CI210 SYSTEM PROGRAMMING

Teaching Scheme	**L:4**		
Evaluation Scheme	**ME**	**ESE**	**Minimum Passing Marks**
	20 Marks	**80 Marks**	**40%**

Course Objectives:

1. To introduce student to the fundamental model of the processing of high level language programs for execution on computer system.
2. To understand and implement functioning of assembler, loader, linkers, macros & compilers.
3. To learn the process management and information management.

Course Contents:

Unit-I : Language Processors (05 Hrs)

System software's, Language processing activities, Fundamentals of language processing (phases and passes of compiler and role of each analyzer), Fundamental of language specification, The simplified instructional computers: - SIC machine architecture, SIC / XE machine architecture.

Unit-II : Compilers (07 Hrs)

Basic compiler functions, Different phases of a compiler, Machine dependent compiler features, Machine independent complier features, Simple one pass compiler, Implementation examples - Java compiler and environment.

Unit-III: Assemblers (08 Hrs)

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 example - MASM assembler.

Unit-IV : Loaders and Linkers (08 Hrs)

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.

Unit-V : Macro Processors **(07 Hrs)**

Basic macro processor functions – Macro definition and expansion, Macro processor algorithm and data structures, Macro processor design options- Recursive macro expansion, General purpose macro processor, Macro processing within language translators, Implementation examples - MASM macro processor.

Unit-VI : Software Tools **(05 Hrs)**

Software tools for program development, Editors, Debug monitors, Programming environment, User interface.

Outcomes: By the end of the course students will be able to

1. Understand different components of system software.
2. Learn intermediate code generation in context of language designing.
3. Know operating system functions such as memory management as pertaining to run time storage management.

Text Books:

1. "*System Software -An Introduction to System Programming",* by Leland L. Beck and D. Manjula (3rd Edition, Pearson Education) ISBN 978-81-7758-555-1.
2. "*System Programming and Operating System*", by D. M. Dhamdhere (2nd Revised Edition, TMH) ISBN 0-07-463579-4

Reference Books:

1. "*System Programming*", by J. J. Donovan (TMH) ISBN 0-07-460482-1
2. "*Compilers Principles, Techniques and Tools*", by A. V. Aho, Ravi Sethi and J. D. Ullman (2nd Edition, Pearson Education) ISBN 978-81-317-2101-8.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

CI211 OBJECT ORIENTED PROGRAMMING WITH C++

Teaching Scheme	**L: 3**	**T:1**	
Evaluation Scheme	**ME**	**ESE**	**Minimum Passing Marks**
	20 Marks	**80 Marks**	**40%**

Course Objectives:

1. To know different programming paradigms.
2. To study and understand the object oriented programming concepts and methodology.
3. To implement object oriented programming concepts in C++.
4. Demonstrate adeptness of object oriented programming in developing solutions to problems demonstrating usage of data abstraction, encapsulation, and inheritance.

Course Contents:

Unit-I : (6 Hrs)

Principles of object-oriented programming:

Software crisis, Software evolution, Different programming paradigms, Procedure-oriented programming, Basic concepts of object-oriented programming, Benefits of OOP, Object-oriented languages, Applications of OOP.

Beginning with C++:

Applications of C++, A simple C++ program, C++ statements, and Structure of C++ program, Creating the source file, Compiling and linking.

Tokens, expressions and control structures:

Introduction, Tokens, Keywords, Identifiers and constants, Basic data types, User-defined data types, Derived data types, Declaration of variables, Dynamic initialization of variables, Reference variables, Operators in C++, Scope resolution operator, Member dereferencing operators, Memory management operators, Manipulators, Type cast operators, Expressions and their types, Special assignment operator. Implicit conversions, Operator precedence, Control structures.

Unit-II : (6 Hrs)

Functions in C++:

Introduction, The main function, Function prototyping, Call by value and call by reference, Return by reference, Inline functions, Default arguments, Function overloading.

Classes and objects:

Introduction, C structures revisited, Specifying a class, Defining member functions, A C++ program with class, Making an outside function inline, Nesting of member functions, Private member functions, Arrays within a class, Memory allocation for objects, Static data members, Static member functions, Arrays of objects, Objects as function arguments, Friendly functions, Returning objects.

Unit-III : (6 Hrs)

Constructors and destructors:

Introduction, Constructors, Parameterized constructors, Multiple constructors in a class, Constructors with default arguments, Dynamic initialization of objects, Copy constructor, Dynamic constructors, and destructors.

Operator overloading and type conversions:

Introduction, Defining operator overloading, Overloading unary operators, Overloading binary operators, Overloading binary operators using friends, Manipulation of strings using operators, Rules for overloading operators, Type conversions.

Unit-IV : (6 Hrs)

Inheritance:

Introduction, Types of inheritance, Virtual base classes, Abstract classes, Constructors in derived classes, Nesting of classes (Container class).

Pointer, Virtual functions and polymorphism:

Introduction to pointers, Pointer to objects, Pointer to derived classes, Virtual functions, Pure virtual functions,Virtual destructors.

Unit-V : (6 Hrs)

Managing console I/O operations:

Introduction, C++ streams, C++ stream classes, Unformatted I/O operations, Formatted console I/O operations, Managing output with manipulators.

Working with files:

Introduction, Classes for file stream operations, Opening and closing of a file, File modes, File pointers and their manipulations, Sequential input and output operations, Error handling during file operations.

Unit-VI : (6 Hrs)

Templates:

Introduction, Class templates, Function templates, Overloading of template functions, Member function templates.

Exception handling:

Introduction, Basics of exception handling, Exception handling mechanism, Throwing mechanism, Catching mechanism, Rethrowing an exception, Specifying exceptions, Exception in constructors and destructors, Exception in operator overloading functions.

Outcomes: By the end of the course students will be able to

1. Understand key features of the object-oriented programming language such as encapsulation (abstraction), inheritance and polymorphism.
2. Apply an object oriented approach to programming and identify potential benefits of object-oriented programming over other approaches.
3. Apply object-oriented concepts in real world applications.

Text Books:

1. *"C++ Programming Language",*by Bjarne Stroustrup (4th Edition, Addison Wesley) ISBN-13: 978-0321563842.
2. *"Object Oriented Programming with C++",* by E. Balagurusamy (6th Edition, McGraw Hill Pvt. Ltd.) ISBN-13:978-1-25-902993-6.

Reference Books:

1. *"Mastering C++",* by K.R.Venugopal,Rajkumar, T.Ravishankar (2nd Edition, McGraw Hill Pvt. Ltd.) ISBN-13:978-93-83286-77-5.
2. *"The Complete Reference C++",* by Herbert Schildt (5th Edition, McGraw Hill Pvt. Ltd.) ISBN-13: 978-0071634809.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

OBJECT ORIENTED PROGRAMMING WITH C++ LAB

Teaching Scheme	**P: 2**		
Evaluation Scheme	**CE**	**ESE**	**Minimum Passing Marks**
	30 Marks	**70 Marks**	**40%**

Term Work:

1. Instructor will frame programming assignments based on the suggested list of assignments using C ++ language.
2. Instructor is expected to incorporate variations in the list.
3. Students will submit term work in the form of a journal that will include at least 12 to 15 programming assignments. Each programming assignment will consists of pseudo algorithm, program listing with proper documentation and printout of the output.
4. 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 programming assignments:

1. Write a program to implement:
   a. Reference variable.
   b. Scope Resolution Operator.
2. Write a program to implement Manipulators.
3. Write a program to implement:
   a. Function prototyping.
   b. Call by Value.

c. Call by Reference.
d. Return by Reference

4. Write a program to implement Function Overloading.
5. Write a program to implement Inline function.
6. Write a program to implement Defaults Arguments.
7. Write a program to implement the concepts of classes.
8. Write a program to implement nesting of member functions.
9. Write a program to implement Static Data Members and Static Member Functions.
10. Write a program to implement Objects as Function Arguments for the addition of two time in the hours and minutes formats.
11. Write a program to implement Friend Function to swap private data members between two classes.
12. Write a program to implement Friend Function to find the sum of two complex numbers.
13. Write a program to demonstrate operator overloading for unary as well as binary operation.
14. Write a program to demonstrate Single, multiple, multilevel, hybrid, hierarchical inheritance.
15. Write a program to demonstrate Virtual base classes.
16. Write a program to implement Array of pointers, pointer to functions, pointer to objects
17. Write a program to implement Array of pointers to objects, this pointer, pointers to derived classes, Virtual functions.
18. Write a program to demonstrate Opening and Closing of file using constructors and open () function.
19. Write a program to implement class templates and function templates.
20. Write a program to demonstrate the concepts of catching and throwing of an exception.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

CI212 NUMERICAL METHODS & SCIENTIFIC COMPUTING

Teaching Scheme	**L:3**	**T:1**	
Evaluation Scheme	**ME**	**ESE**	**Minimum Passing Marks**
	20 Marks	**80 Marks**	**40%**

Course Objectives:

1. To learn the techniques for finding solutions numerically, with reliable and specified accuracy.
2. To introduce students to the mostly used numerical methods in the different engineering fields.
3. To apply numerical methods and probability distribution to obtain approximate solutions to mathematical problems.

Course Contents:

Unit-I : (6 Hrs)

Introduction: Approximations and errors, Error definitions, Significant figures, Accuracy & precision; Roots of equation: Bracketing methods: Bisection method, False position method; Open method: Newton- Raphson method, Secant method.

Unit-II : (8 Hrs)

System of linear algebraic equations: Gauss elimination method, Gauss-Jordan method, LU decomposition method; Matrix inverse using the LU decomposition method.

Interpolation: Newton's forward, backward & central interpolation, Newton's divided difference interpolating polynomial, Lagrange's interpolation and spline interpolation.

Unit-III : (8 Hrs)

Numerical differentiation: Derivatives using Newton's forward & backward difference formula.

Numerical integration: Trapezoidal rule, Simpson's rule and integration with unequal segments.

Differential equations: Euler's method, Modified Euler's method, Runge- Kutta method of 4th order.

Unit-IV : (8 Hrs)

Curve fitting by method of least squares: Fitting a straight line, Fitting a parabola, Exponential curves. Regression Analysis: Linear regression, Lines of regression, Coefficient of correlation, Multiple regression. Testing of hypothesis: Null and alternative hypothesis, Level of significance, One-tailed and two-tailed tests, Tests for large samples (Z-test), Tests for small samples (T- test), Chi-square test.

Unit-V : (6 Hrs)

Probability distribution: Random variables, discrete probability distribution, continuous probability distribution, Uniform distribution, Poisson's distribution, Normal distribution, Exponential distribution and its applications.

Unit-VI : (6 Hrs)

Queuing theory: Queuing systems, The input or arrival pattern, The service pattern and service discipline, Notation, Performance measures, Little's formula, Relation between the probabilities of states.

Outcomes: At the end of this course, the students will be able to

1. Understand the mathematical background of the different numerical methods and probability distributions introduced in the course.
2. Learn the different numerical methods to solve the algebraic equations and a system of linear and nonlinear equations.
3. Understand the different numerical methods for interpolation, differentiation, integration and solving a set of ordinary differential equations.

Text Books:

1. "*Numerical Methods for Engineers",* by Steven C. Chapra (6th Edition, McGraw Hill Book Company.) ISBN-978-1-25-902744-4.
2. "*Higher Engineering Mathematics",* by B. S. Grewal (43rdEdition, Khanna Publishers.) ISBN-978-81-7409-165-5.
3. "*Probability, Statistics with Reliability, Queuing and Computer Science Applications",* by Kishor S. Trivedi (2ndEdition, Wiley India Pvt. Ltd.) ISBN 13-978-81-265-1853-1.

Reference Books:

1. "*Introductory Methods for Numerical Analysis",* by S. S. Sastry (4thEdition, PHI Publication.) ISBN-978-81-203-2761-0.
2. "*Fundamentals of Statistics",* by S.C. Gupta (46thEdition, Himalaya Publishing House.) ISBN-81-8318-417-0.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

NUMERICAL METHODS & SCIENTIFIC COMPUTING LAB

Teaching Scheme	**P: 2**		
Evaluation Scheme	**CE**	**ESE**	**Minimum Passing Marks**
	30 Marks	**70 Marks**	**40%**

Term Work:

1. Instructor will frame programming assignments based on the suggested list of assignments using C language.
2. Instructor is expected to incorporate variations in the list.
3. Students will submit term work in the form of a journal that will include at least 8 to 10 programming assignments. Each programming assignment will consists of pseudo-algorithm, program listing with proper documentation and print out of the output.
4. 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 the problem statement.
4. Innovation & creativity.
5. Team building skills.
6. Technical writing skills.

The suggested list of programming assignments:

1. Write a program to find the root of an equation using following bracketing methods
   a. Bisection method
   b. False position method
2. Write a program to find the root of an equation using following open methods
   a. Newton Raphson method
   b. Secant method

3. Write a program in to implement the following interpolation methods
    a. Newton’s forward interpolation
    b. Newton’s backward interpolation
4. Write a program in to implement Lagrange’s interpolation method.
5. Write a program to implement numerical differentiation using Newton’s forward & backward difference formula.
6. Write a program to implement the following numerical integration methods
    a. Trapezoidal rule
    b. Simpson’s rule
7. Write a program in to implement following methods of solving differential equations
    a. Euler’s method
    b. Modified Euler’s method
8. Write a program to implement Runge-Kutta method of 4th order.
9. Write a program to fit a straight line by the method of least squares.
10. Write a program to fit a parabola by the method of least squares.
11. Write a program to find
    a. Lines of regression
    b. Coefficient of correlation
12. Write a program to implement following probability distributions
    a. Normal distribution
    b. Poisson’s distribution

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

CI 213 PROGRAMMING LAB-II

Teaching Scheme	**P:2**		
Evaluation Scheme	**CE**	**ESE**	**Minimum Passing Marks**
	30 Marks	**70 Marks**	**40%**

Course Objectives

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

Term Work:

1. Instructor will frame programming assignments based on the suggested list of assignments using C or C ++ language.
2. Instructor is expected to incorporate variations in list.
3. Students will submit term work in the form of a journal that will include at least 4 to 5 practical assignments of System Programming and 4 to 5 practical assignments of Computer Algorithms covering complete syllabus. Each programming assignment will consists of pseudo-algorithm, program listing with proper documentation and printout of the output.
4. 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

System Programming Practicals

The following practical assignments should be completed using gcc compiler in Unix/Linux environment:

1. Execution of different commands
    a. File system navigation commands.
    b. File management commands.
    c. Communication commands.
    d. Filtering commands.
2. Learning how to compile a C/C++ program using gcc compiler and Unix/Linux platform.
3. Implementation of macros & nested macros.
4. Design and implementation of 1 pass assembler.
5. Design and implementation of 2 pass assembler.
6. Symbol table generation for input *.c file.
7. Implementation of Toy-code generator.

Computer Algorithm Practicals

1. Write a program using C for implementation of sorting algorithms: (At least 02)
    a. Merge sort.
    b. Quick sort.
    c. Heap sort.

2.Write a program using C for following dynamic programming problems: (At least 01)
    a. Matrix chain multiplication.
    b. Longest common subsequence.
    c. Optimal binary search tree.
    d. 0/1 knapsack problem.

3. Write a program using C for following problems using the greedy approach : (At least 02)
    a. Activity selection problem.
    b. Topological sort.
    c. All pair shortest path.
    d. Kruskals algorithm.
    e. Generation of Huffman code.
    f. Prim’s algorithm.

Swami Ramanand Teerth Marathwada University, Nanded

Second Year U.G. Program in Computer Science & Engineering and Information Technology

Effective from 2016-17

CI214 MINI PROJECT - I

Teaching Scheme	**P:2**		
Evaluation Scheme	**CE**	**ESE**	**Minimum Passing Marks**
	30 Marks	**70 Marks**	**40%**

Evaluation criteria

The continuous evaluation of term work shall be of 30 Marks. The 25 Marks shall be distributed over internal assessments / reviews (at least 02 reviews) during the semester by a review committee. The remaining 05 Marks shall be distributed for attendance. The Head of the department shall constitute the review committee. The student shall make a presentation on the progress made before the committee. The 70 Marks of the practical will be awarded based on the performance in the practical examination conducted by the University at the end of the semester.

General suggestions and expectations / guidelines

- The project shall be developed in C/C++.
- The students may choose the theory concepts they studied in different subjects as project topic.
- Interdisciplinary project proposals and innovative projects are encouraged and more appreciable.
- The project topic can be suggested by the staff member or it can be proposed by the students.
- The project topic shall be approved by the project in-charge.
- The Guides are advised to give projects and suggest project titles focusing more on the current field of research and ensure the level of innovation.

- A project team shall contain a maximum of 2 members.
- The project work should be properly distributed among the team members.
- Students should submit the project documentation at the beginning of the semester consisting of:
  - ✓ Title
  - ✓ Abstract
  - ✓ Modules split-up
  - ✓ Deliverables for each review
  - ✓ Data model (If any)
  - ✓ Details of teammembers
- Reviews for the project work will be conducted at regular intervals by the panel of examiners formed by the Head of the Department.
- The student failing to attend the project review will be subject to strict action as decided by the Head of the Department.
- Throughout the semester at any point of time if students are found to be involved in any of the following:

  - ➢ Using project codes available on the internet
  - ➢ Using project codes developed by someone else
  - ➢ Using project work which is already submitted in other institute or university

  Such students shall be declared failed or penalized as decided by the Examiners.
- The students must arrange regular meetings with the guide and present progress of project work.
- A spiral bound Project report to be prepared as per the guidelines and format given by the department
- The guides are advised to check for the formatting of the presentation and project report.
- Students must submit a report well before the end of the semester.