PANDIT DEENDAYAL PETROLEUM UNIVERSITY GANDHINAGAR
SCHOOL OF TECHNOLOGY

COURSE STRUCTURE FOR B TECH IN COMPUTER ENGINEERING

Semester IV			**B Tech in Computer Engineering**											
Sr. No.	**Course Code**	**Course Name**	**Teaching Scheme**					**Examination Scheme**						
								Theory			**Practical**		**Total**	
			L	**T**	**P**	**C**	**Hrs/Wk**	**CE**	**MS**	**ES**	**CE**	**ES**	**Marks**	
1	MA 202T	Numerical & Statistical Methods	3	1	0	4	4	25	25	50	-	-	100	
2	CP 211T	Design & Analysis of Algorithms	3	1	0	4	4	25	25	50	-	-	100	
3	18CP 211P	Design & Analysis of Algorithms LAB	0	0	2	1	2	-	-	-	25	25	50	
4	18CP218T	Object Oriented Concepts & Programming	3	0	0	3	3	25	25	50	-	-	100	
5	18CP218P	Object Oriented Concepts & Programming LAB	0	0	2	1	2	-	-	-	25	25	50	
6	CP 213T	Computer Networks	4	0	0	4	4	25	25	50	-	-	100	
7	CP 213P	Computer Networks Lab	0	0	2	1	2	-	-	-	25	25	50	
8	CP 214T	Computer Organization & Programming	3	1	0	4	4	25	25	50	-	-	100	
		TOTAL	16	3	6	22	25						650	

MS-Mid Semester; ES-End Semester, CE – Continuous Evaluation

Course Code: MA 202T					Course Name: Numerical & Statistical Methods			
Teaching Scheme					Examination Scheme			
L	T	P	C	Hrs/ Wk	Theory			Total
					Continuous Evaluation	Mid Semester	End Semester	Marks
3	1	0	4	4	25	25	50	100

Prerequisites: Maths III, Computer Programming

Learning objectives:

- Numerical methods provide the technique to solve ordinary differential equations, integrals, algebraic and transcendental equations. The course will also develop an understanding of the elements of error analysis for numerical methods. Ordinary differential equations occur in many scientific disciplines. Thus the course will further develop problem solving skills. This course provides an introduction to probability theory and random variables. In addition the course also covers various distributions – discrete as well as continuous. The students also get to know about the theory of least squares and statistical averages. They also learn about to collect and analyze the data that help in decision making.

Unit wise allocation of course content

UNIT I (10 L, 3 T)

Numerical Solution of System of linear equations & non-linear equations: Solution of transcendental and non-linear equations by Bisection, Regular Falsi, Newton's Raphson and Secant method. Solution of a system of linear simultaneous equations by LU Decomposition, Cholesky Decomposition, Jacobi and Gauss Seidel methods. Concept of Ill conditioned system.

UNIT II (14 L, 5 T)

Interpolation and Numerical Integration: Introduction of Finite differences, Operators, Newton Gregory Forward Interpolation Formula, Newton Gregory Backward Interpolation Formula, Gauss's Forward and Backward Interpolation Formula, Stirling's Central Difference Formula, Lagrange's Interpolation Formula for unevenly spaced data, Inverse Interpolation, Divided Differences, Properties of Divided Differences, Newton's Divided Difference Formula, Relation between Divided Differences and Ordinary Differences. Formulae for Derivatives, Newton-Cotes's Quadrature Formula, Trapezoidal rule, Simpson's one-third rule, Simpson's Three-Eighth rule, Weddle's rule, Romberg's method, Double Integration. Numerical solution of first order ordinary differential equation by Taylor series method, Picard's method, Euler's method, Modified Euler's method and Runge-Kutta (4th order only) method. Multi step methods: Adams-Moulton method and Milne's method.

UNIT III (6 L, 2 T)

Probability: Various approaches of probability-classical, frequency (statistical), subjective and axiomatic. Theorems on probability, conditional probability, Independence, Baye's Theorem. Random variable-discrete and continuous. Distribution function and their properties, probability mass and density functions.

UNIT IV (8 L, 3 T)

Statistics: Mathematical Expectation, Moment Generating Function and its properties. Probability distributions: Bernoulli, Binomial, Negative Binomial, Poisson and Normal Distributions. Theory of least squares and curve fitting. Correlation-Simple, Multiple and Partial, Regression lines and Regression coefficients.

Student centering learning: (The student centering learning contents should be declared at the commencement of semester. It should be maximum 10% ; however exact contents is left to faculty)

Lecture: 38 Hrs

Tutorial: 13 Hrs

Approximate Total: 51 Hrs

Texts and References

1. B.S. Grewal, Numerical Methods in Engineering and Science with Programs in C & C++, Khanna Publishers.
2. S.S. Sastry, Introductory Methods for Numerical Analysis, Prentice Hall of India.
3. M.K. Jain, S.R.K. Iyenger and R.K. Jain, Numerical Methods for Scientific and Engineering Computation, New Age International.
4. S.C. Gupta and V.K. Kapoor, Fundamentals of Mathematical Statistics, S. Chand Publisher .
5. R.K. Jain & S.R.K. Iyenger, Advanced Engineering Mathematics, Narosa.

Course outcome:

At the end of this course students will be able to

1. Understand common numerical methods and how they are used to obtain approximate solutions to otherwise intractable mathematical problems.
2. Derive numerical methods for various mathematical operations and tasks, such as interpolation, differentiation, integration, the solution of linear and nonlinear equations, and the solution of differential equations, and analyze accuracy of numerical methods.
3. Understand basic probability axioms and rules and the moments of discrete and continuous random variables as well as be familiar with common named discrete and continuous random variables.
4. Derive the probability density function of transformations of random variables and use these techniques to generate data from various distributions. Construct the probability distribution of a random variable, based on a real-world situation, and use it to compute various raw and central moments of higher order. Assess which distribution for summarizing a data set are most appropriate and highlight interesting features of the data.

Course Code: CP 211T					Course Name: Design & Analysis of Algorithms			
Teaching Scheme					Examination Scheme			
L	T	P	C	Hrs/ Wk	Theory			Total
					Continuous Evaluation	Mid Semester	End Semester	Marks
3	1	0	4	4	25	25	50	100

Prerequisites: Data and File Structures, Discrete Mathematics

Learning objectives:

- To learn theory of algorithm design and various techniques
- To selectively apply the suitable algorithmic technique to real problems and carry out complexity/performance analysis

Unit wise allocation of course content

UNIT 1 (10 L, 3T)

Elementary Algorithmic: Efficiency of Algorithms, Average & worst-case analysis, Elementary Operation

Analysis Techniques: Empirical, mathematical, Asymptotic analysis and related unconditional and conditional notations.

Analysis of Algorithms: Analyzing control structures: sequencing, "For" loops, Recursive calls, "While" and "repeat" loops, using a barometer, Amortized analysis

UNIT 2 (10 L, 4T)

Solving Recurrences: Intelligent guesswork, Homogeneous recurrences, Inhomogeneous Recurrences, Change of variable, Range transformations, Master Theorem, Recurrence Tree

Data Structures: Heaps, Binomial heaps, Disjoint set structures

Greedy Algorithms: Graphs: Minimum spanning trees-Kruskal's algorithm, Prim's algorithm, Graphs: Shortest paths

UNIT 3 (10 L, 3T)

Divide-and-Conquer: Multiplying large integers, Binary search, sorting: sorting by merging, quick sort, finding the median, Matrix multiplication, Exponentiation

Dynamic Programming: Making Change, The principle of optimality, The Knapsack Problem, Shortest path, Chained matrix multiplication, Approaches using recursion, Memory functions.

UNIT 4 (9 L, 3T)

Branch and Bound, Backtracking: Design of some classical problems using branch and bound and Backtracking approaches.

Brief Overview of NP theory, dealing with higher bounds of computing problems through approximation algorithms.

Student centering learning: (The student centering learning contents should be declared at the commencement of semester. It should be maximum 10% ; however exact contents is left to faculty)

Lecture: 39 Hrs
Tutorial: 13 Hrs
Approximate Total: 52 Hrs

Texts and References:

1. Charles E. Leiserson, Thomas H. Cormen, Ronald L. Rivest, Clifford Stein - Introduction to Algorithms, PHI
2. Gilles Brassard & Paul Bratley, Fundamentals of Algorithmic, PHI
3. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekharan, Fundamentals of Computer Algorithms, Galgotia.

Course Outcomes (COs):

At the end of this course students will be able to

1. Understand need of complexity analysis of the algorithm
2. Learn and apply various methods for solving recurrence relations
3. Design and implement greedy algorithms for solving various problems
4. Design and implement dynamic programming algorithms for solving various problems
5. Design and implement branch and bound techniques for solving various problems
6. Apply algorithmic techniques to design optimal solutions

Lab Code: 18CP 211P					Lab Name: Design & Analysis of Algorithms Lab		
Teaching Scheme					Examination Scheme		
L	T	P	C	Hrs/Wk	Practical		Total
					Continuous Evaluation	End Semester Exam	Marks
0	0	2	1	2	25	25	50

Prerequisites: Computer Programming

Course objectives:

- **Analyze the asymptotic performance of the algorithms**
- **Implement time and space efficient optimized algorithms.**
- **Demonstrate a familiarity with major algorithms and data structures.**
- **Apply important algorithmic design paradigms and methods of analysis.**
- **Synthesize efficient algorithms in common engineering design situations.**
- **Use different algorithms for solving real word problems.**

List of Experiments:

1. List the factors that may influence the space complexity of a program. Write a recursive and non-recursive function to compute nth Fibonacci. Compare the time requirements of non-recursive function with those of recursive version.
2. Program to solve the fractional knapsack using greedy approach.
3. Program to implement the MST using prim's method.
4. Program to implement the MST using kruskal's method.
5. Program to implement the Huffman coding.
6. Program to implement the dijkstra's method of shortest path.
7. Program to implement the making change using greedy strategy.
8. Program to implement the binary search.
9. Program to implement the merge, quick and heap sort. Compare the time complexity for best case, worst case and average case. (Taking very large data set)
10. Program to implement the strassen's matrix multiplication.
11. Program to implement the assembly line scheduling.
12. Program to implement the chained matrix multiplication.
13. Program to implement the Longest Common Sequence.
14. Program to implement the all pair shortest path algorithm.
15. Program to implement the 0/1 knapsack.
16. Program to implement the exponent using dynamic programming.
17. Program to implement the making change using dynamic programming.
18. Program to implement the TSP using backtracking.

Details of Assessment Instruments under LW Practical Component:

- Experiments during lab sessions and record-keeping of lab work (Term Work)
- Assignments / Mini project / Quiz / Practical Test

Course Outcomes (COs):

At the end of this course students will be able to

1. Analyze worst case running times of algorithms using asymptotic analysis.
2. Derive and solve recurrences describing the performance of divide and conquer algorithms.

3. Understand backtracking algorithms and its analysis.
4. Synthesize dynamic programming algorithms and analyze them.
5. Synthesize greedy algorithms and analyze them.
6. Use graph implementation to solve real world problems.

Course Code: 18CP218T					**Course Name: Object Oriented Concepts & Programming**			
Teaching Scheme					**Examination Scheme**			
L	**T**	**P**	**C**	**Hrs/ Wk**	**Theory**			**Total**
					Continuous Evaluation	**Mid Semester**	**End Semester**	**Marks**
3	0	0	3	3	25	25	50	100

Prerequisites: Programming Fundamentals

Learning objectives:

- To improve the analytical skills of Object Oriented programming
- To learn theory and concepts of Object Oriented programming for solving and critical analysis.
- To apply the object oriented design and programming skills in realistic applications using Java programming language.

Unit wise allocation of course content

UNIT 1 (09 L)

Object Oriented Concepts: Object-Oriented Programming vs. Non-Object-Oriented Programming, Classes, Objects, Abstraction, Inheritance, Polymorphism, Encapsulation, Associations, Aggregation and Composition, etc.

Introduction to Java :

Basics of Java programming, Data types, Variables, Operators, Control structures including selection, Looping, Java methods, Overloading, Math class, Arrays in java.

UNIT 2 (10 L)

Objects and Classes :

Basics of objects and classes in java, Constructors, Finalizer, Visibility modifiers, Methods and objects, Inbuilt classes like String, Character, StringBuffer, File, this reference.

Inheritance and Polymorphism :

Inheritance in java, Super and sub class, Overriding, Object class, Polymorphism, Dynamic binding, Generic programming, Casting objects, Instance of operator, Abstract class, Interface in java, Package in java, UTIL package.

UNIT 3 (10 L)

I/O programming :

Text and Binary I/O, Binary I/O classes, Object I/O, Random Access Files.

Exception Handling:

Exception handling with try-catch-finally.

Multithreading in java :

Thread life cycle and methods, Runnable interface, Thread synchronization, Collections in java.

UNIT 4 (10 L)

Event and GUI programming :

Event handling in java, Event types, Mouse and key events, GUI Basics, Panels, Frames, Layout Managers: Flow Layout, Border Layout, Grid Layout, GUI components like Buttons, Check Boxes, Radio Buttons, Labels, Text Fields, Text Areas, Combo Boxes, Lists, Scroll Bars, Sliders, Windows,

Menus, Dialog Box, Applet and its life cycle.

Student centering learning: (The student centering learning contents should be declared at the commencement of semester. It should be maximum 10% ; however exact contents is left to faculty)

Lecture: 39 Hrs

Approximate Total: 39 Hrs

Texts and References:

1. The Complete Reference, Java 2 (Fourth Edition), Herbert Schild, TMH.
2. Introduction to Java Programming (Comprehensive Version), Daniel Liang, Pearson.
3. Programming in Java, Sachin Malhotra & Saurabh Chaudhary, Oxford University Press.
4. Murach's Beginning Java 2, Doug Lowe, Joel Murach and Andrea Steelman, SPD.
5. Core Java Volume-I Fundamentals, Horstmann & Cornell, Pearson Education.
6. Java Programming, D. S. Malik, Cengage Learning.
7. Balaguruswamy, Programming with Java – A primer, Tata McGraw Hill

Course Outcomes (COs):

At the end of this course students will be able to

1. Differentiate between object oriented programming and procedural programming paradigm
2. Understand concepts of object oriented programming like encapsulation, inheritance, polymorphism, etc.
3. Design and Implement solutions using object oriented programming concepts
4. Show competence in the use of the Java programming language in the development of small to medium-sized application programs that demonstrate professionally acceptable coding and performance standard
5. Understand the basic principles of the object-oriented programming
6. Demonstrate an introductory understanding of graphical user interfaces, multithreaded programming, and event-driven programming.

<table>
<tr><td colspan="5">Lab Code 18CP218P</td><td colspan="3">Lab Name: Object Oriented Concepts & Programming Lab</td></tr>
<tr><td colspan="5">Teaching Scheme</td><td colspan="3">Examination Scheme</td></tr>
<tr><td rowspan="2">L</td><td rowspan="2">T</td><td rowspan="2">P</td><td rowspan="2">C</td><td rowspan="2">Hrs/Wk</td><td colspan="2">Practical</td><td>Total</td></tr>
<tr><td>Continuous evaluation</td><td>End semester exam</td><td>Marks</td></tr>
<tr><td>0</td><td>0</td><td>2</td><td>1</td><td>2</td><td>25</td><td>25</td><td>50</td></tr>
</table>

Prerequisites: Computer Programming

Course objectives:

- To learn theory and concepts of Object Oriented programming
- To learn theory and concepts of Object Oriented concepts
- To apply the object oriented design and programming skills in realistic applications

List of Experiments:

1. Set up and get familiar with Java programming environment;
2. Study language features of Java (variables, data types, declarations, loop and branch constructs, etc.)
3. Class and Objects: study and implement classes based application using Java
4. Inheritance: study and implement various types of inheritance in Java.
5. Polymorphism: study and implement various types of Polymorphism in Java;
6. Study and implement Abstract class and Interfaces in Java;
7. Study and implement Exception handling in Java
8. Study and implement multi-threaded application in Java
9. Program to demonstrate I/O Operations.
10. GUI programming using Java Applet, Events and AWT

Course Outcomes (COs):

At the end of this course students will be able to

1. Differentiate between object oriented programming and procedural programming paradigm
2. Understand concepts of object oriented programming like encapsulation, inheritance, polymorphism, etc.
3. Design and Implement solutions using object oriented programming concepts
4. Design and Implement multi threaded applications
5. Design and implement GUI based applications
6. Carry out continuous testing of object oriented application

Course Code: CP 213T					Course Name: Computer Networks			
Teaching Scheme					Examination Scheme			
L	T	P	C	Hrs/ Wk	Theory			Total
					Continuous Evaluation	Mid Semester	End Semester	Marks
4	0	0	4	4	25	25	50	100

Prerequisites: Discrete Mathematics, Data and File Structures

Learning objectives:

- To understand the communication network design
- understand state-of-the-art in network protocols, architectures,
- To learn the design and implementation of network applications

Unit wise allocation of course content

UNIT 1 (10 L)

Introduction: Nuts and Bolts, Performance parameters: throughput, delay, etc., Layered Architecture (OSI and TCP/IP)

Applications: Network application Design, Socket Programming, Client-server applications, Echo and Chat applications, FTP,DNS, Peer to Peer file sharing application

UNIT 2 (16 L)

Data link layer: Introduction, Media access protocols (ALOHA, CSMA based) , Ethernet 802.3, Token ring 802.5, Reliability Issue: sliding window

Internetworking and Routing: Best effort Service, Virtual Circuits, IP Addressing,

UNIT 3 (9 L)

Internetworking and Routing: Routing Issues, Distance Vector and Link State routing, Intra and Inter Autonomous System Routing (OSPF, RIP, BGP), Broadcast and Multicast Routing Issues

UNIT 4 (17 L)

Transport Layer: End to end delivery issues, Reliable data transfers, Congestion Control, Traffic engineering and Quality of service, TCP, UDP

Advanced Topics: QoS over IP, IPV6, Infrastructure-less networks: wireless ad hoc and sensor networks, and Internet of Things (IoT)

Student centering learning: (The student centering learning contents should be declared at the commencement of semester. It should be maximum 10% ; however exact contents is left to faculty)

Lecture: 52 Hrs

Tutorial: 0 Hrs

Approximate Total: 52 Hrs

Texts and References:

1. James Kurose and Keith Rose, "*Computer Networking: A Top Down Approach*", Pearson Education
2. Larry L Peterson and Bruce S Davie, "*Computer Networks: A Systems Approach*", Elsevier
3. Andrew S Tanenbaum, "*Computer Networks*", Pearson Education
4. Behrouz A Forouzan, "*Data Communication and Networking*", McGraw Hill

5. William Stallings, "*Data and Computer Communication*", Pearson Education

Course Outcomes (COs):

At the end of this course students will be able to

1. Understand functionality of layered network architecture
2. Illustrate various types of network topologies, network devices and their functions within a network Solve computational problems in the domain of computer networks
3. Use network simulator tool(s) for testing and analysis of different computer network scenarios
4. Configure various network services to meet specific needs
5. Understand different computer network protocols and implement computer network application addressing specific requirements (scalability, security, etc.)

Lab Code CP 213P					Lab Name: Computer Networks Lab		
Teaching Scheme					Examination Scheme		
					Practical		Total
L	T	P	C	Hrs/Wk	Continuous evaluation	End semester exam	Marks
0	0	3	1.5	3	25	25	50

Prerequisites: Data and File Structures, Discrete Mathematics

Course objectives:

- To understand the communication network design
- understand state-of-the-art in network protocols, architectures,
- To learn the design and implementation of network applications

List of Experiments:

1. To study and prepare LAN cables (cross and straight), to configure LAN and perform Static Routing
2. Introduction to Socket Programming- Design and Implement client-server elements of a few network applications e.g. Echo client and server, Time client and server, Online Quiz and Buzzer Application, etc
3. Configure DHCP in a small LAN and understand its functionality using Wireshark/ Packet Tracer
4. Configure DNS in a small LAN and understand its functionality using Wireshark/ Packet Tracer
5. Understand functionality of HTTP using Wireshark/ Packet Tracer
6. Understand functionality of TCP and UDP using Wireshark/ Packet Tracer
7. Configure virtual LAN and understand its functionality using Wireshark/ Packet Tracer
8. Configure OSPF and BGP in a small LAN
9. Configure and install NS2/NS3 and simulate communication between two nodes
10. Simulation of TCP/UDP connections and performance analysis

Course Outcomes (COs):

At the end of this course students will be able to

1. Understand functionality of layered network architecture
2. Illustrate various types of network topologies, network devices and their functions within a network
3. Solve computational problems in the domain of computer networks
4. Use network simulator tool(s) for testing and analysis of different computer network scenarios
5. Configure various network services to meet specific needs
6. Understand different computer network protocols and implement computer network application addressing specific requirements (scalability, security, etc.)

Course Code: CP 214T					Course Name: Computer Organization & Programming			
Teaching Scheme					Examination Scheme			
L	T	P	C	Hrs/Wk	Theory			Total
					Continuous Evaluation	Mid Semester	End Semester	Marks
3	1	0	4	4	25	25	50	100

Prerequisites: Digital Logic Design

Learning objectives:

- To understand architecture of digital computer and its operations
- To learn design of various functional units of a digital computer
- To learn techniques for interfacing for memory & input/output devices with a digital computer

Unit wise allocation of course content

UNIT 1 (12 L, 4 T)

Basic Structure of Computers: Block Diagram of General Purpose Computers; Detailed Understanding of Each Functional Unit; Data Transfer Across Bus; Simple Bus Structures With Registers and Memory; Details of Address; Control and Data Bus with Interfacing

Instruction Set: Instruction format; Addressing Modes. Instruction Set of A Simple Real World Microprocessor Covering Data Transfer; Arithmetic; Logical; Control; Subroutine; Stack; Basic I/O and Interrupt Operations

UNIT 2 (12 L, 4 T)

Central Processor Unit Design: Single Bus Architecture; Detailed Design of Execution Unit Using Hardwired Control as well as Microprogrammed Control; Horizontal and Vertical Microinstructions; Concept of Nano-programming; Introduction to RISC and CISC Architectures.

Arithmetic Processor Design: Addition; Subtraction; Multiplication and Division Algorithms in Signed Binary Arithmetic for Fixed and Floating Point Representations and Related Design Standards and Issues

UNIT 3 (12 L, 4 T)

Memory and Input-Output Organisation: Types of Memory; Memory Hierarchies; Organisation of Static and Dynamic Semiconductor Memories; Associative Memory Organization; Cache Organisation. Device Interfacing and Selection; Memory and I/O Mapped I/Os; Modes of Data Transfer-Programmed; Interrupt and DMA Driven I/O-Interrupt Types and Priority Schemes; Synchronous and Asynchronous Data Transfer

UNIT 4 (3 L, 1 T)

Pipeline And Vector Processing: Flynn's taxonomy; Parallel Processing; Pipelining; Arithmetic Pipeline; Instruction; Pipeline; RISC Pipeline; Vector Processing; Array Processors, Assembly Language Programming

Student centering learning: (The student centering learning contents should be declared at the commencement of semester. It should be maximum 10% ; however exact contents is left to faculty)

Lecture: 39 Hrs
Tutorial: 13 hrs
Approximate Total: 52 Hrs

Texts and References:

1. M. Morris Mano, "Computer System Architecture", Pearson Education
2. Yale N. Patt, Sanjay J. Patel, "Introduction to Computing Systems" McGraw Hill
3. Hamacher, Vranesic, Zaky, "Computer Organization", McGraw Hill
4. Andrew S. Tanenbaum and Todd Austin, "Structured Computer Organization", Pearson Education
5. N D Jotwani, "Computer system organization", McGraw Hill
6. R.S.Gaonkar, "Microprocessor Architecture, Programming and Applications with 8085A", Penram International
7. Douglas Hall, Microprocessors and Interfacing, TMH

Course Outcomes (COs):

At the end of this course students will be able to

1. Understand the organization of the control unit, arithmetic and logical unit, memory unit and I/O unit
2. Apply knowledge of the processor's internal registers and operations by use of a PC based microprocessor simulator
3. Implement assembly language programs to provide solutions of given problems
4. Design a basic central processing unit
5. Design interfacing of memory and I/O modules with CPU
6. Compare performance of different types of computer architectures