Sanjay Ghodawat University, Kolhapur

School of Technology

Department of Computer Science and Engineering

Structure and Contents for S. Y. B. Tech. Computer Science and Engineering Program

(AY 2020-21) R0

AME/P/80/00

Semester III

Course Code	Course Title	L	T	P	C	Component	Evaluation Scheme			
							Exam	WT	Min. Pass	
CST201 (BS | SS) Version: 1.0	Mathematics for Modern Computing	3	-	-	3	Theory 100	FET	20	40	40
							CAT I	15		
							CAT II	15		
							ESE	50	40	40
CST203R1 (PC | ST) Version: 1.0	Discrete Structures	3	1	-	4	Theory 100	FET	20	40	40
							CAT I	15		
							CAT II	15		
							ESE	50	40	40
CST205 (PC | ST) Version: 1.0	Object Oriented Programming	2	-	4	4	Practical 100	FEP	50	40	40
							POE	50	40	
CST207 (PC | ST) Version: 1.0	Data Structures	3	-	-	3	Theory 100	FET	20	40	40
							CAT I	15		
							CAT II	15		
							ESE	50	40	40
CST209 (PC | ST) Version: 1.0	Data Structures Laboratory	-	-	2	1	Practical 100	FEP	50	40	40
							POE	50	40	
CST211 (PC | ST) Version: 1.0	Software Engineering	3	1	-	4	Theory 100	FET	20	40	40
							CAT I	15		
							CAT II	15		
							ESE	50	40	40
CST213 (PC | ST) Version: 1.0	Microprocessors and Microcontrollers	3	-	-	3	Theory 100	FET	20	40	40
							CAT I	15		
							CAT II	15		
							ESE	50	40	40
CST215 (MC) Version: 1.0	Professional Skill Development - I	-	-	2	NC	Practical 100	FEP	100	40	40
CST217 (MC) Version: 1.0	Environmental Science	1	-	2	NC	Practical 100	FEP	100	40	40
	Total	18	02	10	22		Total Hrs: 30, Total Credits: 22 NC: 2			

Sanjay Ghodawat University, Kolhapur

School of Technology

Department of Computer Science and Engineering

Structure and Contents for S. Y. B. Tech. Computer Science and Engineering Program

(AY 2020-21) R0

AME/P/80/00

Sem. IV										
Course Code	**Course Title**	**L**	**T**	**P**	**C**		**Evaluation Scheme**			
						Component	**Exam**	**WT**	**Min. Pass**	
CST202 (PC | ST) Version: 1.0	Theory of Computation	3	1	-	4	Theory 100	FET	20	40	40
							CAT I	15		
							CAT II	15		
							ESE	50	40	40
CST204 (PC | ST) Version: 1.0	Computer Programming Laboratory - I	2	-	4	4	Practical 100	FEP	50	40	40
							POE	50	40	
CST206 (PC | ST) Version: 1.0	Database Management Systems	3	-	-	3	Theory 100	FET	20	40	40
							CAT I	15		
							CAT II	15		
							ESE	50	40	40
CST208 (PC | ST) Version: 1.0	Database Management Systems Laboratory	-	-	2	1	Practical 100	FEP	50	40	40
							POE	50	40	
CST210 (PC| ST) Version: 1.0	Operating Systems	3	-	-	3	Theory 100	FET	20	40	40
							CAT I	15		
							CAT II	15		
							ESE	50	40	40
CST212 (PC| ST) Version: 1.0	Operating Systems Laboratory	-	-	2	1	Practical 100	FEP	100	40	40
CST214 (PC | ST) Version: 1.0	Computer Networks	3	-	-	3	Theory 100	FET	20	40	40
							CAT I	15		
							CAT II	15		
							ESE	50	40	40
CST216 (PC | ST) Version: 1.0	Computer Networks Laboratory	-	-	2	1	Practical 100	FEP	100	40	40
CST218 (PW | ST) Version: 1.0	Mini Project-I	-	-	2	1	Practical 100	FEP	50	40	40
							POE	50	40	
CST220 (MC) Version: 1.0	Professional Skill Development - II	-	-	2	NC	Practical 100	POE	100	40	40
	Total	14	01	14	21		Total Hrs: 29, Total Credits: 21 NC: 1			

CST201: Mathematics for Modern Computing

(Ver 1.0, Basic Science, School of Science)

Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	WT	Min Pass	
3	-	-	3	Theory 100	FET	20	40	40
					CAT-I	15		
					CAT-II	15		
					ESE	50	40	

Course Description: This course is at odd semester of Second year B Tech Technology for Computer Science Engineering. It is a foundation course in Numerical method and Fuzzy Mathematics and may be pre requisites for other courses. It covers solution of algebraic and transcendental equation by different methods, Numerical differentiation and Numerical integration, Introduction to fuzzy set and Fuzzy equations.

Course Outcomes: At the end of this course students will able to-

CO1 Solve[3] Algebraic and Transcendental Equations.

CO2 Find[2] first and second order derivative by numerical method.

CO3 Evaluate[3] definite integral by numerical method.

CO4 Apply[3] the probability distributions.

CO5 Distinguish[2] between crisp and fuzzy set and to write different terms of fuzzy set.

CO6 Determine[5] A+B, A-B, A.B and A/B also to solve A+X=B, A.X=B

Syllabus (Theory)

Units	Description	Hours
I	**Algebraic and Transcendental Equations:** Introduction, Types of errors, Rules for estimate errors, Roots of Equation by Bisection Method, False position method, Secant method, Newton- Raphson method, multiple roots by Newton method.	8
II	**Numerical differentiation and Integration:** Newton forward and backward difference formulae for equally spaced data, Derivative using stirling formula, Newton's divided difference formula for unequally spaced data,	6
III	**Numerical Integration:** Newton's cotes qudrature formula, Trapezoidal rule, Simpson one third rule, Simpsons three eight rule, waddles' formula and Romberg Integration.	6
IV	**Statistics:** Mean, Mode, median, standard deviation, Random variable, Probability mass function and probability density function, Binomial, Poisson and Normal distributions.	6
V	**Introduction to Fuzzy sets:** Basic concepts of fuzzy sets, Crisp set and Fuzzy set, membership functions, Basic operations on fuzzy sets, Properties of fuzzy sets	6

VI	**Fuzzy Arithmetic:** Fuzzy numbers, Fuzzy cardinality, Operations on Fuzzy numbers, Fuzzy equations of type A + X = B and A.X = B	8

Text Book

1. Dr. B. S. Grewal, "Higher Engineering Mathematics", Khanna Publishers, Delhi.
2. P. N. Wartikar & J. N. Wartikar, "A text book of Applied Mathematics, Vol.-I, II, III", Pune Vidyarthi Griha Prakashan, Pune.

References

1. George J. Klir, Bo Yuan, "Fuzzy sets and Fuzzy Logic".
2. Erwin Kreyszig, "Advanced Engineering Mathematics", Wiley India Pvt. Ltd.
3. E Balguruswamy, "Numerical Methods", Tata McgrawHill Publication Company Ltd., 8th Edition,2002.
4. Dr. V.N.Vedamurthy, "Numerical Methods", Vikas Publication.
5. G.Haribaskaran, "Numerical Methods", Laxmi Publications Pvt.Ltd, New Delhi, 1st Edition, 2006.
6. R.L.Burden and J.D.Faires, "Numerical Analysis Theory and Applications", Cengage Learning India Pvt.Ltd.,New Delhi,1st Edition,2005.

CST203R1 :Discrete Structures							
Ver 1.1, Program Core, School of Technology							
Lect.	Tut.	Pract.	Credits	Evaluation Scheme			
				Component	Exam	Weightage	Pass %
3	1	-	4	Theory 100 Marks	FET	20	40
					CAT I	15	
					CAT II	15	
					ESE	50	40

Prerequisite: Basics of mathematics

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

CO1 **Relate**[3] logical notation to define mathematical concepts such as logic, set theory, relations, functions, probability.

CO2 **Illustrate**[2] the knowledge and skills obtained to solve a variety of discrete mathematical problems

CO3 **Make use of**[3] the concepts and algorithms of graph theory and elementary combinatorial processes.

CO4 **Apply**[3] the appropriate formulas to calculate permutations & combinations.

Syllabus (Theory)

Units	Description	Hours
I.	**Set theory:** Basic concepts of set theory, types of operations on sets, ordered pairs, Cartesian Product, representation of discrete structures, relation, properties of binary relations, matrix and graph representation, partition and covering of set, equivalence relation, composition, POSET and Hasse diagram, Function – types, composition of functions, Inverse function.	7
II.	**Graph theory:** Basic concepts of graph theory, Storage representation and manipulation of Graphs, PERT and related techniques.	7
III.	**Mathematical Logic:** Statements and notations, connectives – negation, Conjunction, disjunction, conditional, bi-conditional, Statement formulas and truth tables well formed formulas, Tautologies, Equivalence of formulas, Duality law, Tautological implications, functionally complete sets of connectives, other connectives, Normal and principal normal forms, completely parenthesized infix and polish notations, Theory of Inference for statement calculus – validity using truth table, rules of inference, consistency of Premises and indirect method of proof.	7
IV.	**Lattices and Boolean algebra:** Introduction to Lattice, definition, Lattice as POSETs, Properties, Lattice , Boolean algebra definition and examples, Boolean functions, representation and minimization of Boolean functions.	7
V.	**Counting:** The Basics of Counting, Permutations and Combinations, Generalized Permutations and Combinations, Generating Permutations and Combinations	7

VI.	**Discrete Probability:** Discrete Probability, Conditional Probability, Bayes' Theorem, Information and Mutual Information.	**7**

Tutorial

One hour per week per batch tutorial is to be utilized to ensure that students have properly learnt the topics covered in the lectures. It should comprise of minimum 8-10 tutorials. Students of different batches should perform different tutorials based on the following guidelines-

1. Concepts of set theory, operations on sets
2. Function and its types
3. Concepts of graph theory, PERT and related techniques
4. Mathematical logic
5. Permutations, Combinations
6. Discrete Probability

Textbooks:

1. J.P Tremblay, R.Manohar, Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw-Hill Publishing Company Limited, 1997

References :

1. J.L. Mott, A. Kandel, T.P. Baker, "Discrete Maths for Computer Scientists & Mathematicians", Second Edition, Prentice Hall of India Pvt Limited, New Delhi, 2009

CST205: Object Oriented Programming

(Ver 1.0, Program Core, School of Technology)

Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	WT	Min. Pass	
2	-	4	4	Practical 100	FEP	50	40	40
					POE	50	40	

Course Description: This course is at odd semester of second year B Tech computer science and engineering. It is a basic foundation course in programming and may be pre requisites for other courses. It covers basics of programming with object oriented concepts and their application.

Prerequisite: Basics of computer programming
Co requisite: None
Course Outcomes: After the end of this course students will able to-

CO1 **Explain**[2] the principles of the object oriented programming concepts.
CO2 **Describe**[1] object oriented programs using arrays, array of objects, pointers, constructors and destructors as required.
CO3 **Demonstrate**[3] the different types of inheritance.
CO4 **Design**[5] C++ program by using features of polymorphism.

Syllabus (Theory)

Units	Description	Hrs
I	**Overview :** Need of Object-Oriented Programming (OOP), Object Oriented Programming Paradigm, Basic Concepts of Object-Oriented Programming, Benefits of OOP, C++ as object oriented programming language, C++ programming Basics, Data Types, Structures, Enumerations, control structures, Class, Object, class and data abstraction, class scope and accessing class members, separating interface from implementation, controlling access to members.	03
II	**Arrays, Pointer and Functions:** Arrays of objects, Pointers to objects, Type checking C++ Pointers, This Pointer, Pointers to derived types, Pointers to class members, Dynamic allocation operators- new & delete operators. Functions- Function, function prototype, accessing function and utility function, Constructors and destructors, Objects and Memory requirements, Static Class members, data abstraction and information hiding, inline function. **Inheritance:** Single Inheritance, Multilevel Inheritance, Multiple Inheritances, Hybrid Inheritance, Hierarchical Inheritance, Virtual base classes.	08
III	**Polymorphism:** Overloading - Function overloading, Overloading constructor function, copy constructors, Operator overloading using friend function, Overloading new & delete operators, overloading some special operators like	05

[],(),->,Comma operator.
Virtual Functions- Pure virtual function, calling virtual function through a base class, Abstract classes, Early vs Late binding.

IV **File and Streams:** Streams, String I/O, Character I/O, Object I/O, I/O with 08 multiple objects, File pointers and redirections. C++ streams, C++ stream classes, RTTI, Namespace fundamentals, STL containers, STL algorithms, STL iterators.
Templates: Templates - Generic classes, Generic functions, Applying generic functions, type name &export keyword, power of templates.
Exception Handling– Fundamentals, Handling derived class exceptions, exception handling options: catching, throwing & handling of the exception.

Practical

Four hours per week per batch practical is to be utilized for writing programs to ensure that students have properly learnt the topics covered in the lectures. This shall include extra problem statements and there implementations to strengthen the programming logic.
It should comprise of minimum 10-12 experiments. Students of different batches should implement different programs based on the following guidelines in UNIX / Linux platform.

1. Write a program to create BOX Class. Write appropriate the volume of a box. Functions to compute declare a static data member and member function to keep track of objects created.
2. Write a C++ program to demonstrate types of Constructor and Destructor.
3. Write a C++ program to dynamically accept numbers and implement matrix multiplication.
4. Define a student class with USN, Name and marks in 3 tests of subject. Declare an array of 10 student objects. Using appropriate functions, find the average of the two better marks for each student. Print the USN, Name and average marks of all the students.
5. Implement a C++ program for set operation –Union and Intersection.
6. Write a C++ program to return maximum of two positive numbers using inline function.
7. Implement a C++ program to add two numbers using friend functions.
8. Write a C++ program to implement single and multilevel inheritance.
9. Implement C++ program to demonstrate hybrid inheritance.
10. Write a C++ program to calculate the area of circle, rectangle and triangle using function overloading
11. Write a C++ program to create a class called STACK using an array of integers. Implement the following operations by overloading the operator '+' and '—'
    i) S1=s1+element; where s1 is an object of the class STACK and element is an integer to be pushed on the top of the stack.
    ii) s1=--s1; where s1 is an object object of the class STACK.'—'.operator pops

the element.

Handle the stack empty and full condition. Also display the contents of the stack after each operation by overloading the << operator.

12. Write a C++ program to understand the need of virtual base class.
13. Write a C++ program to calculate number of lines, words and characters in a file
14. Implement bubble sort technique using function template and virtual function
15. Write a C++ program to implement stack using class template and exception handling.

Text Book

1. The Complete Reference: C++ - Herbert Schildt (TMGH) Fourth Edition.
2. Object-Oriented Programming in C++ - Rajesh K. Shukla (Wiley) India Edition.

References

1. Object Oriented Programming in Turbo C++ - Robert Lafore (Galgotia).
2. Object Oriented Programming with C++ - Sourav Sahay (Oxford) Second Edition.

CST207: Data Structures

(Ver 1.0, Program Core, School of Technology)

Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	WT	Min. Pass	
3	-	-	3	Theory 100	FET	20	40	40
					CAT I	15		
					CAT II	15		
					ESE	50	40	40

Course Description: This course is at odd semester of second year B Tech computer science and engineering. It is a foundation course in computer science and may be pre requisites for other courses in computer science. It covers basics of data storages, data access and their application.

Prerequisite: Computer Programming, Basics of mathematics.
Co requisite: None

Course Outcomes: After the end of this course students will able to-

CO1 **Explain**[2] concepts of C and data structures.
CO2 **Examine**[4] different searching and sorting techniques.
CO3 **Describe**[2] concept of stack.
CO4 **Apply**[3] knowledge of queue for its implementation.
CO5 **Demonstrate**[3] implementation of list.
CO6 **Demonstrate**[3] concept of trees.

Syllabus (Theory)

Units	Description	Hrs
I	**Basics of Data Structures**: Overview of C- Basic data types, control structures, array, function, structure, pointers, Time and Space complexity.	04
II	**Searching and Sorting Techniques:** Linear search, binary search, bubble sort, selection sort, insertion sort, merge sort, quick sort, radix sort, heap sort.	06
III	**Stacks:** Definition, representation, operations, applications of stack. **Queues:** Definition, representation, operations, applications of queue, circular queue, priority queue.	08
IV	**Lists:** Definition representation, operations, singly, doubly and circular linked lists.	06
V	**Trees:** Basic terminology, representation, binary tree, traversal methods, binary search tree.	06

VI	**Graphs:** Basic concept of graph theory, storage representation, graph traversal techniques- BFS and DFS, Graph representation using sparse matrix.	06

Text Book

1. Let us C – Yashwant Kanetkar (BPB).
2. Data Structure using C- A. M. Tanenbaum, Y. Langsam, M. J. Augenstein (PHI).

References

1. Schaum's Outlines Data Structures – Seymour Lipschutz (MGH).
2. Data Structures- A Pseudocode Approach with C – Richard F. Gilberg and Behrouz A. Forouzon 2nd Edition.

CST209: Data Structures Laboratory

(Ver 1.0, Program Core, School of Technology)

Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	WT	Min. Pass	
-	-	2	1	Practical 100	FEP	50	40	40
					POE	50	40	

Course Description: This course is at odd semester of second year B Tech computer science and engineering. It is a foundation course in computer science and may be pre requisites for other courses in computer science. It covers implementation of searching sorting techniques, basics of data storages, data access and their application.

Prerequisite: Computer Programming, Basics of mathematics.
Co requisite: None

Course Outcomes: After the end of this course students will able to-

CO1 **Experiment**[4] different searching and sorting techniques.
CO2 **Experiment**[4] concept of stack.
CO3 **Apply**[3] knowledge of queue for its implementation.
CO4 **Demonstrate**[3] implementation of list.
CO5 **Demonstrate**[3] concept of trees.

Practical

Two hours per week per batch practical is to be utilized for writing programs to ensure that students have properly learnt the topics covered in the lectures. This shall include extra problem statements and there implementations to strengthen the programming logic.

It should comprise of minimum 10-12 experiments. Students of different batches should implement different programs based on the following guidelines-

1. Program for Searching problem, linear search, binary search.
2. Program for Sorting problem, bubble sort, selection sort.
3. Implementation of insertion sort, merge sort.
4. Implementation of quick sort, radix sort.
5. Write a program for hash functions
6. Program to implement Stacks.
7. Program to implement Queues.
8. Program to implement Circular queue, priority queue.
9. Implementation of singly linked lists.
10. Implementation of doubly linked lists.
11. Implementation of circular linked lists.
12. Implementation of stack using linked lists.
13. Implementation of queue using linked lists.

14. Program to implement binary search tree.
15. Program to implement Breadth First Search (BFS).
16. Program to implement Depth First Search (DFS).

Text Book

3. C the Complete Reference, Herbert Schildt, Tata McGraw Hills, 6th edition.
4. Schaum's Outlines Data Structures – Seymour Lipschutz (MGH).

References

3. Data Structure using C- A. M. Tanenbaum, Y. Langsam, M. J. Augenstein (PHI).
4. Data Structures- A Pseudocode Approach with C – Richard F. Gilberg and Behrouz A. Forouzon 2nd Edition.

CST211: Software Engineering

(Ver 1.0, Program Core, School of Technology)

Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	WT	Min. Pass	
3	1	-	4	Theory 100	FET	20	40	40
					CAT I	15		
					CAT II	15		
					ESE	50	40	40

Course Description: This course is at odd semester of second year B Tech computer science and engineering. It is a one of the core course in computer science and may be pre requisites for other courses in computer science. It covers basics of software engineering, software development life cycles and testing strategies.

Prerequisite: This course requires the student should prerequisites with basic concepts of software products; idea about existing software's available in the market and software driven business to all the aspects in day to day life like automation systems, online transactions, online transactions, e-commerce etc. students should know the clear layer between software engineering discipline and other discipline.

Course Outcomes: After the end of this course students will able to-

CO1 **Explain**[2] the requirement of software engineering.
CO2 **Explain**[2] the software life cycle models.
CO3 **Design**[3] the SRS.
CO4 **Distinguish**[4] different roles of software development staffs.
CO5 **Compare**[4] Function Oriented Design & Object Oriented Design.
CO6 **Develop**[5] correct and robust software products.

Syllabus (Theory)

Units	Description	Hrs
I	**Introduction to Software Engineering:** The Evolving Role of Software, Software Characteristics, Categories of Computer Software, The Software Myths.	05
II	**Software Process and Software Development Models:** Software Processes: Software Process Components, The Capability Maturity Model Integration, Process Patterns, Software Development Models: Waterfall Model, Prototyping, Model, Spiral Model, Incremental Model, Time boxing Model, V Model.	07
III	**Requirements Engineering:** Eliciting Requirements, Negotiating Requirements, Validating Requirements, Software Requirement specifications (SRS): Role of SRS, validation of SRS document, Organization of structure, case study.	06

IV Agile Project Management: 06

Core Agile Concepts Overview, Methodologies, The Agile Manifesto Overview, Scrum Methodology, Project (Product; Release) Initiation, Scrum Planning, Scrum Sprint Planning and Executing, Identify Stakeholders.

V **Agile Teams and Team Space**: Overview, Scrum Master/Coach, Product 07
Owner/customer, Team Members/Developers (XP), Develop Epics and Stories, Create Stories, Create Product Backlog, Create Product Roadmap, Sprint Reviews, Closing: Sprint, Release and Product Retrospectives.

VI Testing Strategies: 05

Testing, Unit Testing, Black Box Testing, White Box Testing, Integration Testing, System Testing.

Tutorial

One hour per week per batch tutorial is to be utilized to ensure that students have properly learnt the topics covered in the lectures. It should comprise of minimum 8-10 tutorials. Students of different batches should perform different tutorials based on the following guidelines-

1. Introduction to software engineering.
2. Software process and software development models.
3. Software requirements engineering.
4. Agile Project Management.
5. Agile Teams and Team Space.
6. Design engineering.
7. Software testing strategies.

Text Book

1. Software Engineering: A precise Approach - Pankaj Jalote (Wiley India)
2. Fundamentals of Software Engineering - Rapit Mall (3rd Edition)(PHI)
3. Software Engineering by Jan Sommerville (9th Edition) Pearson
4. Software Engineering Principles & Practices by Rohit Khurana ITLESL (2nd Edition) Vikas Publishing House Pvt. Ltd.
5. Agile Project Management for Dummies by Mark C. Layton

References

1. Software Engineering –A Practitioner's approach Sixth Edition By Roger S. Pressman.
2. Software Engineering - Concepts & Practices -- Ugrasen Suman (Cenage Learning).
3. Software Engineering Fundamentals -Behforooz & Hudson (Oxford : Indian Edition 1st).
4. Agile Project Management: A Complete Beginner's Guide To Agile Project Management by Marcus Ries and Diana Summers.

CST213: Microprocessors and Microcontrollers

(Ver 1.0, Program Core, School of Technology)

Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	WT	Min. Pass	
3	-	-	3	Theory 100	FET	20	40	40
					CAT I	15		
					CAT II	15		
					ESE	50	40	40

Course Description: This course is at odd semester of second year B Tech computer science and engineering. The purpose of this course is to the students can learn the fundamental concepts of microprocessor and microcontroller systems. The student will be able to incorporate these concepts into their electronic designs for other courses where control can be achieved via a microprocessor/controller implementation.

Prerequisite: Basics of digital systems/circuits.

Co requisite: None

Course Outcomes: After the end of this course students will able to-

CO1 **Explain**2 basic 8085 and 8086 microprocessors architecture.
CO2 **Discuss**2 8086 microprocessor architecture with software aspects.
CO3 **Classify**2 microprocessor configurations.
CO4 **Discuss**2 memory interfacing and I/O interfacing with 8085.
CO5 **Explain**2 basics of 8051 microcontrollers.
CO6 **Describe**2 programming aspects of 8051 real time control.

Syllabus (Theory)

Units	Description	Hrs
I	**The 8085 and 8086 Microprocessors:** 8085 Microprocessor architecture- 05 Addressing modes- Instruction set-Programming the 8085	
II	**8086 Software Aspects:** Intel 8086 microprocessor - Architecture - Signals- Instruction Set-Addressing Modes- Assembler Directives- Assembly Language Programming-Procedures-Macros-Interrupts And Interrupt Service Routines- BIOS function calls.	07
III	**Microprocessor Configurations:** Coprocessor Configuration – Closely Coupled Configuration – Loosely Coupled Configuration –8087 Numeric Data Processor – Data Types – Architecture –8089 I/O Processor –Architecture –Communication between CPU and IOP.	06
IV	**I/O Interfacing:** Memory interfacing and I/O interfacing with 8085 – parallel communication interface – serial communication interface – timer- keyboard/display controller – interrupt controller – DMA controller (8237) –	07

applications – stepper motor – temperature control.

V	**Microcontrollers:** Overview of 8051 microcontrollers, Architecture, I/O ports, memory organization, Addressing modes, Instruction set of 8051, Simple programs.	05
VI	**8051 Real Time Control:** Programming timer interrupts, Programming external hardware interrupts. Programming the serial communication interrupts, Programming 8051 timers/counters.	06

Text Book

2. Ramesh S. Gaonkar, "Microprocessor – Architecture, Programming and Applications with the 8085" Penram International Publisher, 5th Ed., 2006.
3. Yn-cheng Liu, Glenn A. Gibson, "Microcomputer systems: The 8086 / 8088 Family architecture, Programming and Design", second edition, Prentice Hall of India, 2006.
4. Kenneth J. Ayala, "The 8051 microcontroller Architecture, Programming and applications" second edition, Penram international.

References

1. Douglas V.Hall, "Microprocessors and Interfacing: Programming and Hardware", second edition, Tata Mc Graw Hill, 2006.
2. A.K.Ray & K.M Bhurchandi, "Advanced Microprocessor and Peripherals – Architecture, Programming and Interfacing", Tata Mc Graw Hill, 2006.
3. Peter Abel, "IBM PC Assembly language and programming", fifth edition, Pearson education / Prentice Hall of India Pvt.Ltd, 2007.
4. Mohamed Ali Mazidi, Janice Gillispie Mazidi, "The 8051 microcontroller and embedded systems using Assembly and C", second edition, Pearson education /Prentice hall of India, 2007.

CST215: Professional Development Skills- I

(Ver 1.0, Sem-III of Second Year B. Tech (Common for All Branches)

Course Description: This course aims to prepare the students for soft skills. The course will help them to understand their potential and set goals accordingly and organize their activities to achieve their set goals. The course also focuses on presentation and public speaking.

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

CO01 : apply[3] self analysis techniques.
CO02 : plan[4] and execute SMART goals.
CO03 : demonstrate[3] team building skills.
CO04 : prepare time table and action plan to achieve set goals.
CO05 : exhibit[3] presentation and public speaking skills.

Syllabus

Units	Description	Hrs
I	**Soft Skills:** What are soft skills? Importance of soft skills, selling your soft skills, identifying and improving your soft skills **Self Analysis:** Importance of knowing yourself, SWOT Analysis, Importance of Self Confidence, Self Esteem	04
II	**Goal Setting:** SMART Goals, Short Term goals, Moderate term goals, Long Term, Life Time Goals	04
III	**Team Building and Teamwork:** Introduction-meaning–aspects of team building, team Vs group, Stages of team building, Characteristics of effective team, role of a team leader, role of team members	04
IV	**Time Management:** Value of time, Diagnosing Time Management, Preparing to do list, Prioritizing work	04
V	**Presentation skills and Public Speaking:** Elements of an effective presentation, Structure of a presentation, Presentation tools, Audience analysis, Language: Articulation, Good pronunciation, Voice quality, Modulation, Accent and Intonation. Extempore and Prepared speeches	04

Note: During the practical sessions, it is expected that the contents of all modules should be delivered to the students of different batches and assignments be given based on the activities discussed as per the modules. Students must demonstrate the acquired skills by means of giving presentations, delivering public speeches, group discussions etc.

References:

1. Wallace & Masters, Personal development for Life & work, Thomson Learning.

2. Barun K. Mitra, Personality Development and Soft- Skills, Oxford University Press.

3. Fred Luthans, Organizational behavior, McGraw Hill.

4. Asa Don Brown, Interpersonal skills in the Workplace, Tate publishing and Enterprises.

Sanjay Ghodawat University, Kolhapur

School of Technology

Department of Computer Science and Engineering

Structure and Contents for S. Y. B. Tech. Computer Science and Engineering Program

(AY 2020-21) R0

AME/P/80/00

CST 217: Environmental Studies (Audit Course)
S. Y. (B. Tech) (Semester III/IV)

Course Code	Course Title	L	T	Pr	C	Evaluation Scheme				
						Component	Exam	WT	Min Passing	
(UC|SS) Version: 10	Environmental Studies	3	-	-	3	Th (100)	FET	20	40	40
							CAT	30		
							ESE	50	40	
(UC|SS) Version: 1.0	Environmental Studies Project	-	-	2	1	Pr (100)	FEP	100	40	40

Contents

Units	Description	Hours
	a) Introduction to environmental studies:	
I	Multidisciplinary nature of environmental studies; Scope and importance; Concept of sustainability and sustainable development.	1
	b) Ecosystem:	
	Concept of ecosystem, Structure and function of ecosystem; Energy flow in an ecosystem.	
	Food chains, food webs and ecological succession.	2
	Structure and function of the following ecosystems: Examples	
	a) Natural Resources: Renewable and Non- Renewable Resources	
II	Land resources and land use change; Land degradation, soil erosion and desertification.	2
	Deforestation: Causes and impacts due to mining, dam building on environment and forests	
	Water: Use and over-exploitation of surface and ground water, floods, droughts	
	Energy resources: Renewable and non renewable energy sources, use of alternate energy sources, growing energy needs, case studies	
	a) Biodiversity and Conservation	
	Levels of biological diversity: genetic, species and ecosystem diversity;	3
	Global biodiversity hot spots. India as a mega-biodiversity nation; Endangered and endemic species of India	
	Threats to biodiversity: Habitat loss, poaching of wildlife, man-wildlife conflicts, biological invasions.	
	Conservation of biodiversity In-situ and Ex-situ conservation of biodiversity.	
	Ecosystem and biodiversity services: Ecological, economic, social, ethical, Aesthetic and Informational value.	

Sanjay Ghodawat University Kolhapur

Structure and Contents for Second Year B. Tech. Computer Science and Engineering Program R0

III **a) Environmental Pollution** 3

- Environmental pollution: types, causes, effects and controls; Air, water, Noise pollution
- Nuclear hazards and human health risks
- Solid waste management: Control measures of urban and industrial waste.

b) Environmental policies and practices 3

- Global issues: Climate change, global warming, ozone layer depletion, acid rain and impacts on human communities and agriculture.
- Environment Laws: Environment Protection Act; Air (Prevention & Control of Pollution) Act; Water (Prevention and control of Pollution) Act; Wildlife Protection Act

IV **a) Human Communities and the Environment** 3

- Human population growth: Impacts on environment, human health and welfare
- Disaster management: floods, earthquake, cyclones and landslides.
- Environmental movements: Chipko, Silent valley, Bishnois of Rajasthan.
- Environmental communication and public awareness, case studies (e.g., CNG vehicles in Delhi).

Environmental Studies Project (Field Work)

(Ver 1.0, Program Core, School of Sciences)

Lect.	Tut.	Practical.	Credits	Evaluation Scheme			
				Component	Exam	WT	Pass
-	-	2	0	Practical (100)	PET	100	Min 40

Field Work

Description

- Visit to an area to document environmental assets: river/ forest/ flora/fauna, etc.
- Visit to a local polluted Site-Urban/Rural/Industrial/Agricultural.
- Study of common plants, insects, birds and basic principles of identification.
- Study of simple ecosystems-pond, river, etc.

References

1. Carson, R. 2002. Silent Spring. Houghton Mifflin Harcourt.
2. Gadgil, M., & Guha, R. 1993. This Fissured Land: An Ecological History of India. Univ. of California Press.
3. Gleeson, B. and Low, N. (eds.) 1999. Global Ethics and Environment, London, Routledge.
4. Gleick, P. H. 1993. Water in Crisis. Pacific Institute for Studies in Dev., Environment & Security. Stockholm Env. Institute, Oxford Univ. Press.
5. Groom, Martha J., Gary K. Meffe, and Carl Ronald Carroll. Principles of Conservation Biology. Sunderland: Sinauer Associates, 2006.
6. Grumbine, R. Edward, and Pandit, M.K. 2013. Threats from India's Himalaya dams. Science, 339: 36-37.
7. McCully, P. 1996. Rivers no more: the environmental effects of dams (pp. 29-64). Zed Books.
8. McNeill, John R. 2000. Something New Under the Sun: An Environmental History of the Twentieth Century.
9. Deeksha Dave, S.S. Katewa, Textbook of Environmental Studies.
10. B.K. Sharma, Environmental Chemistry.
11. Bharucha Erach, The Biodiversity of India, Mapin Publishing pvt. Ltd., Ahmedabad 380013, India, Email:mapin@icenet.net (R)
12. De A.K., Environmental Chemistry, Wiley Wastern Ltd.
13. Trivedi R.K. Handbook of Environmental Laws, Rules, Guidelines, Compliances and Standards, vol. I anfd II, Environmental Media (R)

CST202: Theory of Computation

(Ver 1.0, Program Core, School of Technology)

Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	WT	Min. Pass	
3	1	-	4	Theory 100	FET	20	40	40
					CAT I	15		
					CAT II	15		
					ESE	50	40	40

Course Description: This course is at even semester of second year B Tech computer science and engineering. It is a one of the core course in computer science and may be pre requisites for other courses in computer science. It covers basics of regular languages, finite automata, grammar, languages and concepts Turing machines.

Prerequisite: Discrete structures and basics of mathematics
Co requisite: None

Course Outcomes: After the end of this course students will able to

CO1 **Construct**[5] regular expressions and regular grammars that produce regular languages.
CO2 **Construct**[5] PDA, CFG for regular expression.
CO3 **Design**[5] specify and test deterministic and non deterministic finite automata that recognize regular languages.
CO4 **Illustrate**[3] the concepts of Turing machines and grammars.

Syllabus (Theory)

Units	Description	Hrs
I	**Mathematical Induction, Regular Languages & Finite Automata:** The Principle of Mathematical Induction Recursive Definitions, Definition & types of grammars & languages, Regular expressions and corresponding regular languages, examples and applications, unions, intersection & complements of regular languages, Finite automata-definition and representation, Non-deterministic F.A.,NFA with null transitions, Equivalence of FA's , NFA's and NFA's with null transitions.	07
II	**Kleene's Theorem:** Part I and II statements and proofs, minimum state of FA for a regular language, minimizing number of states in Finite Automata.	03
III	**Grammars and Languages:** Derivation and ambiguity, BNF and CNF notations, Union, Concatenation and *'s of CFLs, Eliminating production and unit productions from CFG, Eliminating useless variables from a context Free Grammar. **Parsing:** Top-Down, Recursive Descent and Bottom-Up Parsing	9
IV	**Push Down Automata:** Definition, Deterministic PDA and types of acceptance, Equivalence of CFG's	05

and PDAs.

V	**CFL's and non CFL's:** Pumping Lemma and examples, intersections and complements.	03
VI	**Turing Machines:** Models of computation, definition of Turing Machine as Language acceptors, combining Turing Machines, Computing a function with a TM **Variations in Turing Machines :** Turing machines with doubly-infinite tapes, more than one tape, Non-deterministic TM and Universal TM.	09

Tutorial

One hour per week per batch tutorial is to be utilized to ensure that students have properly learnt the topics covered in the lectures. It should comprise of minimum 8-10 tutorials. Students of different batches should perform different tutorials based on the following guidelines-

1. Finite automata, NFA, DNFA
2. Kleene's Theorem,
3. Minimum state of FA for a regular language, minimizing number of states in FA
4. BNF and CNF notations
5. Eliminating production and unit productions from CFG
6. Top-Down, Recursive Descent and Bottom-Up Parsing
7. PDA and types of acceptance
8. Pumping Lemma and examples, CFL's and non CFL's
9. Computing a function with a TM
10. Variations in Turing Machines

Text Book

1. Introduction to Languages & theory of computations—John C. Martin (MGH).

References

1. Discrete Mathematical Structures with applications to Computer Science—J .P.Trembley & R.Manohar (MGH)
2. Hopcroft J.E., Motwani R. and Ullman J.D, "Introduction to Automata Theory, Languages and Computations", Second Edition, Pearson Education, 2008.
3. John.C.Martin, "Introduction to Languages and the Theory of Computation" McGraw-Hill Education, 01-May-2010.
4. Michael Sipser, "Introduction to the Theory of Computation" Cengage Learning, 2012.
5. Peter Linz, "An introduction to formal languages and automata", Jones & Bartlett Learning, 2001.

CST204: Computer Programming Laboratory - I

(Ver 1.0, Engineering Science, School of Technology)

Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	WT	Min. Pass	
2	-	4	4	Practical 100	FEP	50	40	40
					POE	50	40	

Course Description: This course is at even semester of second year B Tech computer science and engineering. It is a foundation course in programming and may be pre requisites for other courses. It covers basics of java programming with object oriented concepts and their application.

Prerequisite: Object Oriented programming concepts.
Co requisite: None
Course Outcomes: After the end of this course students will able to-

CO1 **Explain**[2] Fundamental and object oriented concepts of Java.
CO2 **Analyze**[4] Application of Interface, inheritance and packaging in Java.
CO3 **Express**[2] Writing code with Exception handing and I/O programming features.
CO4 **Demonstrate**[3] Architecture and components of GUI development in Java.
CO5 **Demonstrate**[3] Fundamental concept of multithreading and Network Programming in Java.
CO6 **Demonstrate**[3] Collection and database programming in Java.

Syllabus (Theory)

Units	Description	Hrs
I	**Fundamental Programming in Java:** The Java Buzzwords, The Java Programming Environment- JVM, JIT Compiler, Byte Code Concept, HotSpot, A Simple Java Program, Source File Declaration Rules, Comments, Data Types, Variables, Operators, Strings, Input and Output, Control Flow, Big Numbers, Arrays- Jagged Array. **Objects and Classes:** Object-Oriented Programming Concepts, Declaring Classes, Declaring Member Variables, Defining Methods, Constructor, Passing Information to a Method or a Constructor, Creating and using objects, Controlling Access to Class Members, Static Fields and Methods, this keyword, Object Cloning, Class Design Hints.	04
II	**Interface, Inheritance and Packaging:** Interfaces: Defining an Interface, Implementing an Interface, Using an Interface as a Type, Evolving Interfaces and Default Methods. **Inheritance:** Definition, Superclasses, and Subclasses, Overriding and Hiding Methods, Polymorphism, Inheritance Hierarchies, Super keyword, Final Classes and Methods, Abstract Classes and Methods, casting, Design	05

Hints for Inheritance, Nested classes & Inner Classes, finalization and garbage collection.
Packages: Class importing, Creating a Package, Naming a Package, Using Package Members, Managing Source and Class Files. Developing and deploying (executable) Jar File.

III	**Exception and I/O Streams:** Exception: Definition, Dealing with Errors, The Classification of Exceptions, Declaring Checked Exceptions, Throw an Exception, Creating Exception Classes, Catching Exceptions, Catching Multiple Exceptions, Re-throwing and Chaining Exceptions, finally clause, Advantages of Exceptions, Tips for Using Exceptions. **I/O Streams:** Byte Stream – InputStream, OutputStream, DataInputStream, DataOutputStream, FileInputStream, FileOutputStream, Character Streams, BufferedStream, Scanner, File, RandomAccesFile.	05
IV	**Graphical User Interfaces using Swing:** Introduction to the Swing, Swing features, Swing Top Level Containers-Creating a Frame, Positioning a Frame, Displaying Information in a Panel, The Model-View-Controller Design Pattern, The JComponent Class. **Layout Management:** Introduction to Layout Management, APIs for Border Layout, Flow Layout, Grid Layout **Event Handling:** Basics of Event Handling, The AWT Event Hierarchy, Semantic and Low- Level Events in the AWT, Low-Level Event Types **User Interface Components:** Text Input, Choice Components, Menus, Dialog Boxes. Setting the Look and Feel of UI, Introduction to JApplet	04
V	**Networking and Multithreading:** **Networking:** Overview of Networking, Networking Basics, Working with URLs, Creating a URL, Parsing a URL, Reading Directly from a URL, Connecting to a URL, Reading from and Writing to a URL Connection, Sockets, Reading from and Writing to a Socket, Writing the Server Side of a Socket, Datagrams, Writing a Datagram Client and Server. **Multithreading:** Processes and Threads, Runnable Interface and Thread Class , Thread Objects, Defining and Starting a Thread, Pausing Execution with Sleep, Interrupts, Thread States, Thread Properties, Joins, Synchronization	04
VI	**Collection and Database Programming:** **Collections:** Collection Interfaces, Concrete Collections- List, Queue, Set, Map, the Collections Framework. **Database Programming:** The Design of JDBC, The Structured Query Language, JDBC Installation, Basic JDBC Programming Concepts, Query Execution, Scrollable and Updatable Result Sets, Metadata, Row Sets, Transactions.	02

Practical

Four hours per week per batch practical is to be utilized for writing programs to ensure that students have properly learnt the topics covered in the lectures. This shall include extra problem statements and there implementations to strengthen the programming logic.

It should comprise of minimum 10-12 experiments. Students of different batches should implement different programs based on the following guidelines-

1. Implement a program based on constructor.
2. Implement a program using static variable and static method.
3. Implement a program based on interface.
4. Implement a program based on inheritance.
5. Implement a program based on package.
6. Implement a program to solve arithmetic equations with exception handling.
7. Implement a program to perform string conversion with exception handling.
8. Implement a simple file handling program.
9. Implement a program using command line arguments.
10. Implement a program using Swing.
11. Implement a program based on event handling.
12. Implement a program based on multithreading.
13. Implement a program using Database handling.

Text Book

1. Core Java- Volume I Fundamentals: Cay Horstmann and Gary Cornell, Pearson, Eight edition.
2. Core Java- Volume II Advanced Features: Cay Horstmann and Gary Cornell, Pearson, Eight edition.

References

1. JAVA-The Complete Reference: Herbert Schildt, Oracle Press, Mcgraw Hill, Ninth edition.
2. JAVA™ HOW TO PROGRAM, By Deitel Paul , Deitel Harvey.10th Edition, Publisher: PHI Learning.
3. Thinking in Java by Bruce Eckel, Prentice Hall, 4th Edition.
4. A Programmer's guide to JAVA SCJP Certification: Khaleed Mughal and Rolf W. Rasmussen, Addison Wesley, Third edition.

CST206: Database Management System

(Ver 1.0, Program Core, School of Technology)

Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	WT	Min. Pass	
3	-	-	3	Theory 100	FET	20	40	40
					CAT I	15		
					CAT II	15		
					ESE	50	40	40

Course Description: This course is at even semester of second year B Tech computer science and engineering. It is a one of the core course in computer science and may be pre requisites for other courses. It covers basics of database management systems, SQL, database transactions and their application.

Prerequisite: OO Programming
Co requisite: None

Course Outcomes: After the end of this course students will able to

CO1 **Explain**[2] Fundamental Concepts and algorithms related to database.
CO2 **Summarize**[2] familiarity with SQL & DBMS.
CO3 **Apply**[3] basic concepts of Database Design.
CO4 **Illustrate**[3] to organize data in specific order to improve performance.
CO5 **Develop**[3] database transaction, concurrency control and their properties.
CO6 **Discuss**[2] to maintain the integrity of data.

Syllabus (Theory)

Units	Description	Hrs
I	**Introduction:** Purpose of Database System – Views of data – Data Models – Database Languages — Database System Architecture – Database users and Administrator – Entity– Relationship model (E-R model) – E-R Diagrams -- Introduction to relational databases.	06
II	**Relational Model:** The relational Model – The catalog- Types– Keys - Relational Algebra – Domain Relational Calculus – Tuple Relational Calculus - Fundamental operations – Additional Operations- SQL fundamentals - Integrity – Triggers – Security.	06
III	**Database Design:** Functional Dependencies – Non-loss Decomposition – Functional Dependencies –	07

First, Second, Third Normal Forms, Dependency Preservation – Boyce Normal Form Multi-valued Dependencies and Fourth Normal Form – Join Dependencies and Fifth Normal Form.

IV **Transaction Management & Concurrency Control:** 06

Transaction- Concurrency Control- Concurrency Control with Locking Methods- Concurrency Control with Times tamping Methods- Concurrency Control with Optimistic Methods.

V **Data Storage & Indexing:** 06

File Organization- Organization of records in File- Data Dictionary Storage- Database Buffer- Basic Concepts indexing & hashing- Ordered Indices-Multiple-Key Access- Static Hashing- Dynamic Hashing- Bitmap Indices- Index Definition in SQL.

VI **Recovery System:** 05

Failure Classification- Storage-Recovery & atomicity-Recovery Algorithm- Buffer Management-Failure with loss of non- volatile Storage.

Text Book

1. Database Systems- A practical approach to Design, Implementation and Management by Thomos Connolly, Carolyn Begg, 3rd Edition, Pearson Education.
2. Database System Concepts by A. Silberschatz, H.F. Korth, S. Sudarshan, 6th edition, Mc Graw Hill Education.

References

1. C. J. Date, A.Kannan, S.Swamynathan, “An Introduction to Database Systems”, Eighth Edition, Pearson Education, 2006.

CST208: Database Management Systems Laboratory

(Ver 1.0, Program Core, School of Technology)

Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	WT	Min. Pass	
-	-	2	1	Practical 100	FEP	50	40	40
					POE	50	40	

Course Description: This course is at even semester of second year B Tech computer science and engineering. It is a one of the core course in computer science and may be pre requisites for other courses. It covers study and implementation of database management systems, SQL, database transactions and their application.

Prerequisite: OO Programming
Co requisite: None

Course Outcomes: After the end of this course students will able to

CO1 **Operate**3 SQL queries and DBMS.
CO2 **Illustrate**3 to organize data in specific order to improve performance.
CO3 **Develop**5 database transaction, concurrency control and their properties.
CO4 **Experiment**4 to maintain the integrity of data.

Practical

Two hours per week per batch practical is to be utilized for writing programs to ensure that students have properly learnt the topics covered in the lectures. This shall include extra problem statements and there implementations to strengthen the programming logic.
It should comprise of minimum 10-12 experiments. Students of different batches should implement different programs based on the following guidelines-

1. Program to implement Data Definition, Table Creation and Constraints.
2. Program to implement Insert, Select, Update & Delete Commands
3. Program to implement Nested Queries & Join Queries.
4. Program to implement Views.
5. Program to implement High level programming language extensions (Control structures, Procedures and Functions).
6. Program to implement Front end tools.
7. Program to implement Forms.
8. Program to implement Triggers.
9. Program to implement Menu Design.
10. Program to implement Reports.
11. Program to implement Database Design and implementation (Mini Project).

Text Book

1. The Complete Reference: C++ - Herbert Schildt (TMGH) Fourth Edition.
2. Database Systems- A practical approach to Design, Implementation and Management by Thomos Connolly, Carolyn Begg, 3rd Edition, Pearson Education.
3. Database System Concepts by A. Silberschatz, H.F. Korth, S. Sudarshan, 6th edition, Mc Graw Hill Education.

References

1. C. J. Date, A.Kannan, S.Swamynathan, “An Introduction to Database Systems”, Eighth Edition, Pearson Education, 2006.

Sanjay Ghodawat University, Kolhapur

School of Technology

Department of Computer Science and Engineering

Structure and Contents for S. Y. B. Tech. Computer Science and Engineering Program

(AY 2020-21) R0

AME/P/80/00

CST210: Operating Systems

(Ver 1.0, Program Core, School of Technology)

Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	WT	Min. Pass	
3	-	-	3	Theory 100	FET	20	40	40
					CAT I	15		
					CAT II	15		
					ESE	50	40	40

Course Description: This course is at even semester of second year B Tech computer science and engineering. It is a core course in computer science and may be pre requisites for other courses. It covers basics of operating systems, Linux basics, Linux networking and Linux security.

Prerequisite: Computer fundamentals
Co requisite: None

Course Outcomes: After the end of this course students will able to-

CO1 **Identify**[1] different types of operating system.
CO2 **Explain**[2] process management done by operating system.
CO3 **Explain**[2] scheduling of multiple processes by operating system.
CO4 **Discuss**[2] process synchronization.
CO5 **Define**[1] key terms of deadlock and its prevention.
CO6 **Discuss**[2] the memory management done by operating systems.

Syllabus (Theory)

Units	Description	Hrs
I	**Operating Systems Overview** What is an Operating system?, services given by operating system, types of operating system, Batch Processing System, Multiprogramming System, The Time Sharing System, The Real Time Operating System, Distributed operating system	05
II	**Process management and Scheduling**: process overview, process scheduling, operations on processes, inter-process communication, Scheduling, scheduling criteria ,scheduling algorithms	06
III	**Process synchronization:** Critical section problem, semaphores and its implementation, classical problems of synchronization, monitors**.**	05
IV	**Linux Basics:** Introduction to Linux, Linux File System, General usage of Linux kernel & basic commands, Linux users and group, Permissions ior file directory and' users, Searching a file & directory, zipping and unzipping concepts.	06
V	**Linux Networking**: Introduction to Networking in Linux, Network basics &	07

tools, File transfer protocol in Linux, Network File system, Domain Naming Services (DNS).

VI **Linux Security:** Configure and troubleshoot lpv4 and lPv6 address in Linux, 07 Linux Network Namespaces, Dynamic Hosting configuration protocol (DHCP) basics and configuration, Security tasks and Auditing source code, Securing SSH, port scanning with suitable tools.

Text Book

1. Operating System concepts – 5th Edition –by Silberschatz Galvin, John Wiley.
2. Operating Systems a concept based approach by Dhananjay M Dhamdhere, TMGH.
3. The Complete Reference Linux, Sixth Edition, Richard Peterson, TMGH.

References

1. Harvey M. Deitel, “Operating Systems”, Second Edition, Pearson Education Pvt. Ltd, 2002.
2. Andrew S. Tanenbaum, “Modern Operating Systems”, Prentice Hall of India Pvt. Ltd, 2003.
3. William Stallings, “Operating System”, Prentice Hall of India, 4th Edition, 2003.
4. Linux Fundamentals, Paul Cobbaut, CEST, 2015.

CST212: Operating Systems Laboratory

(Ver 1.0, Program Core, School of Technology)

Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	WT	Min. Pass	
-	-	2	1	Practical 100	FEP	100	40	40

Course Description: This course is at even semester of second year B Tech computer science and engineering. It is a core course in computer science and may be pre requisites for other courses. It covers the study and implementation of basics concepts of operating systems, Linux, Linux networking and Linux security.

Prerequisite: Computer fundamentals, Programming Basics
Co requisite: None

Course Outcomes: After the end of this course students will able to-

CO1 **Experiment**$_4$ process management done by operating system.
CO2 **Analyze**$_2$ scheduling of multiple processes by operating system.
CO3 **Practice**$_3$ process synchronization.
CO4 **Demonstrate**$_3$ key terms of deadlock and its prevention.
CO5 **Experiment**$_4$ the memory management done by operating systems.

Practical

Two hours per week per batch practical is to be utilized for writing programs to ensure that students have properly learnt the topics covered in the lectures. This shall include extra problem statements and there implementations to strengthen the programming logic.

It should comprise of minimum 10-12 experiments. Students of different batches should implement different programs based on the following guidelines-

1. Linux commands (basic, file & directory, zip & unzip, networking)
2. Process control system calls (fork (), execv () and wait () system call)
3. Thread management (using pthread library)
4. CPU Scheduling algorithms (FCFS/SJF/ RR/ Priority/Multilevel queue)
5. Interprocess communication (using pipes, FIFO, signal)
6. Process synchronization (Producer-consumer problem using semaphore)
7. Deadlock management technique (Banker's algorithm for deadlock avoidance)
8. File allocation strategies (sequential/indexed/linked)
9. Simulate file organization technique(single level/two level/hierarchical)
10. Memory management algorithm (worst-fit, best fit, first-fit)

Text Book

1. Operating System concepts – 5th Edition –by Silberschatz Galvin, John Wiley.
2. Operating Systems a concept based approach by Dhananjay M Dhamdhere, TMGH.
3. The Complete Reference Linux, Sixth Edition, Richard Peterson, TMGH.

References

1. Harvey M. Deitel, “Operating Systems”, Second Edition, Pearson Education Pvt. Ltd, 2002.
2. Andrew S. Tanenbaum, “Modern Operating Systems”, Prentice Hall of India Pvt. Ltd, 2003.
3. William Stallings, “Operating System”, Prentice Hall of India, 4th Edition, 2003.
4. Linux Fundamentals, Paul Cobbaut, CEST, 2015.

CST214: Computer Networks

(Ver 1.0, Program Core, School of Technology)

Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	WT	Min. Pass	
3	-	-	3	Theory 100	FET	20	40	40
					CAT I	15		
					CAT II	15		
					ESE	50	40	40

Course Description: This course is at even semester of second year B Tech computer science and engineering. It is a basic core course in computer science and may be pre requisites for other courses. It covers basics of data communication, networking, network modeling layers and protocols.

Prerequisite: Basics of data communication and computer programming
Co requisite: None
Course Outcomes: After the end of this course students will able to

CO1 **Explain**[2] of the basic networking terms.
CO2 **Analyze**[4] different computer network modeling layers and addressing schemes.
CO3 **Apply**[3] networking algorithm to find the shortest path in network/ subnet.
CO4 **Demonstrate**[3] performance of TCP & UDP.
CO5 **Examine**[4] different application layer protocols.

Syllabus (Theory)

Units	Description	Hrs
I	**Data Communications** Data Transmission, fiber optics backbones and components, multiplexing.	04
II	**Data Link Layer** Channel access on links, SDMA, TDMA,FDMA ,CDMA, Hybrid Multiple Access, Techniques, Issues in the Data Link Layer ,Framing , Error correction and detection ,Link Level Flow Control ,Medium Access ,Ethernet ,Token Ring , FDDI ,Wireless LAN, Bridges and Switches.	07
III	**Network Layer** Circuit Switching, Packet Switching Virtual Circuit Switching, IP, ARP, DHCP, ICMP, Routing, RIP, OSPF, Sub netting, CIDR, Inter domain Routing – BGP.	06
IV	**Logical Addressing** IPV4 Addressing & IPV6, Basic Features, Inter Domain Multicast, Congestion Avoidance in Network Layer.	07
V	**Transport Layer** User Datagram Protocol (UDP), Transmission Control Protocol (TCP), Flow	06

Control, Congestion Control, Queuing, Discipline Introduction to Quality of services (QOS). Reduction Algorithms, Ensemble Algorithms

VI **Application Layer** 06

Network Architecture, Layers, HTTP, DNS, E-Mail (SMTP, MIME, POP3, IMAP, Web (WWW), FTP, Telnet, SNMP.

Text Book

1. William Stallings, "Data and Computer Communications", Eighth Edition, Pearson Education, 2011.
2. James F. Kurose, Keith W. Ross, "Computer Networking, A Top, Down Approach Featuring the Internet", Third Edition, Pearson Education, 2006.

References

1. Larry L. Peterson, Bruce S. Davie, "Computer Networks: A Systems Approach", Fifth Edition, Morgan Kaufmann Publishers Inc., 2011.
2. Nader F. Mir, "Computer and Communication Networks", First Edition, Pearson Education, 2007.
3. Ying,Dar Lin, Ren,Hung Hwang and Fred Baker, Computer Networks: An Open Source Approach ", McGraw Hill Publisher, 2011.
4. Behrouz A. Forouzan, "Data communication and Networking", Tata McGraw, Hill, 2004.

CST216: Computer Networks Laboratory

(Ver 1.0, Program Core, School of Technology)

Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	WT	Min. Pass	
-	-	2	1	Practical 100	FEP	100	40	40

Course Description: This course is at even semester of second year B Tech computer science and engineering. It is a basic core course in computer science and may be pre requisites for other courses. It covers study and implementation of basics of data communication, networking, network modeling layers and protocols.

Prerequisite: Basics of data communication and computer programming
Co requisite: None
Course Outcomes: After the end of this course students will able to

CO1 **Analyze**$_4$ different computer network modeling layers and addressing schemes.
CO2 **Apply**$_3$ networking algorithm to find the shortest path in network/ subnet.
CO3 **Demonstrate**$_5$ performance of TCP & UDP.
CO4 **Examine**$_4$ different application layer protocols.

Practical

Two hours per week per batch practical is to be utilized for writing programs to ensure that students have properly learnt the topics covered in the lectures. This shall include extra problem statements and there implementations to strengthen the programming logic.

It should comprise of minimum 10-12 experiments. Students of different batches should implement different programs based on the following guidelines-

1. Implementation of file transmission using RS-232.
2. Implementation of file transmission using Stop and Wait / Go Back n / Selective Repeat protocol.
3. Implementation of Hamming code / CRC for error detection / recovery.
4. Developing a file transfer application using TCP (socket program).
5. Developing a file transfer application using and UDP (socket program).
6. Develop applications to demonstrate Congestion control algorithms.
7. Implementation of Cryptographic algorithms.
8. Develop a network application to identify host id and network id of a remote machine in an IPv4 network.
9. Develop a simple email application.
10. Study of DNS, Remote login.
11. Use of dig, ftp, SSH, etc.

Text Book

1. William Stallings, “Data and Computer Communications”, Eighth Edition, Pearson Education, 2011.
2. James F. Kurose, Keith W. Ross, “Computer Networking, A Top, Down Approach Featuring the Internet”, Third Edition, Pearson Education, 2006.

References

1. Larry L. Peterson, Bruce S. Davie, “Computer Networks: A Systems Approach”, Fifth Edition, Morgan Kaufmann Publishers Inc., 2011.
2. Nader F. Mir, “Computer and Communication Networks”, First Edition, Pearson Education, 2007.
3. Ying,Dar Lin, Ren,Hung Hwang and Fred Baker, Computer Networks: An Open Source Approach “, McGraw Hill Publisher, 2011.
4. Behrouz A. Forouzan, “Data communication and Networking”, Tata McGraw, Hill, 2004.

CST218: Mini Project-I

(Ver 1.0, Program Core, School of Technology)

Lect.	Tut.	Pract.	Credits	Evaluation Scheme				
				Component	Exam	WT	Min. Pass	
-	-	4	2	Project 100	FEP	50	40	40
					POE	50	40	

Course Description: This course is at even semester of second year B Tech computer science and engineering. It is a project work and may be pre requisites for other project work.

Prerequisite: Computer programming fundamentals, software engineering
Co requisite: None

Course Outcomes: After the end of this course students will able to

CO1 **Develop**$_5$ the students to use the engineering approach to solve the real time problems.
CO2 **Use**$_3$ of the skills of team building and team work
CO3 **Develop**$_5$ the logical skills and use of appropriate data structures for solving the engineering problems and puzzles.

Syllabus (Practical)

Description	Hrs
The mini project should be undertaken preferably by a group of 3-4 students who will jointly work and implement the project. The mini project must be based upon the problem statements as that of programming contest (Advanced Computing Machines – Inter-Collegiate Programming Contest: ACM-ICPC). The problems can be referred from the web links concerned with ACM-ICPC. The group will select a problem with the approval of the guide and prepare the solution guidelines for its implementation. The same should be put in the form of synopsis (3 to 5 pages), stating the usage of logic, algorithms and suitable data structures necessary for implementation of the solution. Further the group is expected to complete analysis of problem by examining the possible different inputs to the system and the corresponding outputs. The term work submission is to be done in the form of a report containing the details of the problem, solution techniques, implementation details, input-output scenarios and the conclusion. The project must be implemented in C/C++/Java. Graphics is optional for GUI.	26

Practical

Two hours per week per batch practical is to be utilized for project work. Students of different batches should follow following sequence of guidelines-

1. Project topic and title finalization.
2. Submission of proposal for project work (Synopsis).
3. First presentation includes a) Requirements analysis b) Architecture c) Data design d) Algorithm design e) Module identification f) Class properties d) Method identification

(if applicable) g) Level 0 & Level 1 DFD h) Object oriented analysis (UML diagrams).

4. Second presentation.
5. End Semester Review in 3rd Presentation (after 100 % implementation of all modules).
6. Project report preparation.

Text Book

1. Software Engineering : A precise Approach - Pankaj Jalote (Wiley India)
2. Fundamentals of Software Engineering - Rapit Mall (3rd Edition)(PHI)
3. Let Us C- Yashvant Kanetkar (BPB Publications)
4. Object-Oriented Programming in C++ - Rajesh K. Shukla (Wiley) India Edition
5. Core Java- Volume I Fundamentals: Cay Horstmann and Gary Cornell, Pearson, Eight Edition
6. Core Java- Volume II Advanced Features: Cay Horstmann and Gary Cornell, Pearson, Eight edition

Reference

1. C The Complete Reference – Herbert Schildt (Tata McGraw-Hill Edition)
2. The Complete Reference: C++ - Herbert Schildt (TMGH) Fourth Edition
3. JAVA-The Complete Reference: Herbert Schildt, Oracle Press, Mcgraw Hill, Ninth edition.

CST 220: Professional Development Skill - II

(Ver 1.0, Sem-IV of B. Tech (Common for All Branches))

Course Description: This course is the extension of the Professional Development – I course of third semester. The course aims to develop leadership skills and sharpen their decision making skills. The major focus of the course is to prepare students for job.

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

CO01 : demonstrate[3] leadership skills
CO02 : evaluate[6] process and practical ways of decision making
CO03 : judge[6] causes of stress and find remedies to reduce stress
CO04 : apply[3] business etiquettes and ethics
CO05 : exhibit[3] group discussion and Interview skills

Syllabus

Units	Description	Hrs.
I	**Leadership:** Skills for a good Leader, Assessment of Leadership Skills	4
	Creativity: Lateral thinking, vertical thinking, Out of box thinking	
II	**Decision Making:** Importance and necessity of Decision Making, Process and practical way of Decision Making, Weighing Positives & Negatives.	4
III	**Stress Management**: Causes of Stress and its impact, how to manage & distress, Circle of control, Stress Busters.	4
	Emotional Intelligence: What is Emotional Intelligence, dealing with feelings, emotional quotient, Why Emotional Intelligence matters, Emotion Scales? Managing Emotions.	
IV	**Adapting to corporate life**: Corporate Grooming and dressing, Business Etiquette Business Ethics, Dinning Etiquette, Ethics policy	4
V	**Group Discussion:** Group discussions as part of selection process. Structure of a group discussion, Dynamics of group behavior, techniques for effective participation, Team work and use of body language.	4
	Interview: Process, techniques, Pre-In-After the interview preparation.	

References:

1. Wallace & Masters, Personal development for Life & work, Thomson Learning.
2. Barun K. Mitra, Personality Development and Soft- Skills, Oxford University Press.
3. Fred Luthans, Organizational behavior, McGraw Hill.
4. Asa Don Brown, Interpersonal skills in the Workplace, Tate publishing and Enterprises.