Autonomous Programme Structure of
Second Year B. Tech. Information Technology
Academic Year : 2019-2020

S. Y. B. Tech. Information Technology Semester – I										
Course Code	Course Title	Teaching Scheme Hours /Week			Examination Scheme				Marks	Credit
		Lecture	Tutorial	Practical	In Semester	End Semester	Oral	Practical		
IT 2101	Discrete Structures	3	1	0	50	50	0	0	100	4
IT 2102	Digital Systems	3	1	0	50	50	0	0	100	4
IT 2103	Data Structures I	3	0	0	50	50	0	0	100	3
IT 2104	Network Fundamentals	3	1	0	50	50	0	0	100	4
BSH 2101	Principles of Economics and Finance	3	0	0	50	50	0	0	100	3
IT 2105	Digital Systems Laboratory	0	0	2	0	0	0	25	25	1
IT 2106	Data Structures I Laboratory	0	0	4	0	0	0	50	50	2
IT 2107	Web Engineering Technology Laboratory	0	0	2	0	0	0	25	25	1
AC 2101	Self Expression	0	0	2	0	0	0	0	0	No Credit
	Total	**15**	**3**	**10**	**250**	**250**	**0**	**100**	**600**	**22**
	Grand Total	**28**			**600**				**600**	**22**

AC 2101 -- Audit Course: Self Expression

1. Dance
2. Drawing / Painting / Sketching
3. English Communication Skill
4. Film Appreciation
5. Origami
6. Theatre

MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

DEAN ACADEMICS
MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

Principal
MKSSS's Cummins College of Engg. For Women, Karvenagar, Pune-52.

APPROVED BY
Governing Body Members
MKSSS's Cummins College of Engineering for Women
Karvenagar, Pune-411052

IT 2101 Discrete Structures

Teaching Scheme:

Lectures: **3** Hrs/Week

Tutorial: **1** Hr/Week

Examination Scheme:

In-Semester: **50** Marks

End-Semester: **50** Marks

Credits: 4

Course Objectives:

1. Learn the concepts of propositions and propositional logic
2. Learn the concepts of sets operations and functions
3. Learn the fundamentals of counting, permutations and combinations
4. Learn the relations, its representations and properties
5. Learn the concepts of graph, its terminology, representation, connectivity, and its Applications.
6. Learn the concepts of tree, tree traversals and applications

Course Outcomes:

By the end of the course, students should be able to

1. Solve real world problem using sets and functions
2. Use proposition and propositional logic for drawing conclusions
3. Demonstrate the application of discrete structures using relations
4. Apply graphs as models to variety of domains
5. Apply trees in simple applications of computation
6. Evaluate the combinatorial problems

Unit – I: Sets and Functions **(07)**

Sets: Introduction to Power set, Cartesian products ; Set Operations: Introduction, Generalized union and intersection, Computer representation of sets; Functions: Introduction, One-to-One and Onto Functions, Inverse function and Composition of Functions

Unit – II: Propositional Logic **(06)**

Propositional Logic: Introduction, Proposition, Conditional Statements, Truth tables of compound proposition; Propositional equivalences: Introduction, Logical Equivalences, Constructing new logical equivalences; Preliminaries of predicates and quantifiers: Introduction, Predicates, Quantifiers, Negating quantified expressions

Unit – III: Relations **(08)**

Relations and Their Properties: Introduction, functions as relation, relations on set, Properties of relations, combining relations; n-ary Relations and Their Applications: Introduction, n-ary relations, operations on n-ary relations; Representing Relations:

Representing relations using matrices, Representing relations using digraph; Closures of Relations: Introduction, Closures, paths in directed graph, transitive closure, Warshall's algorithm; Equivalence Relations: Introduction, Equivalence relation, Equivalence classes and partition; Partial Orderings: Introduction, Hasse Diagrams, Maximal and Minimal elements, Lattices, discrete numeric functions

Unit – IV: Graphs **(06)**

Graphs and Graph Models , Graph Terminology and Special Types of graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths , Shortest-Path Problems, Planar Graphs, Graph Coloring

Unit – V: Trees **(06)**

Introduction to Trees, Applications of Trees: Introduction, Binary search trees, Prefix codes, Tree Traversal: Preorder, in-order and post-order traversals , Minimum Spanning Trees: Introduction, Prim's algorithm, Kruskal's algorithm

Unit – VI: Counting **(07)**

The Basics of Counting: Introduction, Basic counting principles, Inclusion exclusion principle; The Pigeonhole Principle: Introduction, Generalized pigeonhole principle; Permutations and Combinations: Introduction, permutations, combinations; Binomial

Coefficients and Identities; Generalized Permutations and Combinations: permutation with repetition, combination with repetition; Generating Permutations and Combinations: Generating permutations, generating combinations

Text Books:

1. Kenneth H. Rosen, **"Discrete Mathematics and Its Applications",** Tata McGraw-Hill (7th Edition) (2012)

Reference Books:

1. C. L. Liu, "**Elements of Discrete Mathematics**", Tata McGraw-Hill (2^{nd} Edition)

IT 2102 Digital Systems

Teaching Scheme: **Examination Scheme:**

Lectures: **3** Hrs/Week In-Semester: **50** Marks

Tutorial: **1** Hr/Week End-Semester: **50** Marks

Credits: 4

Course Objectives:

1. To learn and understand basic digital design techniques
2. To develop, design and implement combinational and sequential logic circuits
3. To learn programmable logic devices
4. To introduce computer arithmetic

Course Outcomes:

On completion of the course, student will be able to Explain–

1. Apply knowledge of number systems, codes, Boolean algebra and use necessary A.C, D.C Loading characteristics as well as functioning while designing with logic gates.
2. Use logic function representation for simplification with K-Maps and analyze as well as design Combinational logic circuits using SSI & MSI chips.
3. Analyze Sequential circuits like Flip-Flops (Truth Table, Excitation table), their conversion & design the applications.
4. Identify the Digital Circuits, Input/Outputs to replace by FPGA

Unit – I: Number System and Logic Families (05)

Introduction to digital electronics & Boolean algebra. Number Systems - Binary, Octal, Hexadecimal and their conversions. Signed Binary number representation and Arithmetic's: Signed & True Magnitude, 1's complement, 2's complement representation and arithmetic's.
Codes: BCD, Excess-3, Gray code, Binary Code and their conversion. Switching characteristics of BJT & FET, IC Characteristics. TTL: Standard TTL characteristics, Operation of TTL NAND gate, Subfamilies, totem pole, CMOS: Standard CMOS characteristics, operation of CMOS NAND, Subfamilies, Comparison of TTL & CMOS, Interfacing: TTL to CMOS and CMOS to TTL

Unit – II: Combinational Logic Design (08)

Logic minimization: Representation of truth-table, Simplification of logical functions, Minimization of SOP and POS forms, don't care Conditions. Reduction techniques: K-Maps, Quine - McClusky technique. CLC design using SSI chips – Code converters, Adder, Subtracter, n bit Binary adder, look ahead carry generator. Magnitude comparator. Introduction to MSI functions & chips - Multiplexers, Decoder / Demultiplexer, Encoder, Binary adder. CLC design using MSI chips – BCD & Excess 3 adder & subtracter, Implementation of logic functions using MSI chips

Unit – III: Sequential Logic (06)

Introduction to sequential circuits. Difference between combinational circuits and sequential circuits, memory element – latch. Flip- Flops: Design, truth table, excitation table of SR, JK, D, T flip flops. Study of flip flops with asynchronous and synchronous Preset & Clear, Master Slave configuration, conversion from one type to another type of flip flop. Application of flip-flops – Bounce elimination switch, Counters- asynchronous, synchronous and modulo counters study of modulus n counter ICs & their applications to implement mod counters.

Unit – IV: Sequential Logic Design (08)

Registers- Buffer register, shift register types - SISO, SIPO, PISO & PIPO, applications of shift registers - ring counter, twisted ring counter, study of universal shift register. Sequence generators using counters & shift register, Pseudo Random Binary Sequence Generator. Basic design steps-State diagram, State table, State reduction, State assignment, Mealy and Moore machines representation, Implementation, finite state machine implementation, sequence detector using Moore & Mealy model.

Unit – V: Programmable Logic Devices (06)

Algorithmic State Machines- ASM notations, charts (e.g.- counters, washing machine, lift controller, vending machine), design using multiplexer controller method (e.g.- counters). Introduction to PLD's – ROM, PAL, PLA, Design of 4 variable SOP using PLDs, Basic architecture of SPLD and CPLD, Study of CPLD architecture XC9572, Basic architecture of FPGA, CPLD. Design flow

Unit – VI: Computer Arithmetic (07)

A Brief History of computers, Von Neumann Architecture, Harvard architecture, Bus Interconnection, Scalar Data Types, Fixed and Floating point numbers, Booths algorithm for multiplication and its Hardware Implementation, Division: Restoring and Non Restoring algorithms, IEEE standards of Floating point representations, Floating point arithmetic.

Text Books:

1. R.P. Jain, "Modern Digital Electronics", 3rd Edition, Tata McGraw-Hill, ISBN: 0–07–049492–4

Reference Books:

1. W. Stallings, "**Computer Organization and Architecture: Designing for Performance**", 8th Edition, Prentice Hall of India, 2010, ISBN 13: 978-0-13-607373-4
2. C. Hamacher, V. Zvonko, S. Zaky, "**Computer Organization**", 5th edition, McGraw Hill, 2002, ISBN: 007-120411-3

IT 2103 Data Structures I

Teaching Scheme:

Lectures: **3** Hrs/Week

Examination Scheme:

In-Semester: **50** Marks

End-Semester: **50** Marks

Credits: 3

Course Objectives:

1. To learn logic building using algorithm for problem solving
2. To learn logic building for puzzles and games
3. To learn use of different data structures and algorithm asymptotic notations.
4. To learn use of different searching and sorting techniques
5. To learn linear data structures using sequential organization and recursion concept.
6. To learn linear data structures using linked organization.

Course Outcomes:

1. Apply appropriate programming language constructs to develop logical steps to solve a given real world problem.
2. Select appropriate searching and/or sorting techniques for application development.
3. Analyze algorithm complexities and use appropriate algorithms to solve a given problem
4. Select appropriate sequential and linked organization of data structures to solve a given problem

Unit – I: Introduction to Algorithm and Logic building (06)

Concept of algorithm, Algorithmic thinking and Logic building, Solving specific real world problems such as in numerical methods, quantitative aptitude etc. using Operators, control structures, enumeration, structure, union, macros, arrays, functions and parameter passing, scope rules, string manipulation, matrix operations.

Unit – II: Logic building for Puzzles/ Games and File Organization (06)

Logic for password cracking (Brute Force – all possible permutations), puzzle solving & creation like Sudoku, magic square, eight queen, logical games like mine sweeper, connect dots, tic-tac-toe, debugging, dry-run, understand different codes
File Organization: file operations, keyword search

Unit – III: Introduction to Data structures and Analysis of Algorithms (06)

Introduction to Data Structures: Types of data structures, Abstract Data Types, Analysis of algorithm: frequency count and its importance in analysis of an algorithm, Time complexity & Space complexity of an algorithm, Best, Worst and Average case analysis of algorithm.

Unit – IV: Searching and sorting techniques (08)

Need of searching and sorting, Concept of internal and external sorting, sort stability. Searching methods: Linear and binary search algorithms their comparison and complexity analysis
Sorting methods: Bubble, selection, insertion, merge, quick, bucket sort and their complexity analysis.

Unit – V: Logic building using linear data structures and recursion (06)

Concept of Linear data structures, ordered list, Multidimensional arrays and their storage representation. Sparse matrix using arrays - addition, polynomial representation.
Concept of recursion and logic building using iterative and recursive methods, Recursive algorithms e.g. Factorial, Fibonacci series, etc. Use of implicit stack in recursion

Unit – VI: Linked List (08)

Concept of linked organization, singly linked list, doubly linked list, circular linked list. Linked list as an ADT. Representation and manipulations of polynomials using linked lists, comparison of a sequential and linked memory organization, concept of Generalized Linked List, representation polynomial using GLL.

Text Books:

1. R. Gilberg, B. Forouzan, "**Data Structures: A pseudo code approach with C"**, Cenage Learning

Reference Books:

1. Dennis Ritchie, Kernighan, "**The C Programming Language'**, Prentice Hall
2. E. Horowitz, S. Sahani, S. Anderson-Freed, "**Fundamentals of Data Structures in C"**, Universities Press
3. G. A. V. Pai, "**Data structures and Algorithms"**, McGraw Hill
4. Jon Bentley, **"Programming Pearls"**, Addison Wesley

IT 2104 Network Fundamentals

Teaching Scheme:

Lectures: **3** Hrs/Week

Tutorial: **1** Hr/Week

Examination Scheme:

In-Semester: **50** Marks

End-Semester: **50** Marks

Credits: 4

Course Objectives:

1. To understand fundamentals of communication systems.
2. To acquaint themselves with layered model used in computer networks.
3. To understand OSI and TCP/IP models.
4. To understand analyse MAC layer protocols and LAN technologies.

Course Outcomes:

1. Enumerate the layers of the OSI model and TCP/IP.
2. To differentiate between media access schemes.
3. Design the IP addressing schemes for a computer network.
4. Familiarity with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.

Unit – I: Network – Centric World **(6)**

Communicating in a Network-Centric World, The Architecture of the Internet, Trends in networking, LAN, WAN, MAN, Networking Devices, Network Topologies Point to Point, Point to Multipoint Topologies.

Unit – II: Communicating over the Network **(6)**

The platform for communications, Protocols, OSI Model, TCP/IP Model, Protocol Data Units and Encapsulation, Comparison between OSI and TCP/IP Model, Network Addressing.

Unit – III: Network Layer **(8)**

IP Addressing, Communication from Host to Host ,Network Layer Protocol, Packaging the Transport Layer PDU ,IPv4 Packet Header, Subnetting, Static Routing ,Dynamic Routing ,Routing Protocols

Unit – IV: Ethernet **(6)**

Ethernet Basics, Collision Domain , Broadcast Domain, CSMA/CD , Half- and Full-Duplex Ethernet, Ethernet at the Data Link Layer, Ethernet Addressing , Ethernet Frames ,Channel Bonding, Ethernet at the Physical Layer.

Unit – V: Physical Layer **(7)**

The Theoretical Basis for data communication, Digital Modulation and Multiplexing, The Public Switched Telephone Network and Cable Television, Community Antenna Television, Internet over Cable, Spectrum Allocation, Cable Modems, ADSL Versus Cable, Network Interface.

Unit – VI: Data Link Layer **(7)**

Data Link Layer Design Issues, Error Detection and Error Correction, Sliding Window Protocol, Medium Access Control Sub layer, Channel Allocation Problem, Ethernet MULTIPLE ACCESS PROTOCOLS, ALOHA, Carrier Sense Multiple Access Protocols, Collision-Free Protocols,

Limited-Contention Protocols, Wireless LAN Protocols,

Text Books:

1. Mark A. Dye, Rick McDonald, Antoon W. Rufi, **"Network Fundamentals",** Cisco Press (2008)

Reference Books:

1. Andrew S. Tennabaum, David J. Weatherall **"Computer Networks",** Pearson (5^{th}edition), (2011)

IT 2105 Digital Systems Laboratory

Teaching Scheme: **Examination Scheme:**

Practical : **2** Hrs/Week Practical: **25** Marks

Credits: 1

Course Objectives:

1. To learn and understand basic digital design techniques.
2. To develop design and implementation skills of combinational and sequential logic circuits.
3. To introduce computer Arithmetic

Course Outcomes:

On completion of the course, student will be able to explain–

1. Apply knowledge of number systems, codes, Boolean algebra and use necessary A.C, D.C Loading characteristics as well as functioning while designing with logic gates.
2. Use logic function representation for simplification with K-Maps and analyze as well as design Combinational logic circuits using SSI & MSI chips.
3. Analyze Sequential circuits like Flip-Flops (Truth Table, Excitation table), their conversion & design the applications.
4. Identify the Digital Circuits, Input/Outputs to replace by FPGA, Advanced processor organization

Suggested List of Laboratory Assignments

1. Design (truth table, K-map) and implementation of 4-bit BCD to Excess-3 and Excess-3 to BCD Code converters.
2. Design (truth table, K-map) and implementation of 4 bit BCD & Excess 3 Adder using IC7483.
3. Implementation of logic functions using multiplexer IC 74153 & decoder IC 74138. (Verification, cascading & logic function implementation)
4. Design (State diagram, state table & K map) and implementation of 3 bit Up and Down Asynchronous Counter using master slave JK flip-flop IC 7476
5. Design (State diagram, state table & K map) and implementation of 3 bit Up and Down Synchronous Counter using master slave JK flip-flop IC 7476
6. Design and implementation of Module ‘n’ counter with IC7490 and IC 74191
7. Design (State Diagram, State Table, K Map) and implementation of Sequence Generator using Shift Register IC 74194.
8. Design and implement unsigned binary multiplication (3 bit)

Text Books:

1. R.P. Jain, “**Modern Digital Electronics**”, 3rd Edition, Tata McGraw-Hill, ISBN: 0–07–049492–4

Reference Books:

1. W. Stallings, “**Computer Organization and Architecture: Designing for Performance**”, 8th Edition, Prentice Hall of India, 2010, ISBN 13: 978-0-13-607373-4

2. C. Hamacher, V. Zvonko, S. Zaky, "**Computer Organization**", 5th edition, McGraw Hill, 2002, ISBN: 007-120411-3

IT 2106 – Data Structures I Laboratory

Teaching Scheme: **Examination Scheme:**

Practical : **4** Hrs/Week Practical: **50** Marks

Credits: **2**

Course Objectives:

1. To learn Python constructs
2. To learn algorithm development and analysis of algorithms
3. To learn linear data structures and their applications
4. To learn different searching and sorting techniques
5. To build logic to solve real world problems
6. To learn debugging to understand different codes & detect logical errors

Course Outcomes:

On completion of the course, student will be able to –

1. Implement appropriate searching and/or sorting techniques to solve a given problem
2. Implement algorithms to illustrate use of data structures such as array, linked list
3. Implement algorithms to create and manipulate database using sequential file organization
4. Debug different code snippets

Suggested List of Laboratory Assignments (13 assignments)

Group A Assignments (Python programming) (Any 5)

1. To check whether a given input number is prime or not
2. To develop a password cracker (brute force - permutations)
3. To develop tic-tac-toe game
4. a) Sort the set of strings in ascending order using Bubble sort and descending order by using Selection sort or Insertion sort. (Display pass by pass output) b) Search a particular string using binary search with and without recursion.
5. Implement Quick Sort to sort the given list of numbers. Display corresponding list in each pass. (with and without recursion)
6. Implement a doubly linked list with following options
   a) Insertion of a node at any location b) Deletion of a node from any location c) Display a linked list d) Display in linked list in reverse e) Reverse the linked list without using additional data structure, no data swapping

Group B Assignments (C programming) (Compulsory)

1. There are 10 students in art class and 15 students in dance class. 8 students are enrolled in both activities. (Sets).
   a) Find the students who are enrolled in both the activities
   b) Find the students who are enrolled only in art class
   c) Find all the student without repetition
2. Create a Database for employee salary calculation using array of structures and perform following operations on it:
   a) Create Database b) Display Database (tabular format) c) Add a record d) Search a record e) Modify a record f) Delete a record g) Search can be in different manner e.g. Search all records having percentage more than 70.
3. Implement sequential file and perform following operations:
   a) Display b) Add records c) Search record d) Modify record e) Delete record
4. Implement a singly linked list with following options
   a) Insertion of a node at any location
   b) Deletion of a node from any location
   c) Display a linked list
   d) Display in linked list in reverse
   e) Reverse the linked list without using additional data structure, no data swapping
5. Implement a doubly linked list with following options
   a) Insertion of a node at any location
   b) Deletion of a node from any location
   c) Display a linked list
   d) Display in linked list in reverse
   e) Reverse the linked list without using additional data structure, no data swapping

Group C Assignments (C programming) (Any 2)

1. Solve Simultaneous Equations in Three Variables (Matrix)
2. Implement following operations on string with / without pointers (without using library functions)
   a) Length b) Copy c) Reverse d) String comparison e) Palindrome f) Substring g) Search and replace character h) Password validation i) Code / decode
3. Implement polynomial using CLL and perform
   a) Addition of Polynomials b) Multiplication of polynomials c) Evaluation of polynomial

   Implement Generalized Linked List to create and display operations

Group D Assignment (Any programming language) (Any 1)

One assignment to be carried out by a group of 2 students.
Each group will design an application using the appropriate data structures – to solve real world problem (proof of concept). The group requires getting the application approved by the respective faculty member.

1. Unit / number system conversions
2. Verification of amount in digits and in words (e.g. as given on cheque)
3. Result analysis of class data (e.g. no. of first classes etc.)
4. Implementation of skip list
5. Operations on polynomials (e.g. add, multiply, evaluate)
6. Searching & counting no. of occurrence & location (line no) of a word in a given
7. Searching & counting no. of occurrence & location (line no) of a word in a given text file

8. Implementation of numerical methods (e.g. Runge Kutta)
9. Implementation to generalize Time / Train based aptitude question (R. S. Agarwal)
10. Recursive solution to problems (e.g. Tower of Hanoi)
11. Develop games (e.g. Tic-tac-toe, sudoku)
12. Text editor (Hint – GLL)
13. Code beautifier (e.g. int a ; - keyword shown in blue color, others in black)

Text Books:

1. Steve McConnell, "**Code complete**", Second edition, 2nd ed. Redmond, WA: Microsoft Press, 2007.
2. E. Balagurusamy, "**Introduction to Computing and Problem Solving Using Python**", McGraw Hill, ISBN : 9352602587

Reference Books:

1. E. Horowitz, S. Sahani, S. Anderson-Freed, "**Fundamentals of Data Structures in C**", Universities Press, 2008
2. R. Gilberg, B. Forouzan, "**Data Structures: A pseudo code approach with C**", Cenage Learning, ISBN 9788131503140.
3. Yashwant Kanetkar, "**Pointers in C**", BPB Publication
4. Rance Necaise, "**Data Structures and Algorithms Using Python**", Wiley, ISBN : 9788126562169

IT 2107 Web Engineering Technology Laboratory

Teaching Scheme: **Examination Scheme:**

Practical: **2** Hrs/Week Practical: **25** Marks

Credits: **1**

Course Objectives:

1. To understand various application layer protocols for its implementation in client/server environment

Course Outcomes:

By the end of the course, students should be able to

1. Apply basics of web designing
2. Design a simple web application
3. Implement dynamic web pages
4. Establish client and server-side communication

Group A

A. HTML

Create a registration form using HTML form input elements viz. textbox, text area, radio button and drop down menu, check box, submit, file and reset button. Field should contain name, address, birth-date, qualification, email, phone number, gender, comments, attach photo etc. Use HTML Form elements wherever required. Align all elements using table.

B. CSS

Create a horizontal navigation bar in DIV using external CSS which contain home, about, gallery, enquiry, contacts menus. Also create the same bar in vertical alignment in another DIV in same page.

C. Java Script

1. Write a Java script to create a simple calculator.
2. Write a Java script that read ten numbers and display the count of negative and positive numbers and count of zero from the list.
3. Create form validation program that checks the empty values from that form and alert back using alert function. Use at least 5 components.

D. PHP

1. Create a PHP program in which two values submitted using form and calculate its addition, subtraction, multiplication, modulation, average and division on the same page. Find the greatest number between them and square of each of them using PHP function.
2. Write PHP script to display the squares and cubes of 1 to 10 numbers in tabular format.
3. Write PHP script to validate Email address.
4. Create a login form using session handling in PHP. After successful login display name, address and other details in tabular format of logged user.

E. XML

Write an XML schema that provides tabulated information related to expected height (in cms) and weight (in kgs) for male and female separately for the age groups starting with 5-10 years, 15-20 years, and so on.

Group B

Design and develop web site in group of 2 using above all learnt technology.

Text Books:

1. **'Web Technologies Black Book: HTML, JavaScript, PHP, Java, JSP, XML and AJAX'** by Kogent Learning Solutions Inc.

Reference Books:

1. Steven M. Schafer, '**HTML, XHTML and CSS'**, Fourth Edition, Wiley India Edition. ISBN: 978-81-265-1635-3.