

Subject Code: PCIT-101
Subject Name: Data Structures

Programme: B.Tech. (IT)	L: 3 T: 0 P: 0
Semester: 3	Teaching Hours: 40
Theory/Practical: Theory	Credits: 3
Internal Marks: 40	Percentage of Numerical/Design Problems: 30%
External Marks: 60	Duration of End Semester Exam(ESE): 3hr
Total Marks: 100	Elective Status: Compulsory

Prerequisites: Knowledge of Programming for Problem Solving

Additional Material Allowed in ESE: NIL

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	Analyze and compare algorithms for efficiency using Big-O notation.
CO2	Create and evaluate new algorithms to solve complex engineering problems.
CO3	Illustrate various data structures to solve multi-disciplinary projects.
CO4	Utilize the templates for modularity.
CO5	Compare and classify various data structures
CO6	Demonstrate the reusability of data structures for implementing complex iterative problems.

Detailed Contents

Part-A

Introduction to Algorithms: Definition and brief description of various data structures, operations on data structures, Algorithm development, Complexity analysis, Big O notation, Time space trade-off. **[2 Hours]**

Arrays, Stacks and Queues: Linear and Multi-dimensional arrays and their representation, operations on arrays, Linear Search, Binary Search, Sparse matrices and their storage. Stacks: Array Representation and Implementation of Stacks, Operations on Stacks, Application of

stacks: Conversion of Infix to Prefix and Postfix Expressions, Evaluation of postfix expression using stack, Balanced parenthesis checking. Recursion, Tower of Hanoi Problem. Queues: Sequential representation of queue, linear queue, circular queue, operations on linear and circular queue, dequeue, priority queue. **[12 Hours]**

Linked Lists: Linear linked list, operations on linear linked list, doubly linked list, operations on doubly linked list, Circular Linked list, Garbage collection and Compaction, linked representation of Stack, Linked representation of a Queue, Skip List, Operation done in skip list, Implementing the skip list. **[8 Hours]**

Part-B

Trees and Graphs: Basic terminology of Trees, sequential and linked representations of trees, traversing a binary tree, brief introduction to threaded binary trees, AVL trees and B-trees, Heap Trees. Binary Search Trees: Binary Search Tree (BST), Insertion and Deletion in BST, Complexity of Search Algorithm. Basic terminology of graphs, representation of graphs (adjacency matrix, adjacency list), traversal of a graph (breadth - first search and depth - first search). **[10 Hours]**

Sorting and Hashing Techniques: Selection Sort, Insertion Sort, Bubble Sort, Quick Sort, Merge Sort, Heap Sort, Shell sort. Complexity. Hashing: Hashing Functions, Collision Resolution Techniques, Rehashing, Double hashing. **[8 Hours]**

Text Books

1. Seymour Lipschitz, "Data Structures", Scahum's Outline series, Tata McGraw Hill, 2015
2. Y. Langsam, M. J. Augenstein, A. M. Tanenbaum, "Data Structures using C and C++", PrenticeHall of India.2016
3. Sartaj Sahni, "Data Structures, Algorithms and Applications in C++", Tata McGraw Hill. 2008

Reference Books:

1. Michael T. Goodrich, Roberto Tamassia, David Mount, "Data Structures and Algorithms in C++", Wiley India,2011
2. Mark A. Weiss, Algorithms, "Data Structures, and Problem Solving with C++", Pearson Education. 1995
3. Reema Thareja, "Data Structures using C", Qxford Higher Education.2014

E-Books and online learning material

1. Handbook of Algorithms and Data Structures by [Gaston H. Gonnet](#), [Ricardo Baeza-Yates](#)
<https://users.dcc.uchile.cl/~rbaeza/handbook/hbook.html> Accessed on Feb 24, 2019
2. Martin Gonzalez Rodriquez, “Data Structures: Graphs”, e Book Accessed on Feb 24, 2019
3. Algorithms and Data Structures Lecture Materials by Douglas Wilhelm Harder
<http://www.freebookcentre.net/CompuScience/free-computer-algorithm-books.html> ,
Accessed on Feb 24, 2019
4. Notes on Data Structures and Programming Techniques by James Aspnes
<http://www.cs.yale.edu/homes/aspnes/classes/223/notes.pdf> Accessed on Feb 24, 2019
5. Skip List <https://courses.csail.mit.edu/6.046/spring04/handouts/skiplist.pdf> Accessed on Feb 24, 2019

Online Courses and Video Lectures

1. <http://www.nptelvideos.in/2012/11/data-structures-and-algorithms.html>
Accessed on Feb 24, 2019
2. https://www.youtube.com/watch?v=S47aSEqm_0I&index=1&list=PLgi_VZKxRKrxgFyOutPJpoLFBaQMOpK-
Accessed on Feb 24, 2019
3. <https://www.cs.utexas.edu/users/novak/cs315contents.html> Accessed on Feb 24, 2019
4. <https://lecturenotes.in/subject/81/data-structure-using-c-ds> Accessed on Feb 24, 2019
5. <https://www.youtube.com/channel/UCMNkvKnD3mo3Jj9eTwJlIWw>
Accessed on Feb 24, 2019
6. <https://www.sanfoundry.com/1000-data-structure-questions-answers/>
Accessed on Feb 24, 2019
7. <https://freevideolectures.com/subject/data-structures/> Accessed on Feb 24, 2019

Subject Code: PCIT-102

Subject Name: Object Oriented Programming using C++

Programme: B.Tech. (IT)	L: 3 T: 0 P: 0
Semester: 3	Teaching Hours: 40
Theory/Practical: Theory	Credits: 3
Internal Marks: 40	Percentage of Numerical/Design Problems: 50%
External Marks: 60	Duration of End Semester Exam(ESE): 3 Hours
Total Marks: 100	Elective Status: Compulsory

Prerequisites: Programming for problem solving.

Additional Material Allowed in ESE: NIL

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	Understand the basic concepts of classes, objects and methods as well as basic principles of object-oriented programming.
CO2	Create object oriented design based on the characteristics of an object-oriented programming language: data abstraction and information hiding, overloading and dynamic binding of the messages to the methods.
CO3	Apply the concepts of inheritance and relationship among different objects to generate the hierarchies like generalization and aggregation.
CO4	Investigate the concept of strings, File Handling and Exception handling of Specific Programming Problem
CO5	Function on Multi-disciplinary team by using OOPs experiments and Projects.
CO6	Demonstrate real world applications based on the concepts of OOP in C++..

Part-A

Fundamental Concepts of a Programming Language: basic structure of a program, character set, tokens, keywords and identifiers, constant and variables, data types and sizes, operators and expressions, operator precedence, promotion and type conversion, Control flow (if, if-else,

nested if-else, for, while, do-while, break, switch, continue and goto statements), arrays and strings. **[8 Hours]**

Object Oriented Paradigm: need of object oriented programming, comparison of structured and object oriented development, elements of object oriented programming, encapsulation and data abstraction, inheritance, polymorphism, inheritance, dynamic binding and message communication. **[3 Hours]**

Classes and Objects: introduction to classes and objects, class specification, defining member functions, reference and instance variables, scope resolution operator, objects as arguments and returning object. **[3 Hours]**

Object Initialization and Cleanup : access specifiers, pass by reference, need for constructors and destructors, parameterized constructors, copy constructor, dynamic constructor, destructors, static data and member functions. **[3 Hours]**

Overloading and Inheritance: unary and binary operator overloading, function overloading. Inheritance – need of inheritance, forms of inheritance, inheritance and member accessibility, generalization and aggregation, object composition and delegation, super keyword, overriding. **[3 Hours]**

Part-B

Pointers and Run Time Polymorphism: Abstract classes, need of pointers, pointer to objects, this keyword, pointer to derived classes, friend functions, virtual functions and pure virtual functions. **[5 Hours]**

Exception Handling: error handling, exception handling model, try, throw, catch and multiple catch, nested try, handler throwing the same exception again, handling uncaught exception and user defined exceptions, rules for handling exceptions successfully. **[5 Hours]**

String Handling: creating string objects, extracting characters of a string, string handling functions, streams in C++, formatted and unformatted I/O, formatting using manipulators. **[5 Hours]**

File Handling and Templates: file streams, opening and closing a file, reading and writing a file, error handling during file operations, templates, class and function templates. [5 Hours]

Text Books

1. K R Venugopal, Rajkumar Buyya and T Ravishankar, “Mastering C++”, 2nd Edition, Tata McGraw Hill Publishing Company Ltd, 1999.
2. E. Balagurusamy, “Object Oriented Programming with C++”, 7th Edition, McGraw Hill, 2017
3. Kanetkar P. Yashavant, “Let Us C++”, BPB Publications, 2nd Edition, 2003.
4. Bjarne Stroustrup, “The C++ Programming Language”, 4th Edition , Addison Wesley, 2013.

Reference Books

1. Dietel, H.M. and Dietel, P.J., “C++ How to Program”, 6th Edition, Prentice Hall, 2008.
2. Coohoon and Davidson, C++ Program Design: An introduction to Programming and Object-Oriented Design. Tata McGraw Hill, 3rd Edition. 2003.
3. Herbert Schildt, C++: The Complete Reference, 4th Edition, McGraw - Hill Education, 2017

E-Books and online learning material

1. Tutorials Point (C++ Object Oriented)
https://www.tutorialspoint.com/cplusplus/cpp_object_oriented.htm Accessed on Feb 21, 2019
2. C++ Tutorials (Object Oriented Programming)
<https://www.studytonight.com/cpp/cpp-and-oops-concepts.php> Accessed on Feb 21, 2019
3. GeeksforGeeks (A computer science portal for geeks)
<https://www.geeksforgeeks.org/basic-concepts-of-object-oriented-programming-using-c/>
Accessed on Feb 21, 2019
4. LearnCpp.com (Tutorial to help you master C++ and Object Oriented Programming)
<https://www.learncpp.com/> Accessed on Feb 21, 2019

Online Courses and Video Lectures

1. <http://www.cplusplus.com/> Accessed on Feb 21, 2019
2. <https://nptel.ac.in/courses/106105151> Accessed on Feb 21, 2019

Subject Code: ESIT-101

Subject Name: Digital Circuits and Logic Design

Programme: B.Tech. (IT)	L: 3 T: 0 P: 0
Semester: 3	Teaching Hours: 40
Theory/Practical: Theory	Credits: 3
Internal Marks: 40	Percentage of Numerical/Design Problems: 40%
External Marks: 60	Duration of End Semester Exam(ESE): 3hr
Total Marks: 100	Elective Status: Compulsory

Prerequisites: Knowledge of basic Electronics and Number Systems

Additional Material Allowed in ESE: NIL

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	To understand and examine the structure of various number systems and its application in digital design
CO2	Utilize knowledge of number systems, codes and Boolean algebra to the analysis and design of digital logic circuits
CO3	Formulate and employ a Karnaugh Map to reduce Boolean expressions and logic circuits to their simplest forms
CO4	Identify concepts and terminology of digital logic circuits
CO5	Ability to understand, analyze and design various combinational and sequential circuits.
CO6	To develop skill to build, and troubleshoot digital circuit

Part-A

Number Systems: Binary, Octal, Decimal, and Hexadecimal. Number base conversions, 1's, 2's, n's complements, signed Binary numbers. Binary Arithmetic, Binary codes: Weighted BCD, Gray code, Excess 3 code, ASCII – conversion from one code to another. **[5 Hours]**

Boolean Algebra: Boolean postulates and laws – De-Morgan’s Theorem, Principle of Duality, Boolean expression – Boolean function, Minimization of Boolean expressions – Sum of Products (SOP), Product of Sums (POS), Minterm, Maxterm, Canonical forms, Conversion between canonical forms, Karnaugh map Minimization, Quine-McCluskey method - Don’t care conditions **[5 Hours]**

Logic GATES: AND, OR, NOT, NAND, NOR, Exclusive-OR and Exclusive-NOR. Implementations of Logic Functions using gates, NAND-NOR implementations. Study of logic families like RTL, DTL, DCTL, TTL, MOS, CMOS, ECL and their characteristics. **[7 Hours]**

Part-B

Combinational Circuits: Design procedure – Adders, Subtractors, Serial adder/Subtractor, Parallel adder/ Subtractor Carry look ahead adder, BCD adder, Magnitude Comparator, Multiplexer/Demultiplexer, encoder/decoder, parity checker, code converters. Implementation of combinational logic using MUX. **6 Hours]**

Sequential Circuits: Flip flops SR, JK, T, D and Master slave, Excitation table, Edge triggering, Level Triggering, Realization of one flip flop using other flip flops. Asynchronous/Ripple counters, Synchronous counters, Modulo-n counter, Ring Counters. Classification of sequential circuits-Moore and Mealy, Design of Synchronous counters: state diagram, Circuit implementation. Shift registers **[8 Hours]**

Signal Conversions: Analog & Digital signals. A/D and D/A conversion techniques (Weighted type, R-2R Ladder type, counters Type, Dual Slope type, Successive Approximation type). **[4 Hours]**

Introduction to VHDL :Introduction, Behavioral, Data flow, Structural Models, Simulation Cycles, Process Concurrent Statements, Sequential Statements, Loops, Functions and Procedures, Tools used for Simulation of VHDL.**[5 Hours]**

Text Books

1. Morris Mano M, Michael D. Ciletti, “Digital Design”, Pearson Education, 4th Edition, 2007.
2. Charles H Roth (Jr), Larry L. Kinney, “Fundamentals of Logic Design”, Cengage Learning India Edition, 5th Edition, 2010.
3. Floyd and Jain, “Digital Fundamentals”, Pearson Education, 8th Edition, 2007.

Reference Books

1. Donald P Leach, Albert Paul Malvino, GoutamSaha, “Digital Principles and Applications”, Tata McGraw Hill, 6th Edition, 2008
2. William H. Gothmann, “Digital Electronics”, 2nd Edition, PHI.
3. Anand Kumar, “Fundamental of Digital Circuits” 3rd Edition, PHI, 2014

E-Books and online learning material

1. Digital Logic and Computer Design by Morris Mano
<https://www.pdfdrive.com/digital-logic-and-computer-design-by-m-morris-mano-d34332016.html>
Accessed on Feb 21, 2019

2. Digital Circuits and Systems by NEPTEL
<http://www.freebookcentre.net/electronics-ebooks-download/Digital-Circuits-and-Systems.html>
Accessed on Feb 21, 2019

Online Courses and Video Lectures

1. <https://nptel.ac.in/courses/117106086/1> Accessed on Feb 21, 2019
2. <https://freevideolectures.com/course/3164/digital-electronics> Accessed on Feb 21, 2019
3. <http://www.nptelvideos.in/2012/12/digital-circuits-and-systems.html>
Accessed on Feb 21, 2019
4. <https://swayam.gov.in/courses/4860-july-2018-digital-circuits> Accessed on Feb 20, 2019

Subject Code: PCIT-103

Subject Name: Data Communication and Computer Networks

Programme: B.Tech. (IT)	L: 3 T: 0 P: 0
Semester: 3	Teaching Hours: 40
Theory/Practical: Theory	Credits: 3
Internal Marks: 40	Percentage of Numerical/Design Problems: 20%
External Marks: 60	Duration of End Semester Exam(ESE): 3 Hours
Total Marks: 100	Elective Status: Compulsory

Prerequisites: Programming for Problem Solving.

Additional Material Allowed in ESE: Nil

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	Understand Network essentials, Network Architecture, TCP/IP and OSI model. .
CO2	Analyze and solve networking problems in the area of guided and unguided transmission media.
CO3	Illustrate multi - channel access protocols and IEEE 802standards for LAN and MAN.
CO4	Contrast the design issues and working of protocols at different layers of TCP/IP and OSI models.
CO5	Formulate the various congestion and routing algorithms
CO6	Implement the concepts of N/W security and protocols such as HTTP, FTP, Telnet, DNS.

Detailed Contents

Part-A

Introduction to Computer Networks: Introduction to Computer Networks Data Communication System and its components, Data Flow, Computer network and its goals, Types of computer networks: LAN, MAN, WAN, Wireless and wired networks, broadcast and point to point networks, Network topologies, Network software: concept of layers, protocols, interfaces and services, ISO-OSI reference model, TCP/IP reference model. **[6 Hours]**

Physical Layer: Concept of Analog & Digital Signal, Bandwidth, Transmission Impairments: Attenuation, Distortion, Noise, Data rate limits : Nyquist formula, Shannon Formula, Multiplexing : Frequency Division, Time Division, Wavelength Division, Introduction to Transmission Media : Twisted pair, Coaxial cable, Fiber optics, Wireless transmission (radio, microwave, infrared), Switching: Circuit Switching, Message Switching ,Packet Switching & their comparisons. [7 Hours]

Data Link Layer: Design issues, Framing, Error detection and correction codes: checksum, CRC, hamming code, Data link protocols for noisy and noiseless channels, Sliding Window Protocols: Stop & Wait ARQ, Go-back-N ARQ, Selective repeat ARQ, Data link protocols: HDLC and PPP. [7 Hours]

Part-B

Medium Access Sub-Layer: Static and dynamic channel allocation, Random Access: ALOHA, CSMA protocols, Controlled Access: Polling, Token Passing, IEEE 802.3 frame format, Ethernet cabling, Manchester encoding, collision detection in 802.3, Binary exponential back off algorithm. [6 Hours]

Network Layer: Design issues, IPv4 classful and classless addressing, subnetting, Routing algorithms: distance vector and link state routing, Congestion control: Principles of Congestion Control, Congestion prevention policies, Leaky bucket and token bucket algorithms. [6 Hours]

Transport Layer: Elements of transport protocols: addressing, connection establishment and release, flow control and buffering, multiplexing and de-multiplexing, introduction to TCP/UDP protocols and their comparison [3 Hours]

Application Layer: World Wide Web (WWW), Domain Name System (DNS), E-mail (SMTP, POP3, IMAP, MIME) [2 Hours]

Text Books

1. Forouzan, B.A., “Data communication and Networking”, McGraw Hill , 4th edition , 2006.
2. Tanenbaum , A.S., “Computer Networks”, Prentice Hall , 5th edition ,2010.
3. Stallings, W., “Computer Networking with Internet Protocols and Tech”, Prentice Hall of India , 9th edition, 2010.

Reference Books

1. Comer, D.E., Internetworking with TCP/IP Vol. 1 Principles, Portals and Architecture, Prentice Hall of India ,5th edition, 2005
2. NarasimhaKarumanchi, Elements of Computer Networking: An Integrated Approach (Concepts, Problems and Interview Questions) CareerMonk Publication,1st Edition , 2014.
3. Norman F. Schneidewind, Computer, Network, Software, and Hardware Engineering with Applications, Wiley-IEEE Press
4. Victor Olifer , Computer Networks: Principles, Technologies and Protocols for Network Design Paperback , Wiley 2006

E-Books and online learning material

1. Computer networks book (Forouzan)

<https://doc.lagout.org/network/Data%20Communications%20and%20Networking%20By%20Behrouz%20A.Forouzan.pdf> Accessed on Feb 20, 2019

2. Computer networks book(Tanenbaum)

<http://iips.icci.edu.iq/images/exam/Computer-Networks---A-Tanenbaum---5th-edition.pdf>

Accessed on Feb 20, 2019

Online Courses and Video Lectures

- 1.https://www.tutorialspoint.com/computer_fundamentals/computer_networking.htm

Accessed on Feb 20, 2019

2. <https://www.geeksforgeeks.org/basics-computer-networking/> Accessed on Feb 20, 2019

3. <https://www.studytonight.com/computer-networks/> Accessed on Feb 20, 2019

Subject Code: HSIMT-101

Subject Name: Professional Practice, Laws and Ethics for IT Engineers

Programme: B.Tech. (IT)	L: 3 T: 0 P: 0
Semester: 3	Teaching Hours: 40
Theory/Practical: Theory	Credits: 3
Internal Marks: 40	Percentage of Numerical/Design Problems: NA
External Marks: 60	Duration of End Semester Exam(ESE): 3hr
Total Marks: 100	Elective Status: Compulsory

Prerequisites: Nil

Additional Material Allowed in ESE: Nil

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	Critically analyze and discuss key characteristics and emerging issues of Professional practice and ethics.
CO2	Articulate and reflect on the industry expectations of competence and conduct in IT related professions.
CO3	Awareness of types of ethical challenges and dilemmas confronting in IT field.
CO4	Ability to relate ethical concepts and materials to ethical problems in specific professions and professionalism.
CO5	Interpret and understand the various chapters and sections under Indian IT Act 2000.
CO6	Reckon out the need of Intellectual property rights and copyrights.

Detailed Contents

Part-A

Professional Practice: Definition of Professional practice, What constitutes a profession, Professional relationships, Principles of professional practice, Nature of Professionalism and its place in the field of Information Technology, are IT workers professionals, Industry code of

practice, Benefits of an industry code of practice, Writing a code of conduct, Implementing and review of code of conduct. **[3 Hours]**

Teamwork and conflict management: Skills required for functioning of effectively in a team environment, Methods how industry approaches towards a common goal, Methods for conflict management in building stronger teams. **[3 Hours]**

Professional Ethics: Definition of Ethics, Business Ethics, Corporate Ethics, Engineering Ethics, Personal Ethics; Importance of Integrity, Difference between moral ethics and laws, Ethics in Business world, Contrast ethical and legal issues as related to Information Technology, How IT uses or benefits from social and professional issues. **[3 Hours]**

Ethics for IT workers and Green IT: Significance of social context of IT and adherence to ethical code of conduct, IT, IEEE Code of Ethics, Developing green IT policies, Standards and learn to identify green IT, Evaluate green computing performance metrics, recycling practice, energy uses, Incentives at workplace for implementing green computing, management of computer hazardous material **[4 Hours]**

Risk Identification and evaluation: Analyze the role of risk to an organization, Risk Assessment and appropriate actions, Establishing a security policy, Prevention, Detection, response, Information Privacy, Privacy laws, application and court rulings, Identity theft, Consumer profiling, Treating customer data responsibly, Workplace monitoring **[4 Hours]**

Freedom of Expression: First Amendment rights, controlling access to information on internet, defamation and hate speech, Anonymity on internet, corporate blogging, Social Networking: Social networking ethical issue, Cyber bullying, cyber stalking, uploading of inappropriate materials, Crime in virtual worlds. **[5 Hours]**

Part-B

Laws, IT Act 2000, Intellectual Property and Copyrights: Computer Misuse Act 1990, Overview of IT Act 2000, definition, chapters, Jurisdiction, Penalties, Intellectual property: Introduction – meaning of intellectual property, main forms of IP, Copyright, Trademarks, Patents and Designs, Secrets; Law relating to Copyright in India including Historical evolution of Copy Rights Act, 1957, Meaning of copyright – computer programs, Ownership of copyrights and assignment, Criteria of infringement, Piracy in Internet – Remedies and procedures in India; Law relating to Patents under Patents Act, 1970 including Concept and historical perspective of patents law in India, Patentable inventions with special reference to biotechnology products,

Patent protection for computer programs, Process of obtaining patent, Rights and obligations of patentee, Duration of patents law and policy considerations, Infringement and related remedies.

[10 Hours]

Industry standards and careers in IT: Evaluate skills sets essential to a career in IT, Elements of successful technical resume, significant elements needed for a strong interview for an IT position. Information technology standards, standards organizations, Coding Standards, DBMS standards, Network Standards **[4Hours]**

Plagiarism: Introduction to Plagiarism, Types of Plagiarism, taking the Plauge out of Plagiarism, Copies (and Robberies), Shades of Gray: Reason for Students towards Plagiarism, Reducing the Sentence: Responding to Plagiarism, Acting, and Not Reacting: Giving, Tools to Avoid Plagiarism. **[4 Hours]**

Text Books

1. Ethics in information Technology, 4th Edition, George W Reynoldss, Cengage Learning, 2012.
2. Professional issues in IT, Frank Bott, British Computer Society, UK, 2014.
3. Information Technology law and practice, 3rd Edition, Vakul Sharma, Universal Publisher, 2018.

Reference Books

1. The Complete Guide to Referencing and Avoiding Plagarism (Open Up Study kills) Colin Nevilee, Open University Press; 2007.
2. Professional Ethics, R Subramanian, 2nd Edition, Oxford Press, 2017.

Online learning material

1. Green IT, <https://www-03.ibm.com/press/attachments/GreenIT-final-Mar.4.pdf>, [Accessed on 20 Feb 2019]
2. Carriers in IT, https://www.microtrain.net/files/career_pathways.pdf, [Accessed on 20 Feb 2019]
3. Carrier in IT, <https://www.bls.gov/opub/btn/volume-2/pdf/careers-in-growing-field-of-information-technology-services.pdf>, [Accessed on 20 Feb 2019]

4. <https://www.cl.cam.ac.uk/teaching/0910/PPEthics/PPEnotesRJennings.pdf> [Accessed on 20 Feb 2019]
5. IT Act 2000, <https://www.wipo.int/edocs/lexdocs/laws/en/in/in024en.pdf>, The Gazette of India [Accessed on 20 Feb 2019]
6. IEEE code of Ethics, <https://www.ieee.org/about/corporate/governance/p7-8.html> [Accessed on 20 Feb 2019]
7. Industry Code of Practice, <https://lawpath.com.au/blog/what-is-an-industry-code-of-practice> [Accessed on 20 Feb 2019]
8. IT Standards, <https://www.encyclopedia.com/computing/news-wires-white-papers-and-books/information-technology-standards> [Accessed on 20 Feb 2019]
9. Network Standards: <http://what-when-how.com/data-communications-and-networking/network-standards-data-communications-and-networking/> [Accessed on 20 Feb 2019]
10. Computer Misuse act : <https://www.legislation.gov.uk/ukpga/1990/18/contents> [Accessed on 20 Feb 2019]
11. [Coding Standards, www.cs.nott.ac.uk/~pszcah/G53QAT/Reports09/jjb07u/QAT09Report-jjb07u.doc](http://www.cs.nott.ac.uk/~pszcah/G53QAT/Reports09/jjb07u/QAT09Report-jjb07u.doc) [Accessed on 20 Feb 2019]

Subject Code: LPCIT-101

Subject Name: Data Structures Laboratory

Programme: B.Tech. (IT)	L: 0 T: 0 P: 2
Semester: 3	Teaching Hours: 26
Theory/Practical: Practical	Credits: 2
Internal Marks: 30	Percentage of Numerical/Design Problems: 100%
External Marks: 20	Duration of End Semester Exam(ESE): 1.5 hr
Total Marks: 50	Elective Status: Compulsory

Prerequisites: Fundamentals of Computers.

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	Improve practical skills in designing and implementing data structure algorithms
CO2	Compose algorithms to solve complex engineering problems
CO3	Implement the templates for modularity
CO4	Design graphical user interface for better human computer interaction.
CO5	Utilize data structure algorithms in a better way to solve responsibilities relevant to other professional engineering practices
CO6	Execute projects for individual or team based on data structure algorithms.

Special Instruction related to resources requirement: Code Block / C or C++ compiler

Sr.No. Name of Practical

1. Program to insert a new element at end as well as at a given position in an array.
2. Program to delete an element from a given whose value is given or whose position is given.
3. Program to find the location of a given element using Linear Search
4. Program to find the location of a given element using Binary Search
5. Program to implement push and pop operations on a stack using linear array.
6. Program to convert an infix expression to a postfix expression using stacks.
7. Program to evaluate a postfix expression using stacks.

8. Implement recursive function for Tower of Hanoi problem.
9. Program to implement insertion and deletion operations in a queue using linear array.
10. Program to implement linked list.
11. Program to implement push and pop operations on a stack using linked list.
12. Program to implement push and pop operations on a queue using linked list.
13. Program to sort an array of integers in ascending order using bubble sort.
14. Program to sort an array of integers in ascending order using selection sort.
15. Program to sort an array of integers in ascending order using insertion sort.
16. Program to sort an array of integers in ascending order using quick sort
17. Program to traverse a Binary search tree in Pre-order, In-order and Post-order.
18. Program to traverse graphs using BFS.
19. Program to traverse graphs using DFS.

Mini Project: Student has to do a project assigned from course contents in a group of two or three students. The group of students must submit a project report of 8 to 10 pages (approximately) and the team will have to demonstrate as well as have to give a presentation of the same

Reference Books

1. Fundamentals of Data structures by Ellis Horowitz & Sartaj Sahni, Pub, 2011,AW
2. Data Structures and Program Design in C By Robert Kruse, PHI, 1996
3. Theory & Problems of Data Structures by Jr. Symour Lipschetz, Schaum?s outline by TMH,2003

Subject Code: LPCIT-102

Subject Name: Object Oriented Programming using C++ Laboratory

Programme: B.Tech. (IT)	L: 0 T: 0 P: 2
Semester: 3	Teaching Hours: 26
Theory/Practical: Practical	Credits: 1
Internal Marks: 30	Percentage of Numerical/Design Problems: 100%
External Marks: 20	Duration of End Semester Exam(ESE): 1.5 Hours
Total Marks: 50	Elective Status: Compulsory

Prerequisites: Programming for Problem Solving.

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	Develop the solution to problems demonstrating usage of control and selection structure.
CO2	Learn and develop the solution of real world problems with the practical use of data abstraction and information hiding, constructors and destructors, overloading and dynamic binding of the messages to the methods.
CO3	Implement the inheritance and relationship among different objects based on the concepts of generalization and aggregation.
CO4	Demonstrate various I/O operations and String manipulations.
CO5	Create programs related to exception handling.
CO6	Understand generic programming, template and file handling.

Special Instruction related to resources requirement: Any compiler of C++ may be used for the programming, however the use of Open Source Software is recommended such as Code Blocks or Dev C++ .

Sr.No. Name of Practical

1. Demonstrate the use of data types in C++.
2. Demonstrate the use of unary, binary and special operators in C++.
3. Programming practice on if, if-else, nested if and switch statements in C++.
4. Programming practice on “for”, “do-while”, “while”, “break” and “continue” statements in C++.
5. Demonstrate the use of class, objects, reference variable and scope resolution operator.

6. Programming practice to use objects as arguments and returning objects from functions.
8. Initialization of the member variables using constructors and illustrate the use of destructors.
9. Demonstrate the concept of default constructor, parameterized constructor and copy constructor.
10. Demonstrate the role of “static” keyword C++..
11. Demonstrate the overloading of unary operators and binary operators of C++.
12. Demonstrate the different types of inheritance and illustrate the concept of overriding.
13. Illustrate the concept of Abstract class and abstract functions.
14. Illustrate the concept of virtual functions and pure virtual functions
15. Demonstrate the use of exception handling in C++.
16. Demonstrate string handling functions of C++.
17. Illustrate the concept of streams and file pointers
18. Perform read and write operations on a file.
19. Demonstrate the concept of Templates

Mini Project:

Students are required to prepare a project based on the course contents in a group of two to three students. The usage of concepts like inheritance, exception handling, file handling and string handling for the development of the project is recommended. The group of students must submit a project report of 8 to 10 pages (approximately) and the team will have to demonstrate the project as well as have to give a presentation of the same.

Text Books

1. K R Venugopal, Rajkumar Buyya and T Ravishankar, “Mastering C++”, 2nd Edition, Tata McGraw Hill Publishing Company Ltd, 1999.
2. E. Balagurusamy, “Object Oriented Programming with C++”, 7th Edition, McGraw Hill, 2017
3. Kanetkar P. Yashavant, “Let Us C++”, BPB Publications, 2nd Edition, 2003.
4. Bjarne Stroustrup, “The C++ Programming Language”, 4th Edition , Addison Wesley, 2013.

Reference Books

1. Dietel, H.M. and Dietel, P.J., “C++ How to Program”, 6th Edition, Prentice Hall, 2008.

2. Coohoon and Davidson, C++ Program Design: An introduction to Programming and Object-Oriented Design. Tata McGraw Hill, 3rd Edition. 2003.
3. Herbert Schildt, C++: The Complete Reference, 4th Edition, McGraw - Hill Education, 2017

E-Books and online learning material

1. Tutorials Point (C++ Object Oriented)

https://www.tutorialspoint.com/cplusplus/cpp_object_oriented.htm Accessed on Feb 21, 2019

2. C++ Tutorials (Object Oriented Programming)

<https://www.studytonight.com/cpp/cpp-and-oops-concepts.php> Accessed on Feb 21, 2019

3. GeeksforGeeks (A computer science portal for geeks)

<https://www.geeksforgeeks.org/basic-concepts-of-object-oriented-programming-using-c/>
Accessed on Feb 21, 2019

4. LearnCpp.com (Tutorial to help you master C++ and Object Oriented Programming)

<https://www.learncpp.com/> Accessed on Feb 21, 2019

Online Courses and Video Lectures

1. <http://www.cplusplus.com/> Accessed on Feb 21, 2019

2. <https://nptel.ac.in/courses/106105151> Accessed on Feb 21, 2019

Subject Code: LPCIT-103

Subject Name: Data Communication and Computer Networks Laboratory

Programme: B.Tech. (IT)	L: 0 T: 0 P: 2
Semester: 3	Teaching Hours: 26
Theory/Practical: Practical	Credits: 1
Internal Marks: 30	Percentage of Numerical/Design Problems: 100%
External Marks: 20	Duration of End Semester Exam(ESE): 1.5 hr
Total Marks: 50	Elective Status: Compulsory

Prerequisites: Fundamentals of Computers.

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	Demonstrate the hardware components, transmission media and tools used in computer networks.
CO2	Implement the LAN based on different topologies.
CO3	Execute various networking commands related to troubleshooting.
CO4	Implementation of file and printer sharing.
CO5	Understanding packet tracers.
CO6	Design and execute projects in networking in multi-disciplinary teams.

Special Instruction related to resources requirement: Hardware requirement: RJ-45 connector, twisted pair cable, crimping tool

Software requirement: C language, Open Source Simulation Tools, Packet Tracer

Sr.No. Name of Practical

1. Implementation of Error detection and correction technique.
2. Implementation of stop and wait protocol and sliding window.
3. Implementation of High level data link protocol.
4. To create a scenario and study the performance of network with CSMA Protocol
5. Implementation of Distance vector routing protocol

6. Take an example subnet graph with weights indicating delay between nodes. Now obtain Routing table at each node using distance vector routing algorithm.
7. Implementation and study of Go back-N and selective repeat protocols

Mini Project: Student has to do a project assigned from course contents in a group of two or three students. The group of students must submit a project report of 8 to 10 pages (approximately) and the team will have to demonstrate as well as have to give a presentation of the same

Reference Books

1. <http://www.rpsinstitutions.org/downloads/lab%20manual/cnlab.pdf>
2. Cisco Learning network <https://learningnetwork.cisco.com/thread/118264>
3. <https://www.experts-exchange.com/questions/24163288/How-to-implement-an-ARQ-stop-and-Wait-Protocol.html>

Subject Code: LESIT-101

Subject Name: Digital Control Logic and Circuits Lab

Programme: B.Tech. (IT)	L: 0 T: 0 P: 2
Semester: 4 th Semester	Teaching Hours: 26
Theory/Practical: Practical	Credits: 1
Internal Marks: 30	Percentage of Numerical/Design Problems: 100%
External Marks: 20	Duration of End Semester Exam(ESE): 1.5 hr
Total Marks: 50	Elective Status: Compulsory

Prerequisites: Knowledge of Basic Electronics Engineering and Basic Programming Languages.

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	Demonstrate the logic gates and realization of AND, OR, NOT and XOR functions using universal gates
CO2	Design and implement combinational circuits like half adder/ Full adder, half Subtractor/ Full Subtractor, and, MUX/DEMUX
CO3	Design and implement sequential circuits.
CO4	Compare and contrast the outputs of flip flops and counters using different chips
CO5	Demonstrate the working of Synchronous and Asynchronous Counters.
CO6	Execute the various experiments and projects on individuals and multidisciplinary teams through various logic designs and circuits

Special Instruction related to resources requirement: Digital trainer kits and Digital ICs and Programming Language Software

Sr.No. Name of Practical

- 1 Truth-table verification of OR, AND, NOT, XOR, NAND and NOR gates (using various IC's and programming languages)
2. Realization of OR, AND, NOT and XOR functions using universal gates IC's 7400 and 7402
3. Half Adder / Full Adder: Realization using basic and XOR gates IC's(also using Programming Languages)
4. Half Subtractor / Full Subtractor: Realization using IC's 7400 and 7402(also using Programming Languages)
5. Realisation of IC7483 as Parallel Adder/Subtractor
6. 4-Bit Binary-to-Gray & Gray-to-Binary Code Converter: Realization using Basic, XOR gates and Universal gates

- 7 4-Bit and 8-Bit Comparator: Implementation using IC7485 magnitude comparator chips
8. Multiplexer: Truth-table verification and realization of Half adder and Full adder using IC74153 chip
9. Demultiplexer: Truth-table verification and realization of Half subtractor and Full subtractor using IC74139 chip
10. Flip Flops: Truth-table verification of JK Master Slave FF, T-type and D-type FF using IC7476 chip.
- 11 Asynchronous Counter: Realization of 4-bit up counter and Mod-N counter using IC7490 & IC7493 chip.
- 12 Synchronous Counter: Realization of 4-bit up/down counter and Mod-N counter using IC74192 & IC74193 chip.

Mini Project: Student has to do a project assigned from course contents in a group of two or three students. The group of students must submit a project report of 8 to 10 pages (approximately) and the team will have to demonstrate as well as have to give a presentation of the same

Reference Books:

1. Digital Fundamentals, Thomas L. Floyd, Pearson Education, ISBN:9 788131734483
2. Digital Principles and Applications, Malvino and Leach, TMH

Subject Code: MCIT-101
Subject Name: Environmental Sciences

Programme: B.Tech. (IT)	L: 3 T: 0 P: 0
Semester: 3	Teaching Hours: 24
Theory/Practical: Theory	Credits: 0
Internal Marks: 40+10*	Percentage of Numerical/Design Problems: 00%
External Marks: Nil	Duration of End Semester Exam(ESE): 03hr
Total Marks: 50	Elective Status: Compulsory

Prerequisites: basics of environment.

Additional Material Allowed in ESE: NIL

***10 marks will be awarded based upon the performance in debates/seminar/fields study related to the contemporary issues of the subject.**

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	Measure environmental variables and interpret results.
CO2	Evaluate local, regional and global environmental topics related to resource use and management.
CO3	Propose solutions to environmental problems related to resource use and management.
CO4	Interpret the results of scientific studies of environmental problems.
CO5	Describe threats to global biodiversity, their implications and potential solutions.

Part-A

Natural Resources: Renewable and non renewable resources: Natural resources and associated problems: Forest resources: Use and over-exploitation, deforestation, case studies, Timber extraction, mining, dams and their effects on forests and tribal people [**2 Hours**]

Relational Model: Use and over-utilization of surface and ground water, floods, drought, conflicts over water, dam's benefits and problems, Food Resources: World food problems, changes caused by agriculture and over grazing, effects of modern agriculture, fertilizers- pesticides problems, water logging, salinity, case studies, Land Resources: Land as a resource, land degradation, man induces landslides, soil erosion, and desertification [**3 Hours**]

Eco Systems: Concept of an ecosystem, Structure and function of an ecosystem, Producers, consumers, decomposers, Energy flow in the ecosystems, Ecological succession, Food chains, food webs and ecological pyramids [**3 Hours**]

Environmental Pollution: Definition, causes, effects and control measures of: Air pollution, Water pollution, Soil pollution, Marine pollution, Noise pollution, Thermal pollution, Nuclear hazards [2 Hours]

Impact of Information Technology on Environment and Sustainable Development: Positive and Negative Impacts of IT for Environment, Mobile Phones and Cell Towers, SAR Levels, Effects of Mobile Radiations, Management and Control, IT Impact in Education-Health-Entertainment-Environment-Business-Society, National Management Information System, Environmental Information System, Geographical Information System, Functions of Remote Sensing, Human Health and Safety [4 Hours]

Part-B

Social issues and the Environment: From unsustainable to sustainable development, Water conservation, rain water harvesting, water shed management, Resettlement and rehabilitation of people; its problems and concerns, case studies, Environmental ethics: issues and possible solutions, Climate change, global warming, acid rain, ozone layer depletion, nuclear accidents and holocaust, case studies, Environment protection Act, Air (prevention and control of pollution) Act, Water (prevention and control of pollution) Act, Wildlife protection act, Forest conservation act [5 Hours]

E-Wastage and Green Computing: Impacts of E-Waste on the Environment, Harmful Effects caused by Improper Computer & Electronic Waste Recycling, Global Trade Issues, Information Security, Recycling, Repair, Electronic Waste Substances, Holistic Approaches and Techniques for Green Computing, Impacts of Green Computing, Green Awareness, Green Initiatives in Information Technology, Green Computing Certifications, Issues & Challenges Ahead [5 Hours]

Text Books

1. Textbook of Environmental studies, Erach Bharucha, UGC,2017
Weblink: <https://www.ugc.ac.in/oldpdf/modelcurriculum/env.pdf>
2. Fundamental concepts in Environmental Studies, D D Mishra, S Chand & Co Ltd, 2018
3. Environment Biology by Agarwal, K. C., Nidi Publ. Ltd. Bikaner.2001

References Books

1. Principle of Environment Science by Cunningham, W.P.2009
2. Essentials of Environment Science by Joseph.2006
3. Perspectives in Environmental Studies by Kaushik, A. 2008
4. Elements of Environment Science & Engineering by Meenakshi.2012
5. Elements of Environment Engineering by Duggal.2007

Subject Code: TRIT-101

Subject Name: Seminar and Technical Report Writing

Programme: B.Tech. (IT)	L: 0 T: 0 P: 2
Semester: 3	Teaching Hours: 24
Theory/Practical: Theory	Credits: 1
Internal Marks: 50	Percentage of Numerical/Design Problems: 100%
External Marks: Nil	Duration of End Semester Exam (ESE):
Total Marks: 50	Elective Status: Compulsory

Prerequisites: Basic writing skills

Additional Material Allowed in ESE: NIL

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	Define and agree the purpose of the report.
CO2	Have a clear understanding of the needs of your readers.
CO3	Design a document structure to effectively get your message across.
CO4	Identify the necessary content and have an appropriate layout.
CO5	Use a number of readily available tools to assist with report writing.
CO6	Edit more competently and eliminate avoidable mistakes.
CO7	Reference and quote correctly, and not infringe copyright.

Detailed Contents:

Part-A

Introduction: Structure of technical Report, Presentation, Planning the report, Writing the first draft, Revising the first draft, Diagrams, graphs, tables and mathematics, The report layout, Headings, References to diagrams, graphs, tables and equations, Originality and plagiarism, Finalising the report and proofreading, The Summary, Proofreading. **[3 Hours]**

Presentation Skills: Different ways to fight anxiety, If you don't have anything to say, If something goes wrong, If you forget something, If you make a mistake. **Voice:** Voice and eye contact, Perfect vs Passionate, Tempo and Time, Gestures, **Contact:** Facial Expression, asking

questions, Things not to do, Computer does not start, Working with slides. **Q&A : Recap**, Filtering questions, Tough questions, You're uncomfortable with the answer, Difficult situations. **[2 Hours]**

LaTeX: TeX, LaTeX, Terms regarding TeX, Custom installation with TeX Live, Tables and graphics tools, Automatic installation, Manual installation, Checking package status, External resources, The LaTeX syntax, Compilation. **[3 Hours]**

Text Formatting

Spacing, Hyphenation, Quote-marks, Diacritics and accents, Margin misalignment and interword spacing, Ligatures, Slash marks, Fonts, Formatting macros, Text mode superscript and subscript, Text figures ("old style" numerals), Dashes and hyphens, Ellipsis (...), Ready-made strings **[3 Hours]**

Part B

Paragraph Formatting

Paragraph alignment, Paragraph indent and break, \paragraph line break, Line spacing, Manual breaks, Special paragraphs **[3 Hours]**

Fonts

Introduction, Font families, Available LaTeX Fonts, emphasizing text, Font encoding, Font styles, Local font selection, Arbitrary font size, finding fonts, Using arbitrary system fonts, PDF fonts and properties, List Structures **[3 Hours]**

Tables

The *tabular* environment, Row specification, Spanning, controlling table size, Colors, Width and stretching, Table across several pages, Partial vertical lines, vertically centered images, Footnotes in tables, Professional tables, Sideways tables, Table with legend, the *eqparbox* package, Floating with table **[3 Hours]**

Floats, Figures and Captions

Floats, keeping floats in their place, Captions, lists of figures and tables, Labels and cross-referencing, Wrapping text around figures, Subfloats, Wide figures in two-column documents, Custom floats, Labels in the figures, Footnotes and Margin Notes **[3 Hours]**

Text Books

1. LaTeX, Wikibook , <http://en.wikibooks.org/wiki/LaTeX> ,en.wikibooks.org, 2016

[Accessed on 23 Feb 2019]

2. Presentation skills: Effective Presentation Delivery (Coursera)
[Accessed on 23 Feb 2019]
3. <http://www.sussex.ac.uk/ei/internal/forstudents/engineeringdesign/studyguides/techreportwriting>
[Accessed on 23 Feb 2019]
4. Frank Mittelbach , Michel Goossens, Johannes Braams,David Carlisle, Chris Rowley, “The LaTeX Companion (Tools and Techniques for Computer Typesetting)”, 2nd Edition,Addison-Wesley,2005
5. Stefan Kottwitz, “LaTeX Beginner's Guide” ,1st Edition PACKT, 2011

Reference Books:

1. Davies J.W., “Communication for Engineering Students”, Longman, 1996
2. Van Emden J., “Effective communication for Science and Technology”, Palgrave 2001
3. Van Emden J., “A Handbook of Writing for Engineers”,2nd ed. Macmillan 1998
4. Van Emden J. and Easteal J., “Technical Writing and Speaking, an Introduction”, McGraw-Hill 1996.
5. Pfeiffer W.S., “Pocket Guide to Technical Writing”, Prentice Hall 1998.
6. Eisenberg A., “Effective Technical Communication”, McGraw-Hill 1992.

E-Books and online learning material

1. AH Basson & TW von Backström ,“Guide for Writing Technical Reports”,3rd Edition, Stellenbosch University”,2007
Accessed on Feb 23, 2019
2. “Introduction to LaTeX”, http://home.iitk.ac.in/~kalpant/docs/intro_latex.pdf
Accessed on Feb 23, 2019

Online Courses and Video Lectures

1. “Technical Report Writing for engineers”, <https://www.futurelearn.com/courses/technical-report-writing-for-engineers>
Accessed on Feb 23, 2019
2. “Academic and Research Report Writing”, <https://swayam.gov.in/courses/4635-academic-and-research-report-writing>
Accessed on Feb 23, 2019

Subject Code: BSIT-101

Subject Name: Probability and Statistics

Programme: B.Tech. (IT)	L: 3 T: 0 P: 0
Semester: 4	Teaching Hours: 40
Theory/Practical: Theory	Credits: 3
Internal Marks: 40	Percentage of Numerical/Design Problems: 80%
External Marks: 60	Duration of End Semester Exam(ESE): 3hr
Total Marks: 100	Elective Status: Compulsory

Prerequisites: Mathematics-1 and Mathematics-II

Additional Material Allowed in ESE: Scientific Calculator

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	Demonstrate the measures of central tendency to analyze the given data set
CO2	Create the histogram for a given data set
CO3	Verify the predicted data sets using Binomial, Poisson and normal distribution
CO4	Select a critical value from a normal, t, chi -square, and f distribution
CO5	Predict the linear regression parameters and correlation coefficient
CO6	Apply probability theory via Bayes' Rule
CO7	Formulate the marginal and conditional distributions of bi-variate random variables

Detailed Contents:

Part-A

Introduction to Statistics: Meaning, scope, importance and limitations. Analysis of data: source of data, collection, classification, tabulation, depiction of data. Measures of Central tendency: Arithmetic, weighted, geometric mean, median and mode. Measures of Dispersion: Range, Quartile deviation, Mean deviation, Standard deviation Coefficient of variation, Skewness and Kurtosis. **[7 Hours]**

Sampling Distribution & Testing of Hypothesis: Sampling, Distribution of means and variance , Chi – Square distribution, t – distribution , F – distribution . General concepts of hypothesis, Testing a statistical Hypothesis, One and two tailed tests , critical region , Confidence interval estimation . Single and two sample tests on proportion , mean and variance .
[8 Hours]

Correlation Analysis: Significance, types, Methods of correlation analysis: Scatter diagrams, Graphic method, Karl Pearson's correlation co-efficient, Rank correlation coefficient, Properties of Correlation. Regression analysis: meaning, application of regression analysis, difference between correlation & regression analysis, regression equations, standard error and Regression coefficients. curve fitting.

[7 Hours]

Part-B

Theory of Probability: Definition, basic concepts, events and experiments, random variables, expected value, types of probability, classical approach, relative frequency and subjective approach to probability, theorems of probability, addition, Multiplication and Bays Theorem and its application. **[6 Hours]**

Probability Distributions: Difference between frequency and probability distributions, Binomial, Poisson and normal distribution **[6 Hours]**

Optimization: Matrix calculus, gradient descent, coordinates descent, introduction to convex optimization. **[6 Hours]**

Text Books

1. C. B. Gupta, "An Introduction to Statistical Methods", Vikas Publications House Pvt. Ltd. 2010
2. S. P Gupta, Sultan Chand , "Statistical Methods", Publishers Sultan Chand & Sons 2014.
3. E.Kreyszig , " Advanced Engineering Mathematics" , Wiley International Edition 2006
4. Convex Optimization by Boyd and Vandenberghe

Reference Books

1. Bali , N.P. , " A Text Book On Engineering Mathematics" , Luxmi Publications, New Delhi 2010.

2. HosseinPishro-Nik ,”Introduction to Probability, Statistics, and Random Processes”, Kappa Research, LLC , 2014
3. P. G. Hoel, S. C. Port and C. J. Stone, Introduction to Probability Theory, Universal Book Stall, 2003 (Reprint)
4. W. Feller, An Introduction to Probability Theory and its Applications, Vol. 1, 3rd Ed., Wiley, 1968.

E-Books and online learning material

1. Convex Optimization by Boyd and Vandenberghehttp://stanford.edu/~boyd/cvxbook/bv_cvxslides.pdf Accessed on Feb 21, 2019
2. Probability and Statistics by PrasannaSahoo<http://www.math.louisville.edu/~pksaho01/teaching/Math662TB-09S.pdf> Accessed on Feb 21, 2019

Online Courses and Video Lectures

1. Probability and Statistics by nptelhrd<https://nptel.ac.in/courses/111105041/> Accessed on Feb 21, 2019
2. Probability and Statistics by nptelhrd<https://www.youtube.com/watch?v=VVYLpmKRfQ8&list=PLbMVogVj5nJQrz bAweTVvnH6-vG5A4aN5> Accessed on Feb 21, 2019

Subject Code: PCIT-104

Subject Name: Database Management System

Programme: B.Tech. (IT)	L: 3 T: 0 P: 0
Semester: 4	Teaching Hours: 40
Theory/Practical: Theory	Credits: 3
Internal Marks: 40	Percentage of Numerical/Design Problems: 30%
External Marks: 60	Duration of End Semester Exam(ESE): 3hr
Total Marks: 100	Elective Status: Compulsory

Prerequisites: Knowledge of Programming for Problem Solving

Additional Material Allowed in ESE: NIL

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	Apply knowledge of database system, No Sql database, data mining and SQL structure.
CO2	Identify, formulate database design, Functional dependencies and recovery techniques.
CO3	Use the techniques, skills and tools such as query handling, normalized relations.
CO4	Design Physical and object relational database.
CO5	Investigate various case studies using NoSql.
CO6	Apply the Applications of spatial and multimedia databases for real world.

Detailed Contents:

Part-A

Introduction to Database System: Database Systems versus File Systems, View of Data, Data Models, database languages, Database Users and Administrators. Transaction Management, Components of a Database management Distributed Processing and Client Basic Concepts, Keys, Design Issues, ER Diagrams [**5 Hours**]

Relational Model: Structures of relational databases, Integrity Constraints, Logical database Design, Tables, Views, Data Dictionary. Relational Algebra, Relational Calculus. SQL – Basic Structures, Query Handling, Embedded SQL, Triggers, Security and Authorization. [10 Hours]

Relational Database Design: Functional Dependencies, Multivalued Dependencies, Normal Forms (1NF, 2NF, 3NF, BCNF, 4NF and 5NF), Decomposition into Normalized Relations. [5 Hours]

Part-B

Transaction Management and Concurrency Control: Transaction Management and Concurrency Control ACID properties, failure and recovery, concurrency control, serializability, two phase locking protocols, Timestamp and Validation based protocols, deadlocks, logs and logging protocol [5 Hours]

Recovery Systems: Failure Classification, Recovery and Atomicity, Log Based Recovery, Shadow Paging, Recovery with Concurrent Transactions [4 Hours]

NoSQL Database: Introduction to NoSQL Database, NoSQL Database Terms and Terminology, Evaluating NoSQL, Key Value Stores, Case studies of Metlife, Facebook and Google using NoSQL. [5 Hours]

Advanced Topics: Introduction to Data Mining, Process of Data Mining, Applications of Data Mining, Data Warehousing, Advantages of Data Warehousing, Data Marts, Introduction and Applications of Spatial and Multimedia Databases. [6 Hours]

Text Books:

1. Abraham Silberschatz, S. Sudarshan, Henry F. Korth, “Database System Concepts”, 6th Edition, Tata McGraw - Hill Education, 2011.
2. Shamkant B. Navathe, RamezElmasri, “Fundamentals of Database Systems”, 6th Edition, Addison Wesley Pub Co Inc, 2010.
3. Connolly, “Specifications of Database Systems : A Practical Approach to Design, Implementation and Management”, 4th Edition, Pearson India, 2008.

Reference Books:

1. Essentials of Data Base Management System - Alexis Leon and Mathews Leon - Vikas Publishing Limited, Chennai First Edition, 2009
2. SQL and PL/SQL - SharadMaheswariRuchin Jain - Firewall Media New Dehi First Edition 2010

3. Database Management Systems - Ramon a.Mato-Toledo, Pauline K.Cushman - Schaums'Outline series, TMH, New Delhi Special Indian Edition 2007
4. Data Warehousing - BPB Editorial Board - BPB Publications, New Delhi - First Indian Edition 2004, Reprinted 2008
5. Mastering Database Technologies - Ivan Bayross - BPB Publications, New Delhi - First Indian Edition 2006, Reprinted 2011
6. Database Management Systems - SharadMaheswari, Ruchin Jain - Firewall Media, New Delhi - Second Edition Reprint 2010

E-Books and online learning material

1. Database Management Systems by Ravi Jasuja
<http://www.aduni.org/courses/databases/index.php?view=cw> Accessed on Feb 20, 2019
2. Database Systems by [Hector Garcia-Molina](#), [Jeff Ullman](#), and [Jennifer Widom](#).
<http://infolab.stanford.edu/~ullman/dscb.html> Accessed on Feb 20, 2019
3. Database Management System by GaviNarra
<http://www.freebookcentre.net/database-books-download/Database-Management-System.html>
Accessed on Feb 20, 2019

Online Courses and Video Lectures

1. https://onlinecourses.nptel.ac.in/noc18_cs15/preview Accessed on Feb 20, 2019
2. <https://nptel.ac.in/courses/106105175/2> Accessed on Feb 20, 2019
3. <https://freevideolectures.com/course/2668/database-management-system>
Accessed on Feb 20, 2019
4. <https://www.studytonight.com/dbms/er-to-relational-model.php> Accessed on Feb 20, 2019
5. <https://www.studytonight.com/dbms/database-normalization.php> Accessed on Feb 20, 2019

Subject Code: PCIT-105

Subject Name: Python Programming

Programme: B.Tech. (IT)	L: 3 T: 0 P: 0
Semester: 4	Teaching Hours: 40
Theory/Practical: Theory	Credits: 3
Internal Marks: 40	Percentage of Numerical/Design Problems: 50%
External Marks: 60	Duration of End Semester Exam(ESE): 3hr
Total Marks: 100	Elective Status: Compulsory

Prerequisites: Knowledge of Programming for Problem Solving and OOPS

Additional Material Allowed in ESE: NIL

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	Familiar with Python environment, data types, operators used in Python.
CO2	Compare and contrast Python with other programming languages..
CO3	Learn the use of control structures and numerous native data types with their methods.
CO4	Design user defined functions, modules, and packages.
CO5	Investigate and implement Graphical User Interfaces based programming
CO6	Create and handle files in Python
CO7	Identify the need of object oriented programming features and implement the same to meet real time requirements.

Detailed Contents:

Part-A

Introduction to Python Programming Language:History and Origin of Python Language, Python version numbering scheme, Features of Python, Advantages and disadvantages, Major Applications of Python, Installing Python, Setting up Path and Environment Variables, Getting started with Python Programming, Python Interactive Help Feature, Detecting and Correcting syntax errors, Comparison of Python with other languages. **[5 Hours]**

Software Development, Data types, Operators and Expressions: The software Development Process, Case study: income tax Calculator, strings, Assignment and Comments, numeric Data types and Character sets, Operators in Python, Expressions, Precedence, Associativity, Non Associative Operators, using Functions and modules. **[5 Hours]**

Loops and Selection Statements: Definite Iteration: The for Loop, Formatting Text for Output, Case Study: An Investment Report, Selection: if and if-else Statements, Conditional Iteration: The while Loop, Case Study: Approximating Square Roots. **[6 Hours]**

Strings and Text Files: Accessing Characters and Sub-strings in Strings, Data Encryption, Strings and Number Systems, String Methods, Text Files, Case Study: Text Analysis. **[5 Hours]**

Part-B

Lists and Dictionaries: Lists, Defining Simple Functions, Case Study: Generating Sentences, Dictionaries, Case Study: Non-directive Psychotherapy. **[3 Hours]**

Design with Functions: Function components, Problem Solving with Top-Down Design, Design with Recursive Functions, Case Study: Gathering Information from a File System, Managing a Program's Namespace, Higher-Order Functions. **[4 Hours]**

Graphical User Interfaces: The Behavior of Terminal-Based Programs and GUI –Based Programs, Coding Simple GUI-Based Programs, Windows and Window Components, Command Buttons and Responding to Events, Input and Output with Entry Fields, Defining and Using Instance Variables, Case Study: The Guessing Game Revisited, Useful GUI Resources.

[6 Hours]

Design with Classes: Getting Inside Objects and Classes, Case Study: Playing the Game of Craps, Data-Modeling Examples, Case Study: An ATM, Building a New Data Structure: The Two-Dimensional Grid, Case Study: Data Encryption with a Block Cipher, Structuring Classes with Inheritance and Polymorphism. **[6 Hours]**

Text Books

1. Kenneth A. Lambert, Martin Osborne, “Fundamentals of Python First Programs”, 2nd Edition, Cengage Learning, 2019.
2. Kenneth A. Lambert, Martin Osborne, “Fundamentals of Python: First Programs through Data Structures”, 1st Edition, Cengage Learning, 2010.

Reference Books

1. Martin C. Brown, "Python The complete Reference", McGraw Hill Education.
2. Allen B. Downey, "Think Python", 1st Edition, Green Tea Press, 2012.

E-Books and online learning material

1. Fundamentals of Python First Programs by Kenneth A. Lambert
<https://archive.org/details/FundamentalsOfPythonFirstPrograms2ndEdition> Accessed on Feb. 20, 2019
2. Learning Python by Fabrizio Romano
<http://www.allitebooks.com/learning-python/> Accessed on Feb 20, 2019
3. Python Essential Reference by Beazley
<http://lib.myilibrary.com?id=267855> Accessed on Feb 20, 2019
4. Starting Out with Python by Gaddis
<http://lib.myilibrary.com?id=265135> Accessed on Feb 20, 2019
5. Beginning Programming with Python For Dummies by John Paul Mueller
<http://www.allitebooks.com/beginning-programming-with-python-for-dummies-2nd-edition>
Accessed on Feb 20, 2019

Online Courses and Video Lectures

1. <https://nptel.ac.in/courses/115104095> Accessed on Feb 20, 2019
2. <https://nptel.ac.in/courses/106106145> Accessed on Feb 20, 2019
3. <https://nptel.ac.in/courses/106106182> Accessed on Feb 20, 2019
4. <https://www.coursera.org/learn/python> Accessed on Feb 20, 2019
5. <https://www.coursera.org/learn/interactive-python-1> Accessed on Feb 20, 2019

Subject Code: PCIT-106

Subject Name: Operating System

Programme: B.Tech. (IT)	L: 3 T: 0 P: 0
Semester: 4 th Semester	Teaching Hours: 40
Theory/Practical: Theory	Credits: 3
Internal Marks: 40	Percentage of Numerical/Design Problems: 20%
External Marks: 60	Duration of End Semester Exam(ESE): 3hr
Total Marks: 100	Elective Status: Compulsory

Pre-requisites: Knowledge of Programming for Problem Solving and DCLD.

Additional Material Allowed in ESE: NIL

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1.	Exemplify various types of Operating Systems, deadlocks, Process, File and Memory management.
CO2.	Implement various deadlock scheduling algorithms.
CO3.	Analyze and apply various memory and file management mechanisms.
CO4.	Classify various page replacement algorithms for demand paging.
CO5.	Use different disk scheduling algorithm for better utilization of external memory.
CO6.	Examine the case studies of different Operating Systems to recapitulate the concepts of Operating System.

Detailed Contents

Part - A

Introduction: Introduction to Operating systems, Different types of operating systems - Batch, Multi-programmed, Time sharing, Real time, Distributed, Parallel. Functions of kernel and shell, General structure of Operating System, O/S services, System calls. **[5 Hours]**

Process Management: Concept of processes and threads, Process states, Process control block, Process scheduling, Scheduling Algorithms, Inter Process Communication, Process synchronization – Critical sections, Mutual Exclusion, Semaphores. **[8 Hours]**

Deadlocks: Introduction to deadlocks, Conditions for deadlock, Resource allocation graphs, Deadlock prevention and avoidance, Deadlock detection and recovery. [5 Hours]

Part - B

Memory Management: Background, Overlays, Logical versus physical address space, Memory management policies, Fragmentation types, Partitioned memory managements, Paging, Segmentation, Segmentation with paging, Need of Virtual memories, Demand Paging, Page replacement Algorithms – FIFO, Optimal, LRU. Thrashing, Cause of Thrashing, Local and Global page replacement. [9 Hours]

Secondary Storage: Disk structure, Disk scheduling – FCFS, SSTF, SCAN, C-SCAN, LOOK, CLOOK. Disk Management, Disk Formatting, Boot blocks, Bad blocks. [4 Hours]

File Management: Concept of files, File types, Access methods, File attributes, File operations, Allocation methods – Contiguous, Linked, Indexed. File System Architecture, Layered Architecture, Protection mechanisms. [5 Hours]

Case Studies: Windows, UNIX and LINUX. [4 Hours]

Text Books

1. A.Silberschatz and Peter B. Galvin, “Operating System Concepts”, Addison Wesley.
2. Dhamdhere, “Systems Programming & Operating Systems”, Tata McGraw Hill.
3. GaryNutt, “Operating Systems Concepts”, Pearson Education Ltd.

Reference Books

1. Andrews S. Tanenbaum, “Modern Operating Systems”, Pearson Education (2015) 4th Edition.
2. Pramod Chandra and P. Bhatt, “An introduction to operating systems concepts & Practices”, PHI Publication.
3. Tanenbaum A.S., “Operating System Design & Implementation”, Pearson Education.
4. Pramod Chandra and P. Bhatt, “An introduction to operating systems concepts & Practices”, Prentice Hall of India Publication.
5. Godbole, “Operating systems”, Tata McGraw Hill.

E-Books and online learning material

1. Operating System Concepts by Silberschatz, Galvin and Gagne

http://www.uobabylon.edu.iq/download/M.S%202013-2014/Operating_System_Concepts,_8th_Edition%5BA4%5D.pdf Accessed on 20.2.2019

2. Introduction to Operating System

<https://nptel.ac.in/courses/106108101/1> Accessed on 20.2.2019

Subject Code: PCIT-107
Subject Name: Web Technologies

Programme: B.Tech. (IT)	L: 3 T: 0 P: 0
Semester: 4	Teaching Hours: 40
Theory/Practical: Theory	Credits: 3
Internal Marks: 40	Percentage of Numerical/Design Problems: 40
External Marks: 60	Duration of End Semester Exam(ESE): 3 Hours
Total Marks: 100	Elective Status: Compulsory

Prerequisites: Programming for Problem Solving, Data Communication and Computer Network, Object Oriented Programming using C++.

Additional Material Allowed in ESE: NIL

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	Understand the basic tools required for Web designing and applications
CO2	Build HTML5 and CSS3 for designing interactive web pages.
CO3	Analyze the basic operations of an AJAX application
CO4	Develop an interactive website using jQuery.
CO5	Acquire the basic usage of PHP construct and its integration with database for developing web modules like, login module, session authentication
CO6	Create and design dynamic web application using contemporary development tools like, MVC framework.

Detailed Contents

Part-A

Introduction to HTML: Introduction to web browsers, HTML, XML and XHTML, basic page markup, absolute and relative links, ordered and unordered lists, embedding images and controlling appearance, table creation and use, frames, forms. **[6 Hours]**

Style Sheets: Introduction to Cascading Style Sheets(CSS), features, core syntax, separating style from structure with style sheets: internal style specifications within HTML, external linked style specification using CSS, page and site design considerations. **[5 Hours]**

JavaScript, jQuery and AJAX : Introduction to the JavaScript, Document Object Model, jQuery, including and customization of jQuery, jQuery syntax and selectors, handling events, special effects and manipulating DOM, jQuery without Selectors, plugins, using jQuery for slider design, concept of AJAX, using XMLHttpRequest, sending and receiving data using GET and POST methods. **[9 Hours]**

Part-B

HTML5 & CSS3: Introduction to HTML5, the Canvas, audio and video, forms, local storage, web workers, geolocation and GPS Services, advanced CSS with CSS3: CSS3 backgrounds, CSS flexbox, CSS3 borders, multicolumn layout, text effects, web fonts, 3D transformations, transitions, deploying HTML5 and CSS3 using bootstrap framework. **[8 Hours]**

PHP5: Introduction to PHP, basic syntax and variable declaration, expressions and control flow in PHP, PHP array, inbuilt and user defined PHP functions, creating classes, constructors and objects. assessing functions using objects, implementing inheritance, using MySQL with PHP for database related operations, advantages of PHP framework and basic knowledge about industry used frameworks, features of CodeIgniter framework . **[12 Hours]**

Text Books

1. Robin Nixon, “Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5”, O’Reilly, 4th Edition, 2015.
2. Kogent Learning Solutions Inc. “HTML5 Black Book: Covers CSS3, Javascript, XML, XHTML, AJAX, PHP and jQuery”, Dreamtech Press, 2011.
3. W. Jason Gilmore, “Beginning PHP and MySQL: From Novice to Professional”, 4th Edition, Dreamtech Press, 2010
4. Richard York, “Beginning Javascript and CSS Development with jQuery”, John Wiley and Sons Inc., 2009.
5. Audra Hendrix, “AJAX and PHP: Building Modern Web Applications”, Shroff/ Packt, 2nd Edition, 2015.

Reference Books

1. Adam Trachtenberg, “PHP Cookbook: Solutions & Examples for PHP Programmers”, 3rd Edition, Shroff/ O’reilly

2. William Sanders, “Learning PHP Design Patterns”, O’Reilly, 1st Edition, 2013
3. Jack Herrington, “PHP Hacks: Tips & Tools for creating for Dynamic Web Sites”, John C. Maxwell, 1st Edition, 2006.
4. Adam Freeman, “Pro jQuery 2.0”, Apress, 2013.
5. Nicholas C. Zakasm, Jeremy McPeak and Joe Faweett “Professional AJAX”, Wiley India Pvt. Ltd., 2nd edition, 2007.
6. Thomas Mayer, “Professional Codeigniter”, John Wiley and Sons Inc., 2008.

E-Books and online learning material

1. Tutorial on HTML,CSS,JQuery,AJAX,PHP

<https://www.w3schools.com/html/default.asp>

Accessed on Feb 22, 2019

2. Tutorial on Web Technologies

<https://www.tutorialride.com/web-technologies.htm>

Accessed on Feb 22, 2019

3. Web Tutorials (Web Development Technologies)

https://www.tutorialspoint.com/web_development_tutorials.htm

Accessed on Feb 22, 2019

Subject Code: PCIT-108

Subject Name: Computer Architecture and Microprocessors

Programme: B.Tech. (IT)	L: 3 T: 0 P: 0
Semester: 4	Teaching Hours: 40
Theory/Practical: Theory	Credits: 3
Internal Marks: 40	Percentage of Numerical/Design Problems: 20%
External Marks: 60	Duration of End Semester Exam(ESE): 3hr
Total Marks: 100	Elective Status: Compulsory

Prerequisites: Digital Circuit and Logic Design

Additional Material Allowed in ESE: NIL

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	Identify computer systems, memory organization, Microprocessor and assembly language programming
CO2	Clarify instruction formats, RISC and CISC architecture and different addressing modes
CO3	Solve basic binary math operations by using the instructions of microprocessor
CO4	Compare between pipelining and parallelism
CO5	Design structured, well commented, understandable assembly language programs to provide solutions to real-world problems
CO6	Classify the trends and developments of microprocessor technology

Detailed Contents

Part-A

Basic Computer Organization and Design: Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory Reference Instructions, I/O and Interrupt, Design of Basic Computer, Design of Accumulator Logic [**5 Hours**]

Programming the Basic Computer: Machine Language, Assembly Language, Assembler, Program Loops, Programming Arithmetic and Logic Operations, Subroutines, I/O Programming [4 Hours]

Central Processing Unit: General Register Organization, Stack Organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Program Control, Reduced Instruction Set Computer, Complex Instruction Set Computer [5 Hours]

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processors [5 Hours]

Part-B

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory, Memory Management Hardware [5 Hours]

Multiprocessors: Characteristics of Multiprocessors, Interconnection Structures, Interprocessor Arbitration, Interprocessor Communication and Synchronization, Cache Coherence [4 Hours]

Microprocessor and Microcontroller: 8051 Architecture, Pin Diagram of 8051, External Memory Interfacing, Stacks, Interfacing 8051 to LCD, Parallel and Serial ADC, DAC, Stepper Motor Interfacing and DC Motor Interfacing [8 Hours]

Applications and Trends of Microprocessor Technology: Memory and MPU Design, Development and Troubleshooting Tools, High-End-High-Performance Processors, Embedded Systems [4 Hours]

Text Books:

1. M. Morris Mano, “Computer System Architecture”, 3rd Edition, Pearson Education, 2015.
2. William Stallings, “Computer Organization and Architecture”, 9th Edition, Pearson Education, 2017.
3. Ramesh Gaonkar, “Microprocessor Architecture, Programming, and Applications with 8085”, 6th Edition, Penram International Publication, 2015.

Reference Books:

1. B. Ram, “Microprocessors and Microcomputers”, 4th Edition, Dhanpat Rai Publications, 2000.
2. K. Vani, “Computer Architecture with MIPS”, 1st Edition, Notion Press, 2017.

3. SachinKadam, “Computer Architecture and Maintenance”, 1st Edition, Packt Publishers, 2017.
4. Muhammad Ali Muzidi and Janice GillispieMuzidi, “The 8051 Microcontroller and Embedded Systems using Assembly and C”, 1st Edition, Hanel Publishers, 2017.

E-Books and online learning material

1. Computer Architecture by Wikipedia

https://en.wikipedia.org/wiki/Computer_architecture Accessed on Feb 22, 2019

2. Computer Architecture by Princeton University

<https://www.coursera.org/learn/comparch> Accessed on Feb 22, 2019

3. Computer Organization by Study Tonight

<https://www.studytonight.com/computer-architecture/architecture-of-computer-system>

Accessed on Feb 22, 2019

Online Courses and Video Lectures

1. <https://www.youtube.com/watch?v=4TzMyXmzL8M> Accessed on Feb 22, 2019

2. <https://www.youtube.com/watch?v=So9SR3qpWsM> Accessed on Feb 20, 2019

Subject Code: LPCIT-104

Subject Name: Database Management System Laboratory

Programme: B.Tech. (IT)	L: 0 T: 0 P: 2
Semester: 4th Semester	Teaching Hours: 26
Theory/Practical: Practical	Credits: 1
Internal Marks: 30	Percentage of Numerical/Design Problems: 100%
External Marks: 20	Duration of End Semester Exam(ESE): 1.5 hr
Total Marks: 50	Elective Status: Compulsory

Prerequisites: Fundamentals of Computers.

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	Apply knowledge of SQL and create tables, views.
CO2	Identify, formulate database design using single row function and group function
CO3	Displaying data from multiple tables
CO4	Design single row and multiple row sub queries, advanced sub queries
CO5	Investigate MongoDB using NoSql
CO6	Create an application of DBMS in multidisciplinary environment

Special Instruction related to resources requirement: Except practical number 10, any DBMS software like MySQL/Mariadb, Oracle etc. can be used

Sr.No. Name of Practical

1. **Creating and Managing Tables:** Create table statement; referencing another user's tables; the DEFAULT option; data types; alter table statement; adding a column; modifying a column; dropping a column; dropping a table; truncating a table.
2. **Writing Basic SQL SELECT Statements :** Basic SELECT Statement; selecting - all columns, specific columns; using arithmetic operators; operator precedence; using parenthesis; defining a NULL Value; using column aliases; concatenation operator; eliminating duplicate rows; displaying table structure.
3. **Restricting and Sorting Data:** Limiting rows using a selection; character strings and dates; comparison conditions; using the BETWEEN condition; IN condition; LIKE

condition; NULL conditions; logical conditions- AND, OR and NOT operators; rules of precedence; ORDER BY clause; sorting – ascending, descending order.

4. **Manipulating Data:** Data manipulation language; adding a new row to a table; inserting- new rows, rows with NULL values, specific date values; updating rows in a table; updating two columns; updating rows based on another table; removing a row from a table deleting rows from a table; deleting rows based on another table.
5. **Single Row Functions:** Character functions - case manipulation and character manipulation functions; number functions, date functions; using arithmetic operators with dates; date functions, conversion functions.
6. **Displaying Data from Multiple Tables:** Cartesian products; different types of joins specific to the software package; SQL compliant joins
7. **Aggregating Data Using Group Functions:** Group functions for various statistical metrics; creating groups of data by GROUP BY clause; grouping by more than one column; excluding group results- HAVING Clause.
8. **Subqueries:** Single-row subqueries; multiple-row subqueries; using group function in a subquery; HAVING clause with subqueries; usage of operators in multiple-row subqueries
9. **Creating Views:** Simple views and complex views; creating a view; retrieving data from view; querying a view; modifying a view; removing a view; inline views.
10. **Overview of MongoDB: A NoSQL database:** Create and drop-database, collection; data types; insert document; query document; logical operators; update document; delete document; projection; limit records; sort documents.
11. **Mini Project** By using standard database design rules, database has to be designed for a specific assigned problem to a group of two to three students. ER diagram related to project with an open source database tool like MySQL workbench must also be prepared. The group of students must submit a project report of 8 to 10 pages (approximately) and the team will have to demonstrate as well as have to give a presentation of the same.

Reference Books:

1. Introduction to Oracle 9i:SQL by Nancy Greenberg and Priya Nathan , publisher Sheryl Dominigue edition June 2001.
2. SQL, PL/SQL: The Programming Language of Oracle by Ivan Bayross , publisher BPB, edition December 2010

Subject Code: LPCIT-105

Subject Name: Python Programming Laboratory

Programme: B.Tech. (IT)	L: 0 T: 0 P: 2
Semester: 4	Teaching Hours: 26
Theory/Practical: Practical	Credits: 1
Internal Marks: 30	Percentage of Numerical/Design Problems: 100%
External Marks: 20	Duration of End Semester Exam(ESE): 1.5 hr
Total Marks: 50	Elective Status: Compulsory

Prerequisites: Basic knowledge of programming in C, C++.

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	Apply knowledge of data types, operators and expression.
CO2	Identify and implement the concept of strings, loops, selection statements.
CO3	Formulate and Identify the use of list and dictionaries.
CO4	Able to solve the problem using the concept functions.
CO5	Investigate and implement Graphical User Interfaces programs
CO6	Create an application using classes for multidisciplinary fields

Special Instruction related to resources requirement: The students need to implement any three projects from each of the below mentioned topics.

Data types, Operators and Expressions: (Do any three projects from the following)

In each of the projects that follow, you should write a program that contains an introductory docstring. This documentation should describe what the program will do (analysis) and how it will do it (design the program in the form of a pseudo code algorithm). Include suitable prompts for all inputs, and label all outputs appropriately. After you have coded a program, be sure to test it with a reasonable set of legitimate inputs.

1. The tax calculator program of the case study outputs a floating-point number that might show more than two digits of precision. Use the **round** function to modify the program to display at most two digits of precision in the output number.

2. You can calculate the surface area of a cube if you know the length of an edge. Write a program that takes the length of an edge (an integer) as input and prints the cube's surface area as output.
3. Five Star Retro Video rents VHS tapes and DVDs to the same connoisseurs who like to buy LP record albums. The store rents new videos for \$3.00 a night, and oldies for \$2.00 a night. Write a program that the clerks at Five Star Retro Video can use to calculate the total charge for a customer's video rentals. The program should prompt the user for the number of each type of video and output the total
4. Write a program that takes the radius of a sphere (a floating-point number) as input and then outputs the sphere's diameter, circumference, surface area, and volume.
5. An object's momentum is its mass multiplied by its velocity. Write a program that accepts an object's mass (in kilograms) and velocity (in meters per second) as inputs and then outputs its momentum.
6. The kinetic energy of a moving object is given by the formula $KE = (1 / 2)mv^2$ where m is the object's mass and v is its velocity. Modify the program you created in Project 5 so that it prints the object's kinetic energy as well as its momentum.
7. Write a program that calculates and prints the number of minutes in a year.
8. Light travels at $3 * 10^8$ meters per second. A light-year is the distance a light beam travels in one year.
9. An employee's total weekly pay equals the hourly wage multiplied by the total number of regular hours plus any overtime pay. Overtime pay equals the total overtime hours multiplied by 1.5 times the hourly wage. Write a program that takes as inputs the hourly wage, total regular hours, and total overtime hours and displays an employee's total weekly pay.
10. Write a program that calculates and displays the value of a light-year. Write a program that takes as input a number of kilometers and prints the corresponding number of nautical miles. Use the following approximations:
 - A kilometer represents 1/10,000 of the distance between the North Pole and the equator.
 - There are 90 degrees, containing 60 minutes of arc each, between the North Pole and the equator.

- A nautical mile is 1 minute of an arc.

Loops and Selection Statements :(Do any three projects from the following)

1. Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is an equilateral triangle.
2. Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is a right triangle. Recall from the Pythagorean theorem that in a right triangle, the square of one side equals the sum of the squares of the other two sides.
3. A standard science experiment is to drop a ball and see how high it bounces. Once the “bounciness” of the ball has been determined, the ratio gives a bounciness index. For example, if a ball dropped from a height of 10 feet bounces 6 feet high, the index is 0.6, and the total distance traveled by the ball is 16 feet after one bounce. If the ball were to continue bouncing, the distance after two bounces would be $10\text{ ft} + 6\text{ ft} + 6\text{ ft} + 3.6\text{ ft} = 25.6\text{ ft}$. Note that the distance traveled for each successive bounce is the distance to the floor plus 0.6 of that distance as the ball comes back up. Write a program that lets the user enter the initial height from which the ball is dropped and the number of times the ball is allowed to continue bouncing. Output should be the total distance traveled by the ball.
4. A local biologist needs a program to predict population growth. The inputs would be the initial number of organisms, the rate of growth (a real number greater than 0), the number of hours it takes to achieve this rate, and a number of hours during which the population grows. For example, one might start with a population of 500 organisms, a growth rate of 2, and a growth period to achieve this rate of 6 hours. Assuming that none of the organisms die, this would imply that this population would double in size every 6 hours. Thus, after allowing 6 hours for growth, we would have 1000 organisms, and after 12 hours, we would have 2000 organisms. Write a program that takes these inputs and displays a prediction of the total population.
5. The German mathematician Gottfried Leibniz developed the following method to approximate the value of π : $\pi/4 = 1 - 1/3 + 1/5 - 1/7 + \dots$

Write a program that allows the user to specify the number of iterations use in this approximation and that displays the resulting value.

6. Teachers in most school districts are paid on a schedule that provides a salary based on their number of years of teaching experience. For example, a beginning teacher in the Lexington School District might be paid \$30,000 the first year. For each year of experience after this first year, up to 10 years, the teacher receives a 2% increase over the preceding value. Write a program that displays a salary schedule, in tabular format, for teachers in a school district. The inputs are the starting salary, the percentage increase, and the number of years in the schedule. Each row in the schedule should contain the year number and the salary for that year.
7. The greatest common divisor of two positive integers, A and B, is the largest number that can be evenly divided into both of them. Euclid's algorithm can be used to find the greatest common divisor (GCD) of two positive integers. You can use this algorithm in the following manner:
 - a. Compute the remainder of dividing the larger number by the smaller number.
 - b. Replace the larger number with the smaller number and the smaller number with the remainder.
 - c. Repeat this process until the smaller number is zero.

The larger number at this point is the GCD of A and B. Write a program that lets the user enter two integers and then prints each step in the process of using the Euclidean algorithm to find their GCD.

8. Write a program that receives a series of numbers from the user and allows the user to press the enter key to indicate that he or she is finished providing inputs. After the user presses the enter key, the program should print the sum of the numbers and their average.
9. The credit plan at TidBit Computer Store specifies a 10% down payment and an annual interest rate of 12%. Monthly payments are 5% of the listed purchase price, minus the down payment. Write a program that takes the purchase price as input. The program should display a table, with appropriate headers, of a payment schedule for the lifetime of the loan. Each row of the table should contain the following items:
 - the month number (beginning with 1)

- the current total balance owed
- the interest owed for that month
- the amount of principal owed for that month
- the payment for that month
- the balance remaining after payment

The amount of interest for a month is equal to $\text{balance} * \text{rate} / 12$. The amount of principal for a month is equal to the monthly payment minus the interest owed.

10. In the game of Lucky Sevens, the player rolls a pair of dice. If the dots add up to 7, the player wins \$4; otherwise, the player loses \$1. Suppose that, to entice the gullible, a casino tells players that there are lots of ways to win: (1, 6), (2, 5), and so on. A little mathematical analysis reveals that there are not enough ways to win to make the game worthwhile; however, because many people's eyes glaze over at the first mention of mathematics, your challenge is to write a program that demonstrates the futility of playing the game. Your program should take as input the amount of money that the player wants to put into the pot, and play the game until the pot is empty. At that point, the program should print the number of rolls it took to break the player, as well as maximum amount of money in the pot.

Strings and Text Files:(Do any three projects from the following)

1. Write a script that inputs a line of plaintext and a distance value and outputs an encrypted text using a Caesar cipher. The script should work for any printable characters.
2. Write a script that inputs a line of encrypted text and a distance value and outputs plaintext using a Caesar cipher. The script should work for any printable characters.
3. Modify the scripts of Projects 1 and 2 to encrypt and decrypt entire files of text.
4. A bit shift is a procedure whereby the bits in a bit string are moved to the left or to the right. For example, we can shift the bits in the string 1011 two places to the left to produce the string 1110. Note that the leftmost two bits are wrapped around to the right side of the string in this operation. Define two scripts, **shiftLeft.py** and **shiftRight.py**, that expect a bit string as an input. The script **shiftLeft** shifts the bits in its input one

place to the left, wrapping the leftmost bit to the rightmost position. The script **shiftRight** performs the inverse operation. Each script prints the resulting string.

5. Write a script named **copyfile.py**. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.
6. Write a script named **numberlines.py**. This script creates a program listing from a source program. This script should prompt the user for the names of two files. The input filename could be the name of the script itself, but be careful to use a different output filename! The script copies the lines of text from the input file to the output file, numbering each line as it goes. The line numbers should be right-justified in 4 columns, so that the format of a line in the output file looks like this example:

1>This is the first line of text.

7. Write a script named **dif.py**. This script should prompt the user for the names of two text files and compare the contents of the two files to see if they are the same. If they are, the script should simply output "**Yes**". If they are not, the script should output "**No**", followed by the first lines of each file that differ from each other. The input loop should read and compare lines from each file. The loop should break as soon as a pair of different lines is found.
8. Jack just completed the program for the Flesch text analysis from this chapter's case study. His supervisor, Jill, has discovered an error in his code. The error causes the program to count a syllable containing consecutive vowels as multiple syllables. Suggest a solution to this problem in Jack's code and modify the program so that it handles these cases correctly.
9. The Payroll Department keeps a list of employee information for each pay period in a text file. The format of each line of the file is the following:

<last name><hourly wage><hours worked>

Write a program that inputs a filename from the user and prints to the terminal a report of the wages paid to the employees for the given period. The report should be in tabular

format with the appropriate header. Each line should contain an employee's name, the hours worked, and the wages paid for that period.

Lists and Dictionaries :(Do any three projects from the following)

1. Write a program that allows the user to navigate the lines of text in a file. The program should prompt the user for a filename and input the lines of text into a list. The program then enters a loop in which it prints the number of lines in the file and prompts the user for a line number. Actual line numbers range from 1 to the number of lines in the file. If the input is 0, the program quits. Otherwise, the program prints the line associated with that number.
2. Make the following modifications to the original sentence-generator program:
 - a. The prepositional phrase is optional. (It can appear with a certain probability.)
 - b. A conjunction and a second independent clause are optional: The boy took a drink and the girl played baseball.
 - c. An adjective is optional: The girl kicked the red ball with a sore foot.You should add new variables for the sets of adjectives and conjunction
3. Define a function **decimalToRept** that returns the representation of an integer in a given base. The two arguments should be the integer and the base. The function should return a string. It should use a lookup table that associates integers with digits. Include a main function that tests the conversion function with numbers in several bases.
4. Write a program that inputs a text file. The program should print the unique words in the file in alphabetical order
5. A file concordance tracks the unique words in a file and their frequencies. Write a program that displays a concordance for a file. The program should output the unique words and their frequencies in alphabetical order. Variations are to track sequences of two words and their frequencies, or n words and their frequencies.
6. Conversations often shift focus to earlier topics. Modify the therapist program to support this capability. Add each patient input to a history list. Then, occasionally choose an element at random from this list, change persons, and prepend (add at the beginning) the qualifier "Earlier you said that" to this reply. Make sure that this option is triggered only after several exchanges have occurred.

Design with Functions: (Do any three projects from the following)

1. A list is sorted in ascending order if it is empty or each item except the last one is less than or equal to its successor. Define a predicate `isSorted` that expects a list as an argument and returns `True` if the list is sorted, or returns `False` otherwise.
(Hint: For a list of length 2 or greater, loop through the list and compare pairs of items, from left to right, and return `False` if the first item in a pair is greater.)
2. Add a command to this chapter's case study program that allows the user to view the contents of a file in the current working directory. When the command is selected, the program should display a list of filenames and a prompt for the name of the file to be viewed. Be sure to include error recovery.
3. Write a recursive function that expects a pathname as an argument. The pathname can be either the name of a file or the name of a directory. If the pathname refers to a file, its name is displayed, followed by its contents. Otherwise, if the pathname refers to a directory, the function is applied to each name in the directory. Test this function in a new program.
4. Write a program that computes and prints the average of the numbers in a text file. You should make use of two higher-order functions to simplify the design.
5. Define and test a function `myRange`. This function should behave like Python's standard `range` function, with the required and optional arguments, but it should return a list. Do not use the `range` function in your implementation! (Hints: Study Python's help on `range` to determine the names, positions, and what to do with your function's parameters. Use a default value of `None` for the two optional parameters. If these parameters both equal `None`, then the function has been called with just the stop value. If just the third parameter equals `None`, then the function has been called with a start value as well. Thus, the first part of the function's code establishes what the values of the parameters are or should be. The rest of the code uses those values to build a list by counting up or down.)
6. Lee has discovered what he thinks is a clever recursive strategy for printing the elements in a sequence (string, tuple, or list). He reasons that he can get at the first element in a sequence using the 0 index, and he can obtain a sequence of the rest of the elements by

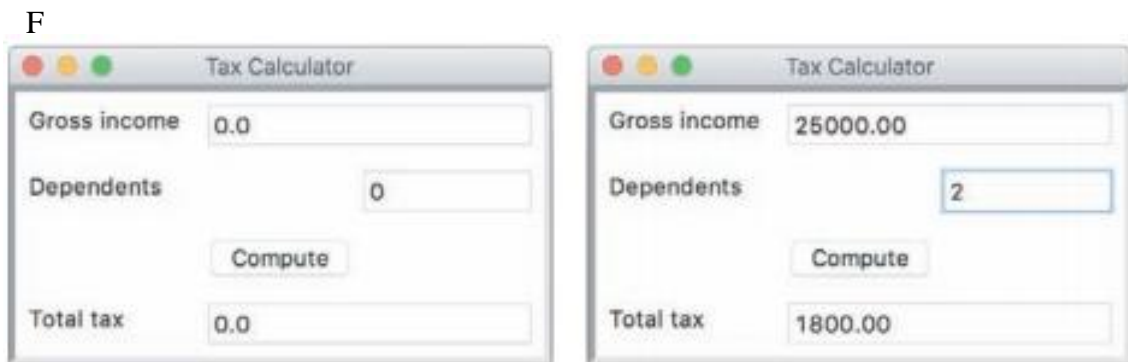
slicing from index 1. This strategy is realized in a function that expects just the sequence as an argument. If the sequence is not empty, the first element in the sequence is printed and then a recursive call is executed. On each recursive call, the sequence argument is sliced using the range 1:.. Here is Lee's function definition:

```
def printAll(seq):
    if seq:
        print(seq[0])
        printAll(seq[1:])
```

Write a script that tests this function and add code to trace the argument on each call. Does this function work as expected? If so, explain how it actually works, and describe any hidden costs in running it.

Graphical User Interfaces: (Do any three projects from the following)

1. Write GUI-based program that implements the tax calculator program shown in Figure 1.



1

2. Write a GUI-based program that allows the user to convert temperature values between degrees Fahrenheit and degrees Celsius. The interface should have labeled entry fields for these two values. These components should be arranged in a grid where the labels occupy the first row and the corresponding fields occupy the second row. At start-up, the Fahrenheit field should contain 32.0, and the Celsius field should contain 0.0. The third row in the window contains two command buttons, labeled >>>> and <<<<. When the user presses the first button, the program should use the data in the Fahrenheit field to

compute the Celsius value, which should then be output to the Celsius field. The second button should perform the inverse function

3. Modify the temperature conversion program so that it responds to the user's press of the return or enter key. If the user presses this key when the insertion point is in a given field, the action which uses that field for input is triggered
4. Write a GUI-based program that plays a guess-the-number game in which the roles of the computer and the user are the reverse of what they are in the Case Study of this chapter. In this version of the game, the computer guesses a number between 1 and 100 and the user provides the responses. The window should display the computer's guesses with a label. The user enters a hint in response, by selecting one of a set of command buttons labeled Too small, Too large, and Correct. When the game is over, you should disable these buttons and wait for the user to click New game, as before.
5. Add radio button options for filing status to the tax calculator program of Project 1. The user selects one of these options to determine the tax rate. The Single option's rate is 20%. The Married option is 15%. The Divorced option is 10%. The default option is Single.
6. Write a GUI-based program that allows the user to open, edit, and save text files. The GUI should include a labeled entry field for the filename and a multiline text widget for the text of the file. The user should be able to scroll through the text by manipulating a vertical scrollbar. Include command buttons labeled **Open**, **Save**, and **New** that allow the user to open, save, and create new files. The **New** command should then clear the text widget and the entry widget.
7. Write a GUI-based program that implements an image browser for your computer's file system. The look, feel, and program logic should be like those of the simple text file browser developed in this chapter. The file dialog should filter for GIF image files, and create and open a **PhotoImage** when a file is accessed.

Design with Classes:(Do any three projects from the following)

1. Add three methods to the **Student** class that compare two **Student** objects. One method should test for equality. A second method should test for less than. The third method should test for greater than or equal to. In each case, the method returns the result of the

comparison of the two students' names. Include a main function that tests all of the comparison operators.

2. This project assumes that you have completed Project 1. Place several **Student** objects into a list and shuffle it. Then run the **sort** method with this list and display all of the students' information.
3. The **str** method of the **Bank** class returns a string containing the accounts in random order. Design and implement a change that causes the accounts to be placed in the string by order of name. (Hint: You will also have to define some methods in the **SavingsAccount** class.)
4. The ATM program allows a user an indefinite number of attempts to log in. Fix the program so that it displays a popup message that the police will be called after a user has had three successive failures. The program should also disable the login button when this happens.
5. The **play** method in the **Player** class of the craps game plays an entire game without interaction with the user. Revise the **Player** class so that its user can make individual rolls of the dice and view the results after each roll. The **Player** class no longer accumulates a list of rolls, but saves the string representation of each roll after it is made. Add new methods **rollDice**, **getRollsCount**, **isWinner**, and **isLoser** to the **Player** class. The last three methods allow the user to obtain the number of rolls and to determine whether there is a winner or a loser. The last two methods are associated with new Boolean instance variables. Two other instance variables track the number of rolls and the string representation of the most recent roll. Another instance variable tracks whether or not the first roll has occurred. At instantiation, the **roll**, **rollsCount**, **atStartup**, **winner**, and **loser** variables are set to their appropriate initial values. All game logic is now in the **rollDice** method. This method rolls the dice once, updates the state of the Player object, and returns a tuple of the values of the dice for that roll. Include in the module the **playOneGame** and **playManyGames** functions, suitably updated for the new interface to the **Player** class.
6. In many card games, cards are either face up or face down. Add a new instance variable named **faceup** to the **Card** class to track this attribute of a card. Its default value is **False**.

Then add a **turn** method to turn the card over. This method resets the **faceup** variable to its logical negation.

- Convert the **DiceDemo** program discussed in this chapter to a completed craps game application, using the **Player** data model class you developed in Project 6. A screen shot of a possible window is shown in Figure 2.

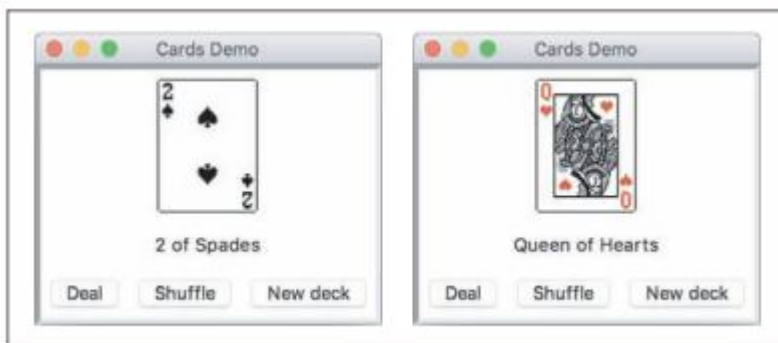


Figure 1.



Figure 2.

- Computer card games are more fun if you can see the images of the cards in a window, as shown in the screen shot in Figure 3. Assume that the 52 images for a deck of cards are in a DECK folder, with the file naming scheme **<rank number><suit letter>.gif**. Thus, for example, the image for the Ace of Hearts is in a file named **1h.gif**, and the image for the King of Spades is in a file named **13s.gif**. Furthermore, there is an image file named **b.gif** for the backside image of all the cards. This will be the card's image if its faceup variable is **False**. Using the **DiceDemo** program as a role model, write a GUI program that allows you to deal and view cards from a deck. Be sure to define a helper method that takes a **Card** object as an argument and returns its associated image, and remember to turn the cards as you deal them.
- Geometric shapes can be modeled as classes. Develop classes for line segments, circles, and rectangles. Each shape object should contain a **Turtle** object and a color that allow the shape to be drawn in a Turtle graphics window . Factor the code for these features

(instance variables and methods) into an abstract **Shape** class. The **Circle**, **Rectangle**, and **Line** classes are all subclasses of **Shape**. These subclasses include other information about the specific types of shapes, such as a radius or a corner point and a **draw** method. Write a script that uses several instances of the different shape classes to draw a house and a stick figure.

Reference Books:

1. Kenneth A. Lambert, Martin Osborne, “Fundamentals of Python First Programs”, 2nd Edition, Cengage Learning, 2019.
2. Kenneth A. Lambert, Martin Osborne, “Fundamentals of Python: First Programs through Data Structures”, 1st Edition, Cengage Learning, 2010.

Subject Code: LPCIT-106

Subject Name: Operating System and Microprocessors Lab

Programme: B.Tech. (IT)	L: 0 T: 0 P: 2
Semester: 4	Teaching Hours: 26
Theory/Practical: Practical	Credits: 1
Internal Marks: 30	Percentage of Numerical/Design Problems: 100%
External Marks: 20	Duration of End Semester Exam(ESE): 2 hr
Total Marks: 50	Elective Status: Compulsory

Prerequisite: Fundamentals of Computer with any basic programming language

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	Carry out the installation of UNIX, LINUX and Windows Operating System.
CO2	Analyze the concept of Virtual Machine and install Operating System through it.
CO3	Implement shell programming and using vi editor.
CO4	Solve basic binary math operations using the instructions of microprocessor
CO5	Construct code and debugs Assembly Language programs to implement simple programs.
CO6	Perform various operations using the microcontroller.

Special Instruction related to resources requirement: For practical numbers 7-11, microprocessors/ GNUSim8085 software etc. can be used. For practical numbers 12 EdSim51 software or other microcontroller etc. can be used.

Sr

No.

Name of Practical

1. Installation Process of various Operating Systems.
2. Virtualization, Installation of Virtual Machine Software and installation of Operating System on Virtual Machine.
3. Basics of Shell programming and various types of shells.
4. Implementation of shell variables and shell keywords.
5. Implement conditional statements, looping statement and case statement in Shell programming.
6. Study and usage of vi Editor.
7. To learn and Perform addition operation for two 8-bit; sum is 8-bit and 16 bit respectively
8. Perform subtraction operation of two 8-bit and 16-bit numbers.

9. Find 1's complement of 8 bit and 16 bit number.
10. Find 2's complement of 8 bit and 16 bit number.
11. Find sum of series of 8 bit numbers
12. Addition and subtraction of 8-bit number using 8051

Reference Books:

1. Das Sumithbha, "UNIX CONCEPTS AND APPLICATIONS", 4th Edition, Tata McGraw Hill, 2017.
2. Ramesh Gaonkar, "Microprocessor Architecture, Programming, and Applications with 8085", 6th Edition, Penram International Publication, 2015.
3. Muhammad Ali Muzidi and Janice Gillispie Muzidi, "The 8051 Microcontroller and Embedded Systems using Assembly and C", 1st Edition, Hanel Publishers, 2017.

Subject Code: LPCIT-107

Subject Name: Web Technologies Laboratory

Programme: B.Tech. (IT)	L: 0 T: 0 P: 2
Semester: 4	Teaching Hours: 26
Theory/Practical: Practical	Credits: 1
Internal Marks: 30	Percentage of Numerical/Design Problems: 100%
External Marks: 20	Duration of End Semester Exam(ESE): 1.5 Hours
Total Marks: 50	Elective Status: Compulsory

Prerequisites: Programming for Problem Solving, Data Communication and Computer Network, Object Oriented Programming using C++.

On Completion of the course, the student will have the ability to:

CO#	Course Outcome
CO1	Design and Construct Static Web Pages using Basic HTML Tags and CSS for varied application areas.
CO2	Implement HTML5 and CSS3 using Bootstrap Framework.
CO3	Creating Web pages integrating with database for developing web modules like, login module, session authentication.
CO4	Use CodeIgniter Framework to create a unique theme and/or child theme.
CO5	Create and design dynamic web application using contemporary development tools like, MVC framework.
CO6	Function on diverse teams to identify, formulate and design web based project using PHP script in multi-disciplinary environment

Special Instruction related to resources requirement: Use of Open Source Software is recommended.

Sr.No. Name of Practical

1. Create a simple web page by writing HTML using a simple text editor, Notepad.

Demonstrate the following components of the web page:

Page titles and Headings

Paragraphs and Inline images

****Note: Validation of your HTML can be done at: <http://validator.w3.org/>

2. Demonstrate the use of Links, Lists and Tables in HTML. You should be able to link

separate pages and create named links within a document, using them to build a “table of contents”.

3. Create simple Forms in HTML and demonstrate the use of various form elements like input box, textarea, submit and radio buttons etc.
4. Demonstrate the use of cascading style sheets (CSS) (inline, internal and external) to specify various aspects of style, such as colours and text fonts and sizes, in HTML document.
5. Create an html file to implement the concept of document object model, different operations and event handling using JavaScript.
6. Demonstrate the use of various selectors, filters and event handling in jQuery.
7. Demonstrate the use of AJAX to retrieve and manipulate the web page content
8. Demonstrate the use of GET and POST methods of AJAX.
9. Creation of Web pages using HTML5 and CSS3.
10. Demonstrate the use of Bootstrap Framework.
11. Setup of development server like XAMP/ WAMP in Windows and Linux.
12. Creating web pages using PHP.
13. Handling database queries with PHP.
14. Setup of CodeIgniter framework and to study its different components.
15. **Mini Project:** Students are required to prepare a web based project on the course contents in a group of two to three students. The usage of concepts like authentication, sessions and database oriented web pages for the development of the project is recommended. The group of students must submit a project report of 8 to 10 pages (approximately) and the team will have to demonstrate the project as well as have to give a presentation of the same.

Text Books

1. Robin Nixon, “Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5”, O’Reilly, 4th Edition, 2015.
2. Kogent Learning Solutions Inc. “HTML5 Black Book: Covers CSS3, Javascript, XML, XHTML, AJAX, PHP and jQuery”, Dreamtech Press, 2011.

3. W. Jason Gilmore, “Beginning PHP and MySQL: From Novice to Professional”, 4th Edition, Dreamtech Press, 2010
4. Richard York, “Beginning Javascript and CSS Development with jQuery”, John Wiley and Sons Inc., 2009.
5. Audra Hendrix, “AJAX and PHP: Building Modern Web Applications”, Shroff/ Packt, 2nd Edition, 2015.

Reference Books

1. Adam Trachtenberg, “PHP Cookbook: Solutions & Examples for PHP Programmers”, 3rd Edition, Shroff/ O’reilly
2. William Sanders, “Learning PHP Design Patterns”, O’Reilly, 1st Edition, 2013
3. Jack Herrington, “PHP Hacks: Tips & Tools for creating for Dynamic Web Sites”, John C. Maxwell, 1st Edition, 2006.
4. Adam Freeman, “Pro jQuery 2.0”, Apress, 2013.
5. Nicholas C. Zakasm, Jeremy McPeak and Joe Faweett “Professional AJAX”, Wiley India Pvt. Ltd., 2nd edition, 2007.
6. Thomas Mayer, “Professional Codeigniter”, John Wiley and Sons Inc., 2008.
7. Brad Williams, David Damstra and Hal Stern, “Professional Wordpress: Design and Development”, Wiley India Private Ltd., 2nd edition, 2013.

E-Books and online learning material

1. Tutorial on HTML,CSS,JQuery,AJAX,PHP

<https://www.w3schools.com/html/default.asp>

Accessed on Feb 22, 2019

2. Tutorial on Web Technologies

<https://www.tutorialride.com/web-technologies.htm>

Accessed on Feb 22, 2019

3. Web Tutorials (Web Development Technologies)

https://www.tutorialspoint.com/web_development_tutorials.htm

Accessed on Feb 22, 2019