Guru Nanak Dev Engineering College, Ludhiana

Department of Computer Science & Engineering

Scheme and Syllabus

B. Tech. Computer Science & Engineering

(2018 Batch Onwards)

Course: B.Tech. Computer Science and Engineering

3rd Semester

Category	Course Code	Course Name	Subject Type (Theory / Practical)	Load Allocation (per week)			Marks Distribution		Total Marks	Credits
				L	T	P	Internal	External		
Professional Core Courses	PCCS-101	Object Oriented Programming	Theory	3	0	0	40	60	100	3
Professional Core Courses	PCCS-102	Computer Networks	Theory	3	0	0	40	60	100	3
Engineering Science Courses	ESCS-101	Digital Electronics	Theory	3	0	0	40	60	100	3
Basic Science Course	BSCS-101	Mathematics-III	Theory	3	1	0	40	60	100	4
Humanities and Social Sciences including Management Courses	HSMCS-101	Human values and Professional Ethics	Theory	3	0	0	40	60	100	3
Professional Core Courses	LPCCS-101	Object Oriented Programming Laboratory	Practical	0	0	4	30	20	50	2
Professional Core Courses	LPCCS-102	Computer Networks Laboratory	Practical	0	0	2	30	20	50	1
Engineering Science Courses	LESCS-101	Digital Electronics Laboratory	Practical	0	0	2	30	20	50	1
Training	TR-101	Training-I	Practical	0	0	0	60	40	100	1
Seminar/Project	PRCS-101	Seminar and Technical Report Writing	Practical	0	0	2	50	0	50	1
Mentoring and Professional Development #		Mentoring and Professional Development	Practical	0	0	1	-	-	-	-
Total				**15**	**1**	**11**	**400**	**400**	**800**	**22**
			Contact Hours = 26+1#							

*Evaluation of 4 weeks institutional/industrial training held after 2nd semester in the institute.

# There will be one period per week for Mentoring and Professional Development; final evaluation of this course will be done based on the combined assessment of odd and even semester of respective year of study.

4th Semester

Category	Course Code	Course Name	Subject Type (Theory / Practical)	Load Allocation (per week)			Marks Distribution		Total Marks	Credits
				L	T	P	Internal	External		
Professional Core Courses	PCCS-103	Discrete Mathematics	Theory	3	1	0	40	60	100	4
Professional Core Courses	PCCS-104	Computer Architecture and Microprocessor	Theory	3	0	0	40	60	100	3
Professional Core Courses	PCCS-105	Operating Systems	Theory	3	1	0	40	60	100	4
Professional Core Courses	PCCS-106	Data Structures	Theory	3	0	0	40	60	100	3
Professional Core Courses	PCCS-107	Software Engineering	Theory	3	1	0	40	60	100	4
Professional Core Courses	LPCCS-103	Computer Architecture and Microprocessor Laboratory	Practical	0	0	2	30	20	50	1
Professional Core Courses	LPCCS-104	Operating Systems Laboratory	Practical	0	0	2	30	20	50	1
Professional Core Courses	LPCCS-105	Data Structures Laboratory	Practical	0	0	4	30	20	50	2
Mandatory Courses	MCI-101	Environmental Sciences	Theory	2	0	0	50	0	50	0
Mentoring and Professional Development #	MPD-102	Mentoring and Professional Development	Practical	0	0	1	100	0	100	1
Total				**17**	**3**	**9**	**440**	**360**	**800**	**23**
			Contact Hours = 28+1#							

# There will be one period per week for Mentoring and Professional Development; final evaluation of this course will be done based on the combined assessment of odd and even semester of respective year of study.

Subject Code: PCCS-101

Subject Name: Object Oriented Programming

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

Additional Material Allowed in ESE: [NIL]

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

CO#	Course Outcomes (CO)
1	Develop an understanding of object oriented programming principles and object oriented design.
2	Use of operators, control structures, and data types with their methods.
3	Make use of arrays and string handling methods.
4	Design user defined functions, modules, and packages.
5	Investigate and implement polymorphism, inheritance, dynamic memory management and exception handling techniques to solve problems.
6	Create and handle files in object oriented programming.

Detailed Contents:

Part-A

Object-Oriented Programming Concepts: Introduction, Comparison between procedural programming paradigm and object-oriented programming paradigm, Features of object-oriented programming: Encapsulation, Class, Object, Abstraction, Data hiding, polymorphism, and Inheritance. Introduction of object oriented design. **[3 Hours]**

Data Types, Operators, and Control Structures: Basic data types, Derived data types, Keywords, Identifiers, Constants and variables, Type casting, Operators, and Operator precedence. Control Structures: if statement, switch-case, for, while and do-while loops, break and continue statement. **[6 Hours]**

Classes and Objects: Implementation of a class, Creating class objects, Operations on objects, Relationship among objects, Accessing class members, Access specifiers, Constructor and destructor, Types of constructor, Static members, Empty classes, Nested classes, Local classes, Abstract classes, Container classes. **[5 Hours]**

Functions, Arrays, and String Handling: Function components, Default arguments, Passing parameters, Function prototyping, Call by value, Call by reference, Return by reference, Inline functions, Friend functions, Static functions, Recursion, Array declaration, Types of arrays, Array of objects, String handling. **[6 Hours]**

Part-B

Polymorphism and Type Conversion: Introduction, Concept of binding – Early binding and late binding, Virtual functions, Pure virtual functions, Operator Overloading, Rules for overloading operators, Overloading of various operators, Function overloading, Constructor overloading, Type conversion – Basic type to class type, Class type to basic type, Class type to another class type. **[6 Hours]**

Inheritance: Introduction, defining derived classes, Types of inheritance, Ambiguity in multiple and multipath inheritance, Virtual base class, Objects slicing, Overriding member functions, Object composition and delegation. **[5 Hours]**

Dynamic Memory Management using Pointers: Declaring and initializing pointers, Accessing data through pointers, Pointer arithmetic, Memory allocation –Static and Dynamic, Dynamic memory management using new and delete operators, Pointer to an object, this pointer, Pointer related problems – Dangling/wild pointers, Null pointer assignment, Memory leak and Allocation failures. **[5 Hours]**

Exceptions Handling: Review of traditional error handling, Basics of exception handling, Exception handling mechanism, Throwing mechanism, Catching mechanism, Rethrowing an exception, Specifying exceptions. **[2 Hours]**

Files Handling: File streams, Hierarchy of file stream classes, Error handling during file operations, Reading/writing of files, Accessing records randomly, Updating files. **[2 Hours]**

Text Books

1. R. Lafore, “Object Oriented Programming in C++”, Waite Group.
2. E. Balagurusamy, “Object Oriented Programming with C++”, Tata McGraw Hill.
3. P YashavantKanetkar, “Let Us C++”, BPB Publications.
4. Bjarne Stroustrup, “The C++ Programming Language”, Addison Wesley.

Reference Books

1. Herbert Schildt, “The Complete Reference to C++ Language”, McGraw Hill-Osborne.
2. B.F.Lippman, “C++ Primer”, Addison Wesley.
3. Farrell, “Object Oriented using C++”, Cengage Learning.
4. Barbara Liskov, Program Development in Java, Addison-Wesley.

E-Books and online learning material

1. E. Balagurusamy, “Object Oriented Programming with C++”, Tata McGraw Hill. http://www.mldcollege.com/panel/programs/Object%20Oriented%20Programming%20with%20C__-Bal%20-%20E.Balagurusamy.pdf

Online Courses and Video Lectures

1. https://nptel.ac.in/courses/106101208/1	Accessed on August, 20, 2019
2. https://nptel.ac.in/courses/106101208/18	Accessed on August, 20, 2019
3. https://nptel.ac.in/courses/106101208/20	Accessed on August, 20, 2019
` 4. https://nptel.ac.in/courses/106101208/21	Accessed on August, 20, 2019
5. https://nptel.ac.in/courses/106101208/23	Accessed on August, 20, 2019
6. https://nptel.ac.in/courses/106101208/25	Accessed on August, 20, 2019

Subject Code: PCCS-102

Subject Name: Computer Networks

Programme: B.Tech. CSE	**L:** 3 **T:** 0 **P:** 0
Semester: 3	**Teaching Hours:** 38
Theory/Practical: Theory	**Credits:** 3
Internal Marks: 40	**Percentage of Numerical/Design/Programming Problems:** 40%
External Marks: 60	**Duration of End Semester Exam (ESE):** 3 hrs
Total Marks: 100	**Elective Status:** Compulsory

Prerequisites: Knowledge of Computer System fundamentals.

Additional Material Allowed in ESE: [NIL]

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

CO#	Course Outcomes (CO)
1	Develop an understanding of modern network architectures from a design and performance.
2	Understand the major concepts involved in wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs).
3	Analyze various protocols to develop network related applications for future needs.
4	Apply the knowledge of different network designs and various logical models of networking to solve problems of communication over different medium.
5	Utilize knowledge of routing and congestion control algorithms to overcome various issues over different complex networking structures.
6	Discuss algorithms for medium access sub layer to avoid collision and error problems over different types of networks.

Detailed Contents:

Part-A

Data Communication Components: Representation of data and data flow, Various Network Topologies, Protocols and Standards, LAN: Wired LAN, Wireless LANs, Connecting LAN and Virtual LAN, Techniques for Bandwidth utilization: Multiplexing- Frequency division, Time division and Wave division, Concepts on spread spectrum, OSI model, TCP/IP reference model and their comparison. **[6 Hours]**

Physical Layer: Concept of analog and digital systems, Transmission Media, Transmission impairments and Data rate limits- Nyquist formula, Shannon formula, Switching- Circuit, Message and Packet switching. **[7 Hours]**

Data Link Layer and Medium Access Sub Layer: Error Detection and Error Correction- Fundamentals, Block coding, Hamming Distance, CRC; Flow Control and Error control protocols-Stop and Wait, Go back–N ARQ, Selective Repeat ARQ, Sliding Window, Piggybacking, Random Access, Multiple access protocols- Pure ALOHA, Slotted ALOHA, CSMA/CD, CDMA/CA. **[7 Hours]**

Part-B

Network Layer: Logical addressing- IPV4, IPV6; Address mapping- ARP, RARP, BOOTP and DHCP–Delivery, Routing algorithms, Congestion control policies, Leaky bucket and token bucket algorithms. **[6 Hours]**

Transport Layer: Design issues, Elements of transport Protocols- Connection establishment and release, Process to Process Communication, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), flow control. **[6 Hours]**

Session, Presentation and Application Layer: Session Layer- Design issue, remote procedure call. Presentation Layer- Design issue, Data compression techniques. Application Layer- Domain Name Space (DNS), DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP. **[6 Hours]**

Text Books

1. Andrew S. Tanenbaum, "Computer Networks", Pearson Education.
2. Behrouz A. Forouzan, "Data Communication &Networking", Tata McGraw Hill.

Reference Books

1. Douglas E. Comer, "Internetworking with TCP/IP", Volume-I, Prentice Hall, India.

2. W. Stallings, “Data and Computer Communication”, Prentice Hall of India.
3. James F. Kurose and Keith W. Ross, “Computer Networking”, Pearson Education.

E-Books and online learning material

1. An Introduction to Computer Networks by Peter L Dordal, Department of Computer Science, Loyola University Chicago.

   http://intronetworks.cs.luc.edu/current/ComputerNetworks.pdf

Online Courses and Video Lectures

1. https://nptel.ac.in/courses/106105081/	Accessed on May. 15, 2019
2. https://nptel.ac.in/courses/106105081/2	Accessed on May. 15, 2019
3. https://nptel.ac.in/courses/106105081/5	Accessed on May. 15, 2019
` 4. https://nptel.ac.in/courses/106105081/16	Accessed on May. 15, 2019
5. https://nptel.ac.in/courses/106105081/31	Accessed on May. 15, 2019

Subject Code: ESCS-101

Subject Name: Digital Electronics

Programme: B.Tech. (CSE)	**L:** 3 **T:** 0 **P:** 0
Semester: 3	**Teaching Hours:** 36
Theory/Practical: Theory	**Credits:** 3
Internal Marks: 40	**Percentage of Numerical/Design/Programming Problems:** 50%
External Marks: 60	**Duration of End Semester Exam (ESE):** 3 hrs
Total Marks: 100	**Elective Status:** Compulsory

Prerequisites: NIL

Additional Material Allowed in ESE: [Scientific Calculator or NIL]

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

CO#	Course Outcomes (CO)
1	Understand the relationships between Boolean algebra, combinational logic, and sequential logic.
2	Solve combinational logic problem formulation and logic optimization.
3	Construct digital logic circuits using gates and state-of-the art MUX, ROM, PLA and PAL units
4	Create profound analysis and design of synchronous and asynchronous sequential circuits
5	Design and inspect digital circuits to meet desired needs within realistic constraints.
6	Develop skills to build and troubleshoot digital circuits.

Detailed Contents:

Part-A

Number Systems: Binary, Octal, Decimal, Hexadecimal. Number base conversions, 1’s, 2’s, rth’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. **[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. **[5 Hours]**

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]**

Part-B

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 machines: state diagram, Circuit implementation. Shift registers. **[6 Hours]**

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

Introduction to Design with PLDs: Introduction to programmable logic devices-Programmable Logic Array (PLA), Programmable Array Logic (PAL), Field Programmable Gate Arrays (FPGA). **[5 Hours]**

Text Books

1. M. Morris Mano, “Digital Design”, Prentice Hall of India Pvt. Ltd./Pearson Education (Singapore) Pvt. Ltd., New Delhi.
2. John F.Wakerly, “Digital Design”, Pearson/PHI.
3. John M. Yarbrough, “Digital Logic Applications and Design”, Thomson Learning.
4. Charles H.Roth., “Fundamentals of Logic Design”, Thomson Learning.
5. A. Anand Kumar, “Fundamentals of Digital Circuits”, PHI.

Reference Books

1. Donald P. Leach and Albert Paul Malvino, “Digital Principles and Applications”, TMH.
2. William H. Gothmann, “Digital Electronics”, PHI.
3. A. K. Maini, “Digital Electronics: Principles, Devices and Applications, Wiley, 2007.

E-Books and online learning material

1. Web packages for HDL, GHDL, FreeHDL
2. PSpices and NGSpice
3. Xcircuit and Scilab
4. NPTEL website and IITs virtual laboratory

Online Courses and Video Lectures

1. https://nptel.ac.in/courses/117106086/32	Accessed on May 19, 2019
2. https://nptel.ac.in/courses/108105113/	Accessed on May 19, 2019
3. https://www.coursera.org/learn/digital-systems	Accessed on May 19, 2019

Subject Code: BSCS-101

Subject Name: Mathematics III

Programme: B.Tech.(CSE**)**	**L:** 3 **T:** 1 **P:** 0
Semester: 3	**Teaching Hours:** 40
Theory/Practical: Theory	**Credits:** 4
Internal Marks: 40	**Percentage of Numerical/Design/Programming Problems:** 90%
External Marks: 60	**Duration of End Semester Exam (ESE):** 3 hrs
Total Marks: 100	**Elective Status:** Compulsory

Prerequisites: Knowledge of partial differentiation, probability and statistics.

Additional material allowed in ESE: Scientific calculator, log tables, probability distribution tables, statistical tables or NIL

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

CO#	Course Outcomes (CO)
1	Understand Partial Differential Equations and their solutions techniques.
2	Understand Analytic functions and evaluation of derivative of functions of complex variable.
3	Evaluate integration of functions of complex variables.
4	Analyze probability spaces, random variables and different probability distribution.
5	Fit the given data into best fit curve.
6	Apply statistical methods for analyzing experimental data.

Detailed Contents:

Part-A

Differential calculus of complex variables: Separation of elementary functions of complex variables, Cauchy-Riemann equations, analytic functions, elementary analytic functions (exponential, trigonometric, logarithm) and their properties, harmonic functions, finding harmonic conjugate. **[8 Hours]**

Integral Calculus of functions of complex variables: Complex integrals, Cauchy-Goursat theorem (without proof), Cauchy Integral formula (without proof) Taylor’s series, zeros of analytic functions, singularities, Laurent’s series; Residues, Cauchy Residue theorem (without proof), Evaluation of definite integral involving sine and cosine, Mobius transformations and their properties. **[8 Hours]**

Linear Systems: Gauss’s elimination method and Gauss’s Jordan method **[3 Hours]**

Part-B

Applied Statistics: Curve fitting by the method of least squares- fitting of straight lines and second degree parabolas **[5 Hours]**

Probability Distributions: Probability spaces, Discrete random variables, Poisson and binomial distribution. Continuous random variables and their properties, distribution functions and densities, normal, exponential and gamma densities. **[8 Hours]**

Statistics: Test of significance: Large sample test for single proportion, difference of proportions, single mean, difference of means, and difference of standard deviations. Small sample test for single mean and difference of means, test for ratio of variances - Chi-square test for goodness of fit and independence of attributes. **[8 Hours]**

Textbooks:

1. Erwin Kreyszig, Advanced Engineering Mathematics, John Wiley & Sons.
2. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications.
3. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers.

Reference books:

1. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, Pearson.
2. P. G. Hoel, S. C. Port and C. J. Stone, Introduction to Probability Theory, Universal Book Stall.
3.. S. Ross, A First Course in Probability, Pearson Education India.
4. J. W. Brown and R. V. Churchill, Complex Variables and Applications, Mc-Graw Hill

Subject Code: HSMCS-101

Subject Name: Human values and Professional Ethics

Programme: B.Tech. (CSE**)**	**L:** 3 **T:** 0 **P:** 0
Semester: 3	**Teaching Hours:** 38
Theory/Practical: Theory	**Credits:** 3
Internal Marks: 40	**Percentage of Numerical/Design/Programming Problems:** Nil
External Marks: 60	**Duration of End Semester Exam (ESE):** 3 hrs
Total Marks: 100	**Elective Status:** Compulsory

Additional Material Allowed in ESE: NIL

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

CO#	Course Outcomes (CO)
1	Discriminate between valuable and superficial in the life.
2	Encourages students to discover what they consider valuable.
3	Understand the value required to be a good human being and apply these values in real life.
4	Evaluate and modify the behavior.
5	Understand fundamental and organizational duties and protect individual and social rights.
6	Know about professional behavior, values and guiding principles.

Detailed Contents:

Part-A

Ethics and values: Importance of Ethics and values, Difference between moral, ethics and values , Nature of Values, The Structure of Value Relations , Integrity, Work ethics, Service learning, Virtues, Respect for others, Living peacefully, caring , Sharing , Honesty, Courage, Valuing time , Cooperation, Commitment, Empathy, Self confidence, Challenges in the work place, spirituality. **[8 Hours]**

Value education: Need for value education, Basic guidelines, Self Exploration, Values in family and Harmony in existence, Values across cultures. **[6 Hours]**

Personality and behavior development: God and scientific attitude, positive thinking, Integrity and discipline, punctuality, Aware of self destructive habits, Association and cooperation, Doing best, motivation Theories and Case study, Johari Window, Leadership Styles and Theories, Win-Win policy, SWOT Analysis **[6 Hours]**

Part-B

Human rights: Definition, Fundamental rights and duties, Regional, national and Universal protection of human rights, human rights and vulnerable groups. **[5 Hours]**

Professional ethics: Introduction, Objectives and types of professional ethics, Personal vs. Professional Ethics. Ethics in Profession, Ethics for employees, Rights of an Employee, Whistle-blower Policy, Code of Conduct, code of ethics, global issues (Technology revolution, international trade, globalization, environmental ethics, war ethics and intellectual property rights). **[8 Hours]**

Engineering Ethics: Scope and approach, Steps to Deal with Issues, Types of Inquiries, Moral Dilemma, Steps to Solve Dilemma, Engineering as Social Experimentation, Engineers as responsible experimenters. **[5 Hours]**

Text Books

1. John Berry, Janek, Pandey; Poortinga, Ype ‘Handbook of Cross-cultural Psychology’, Boston.
2. R.S Naagarazan , “Professional Ethics and Human Values”, New Age Publisher.
3. P.L. Dhar, R.R. Gaur, ‘Science and Humanism’, Common Wealth Publishers.
4. Tanu Shukla, Anupam Yadav , Gajendra Singh Chauhan, ”Human Values and Professional Ethics”, First edition, 2017.
5. M. Govindrajran, S. Natrajan & V.S. Senthil Kumar, “Engineering Ethics (including Human Values)”, Prentice Hall of India Ltd.
6. Premvir Kapoor, “Professional Ethics and Human Values”, Khanna Publishing.

Reference Books

1. E.G. Seebauer& Robert L. Berry, ‘Fundamentals of Ethics for Scientists & Engineers’, Oxford University Press.
2. R.R. Gaur , R Sangal, G P Bagria, “A Foundation Course in Human Values and Professionals Ethics”, Excel Publishers.
3. Vaishali R Khosla, Kavita Bhagat, “Human Values And Professional Ethics”, Technical Publications.

E-Books and online learning material

1. Human values and Professional Ethics

https://crescent.education/wp-content/uploads/2018/12/Crescent-human-values-professional-ethics.pdf

2. Professional Ethics and Human Values

Noteshttps://www.academia.edu/8844628/Professional_Ethics_and_Human_Values_Notes

3. Engineering Ethics Tutorial

https://www.tutorialspoint.com/engineering_ethics/index.htm

4. Professional Ethics and Values in Engineering

https://www.srecwarangal.ac.in/cse-downloads/peve-unit-1.pdf

Online Courses and Video Lectures

1. https://nptel.ac.in/courses/109104068/30 Accessed on June 27, 2019

2. https://www.youtube.com/watch?v=3-UEi_djb7w Accessed on June 27, 2019

3. https://www.youtube.com/watch?v=VVsrsOaEcFQ Accessed on June 27, 2019

4. https://www.youtube.com/watch?v=-iJIrYreix0 Accessed on June 27, 2019

Subject Code: LPCCS-101

Subject Name: Object Oriented Programming Laboratory

Programme: B.Tech. (CSE)	**L:** 0 **T:** 0 **P:** 4
Semester: 3	**Teaching Hours:** 42
Theory/Practical: Practical	**Credits:** 2
Internal Marks: 30	**Percentage of Numerical/Design/Programming Problems:**100%
External Marks: 20	**Duration of End Semester Exam (ESE):** 2 hrs
Total Marks: 50	**Elective Status:** Compulsory

Prerequisites: NIL

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

CO#	Course Outcomes (CO)
1	Compare and contrast object oriented programming paradigm with procedure oriented programming paradigm.
2	Design and implement efficient programs to solve computing problems in a high level programming language.
3	Utilize knowledge of different object oriented principles to identify and apply the appropriate techniques in problem solving.
4	Apply the knowledge acquired to troubleshoot programming related problems.
5	Utilize the knowledge and principles of object oriented programming while working in multidisciplinary teams.
6	Design and develop projects using object oriented tools and techniques.

[Control statements]

1. Demonstrate the use of conditional control statements like if, if-else, if-else ladder, nested if-else, and switch-case statement.

2. Illustrate the use of loop control statements like for, while, and do-while.

3. Write a program to demonstrate the use of break and continue statement.

[Arrays and Strings]

4. Demonstrate the use of one dimensional and two dimensional arrays by using suitable programs.

5. Illustrate the use of various string handling functions.

[Classes and Objects]

6. Program to illustrate the concept of classes and object.

7. Program to illustrate the concept of nesting of member functions.

8. Program to show the working of static members (static functions and static variables) in a class.

9. Program to demonstrate the use of friend functions.

[Constructors and Destructors]

10. Program to illustrate the concept of default constructor, parameterized constructor, and copy constructor.

11. Program to illustrate the concept of destructors.

[Polymorphism]

12. Program to demonstrate the concept of operator overloading

13. Program to illustrate the concept of function overloading and constructor overloading.

14. Program to illustrate the concept of virtual functions and pure virtual functions.

[Inheritance]

15. Program to illustrate the concept of inheritance.

16. Program to illustrate the concept of ambiguity in multiple inheritance.

17. Program to illustrate the order of execution of constructors and destructors in inheritance.

18. Program to demonstrate the concept of function overriding.

[Exception handling]

19. Program to illustrate the exception handling mechanism.

[File handling]

20. Program to illustrate the concept of file pointers.

21. Program to perform read and write operations on a file.

Any one project

Banking System Project

Description: The BANKING SYSTEM project has account class with data members like account number, name, deposit, withdraw amount and type of account. Customer data is stored in a binary file. A customer can deposit and withdraw amount in his account. User can create, modify and delete account.

Library Management System Project

The LIBRARY MANAGEMENT SYSTEM project has book and student class with data members like book no, bookname, authorname. Books record is stored in a binary file. A student can issue book and deposit it within 15 days. Student is allowed to issue only one book. Student Records are stored in binary file. Administrator can add, modify or delete record.

Reference Material

Manuals available in Lab

Subject Code: LPCCS-102

Subject Name: Computer Networks Laboratory

Programme: B.Tech. CSE	**L:** 0 **T:** 0 **P:** 2
Semester: 3	**Teaching Hours:** 24
Theory/Practical: Practical	**Credits:** 01
Internal Marks: 30	**Percentage of Numerical/Design/Programming Problems:**100%
External Marks: 20	**Duration of End Semester Exam (ESE):** 2 hrs
Total Marks: 50	**Elective Status:** Compulsory

Prerequisites: Fundamentals of Computer System.

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

CO#	Course Outcomes (CO)
1	Analyze and configure protocols concerning various network technologies over different mediums and layers.
2	Apply the knowledge of different network components, transmission mediums and tools to solve various problems of communication.
3	Design and develop different network design and logical models of networking to solve network related problems.
4	Utilize knowledge of modern network simulation tools to propose solution for efficient working of networks for real world problems.
5	Make use of various troubleshooting methods to overcome networking problems.
6	Function in multidisciplinary teams through groups while working in different network environments with the help of resource sharing.

Special Instruction related to resources requirement: Except practical number 10.

Sr.No.	Name of Practical
1.	Familiarization with networking components, transmission media, tools and devices: LAN Adapters, Hubs, Switches, Routers etc.
2.	Study of various LAN topologies and their creation using network devices, cables and

	computers, Preparing straight and cross cables.
3.	Configuration of TCP/IP Protocols in Windows and Linux.
4.	Implementation of resource (file, printer, etc.) sharing.
5.	Designing and implementing class A, B and C networks.
6.	Subnet planning and its implementation.
7.	To configure dynamic IP address for a computer connected to a LAN.
8.	Use of commands like ping, ipconfig for trouble shooting network related problems.
9.	Develop a program to compute the Hamming Distance between any two code words.
10.	To configure proxy server, Familiarization with network simulation tools.

Reference Material

Manuals available in Lab.

Subject Code: LESCS-101

Subject Name: Digital Electronics Laboratory

Programme: B.Tech. (CSE)	**L:** 0 **T:** 0 **P:**2
Semester: 3	**Teaching Hours: 24**
Theory/Practical: Practical	**Credits:** 1
Internal Marks: 30	**Percentage of Numerical/Design/Programming Problems:**100%
External Marks: 20	**Duration of End Semester Exam (ESE):** 2 hrs
Total Marks: 50	**Elective Status:** Compulsory

Prerequisites: NIL

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

CO#	Course Outcomes (CO)
1	Identify and apply the knowledge of logic gates and integrated circuits to solve related problems.
2	Design and implement combinational & sequential circuits for engineering problems.
3	Choose and compare the usage of appropriate techniques and tools to solve digital circuits problem.
4	Apply the knowledge acquired to demonstrate the usage of digital circuits in computers at large.
5	Utilize the knowledge and principles of digital electronics while working in multidisciplinary team formation.
6	Design simple digital systems based on these digital abstractions, using the "digital paradigm".

Resources required: Digital Trainer Board, ICs, connecting wires.

S. No.	Name of Practical
1.	Truth-table verification of OR, AND, NOT, XOR, NAND and NOR gates using various IC’s.
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.
4.	Half subtractor / Full subtractor: Realization using IC’s 7400 and 7402.
5.	Realization of IC7483 as Parallel adder/subtractor.
6.	4-Bit Binary-to-Gray and 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 and IC74193 chip.

A mini project such as burglar alarm, fire alarm, traffic alert system etc.

Reference Material

Manuals available in Lab.

Subject Code: PRCS-101

Subject Name: Seminar and Technical Report Writing

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

Additional Material Allowed in ESE: NA

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

CO#	Course Outcomes (CO)
1.	Understand the basic components of definitions, descriptions, process explanations, and other common forms of technical writing.
2.	Analyze and critique various speech techniques, content, purpose, strengths and weaknesses.
3.	Reference and quote correctly, and not infringe copyright.
4.	Practice the unique qualities of professional rhetoric, writing and presentation style.
5.	Use a technical report to communicate information.
6.	Follow the stages of the writing process (prewriting/writing/rewriting) and apply them to technical and workplace writing tasks.

Part-A

Technical report: Importance of technical report, Structure of technical report, Planning the report, Diagrams, graphs, tables and mathematics, Citing and referencing, References to diagrams, graphs, tables and equations, Originality and plagiarism, Finalising the report and proofreading. [**2 Hours**]

Presentation Skills: Plan, structure and prepare presentation, Voice and body language, Conversational style, Rules for effective speaking, Enhancing improvising skills, Managing a challenging audience, Using visuals and metaphors to make presentation more inspiring. **[2 Hours]**

Latex: Installation of the software LaTeX, Latex compilation, Creating a document- preamble of a document, Basic formatting- abstract, paragraphs and newlines, Headers and footers, spacing, hyphenation, Bold, italics and underlining text, Paragraph alignment and indentation, Lists- Unordered, ordered, nested **[3 Hours]**

Part-B

Tables: The tabular environment, Tables with fixed length, Combining rows and columns, Multi-page tables, Positioning tables, Captions, labels and references, Line width and cell padding, colors, Sideways tables, Table with legend. **[3 Hours]**

Figures: Insertion, Changing the image size and rotating the picture, captions, lists of figures and tables, Captions, labels and references, Wrapping text around figures, Subfigures, Wide figures in two-column documents. **[3 Hours]**

References: Bibliography management in LaTeX, Creating a .bib file, citation styles, citation of references in the text. **[2 Hours]**

Text Books

7. John Seely. The Oxford Guide to Effective Writing and Speaking. Oxford University Press.
8. Frank Mittelbach , Michel Goossens, Johannes Braams, David Carlisle, Chris Rowley, "The LaTeX Companion (Tools and Techniques for Computer Typesetting)", Addison-Wesley.
9. Stefan Kottwitz, "LaTeX Beginner's Guide", PACKT.

Reference Books

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

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/technicalreport-writing-for-engineers. Accessed on April 26th 2019.
2. “Technical Writing”, https://www.coursera.org/learn/technical-writing. Accessed on April 26th 2019.

Subject Code: PCCS-103

Subject Name: Discrete Mathematics

Programme: B.Tech. (CSE)	**L:** 3 **T:** 1 **P: 0**
Semester: 4	**Teaching Hours:** 35
Theory/Practical: Theory	**Credits:** 4
Internal Marks: 40	**Percentage of Numerical/Design/Programming Problems:** 90%
External Marks: 60	**Duration of End Semester Exam (ESE):** 3 hrs
Total Marks: 100	**Elective Status:** Compulsory

Prerequisites: Familiarization with sequences and series, sets, functions and derivatives.

Additional Material Allowed in ESE: [Scientific Calculator]

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

CO#	Course Outcomes (CO)
1	Apply knowledge of mathematical proofs, techniques and algorithms to solve complex engineering problem.
2	Prove elementary properties of modular arithmetic and explain their application in analysis and interpretation of data and synthesis of information to provide valid conclusions.
3	Create, select and apply appropriate techniques to model real world problems using graphs.
4	Identify and formulate solutions of engineering problems related to counting and probability theory**.**
5	Utilize the importance of discrete structures towards simulation of problems in multidisciplinary environments.
6	Formulate a logical statement in terms of a symbolic expression and evaluate the truth value of compound statement.

Detailed Contents:

Part-A

Fundamentals of Sets, Relations and Functions: Sets – Operations on sets, Subsets, Types of sets, Ordered pairs, Proofs of general identities of sets, Classes of sets and partitions, Countable and uncountable sets. Relations –Representations of relations, Types of relations, Composition of relations, Closure properties of relations, Equivalence relations, Compatibility relations, Partial order relations. Functions – Introduction and types of functions, Sum and product of functions, Hashing functions, Recursively defined functions. [**7 Hours**]

Prepositional and Predicate Logic: Prepositions and compound prepositions, Logical connectives, Truth tables, Logical implication and logical equivalence, Normal forms– Conjunctive and Disjunctive, Validity of well-formed formula, Propositional inference rules–Modus ponens and modus tollens. Predicate logic, Universal and existential quantification, Limitations of propositional and predicate logic. **[3 Hours]**

Combinatorial Mathematics: Basic counting principles, Permutations and combinations, Pigeonhole principle, Inclusion and exclusion principle, Recurrence relations – Solving homogeneous and non-homogeneous recurrence relations, Sequences, Generating function. **[7 Hours]**

Part-B

Algebraic Structures and Morphism : Algebraic structures with one binary operation, Properties of an operation, Congruence relation , Semi groups, Monoids, Groups, Substructures, Cyclic groups, Cosets, Normal subgroups, Dihedral groups, Permutation Groups. Homomorphism and isomorphism of groups, Applications of groups. Algebraic structures with two binary operation, Rings – Introduction, Abelian ring, Ring with unity, Multiplicative inverse, Subrings, Homomorphism of rings. Integral Domain, Ideals. **[8 Hours]**

Boolean Algebra: Boolean algebra, Boolean sub-algebra, Boolean rings, Application of Boolean algebra (Logic implications, Logic gates, Karnaugh-map). **[2 Hours]**

Graphs and Trees: Graphs – Definition, degree, Connectivity, path, cycle, Directed and undirected, Sub Graph, Bi-connected component and Articulation points. Eulerian chains and cycles, Hamiltonian chains and cycles, Shortest paths algorithms – Dijkstra'salgorithm, Warshall's algorithm. Rooted trees, Spanning tree algorithms – Kruskal's algorithm, Prim's algorithm. Graph coloring, Map Coloring, Chromatic number, Planar graphs, Euler's formula, Isomorphism and homomorphism of graphs, Applications of graph theory. **[8 Hours]**

Text Books

1. S. Lipschutz and M.Lipson, "Schaum's Outline of Discrete Mathematics", Tata McGraw Hill.

2. A. Doerr and K. Levarseur, “Applied Discrete Structures for Computer Science”, Pearson Education, Inc.
3. K.H. Rosen, “Discrete Mathematics and its applications”, Tata McGraw Hill.

Reference Books

1. C.L. Liu , “ Elements of Discrete Mathematics” , Tata McGraw Hill.
2. Susanna S. Epp, Discrete Mathematics with Applications, Wadsworth Publishing Co. Inc.

E-Books and online learning material

1. Discrete Mathematics and its applications by Kenneth H. Rosen
https://mathcs.clarku.edu/~djoyce/ma114/Rosen6E.pdf 6th Edition Accessed on Feb. 27, 2019

2. Discrete Mathematics: An Open Introduction by Oscar Levin
https://open.umn.edu/opentextbooks/textbooks/discrete-mathematics-an-open-introduction Accessed on Feb. 27, 2019

3. A Course in Discrete Structures by Rafael Pass Wei-Lung Dustin Tseng
https://www.cs.cornell.edu/~rafael/discmath.pdf Accessed on Feb. 27, 2019

Online Courses and Video Lectures

1 https://nptel.ac.in/courses/106106094 Accessed on Feb. 27, 2019

2. https://nptel.ac.in/courses/111107058 Accessed on Feb. 27, 2019

3. https://nptel.ac.in/courses/111104026 Accessed on Feb. 27, 2019

4. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-042j-mathematics-for-computer-science-fall-2010/video-lectures Accessed on Feb. 27, 2019

Subject Code: PCCS-104

Subject Name: Computer Architecture and Microprocessors

Programme: B.Tech. (CSE)	**L:** 3 **T:** 0 **P:** 0
Semester: 4	**Teaching Hours:** 38
Theory/Practical: Theory	**Credits:** 3
Internal Marks: 40	**Percentage of Numerical/Design Problems:** 20%
External Marks: 60	**Duration of End Semester Exam (ESE):** 3 hrs
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

Data Representation: Data types, Complements, Fixed point representation, Floating point representation, Error detection and correction. [**3 Hours]**

Register Transfer and Micro-operations: Addition, Subtraction, Multiplication and division algorithms and hardware, Register transfer language and operations, Arithmetic micro-operations, Logic micro-operations, Shift micro-operations, Arithmetic logic shift unit. [**4 Hours]**

Computer Organization and Design: Instruction codes, Computer registers, Computer instructions, Timing and control, Instruction cycle, Memory reference instructions, Input/ Output and interrupts, Design and working of a complete basic computer, Control functions, Design of accumulator logic. [**4 Hours]**

Central Processing Unit and Input-Output Organization: General register organization, Stack organization, Addressing modes, RISC and CISC architecture, I/O interface, Asynchronous data transfer, Modes of transfer, Priority interrupt, DMA, I/O processor. [**4 Hours]**

Memory Organization: Memory hierarchy, Main memory, Auxiliary memory, Associative memory, Cache memory, Virtual memory, Memory management hardware. [**4 Hours]**

Part-B

Microprocessor Architecture: Introduction to microprocessors, 8085 microprocessor architecture – Bus structure, Register organization. [**5 Hours]**

Programming with 8085: Addressing modes, Instruction classification, Instruction formats, Data transfer operations, Arithmetic operations, Logical operations, Branch operations, Stack and subroutine operations, looping, counting and indexing operations. [**5 Hours]**

Interfacing: Memory and I/O mapped I/O, Programmable interfaces – 8255 programmable peripheral interface, 8259 interrupt controller, and 8237 DMA controller. [**5 Hours]**

Microprocessor Applications: Interfacing of keyboards and seven segment LED display, Study of traffic light system, stepper motor controller. [**4 Hours]**

Text Books:

1. M. Morris Mano, "Computer System Architecture", Pearson Education.
2. William Stallings, "Computer Organization and Architecture", Pearson Education.

3. Ramesh Gaonkar, “Microprocessor Architecture, Programming, and Applications with 8085”, Penram International Publication.

Reference Books:

1. B. Ram, “Microprocessors and Microcomputers”, Dhanpat Rai Publications.
2. K. Vani, “Computer Architecture with MIPS”, Notion Press.
3. A.P. Mathur, “Introduction to Microprocessors”, Tata McGrawHill.
4. 4 P.K. Ghosh and P.R. Sridhar, “0000 To 8085: Introduction to Microprocessors for Engineers and Scientists”, PHI Learning.

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

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: PCCS-105

Subject Name: Operating Systems

Programme: B.Tech. (CSE**)**	**L:** 3 **T: 1 P:** 0
Semester: 4	**Teaching Hours:** 36
Theory/Practical: Theory	**Credits:** 4
Internal Marks: 40	**Percentage of Numerical/Design/Programming Problems:** 20%
External Marks: 60	**Duration of End Semester Exam (ESE):** 3 hrs
Total Marks: 100	**Elective Status:** Compulsory

Prerequisites: Basic knowledge of computer fundamentals and computer system architecture.

Additional Material Allowed in ESE:

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

CO#	Course Outcomes (CO)
1	Understand the mechanisms of OS to handle processes and threads and their communication.
2	Compare and contrast the mechanisms involved in memory management Techniques
3	Use the components of Operating System in OS design
4	Evaluate different scheduling Techniques.
5	Investigate basic concepts towards process synchronization and related issues.
6	Understand the structure and organization of file system.

Detailed Contents:

Part - A

Introduction: Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS-Layered, Monolithic, Microkernel Operating

Systems, Case study on UNIX and WINDOWS Operating System. **[5 Hours]**

Process management: Concept of processes and threads, Definition, Process and Program, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching Thread: Definition, Various states, Benefits of threads. **[4 Hours]**

Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non pre-emptive, FCFS, SJF, RR; Multiprocessor scheduling: Real Time scheduling. **[6 Hours]**

Inter-process Communication: Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson’s Solution, The Producer\ Consumer Problem, Semaphore **[4 Hours]** .

Part - B

Deadlocks: Introduction to deadlocks, Conditions for deadlock, Resource allocation graphs, Deadlock prevention and avoidance, Deadlock detection and recovery. **[4 Hours]**

Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation – Fixed and variable partition–Internal and External fragmentation and Compaction; Paging: Principle of operation – Page allocation – Hardware support for paging, Protection and sharing, Disadvantages of paging, Segmentation. **[6 Hours]**

File Management: Concept of File, Access methods, File types, File operation, Directory structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance. **[4 Hours]**

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

Text Books

1. A Silberschatz and Peter B. Galvin, “Operating System Concepts&quot; Addison Wesley.
2. GaryNutt, “Operating Systems Concepts”, Pearson Education Ltd.

Reference Books

1. Dhamdhere, “Systems Programming &amp; Operating Systems” Tata McGraw Hill.
2. Tanenbaum A. S “Operating System Design &amp; Implementation” Pearson Education.
3. Bhatt and Chandra “An introduction to operating systems concepts &amp; Practices” Prentice Hall of India Publication.

E-Books and online learning material

1. http://www.uobabylon.edu.iq/download/M.S%202013-2014/Operating_System_Concepts,_8th_Edition%5BA4%5D.pdf Accessed on Aug. 05, 2019
2. http://dinus.ac.id/repository/docs/ajar/Operating_System.pdf Accessed on Aug. 05, 2019

Online Courses and Video Lectures

1. https://nptel.ac.in/courses/106106144/ Accessed on Aug. 05, 2019
2. https://www.coursera.org/learn/os-power-user Accessed on Aug. 05, 2019

Subject Code: PCCS-106

Subject Name: Data Structures

Programme: B.Tech. (CSE**)**	**L:** 3 **T:** 0 **P:** 0
Semester: 4	**Teaching Hours:** 37
Theory/Practical: Theory	**Credits:** 3
Internal Marks: 40	**Percentage of Numerical/Design/Programming Problems:** 50%
External Marks: 60	**Duration of End Semester Exam (ESE):** 3hrs
Total Marks: 100	**Elective Status:** Compulsory

Prerequisites: Knowledge of Programming for Problem Solving and OOPS

Additional Material Allowed in ESE: [Scientific Calculator]

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

CO#	**Course Outcomes (CO)**
1	Apply knowledge of statistics and programming skills to solve complex engineering problems related to data structures.
2	Make use of Research based knowledge to identify the appropriate data structure and provide better solution to reduce space and time complexity.
3	Identify, Formulate and analyse data structure to develop skills and understand their applications to perform operations on it.
4	Design appropriate algorithm for autonomous realization of sub-programs to model complex engineering activities.
5	Demonstrate various methods of organizing large amounts of data and recognize systematic way to retrieve data and solve problems.
6	Formulate new solutions for programming problems or improve existing code using learned algorithms and data structures.

Detailed Contents:

Part-A

Basic concepts: Concept of data type, Linear and non-linear data structures, Data structures versus data types, Operations on data structures, Algorithm complexity and Asymptotic notations. **[2 Hours]**

Arrays: Linear and multi-dimensional arrays and their representation, Operations on arrays, Sparse matrices and their storage. **[2 Hours]**

Stacks: Sequential representation of stacks, Operations on stacks, Application of stacks – parenthesis checker, Evaluation of postfix expressions, Conversion from infix to postfix, Conversion from infix to prefix representation, Tower of Hanoi problem, implementing recursive functions. **[5 Hours]**

Queues: Sequential representation of queue, Types of queue- Linear Queue, Circular Queue, Deque, Priority Queue, Operations on each types of Queues and their algorithms, Applications of Queues. **[4 Hours]**

Linked List: Definition and representation of Linked list, Types of Linked list- Linear linked list, Doubly linked list, Circular linked list and Header linked list and their operations, Application of linked lists, Garbage collection and compaction, Linked representation of Stack and Queues and their algorithm. **[6 Hours]**

Part-B

Trees: Basic terminology, Sequential and linked representations of trees, Different types of Trees- Binary Tree, Binary search tree, Threaded binary tree, AVL tree and B-tree. Operations on each of the trees. Application of Binary Trees. **[5 Hours]**

Graphs: Basic terminology, Representation of graphs – Adjacency matrix, Adjacency list. Operations on graph, Traversal of a graph – Breadth first search, Depth first search. Shortest path algorithms – Dijkstra’s and Floyd. Minimum spanning tree – Prim and Kruskal. Applications of graphs. **[4 Hours]**

Heaps: Representing a heap in memory, Operations on heaps, Application of heap in implementing priority queue and Heap sort algorithm. **[2 Hours]**

Hashing and Hash Tables: Introduction to hash table, Hash functions, Concept of collision and its resolution using open addressing and separate chaining, Double hashing, Rehashing. **[2 Hours]**

Searching and Sorting: Linear and binary search techniques, Sorting methods – Bubble sort, Selection sort, Insertion sort, Quick sort, Merge sort, Shell sort and radix sort. Complexities of searching and sorting algorithms. **[5 Hours]**

Text Books

1. Seymour Lipschutz, "Data Structures", Schaum's Outline Series, Tata McGraw Hill.
2. Sartaj Sahni, "Data Structures, Algorithms and Applications in C++", Tata McGraw Hill.

Reference Books

1. Michael T. Goodrich, Roberto Tamassia, & David Mount, "Data Structures and Algorithms in C++", Wiley India.
2. Kruse, "Data Structures &ProgramDesign", Prentice Hall of India.
3. Y. Langsa, M.J. Augenstein, A.M. Tanenbaum, "Data structures using C and C++", Prentice Hall of India.
4. Vishal Goyal, LaliGoyal, Pawan Kumar, " Simplified Approach to Data Structures", Shroff Publications and Distributors

E-Books and online learning material

1. Data Structures and Algorithms: by Granville Barnett, and Luca Del Tongo. https://apps2.mdp.ac.id/perpustakaan/ebook/Karya%20Umum/Dsa.pdf
2. Data Structures and Algorithms in JAVA :by Michael T. Goodrich and Roberto Tamassiahttp://enos.itcollege.ee/~jpoial/algorithms/GT/Data%20Structures%20and%20Algorithms%20in%20Java%20Fourth%20Edition.pdf

Online Courses and Video Lectures

1. https://nptel.ac.in/courses/106102064/
2. https://nptel.ac.in/courses/106106133/
3. https://nptel.ac.in/courses/106106145/
4. https://www.youtube.com/watch?reload=9&v=YWnBbNj_G-U

Subject Code: PCCS-107

Subject Name: Software Engineering

Programme: B.Tech. (CSE**)**	**L:** 3 **T:** 1 **P:** 0
Semester:	**Teaching Hours:** 36
Theory/Practical: Theory	**Credits:** 4
Internal Marks: 40	**Percentage of Numerical/Design/Programming Problems:** 10%
External Marks: 60	**Duration of End Semester Exam (ESE):** 3 hrs
Total Marks: 100	**Elective Status:** Compulsory

Prerequisites: Knowledge of System Analysis

Additional Material Allowed in ESE: NIL

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

CO#	Course Outcomes (CO)
1	Plan a software engineering process life cycle, including the specification, design, and implementation.
2	Elicit, analyze and specify software requirements through a productive working relationship with various stakeholders of the project.
3	Analyze and translate a specification into a design, and then realize that design practically, using an appropriate software engineering methodology.
4	Develop the code from the design and effectively apply relevant standards for quality management and practice.
5	Formulate a testing strategy for a software system, employing techniques such as unit testing, test driven development and functional testing.
6	Identify modern engineering tools necessary for software reengineering and reverse engineering.

Detailed Contents:

Part-A

Introduction: Evolution and impact of software engineering, Software myths, Software application domains, Software crisis – Problem and causes. **[4 Hours]**

Software Process Models: Software process, Software process models – Waterfall model, Prototype model, Spiral model, Evolutionary model, RAD model, V-model and Component based model. **[4 Hours]**

Requirements Engineering: Feasibility study, Problem analysis, Requirement elicitation and specification, Functional and non-functional requirements, Software requirements specification document, Requirement validation and management. **[4 Hours]**

Project Management and Risk Analysis: Project planning, Cost estimation techniques– Size metrics, Empirical estimation, Heuristic estimation and analytical estimation, Project monitoring and control– Work breakdown structure, Activity chart, Gantt charts, PERT charts, Critical path method, Manpower management, Risk management- Identification, Analysis, Planning and Monitoring. **[6 Hours]**

Part-B

Software Design: Modular design– Coupling, Cohesion and abstraction, Function oriented design– Data flow diagrams, Structure chart, Object oriented design–Objects and object classes, Relationships between classes, User interface design. **[7 Hours]**

Coding & Testing: Coding standards and code reviews, Testing – Need of testing, Unit testing, Integration testing, System testing, White-Box testing, Black-box testing, Alpha, Beta and acceptance testing, Smoke testing, Sanity testing, Regression testing, Cyclometric Complexity. Verification and validation. **[6 Hours]**

Maintenance and Re-engineering: Software maintenance, Software re-engineering, Reverse engineering, Forward engineering, PSP and Six sigma. **[5 Hours]**

Text Books

1. Roger S. Pressman R., “Software Engineering, A Practitioner’s Approach”, McGraw Hill International.
2. Rajib Mall, “Fundamentals of Software Engineering”, Prentice Hall of India.

Reference Books

1. Ian Sommerville , Software Engineering, Addison-Wesley Publishing Company .
2. Jalote P., “An Integrated Approach to Software Engineering”, Narosa.

E-Books and online learning material

1. https://nptel.ac.in/courses/Webcourse contents/IIT%20Kharagpur/Soft%20Engg/New_index1.html Accessed on March 1, 2019
2. https://nptel.ac.in/downloads/106105087/ Accessed on March 1, 2019

Online Courses and Video Lectures

1. https://onlinecourses.nptel.ac.in/noc18_cs43/preview Accessed on March 1, 2019

Subject Code: LPCCS-103

Subject Name: Computer Architecture and Microprocessors Laboratory

Programme: B.Tech.(CSE)	**L:** 0 **T:** 0 **P:** 2
Semester: 4	**Teaching Hours:** 24
Theory/Practical: Practical	**Credits:** 1
Internal Marks: 30	**Percentage of Numerical/Design/Programming Problems:**100%
External Marks: 20	**Duration of End Semester Exam (ESE):** 2 hrs
Total Marks: 50	**Elective Status:** Compulsory

Prerequisites: Fundamentals of Computers.

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

CO#	Course Outcomes (CO)
1	Utilize the concept of binary & hexadecimal number systems including computer arithmetic.
2	Demonstrate the error detection & correction mechanism in computer architecture.
3	Understand the functional units of the processor such as the register file and arithmetic logical unit.
4	Examine the fundamentals of assembly language programming.
5	Understand the concept of computer arithmetic instruction set by designing code for arithmetic, logical and data transfer operations.
6	Solve basic binary math operations by using the instructions of microprocessor.

Special Instruction related to resources requirement: GNUsim8085 simulator should be installed to perform 4 to 11 practicals.

Sr.No.	Name of Practical
1.	**Combinational Circuit:** To study Half Adder.

2.	**Combinational Circuit:** To study Full Adder (7483).
3.	**Register Transfer and Micro-operations:** Write a program for binary multiplication.
4.	**Memory Organisation:** Write programs to simulate the mapping techniques of Cache memory. a. Direct Mapped cache b. Associative Mapped cache c. Set Associative Mapped cache
5.	**Functional design of ALU:** Analyzing the architecture and design of ALU, study the working of ALU and examine its functionality.
6.	**Working of 8085 simulator GNUsim8085 :** Introduction to 8085 microprocessor, study of components of GNUsim8085 and step wise assembly program execution using it.
7.	**Complete instruction set of 8085 :** Representation of instruction in computer system, Types of instructions - control instructions, logical instructions, branching, arithmetic and data transfer instructions.
8.	**Data transfer instructions:** Assembly language code in GNUsim8085 to implement various data transfer instructions like MOV, MVI, LXI, LDA, STA, IN, OUT etc.
9.	**Arithmetic instructions:** Assembly language code in GNUsim8085 to implement various arithmetic instructions involving immediate addition, subtraction, increment, decrement operations.
10.	**Addition and subtraction using flag:** assembly language code in GNUsim8085 to add and subtract two 8 bit numbers stored in memory and also set various flags corresponding to the result.
11.	**Logical instructions:** Assembly language code in GNUsim8085 to implement various logical instructions involving comparing of memory and register contents, logical operations - AND, OR, XOR and rotate operations.
12.	**Stack and branch instructions:** Assembly language code in GNUsim8085 to implement

	various stack and branch instructions involving insertion and deletion of contents into stack, conditional and unconditional jump, call and return.

R**eference Material**

Manuals available in Lab.

Subject Code: LPCCS-104

Subject Name: Operating Systems Laboratory

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

Prerequisites: Fundamentals of Computers.

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

CO#	Course Outcomes (CO)
1	Analyse the services, architectures and principles used in the design of modern operating systems.
2	Execute Linux commands for files and directories, creating and viewing files, File comparisons and Disk related commands.
3	Utilize the concept of virtualization for creating a virtual machine and installing operating system on virtual machine.
4	Demonstrate shell programming by using shell variables and shell keywords for automated system tasks.
5	Identify the key characteristics of multiple approaches used for the design and development of the operating system.
6	Apply system commands for performing the file manipulation, program execution, and printing text.

Special Instruction related to resources requirement: Any programming language like C, C++, Java can be used to simulate the programs.

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.	Overview of single user systems, network operating system and multiuser system.
4.	Write a program for the simulation of following CPU scheduling algorithms to find turnaround time and waiting time. a) FCFS b) SJF c) Round Robin d) Priority
5.	Write a program for the simulation of producer-consumer problem using semaphores.
6.	Write a program for the simulation of Banker's algorithm for the purpose of deadlock avoidance.
7.	Write a program for the simulation of following contiguous memory allocation techniques a) Worst-fit b) Best-fit c) First-fit
8.	Write a program for the simulation of following page replacement algorithms a) FIFO b) LRU c) Optimal
9.	Write a program for the simulation of following disk scheduling algorithms a) FCFS b) SCAN c) C-SCAN
10.	Write a program for the simulation of following file allocation strategies a) Sequential b) Indexed c) Linked
11.	To study the features of Windows and Linux operating system.
12.	Execute various basic Linux commands, commands for files and directories, creating and viewing files, File comparisons, Disk related commands.
13.	Basics of Shell programming, various types of shell, Shell Programming in bash.
14.	Implement conditional statements, looping statement, case statements and functions in Shell programming

Reference Material

Manuals available in Lab.

Subject Code: LPCCS-105

Subject Name: Data Structures Laboratory

Programme: B.Tech. (CSE)	**L:** 0 **T:** 0 **P:** 4
Semester: 4	**Teaching Hours:** 40
Theory/Practical: Practical	**Credits:** 2
Internal Marks: 30	**Percentage of Numerical/Design/Programming Problems:**100%
External Marks: 20	**Duration of End Semester Exam (ESE):** 2 hrs
Total Marks: 50	**Elective Status:** Compulsory

Prerequisites: Fundamentals of Computers.

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

CO#	Course Outcomes(CO)
1	Apply knowledge of mathematics and programming skills to implement and analyze different data structures.
2	Evaluate and analyze the time and space complexity of linear and non linear data structures.
3	Design and implement efficient algorithms to solve computing problems in a high level programming language.
4	Utilize knowledge of different data structures to identify and apply the appropriate data structures to solve a real world problem.
5	Compare and analyze different solutions of complex engineering activities with an understanding of their advantages and limitations.
6	Developing an awareness of the data structure for storing data and handling various operations on different applications in the broadest context of technology change.

Special Instruction related to resources requirement: Any programming language like C, C++, can be used for the programs.

Sr.No.	Name of Practical
1.	Design, Develop and Implement a menu driven Program for the following Array operations a. Creating an Array of N Integer Elements b. Display of Array Elements with Suitable Headings c. Inserting an Element (ELEM) at a given valid Position (POS) d. Deleting an Element at a given valid Position(POS) e. Exit.
2.	Design, Develop and Implement a menu driven Program for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX) a. Push an Element on to Stack b. Pop an Element from Stack c. Demonstrate how Stack can be used to check Palindrome d. Demonstrate Overflow and Underflow situations on Stack Display the status of Stack f. Exit Support the program with appropriate functions for each of the above operations
3.	Design, Develop and Implement a Program for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, %(Remainder), ^(Power) and alphanumeric operands.
4.	Design, Develop and Implement a Program for the following Stack Applications a. Evaluation of Suffix expression with single digit operands and operators: +, -, *, /, %, ^ b. Solving Tower of Hanoi problem with n disks
5.	Design, Develop and Implement a menu driven Program for the following operations on Circular QUEUE of Characters (Array Implementation of Queue with maximum size MAX) a. Insert an Element on to Circular QUEUE b. Delete an Element from Circular QUEUE c. Demonstrate Overflow and Underflow situations on Circular QUEUE d. Display the status of Circular QUEUE

	e. Exit Support the program with appropriate functions for each of the above operations .
6.	Design, Develop and Implement a menu driven Program for the following operations on Singly Linked List (SLL) of Student Data with the fields: USN, Name, Branch, Sem, PhNo a. Create a SLL of N Students Data by using front insertion. b. Display the status of SLL and count the number of nodes in it c. Perform Insertion / Deletion at End of SLL d. Perform Insertion / Deletion at Front of SLL(Demonstration of stack) e. Exit
7.	Design, Develop and Implement a menu driven Program for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: SSN, Name, Dept, Designation, Sal, PhNo a. Create a DLL of N Employees Data by using end insertion. b. Display the status of DLL and count the number of nodes in it c. Perform Insertion and Deletion at End of DLL d. Perform Insertion and Deletion at Front of DLL e. Demonstrate how this DLL can be used as Double Ended Queue f. Exit
8.	Design, Develop and Implement a Program for the following operationson Singly Circular Linked List (SCLL) with header nodes a. Find the sum of two polynomials POLY1(x,y,z) and POLY2(x,y,z) and store the result in POLYSUM(x,y,z)
9.	Design, Develop and Implement a menu driven Program for the following operations on Binary Search Tree (BST) of Integers a. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2 b. Traverse the BST in Inorder, Preorder and Post Order c. Search the BST for a given element (KEY) and report the appropriate message e. Exit
10.	Design, Develop and Implement a Program for the following operations on Graph(G) of Cities

	a. Create a Graph of N cities using Adjacency Matrix. b. Print all the nodes reachable from a given starting node in a digraph using DFS/BFS method
11.	Write a Program to finds the position of an element in an array using Linear Search Algorithm and Binary search Algorithm.
12.	Write a program to sort list using different sorting algorithms (bubble, selection, insertion, radix, merge and quick sort) and compare them.

Reference Material

Manuals available in Lab.