Curriculum for B.Tech. 2020

Computer Science and Engineering

Indian Institute of Information Technology, Design and Manufacturing, Kancheepuram

Chennai-600 127

Curriculum for B.Tech. Computer Science and Engineering 2020 Batch

Semester 1					
Category	Course Name	L	T	P	C
BSC	Calculus	3	1	0	4
BSC	Engineering Electromagnetics	3	0	0	3
BEC	Electrical Circuits for Engineers	3	1	0	4
BEC	Problem Solving and Programming	3	0	0	3
BEC	Materials for Engineers	3	0	0	3
DSC	Foundation for Engineering and Product Design	1	2	0	3
BSC	Engineering Electromagnetics Practice	0	0	3	1.5
BEC	Problem Solving and Programming Practice	0	0	3	1.5
HSC	Effective Language and Communication Skills	1	0	2	2
HSC	NSO/NCC/SSG/NSS	0	0	2	P/F
					25.0
Semester 2					
Category	Course Name	L	T	P	C
BSC	Differential Equations	3	1	0	4
SEC	Science Elective 1	3	1	0	4
BEC	Engineering Graphics	2	0	4	4
ITC	Data Structures and Algorithms	3	0	0	3
DSC	Sociology of Design	1	2	0	3
ITC	Design and Manufacturing Lab	0	0	2	1
PCC	Discrete Structures for Computer Science	3	1	0	4
ITC	Data Structures and Algorithms Practice	0	0	4	2
HSC	NSO/NCC/SSG/NSS	0	0	2	P/F
HSC	Earth, Environment and Design	1	0	0	P/F
					25.0
Semester 3					
Category	Course Name	L	T	P	C
SEC	Science Elective 2	3	1	0	4
DSC	Systems Thinking for Design	1	2	0	3
PCC	Object Oriented Programming	2	0	4	4
PCC	Digital System Design	3	1	0	4
PCC	Design and Analysis of Algorithms	3	1	0	4
PCC	Digital System Design practice	0	0	4	2
PCC	Design and Analysis of Algorithms practice	0	0	4	2
HSC	Indian Constitution, Essence of Indian Traditional Knowledge	1	0	0	P/F
					23.0
Semester 4					
Category	Course Name	L	T	P	C
SEC	Science Elective 3	3	1	0	4
DSC	Smart Product Design	1	2	0	3
PCC	Computer Organization and Architecture	3	1	0	4
PCC	Database Systems	3	1	0	4
PCC	Theory of Computation	3	1	0	4
PCC	Computer Organization and Architecture practice	0	0	4	2
PCC	Database Systems practice	0	0	4	2
HSC	Human Values and Stress Management	1	0	0	P / F
					23.0
Semester 5					
Category	Course Name	L	T	P	C
ITC	Data Science: An Applied Perspective	3	0	2	4
DSC	Entrepreneurship and Management Functions	1	2	0	3
PCC	Operating Systems	3	1	0	4
PCC	Computer Networks	3	1	0	4
PCC	Compiler Design	3	1	0	4
PCC	Operating Systems practice	0	0	4	2
PCC	Computers Networks practice	0	0	4	2
PCC	Compiler Design Practice	0	0	4	2
HSC	Professional Ethics and Organizational Behaviour	1	0	0	P/F
					25.0

Curriculum for B.Tech. Computer Science and Engineering 2020 Batch

Semester 6					
Category	Course Name	L	T	P	C
DSC	Prototyping and Testing	1	2	0	3
PEC	Professional Elective 1	3	1	0	4
PEC	Professional Elective 2	3	1	0	4
PEC	Professional Elective 3	3	1	0	4
ELC	Elective 1	3	1	0	4
ELC	Elective 2	3	1	0	4
HSC	Professional Communication	1	0	2	2
HSC	Intellectual Property Rights	1	0	0	P/F
					25.0
Summer					
PCD	Internship				P/F
Semester 7					
Category	Course Name	L	T	P	C
ELC	Elective 3	3	1	0	4
ELC	Elective 4	3	1	0	4
ELC	Elective 5	3	1	0	4
					12.0
Semester 8					
Category	Course Name	L	T	P	C
ELC	Elective 6	3	1	0	4
PCD	Project/Course work	0	0	16	8
					12.0

Semester wise Credit Distribution	Credits									
Category	S1	S2	S3	S4	S5	S6	S7	S8	Total	%
Basic Science Course (BSC)	8.5	4	0	0	0	0	0	0	12.5	7.4
Science Elective Course (SEC)	0	4	4	4	0	0	0	0	12	7.1
Basic Engineering Course (BEC)	11.5	4	0	0	0	0	0	0	15.5	9.1
Design Course (DSC)	3	3	3	3	3	3	0	0	18	10.6
IT Skill Course (ITC)	0	6	0	0	4	0	0	0	10	5.9
Professional Core Course (PCC)	0	4	16	16	18	0	0	0	54	31.8
Professional Elective Course (PEC)	0	0	0	0	0	12	0	0	12	7.1
Elective Course (ELC)	0	0	0	0	0	8	12	4	24	14.1
Humanities and Social Science Course (HSC)	2	0	0	0	0	2	0	0	4	2.4
Professional Career Development (PCD)	0	0	0	0	0	0	0	8	8	4.7
Total	25.0	25.0	23.0	23.0	25.0	25.0	12.0	12.0	170.0	100.0
	25.0	50.0	73.0	96.0	121.0	146.0	158.0	170.0	**170.0**	

<table>
<tr><td>Course Title</td><td>Object Oriented Programming</td><td>Course No</td><td colspan="5"></td></tr>
<tr><td rowspan="2">Department/ Specialization</td><td rowspan="2">Computer Science and Engineering</td><td rowspan="2">Credits</td><td>L</td><td>T</td><td>P</td><td colspan="2">C</td></tr>
<tr><td>2</td><td>0</td><td>4</td><td colspan="2">4</td></tr>
<tr><td>Faculty proposing the course</td><td>Faculty, Department of CSE</td><td>Status</td><td>Core</td><td>■</td><td>Elective</td><td colspan="2">□</td></tr>
<tr><td>Offered for</td><td>B.Tech</td><td>Type</td><td>New</td><td>■</td><td>Revision</td><td colspan="2">□</td></tr>
<tr><td>To take effect from</td><td>July 2021</td><td rowspan="2">Submitted for approval</td><td colspan="5" rowspan="2">44th Senate</td></tr>
<tr><td>Prerequisite</td><td>Nil</td></tr>
<tr><td>Learning Objectives</td><td colspan="7">The course introduces students to the object oriented programming paradigm and its benefits in application development. Both C++ and Java would be used as implementation platforms for the various object oriented features.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="7">• To understand Object Oriented Concepts for Software Design
• To analyze various aspects of Software Design in a reusable and secure fashion
• To create applications supporting a command line & graphical user interface in Object Oriented fashion.</td></tr>
<tr><td>Course Contents (with approximate breakup of hours for lecture/ tutorial/practice)</td><td colspan="7">• Object oriented programming - Encapsulation – Constructors – Destructors - Composition – Friend functions/classes – this pointer – Dynamic memory management (8L)
• Operator overloading Reusability – Inheritance – Base & derived classes – Protected members – Constructors –Destructors in derived classes – public/private/protected inheritance – Polymorphism (9L)
• Virtual functions - Templates – Function & Class templates – Streams – Stream input Output Stream format states – Manipulators – Exception handling – Re–throwing exceptions –specifications–and exception handling – Inheritance – STL (9L)
• Event Handling, Applets, – Frames, Buttons, Menu – Visual design layout, Multithreading, Networking, Database connectivity support (10L)
• Practice component will test drive the concepts covered in theory using C++/Java approximately for 14 sessions in the semester [Overall 36 Hours Theory + 28 Hours for lab]</td></tr>
<tr><td>Essential Reading</td><td colspan="7">1. Deitel P J and Deitel H M, C : How To Program, Prentice Hall, 10th Edn, 2016, ISBN 9780131596825
2. Deitel P J and Deitel H M, Java: How To Program, Prentice Hall, 9th Edn, 2016, ISBN 978-0132575669</td></tr>
<tr><td>Supplementary Reading</td><td colspan="7">1. David Flanagan, Java in a Nutshell, 5th Edition, O'Rielly, 2005, ISBN 9780596007737
2. Herbert Schildt, Java: A Beginners Guide, 9th Edition, McGraw Hill, 2014, ISBN 9781260440218
3. Herbet Schildt, Teach Yourself C++, 4th Edition, Tata McGraw Hill, 2003, ISBN 978-0070532465</td></tr>
</table>

<table>
<tr><td>Course Title</td><td>Digital System Design</td><td>Course No</td><td colspan="5"></td></tr>
<tr><td rowspan="2">Department/ Specialization</td><td rowspan="2">Computer Science and Engineering</td><td rowspan="2">Credits</td><td>L</td><td>T</td><td>P</td><td colspan="2">C</td></tr>
<tr><td>3</td><td>1</td><td>0</td><td colspan="2">4</td></tr>
<tr><td>Faculty proposing the course</td><td>Faculty, Department of CSE</td><td>Status</td><td>Core</td><td>■</td><td>Elective</td><td colspan="2">□</td></tr>
<tr><td>Offered for</td><td>B.Tech</td><td>Type</td><td>New</td><td>■</td><td>Revision</td><td colspan="2">□</td></tr>
<tr><td>To take effect from</td><td>July 2021</td><td rowspan="2">Submitted for approval</td><td colspan="5" rowspan="2">44th Senate</td></tr>
<tr><td>Prerequisite</td><td>Nil</td></tr>
<tr><td>Learning Objectives</td><td colspan="7">To introduce the basic understanding of digital representation, Boolean algebra and the operation of the logic components, combinational and sequential circuits, and to introduce the analog device concepts like diode, FET and op-amp.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="7">● To understand Digital Number systems, fixed and floating point representation and arithmetic operations.
● To use Boolean Algebra and Switching theory for Logic minimization.
● To implement Combinational Circuits using Primitive gates and logic functions.
● To implement sequential circuit elements and finite state machines.
● To design various circuits using Op-Amp 741 such as summing, difference, average, logarithmic amplifiers etc.</td></tr>
<tr><td>Course Contents (with approximate breakup of hours for lecture/ tutorial/practice)</td><td colspan="7">• Digital Circuits: Number Representation: Fixed point and floating point, 1’s and 2’s complement. Switching Theory: Boolean algebra, Switching functions, Truth Tables and Algebraic forms, Simplification of Boolean expressions – Algebraic methods, canonical forms and Minimization of functions using K-Maps. (5L,1T)
• Binary Codes: BCD, Gray, Excess 3, Alpha Numeric codes and conversion circuits. (3L,1T)
• Arithmetic circuits: Binary adders and subtractors, multipliers and division, ALU. (5L,2T)
• Synthesis of combinational logic functions using MSIs: mux/demux, decoders/encoders, Priority encoders, Comparators. (2L,2T)
• Sequential Circuits: Latches and Flip-Flops: SR, JK, D, T; Excitation tables. (2L,1T)
• Shift Registers, Counters, Random Access Memory. (3L,1T)
• Synchronous sequential circuits: Finite State Machines- Mealy & Moore types- Basic design steps- Design of counters, sequence generators, and sequence detectors - Design of simple synchronous machines – state minimization. (8L,3T)
• Analog Circuits: Diodes – Basics and Circuits – Clippers, Clampers, rectifiers. (3L,1T)
• Operational amplifiers (op-amp) – Basics and op-amp circuits – non inverting and inverting amplifiers – Signal offset. (4L,1T)
• Analog to Digital and Digital to Analog Conversion and circuits, Applications of Digital ICS: 555 Timer, V to F converters, Introduction to Logic Families, Noise in Digital System. (7L,1T)</td></tr>
<tr><td>Essential Reading</td><td colspan="7">1. M. Mano and C. Kime, “Logic and Computer Design Fundamentals,” Prentice Hall, Upper Saddle River, NJ, 4 th Edition, ISBN-13 : 978-9332518728, 2008.
2. B. Razavi, “Fundamentals of Microelectronics,” Wiley Student Edition, ISBN: 978-1-118-15632-2, 2010.</td></tr>
<tr><td>Supplementary Reading</td><td colspan="7">1. Sedra and Smith, Microelectronic Circuits, 7 th Edition, ISBN-13 : 978-0198089131, Oxford University Press, 2013.
2. J. F. Wakerly, “Digital Design - Principles and Practices,” 3 rd Edition, Pearson, ISBN-13 : 978-9332508125, 2008.
3. M. M. Mano, “Digital Design,” PHI, ISBN-13: 978-0-13-277420-8, 1979.
4. S. Franco, “Design with Operational Amplifiers and Analog Integrated Circuits,” McGraw-Hill Series in Electrical and Computer Engineering, 4th Edition, ISBN-13 : 978-0072320848, 2015.
5. R. J. Tocci, N. S. Widmer, and G. L. Moss, “Digital Systems Principles and applications,” Pearson Prentice Hall,10 th Edition, ISBN-13 : 978-0135103821, 2010.</td></tr>
</table>

<table>
<tr><td>Course Title</td><td colspan="2">Design and Analysis of Algorithms</td><td>Course No</td><td colspan="6"></td></tr>
<tr><td rowspan="2">Department/ Specialization</td><td colspan="2" rowspan="2">Computer Science and Engineering</td><td rowspan="2">Credits</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td>3</td><td>1</td><td>0</td><td>4</td></tr>
<tr><td>Faculty proposing the course</td><td colspan="2">Faculty, Department of CSE</td><td>Status</td><td>Core</td><td>■</td><td>Elective</td><td>□</td></tr>
<tr><td>Offered for</td><td colspan="2">B.Tech</td><td>Type</td><td>New</td><td>■</td><td>Revision</td><td>□</td></tr>
<tr><td>To take effect from</td><td colspan="2">July 2021</td><td rowspan="2">Submitted for approval</td><td colspan="4" rowspan="2">44th Senate</td></tr>
<tr><td>Prerequisite</td><td colspan="2">Nil</td></tr>
<tr><td>Learning Objectives</td><td colspan="7">• To design time or space efficient algorithms using well known paradigms.
• To understand the limitations of computing machines.
• To explore tractable vs intractable problems.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="7">• To design efficient algorithms using paradigms such as divide and conquer, dynamic programming, greedy method etc.
• To differentiate easy vs hard problems.
• To design polynomial-time algorithms with proof of correctness.</td></tr>
<tr><td>Course Contents (with approximate breakup of hours for lecture/ tutorial/practice)</td><td colspan="7">• Review of time/space complexity – recurrence relations – recurrence tree method – masters theorem (5L,2T)

• Incremental and decremental strategies – divide and conquer – case studies – lower bounds for sorting (5L,3T)

• Greedy Method – Container loading – knapsack – scheduling – coin change – proof of correctness (8L,2T)

• Dynamic programming – matrix chain, optimal binary search tree, travelling salesman, LCS, knapsack , greedy vs dynamic programming – Principle of optimality, overlapping subproblems – Dynamic programming vs Divide and Conquer (8L,2T)

• Graph algorithms – Topological sort – Shortest path algorithms – Dijskstra’s Algorithm, – Bellman-Ford’s Algorithm – minimum spanning tree – Principle of optimality (8L,2T)

• Tractability - Introduction to NP-completeness – NP, NP-hardness , polynomial-time reductions (6L,1T)
• Coping with intractable problems - Branch and bound – Back tracking – case studies (5L,1T)
• Solvable vs Unsolvable problems – Halting problem, Reducibility to Halting problem (3L)</td></tr>
<tr><td>Essential Reading</td><td colspan="7">1. T. H. Cormen, C. E. Leiserson, and R. L. Rivest, “Introduction to Algorithms,” Prentice Hall India, 2 nd Edition, 2001. ISBN 978-0-262-53305-8
2. E. Horowitz, S. Sahni, and S. Rajasekaran, “Computer Algorithms,” 2 nd Edition, Galgotia Publications, 2007. ISBN 0-7167-8316-9</td></tr>
<tr><td>Supplementary Reading</td><td colspan="7">1. Aho, Hopcroft, and Ullmann, “Data Structures & Algorithms,” Addison Wesley, 1983. ISBN13: 9780201000238
2. Algorithm Design , Eva Tardos and Kleinberg, Pearson, 2006, ISBN-13 : 978-0321295354</td></tr>
</table>

<table>
<tr><td>Course Title</td><td>Digital System Design Practice</td><td>Course No</td><td colspan="5"></td></tr>
<tr><td rowspan="2">Department/ Specialization</td><td rowspan="2">Computer Science and Engineering</td><td rowspan="2">Credits</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td>0</td><td>0</td><td>4</td><td>2</td></tr>
<tr><td>Faculty proposing the course</td><td>Faculty, Department of CSE</td><td>Status</td><td>Core</td><td>■</td><td>Elective</td><td>□</td></tr>
<tr><td>Offered for</td><td>B.Tech</td><td>Type</td><td>New</td><td>■</td><td>Revision</td><td>□</td></tr>
<tr><td>To take effect from</td><td>July 2021</td><td rowspan="2">Submitted for approval</td><td colspan="4" rowspan="2">44th Senate</td></tr>
<tr><td>Prerequisite</td><td>Nil</td></tr>
<tr><td>Learning Objectives</td><td colspan="6">To provide hands on design and implementation of analog and digital circuits. Students will build simple digital systems on general purpose PCBs.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="6">● To implement and verify logic circuits
● To implement and verify arithmetic circuits using discrete components
● To implement and verify digital systems using Combinational/ Sequential elements
● To implement and verify analog circuits</td></tr>
<tr><td>Course Contents (with approximate breakup of hours for lecture/ tutorial/practice)</td><td colspan="6">• Design and implementation of logic functions, combinational circuits (code converters, half & full adders, comparator, ripple carry adder, priority encoder, Decoders, Seven segment display, multiplexer)
• Design of sequential Circuits.
• Design of 4-bit ALU (Adder, subtractor, logic and shift operations).
• Design project
• Static characteristics of rectifiers and filters, clipping and clamping circuits, Op-Amp based amplifier circuits.
• Design and implementation of a digital system.</td></tr>
<tr><td>Essential Reading</td><td colspan="6">1. S. Franco, “Design with Operational Amplifiers and Analog Integrated Circuits,” McGraw-Hill Series in Electrical and Computer Engineering, 4th Edition, ISBN-13 : 978-0072320848, 2015.
2. S. Brown and Z. Vranesic, “Fundamentals of Digital Logic with VHDL Design,”TMH, 3 rd Edition, ISBN-13 : 978-0077221430, 2008.</td></tr>
<tr><td>Supplementary Reading</td><td colspan="6">1. R.J. Tocci, N. S.Widmer, and G. L. Moss, “Digital Systems Principles and applications,” Pearson Prentice Hall, 10 th Edition, ISBN-13 : 978-0135103821, 2010.
2. D. A. Neaman, “Electronic Circuits,” TMH, 4 th Edition,ISBN-13 : 978-0070634336, 2006.</td></tr>
</table>

<table>
<tr><td>Course Title</td><td>Design and Analysis of Algorithms Practice</td><td>Course No</td><td colspan="6"></td></tr>
<tr><td rowspan="2">Department/ Specialization</td><td rowspan="2">Computer Science and Engineering</td><td rowspan="2">Credits</td><td>L</td><td>T</td><td>P</td><td colspan="2">C</td></tr>
<tr><td>0</td><td>0</td><td>4</td><td colspan="2">2</td></tr>
<tr><td>Faculty proposing the course</td><td>Faculty, Department of CSE</td><td>Status</td><td>Core</td><td>■</td><td>Elective</td><td colspan="2">□</td></tr>
<tr><td>Offered for</td><td>B.Tech</td><td>Type</td><td>New</td><td>■</td><td>Revision</td><td colspan="2">□</td></tr>
<tr><td>To take effect from</td><td>July 2021</td><td rowspan="2">Submitted for approval</td><td colspan="5" rowspan="2">44th Senate</td></tr>
<tr><td>Prerequisite</td><td>Nil</td></tr>
<tr><td>Learning Objectives</td><td colspan="7">• To design time or space efficient algorithms using well known paradigms.
• To understand the limitations of computing machines.
• To explore tractable vs intractable problems.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="7">• To design efficient algorithms using paradigms such as divide and conquer, dynamic programming, greedy method etc.
• To differentiate easy vs hard problems.
• To design polynomial-time algorithms with proof of correctness.</td></tr>
<tr><td>Course Contents (with approximate breakup of hours for lecture/ tutorial/practice)</td><td colspan="7">• The laboratory component will require the student to write computer programs using a careful choice of data structures and algorithmic paradigms (in C++/Java language) from scratch, based on the concepts learnt in the theory course.

• Case studies in respect of different paradigms discussed in theory shall be implemented in C++/Java

• Paradigms – Divide and conquer, dynamic programming, greedy, backtracking.</td></tr>
<tr><td>Essential Reading</td><td colspan="7">1. T. H. Cormen, C. E. Leiserson, and R. L. Rivest, “Introduction to Algorithms,” Prentice Hall India, 2 nd Edition, 2001. ISBN 978-0-262-53305-8
2. E. Horowitz, S. Sahni, and S. Rajasekaran, “Computer Algorithms,” 2 nd Edition, Galgotia Publications, 2007. ISBN 0-7167-8316-9</td></tr>
<tr><td>Supplementary Reading</td><td colspan="7">1. Aho, Hopcroft, and Ullmann, “Data Structures & Algorithms,” Addison Wesley, 1983. ISBN13: 9780201000238
2. Algorithm Design , Eva Tardos and Kleinberg, Pearson, 2006, ISBN-13 : 978-0321295354</td></tr>
</table>

<table>
<tr><td>Course Title</td><td>Computer Organization and Architecture</td><td>Course No</td><td colspan="8"></td></tr>
<tr><td rowspan="2">Department/ Specialization</td><td rowspan="2">Computer Science and Engineering</td><td rowspan="2">Credits</td><td>L</td><td>T</td><td>P</td><td colspan="4">C</td></tr>
<tr><td>3</td><td>1</td><td>0</td><td colspan="4">4</td></tr>
<tr><td>Faculty proposing the course</td><td>Faculty, Department of CSE</td><td>Status</td><td>Core</td><td>■</td><td colspan="3">Elective</td><td>□</td></tr>
<tr><td>Offered for</td><td>B.Tech</td><td>Type</td><td>New</td><td>■</td><td colspan="3">Revision</td><td>□</td></tr>
<tr><td>To take effect from</td><td>July 2021</td><td rowspan="2">Submitted for approval</td><td colspan="6" rowspan="2">44th Senate</td></tr>
<tr><td>Prerequisite</td><td>Nil</td></tr>
<tr><td>Learning Objectives</td><td colspan="9">The course aims to introduce various aspects of computer organization such as Instruction format, Instruction codes, Addressing Modes, processor design and hierarchical memory design, Input and Output Interface design using Programmed Controlled and Interrupt Control way</td></tr>
<tr><td>Learning Outcomes</td><td colspan="9">● Understand the organization of a Computer system and ISAs
● Apply the knowledge of combinational and sequential logical circuits to design computer architecture.
● Understand the input / output and Memory related concepts.
● Analyze the performance of different scalar Computers
● Develop the Pipelining Concept for a given set of Instructions
● Distinguish the performance of pipelining and non pipelining environment in a processor</td></tr>
<tr><td>Course Contents (with approximate breakup of hours for lecture/ tutorial/practice)</td><td colspan="9">• Introduction: function and structure of a computer, functional components of a computer, performance of a computer system. Instruction set architectures – CISC and RISC architectures.(5L,1T)
• Instructions: Language of the Computer, Operations of the Computer Hardware, Operands of the Computer Hardware, Representing Instructions in the Computer, Logical Operations Instructions for Making Decisions, addressing Modes, Parallelism & Instructions. (5L,1T)
• Arithmetic Design: – Carry look ahead adder, Wallace tree multiplier, Floating–point adder/subtractor, Division. (5L,2T)
• The Processor: Logic Design Conventions, Building a Datapath, A Simple Implementation Scheme (3L,1T)
• An Overview of Pipelining, Pipelined Data path and Control, Data Hazards: Forwarding versus Stalling, Control Hazards, Exceptions and Parallelism via Instructions. (7L,2T)
• Memory Hierarchy: Introduction, Memory Technologies (SRAM, DRAM), The Basics of Caches, Measuring and Improving Cache Performance, Dependable Memory, Virtual Machines, Virtual Memory, A Common Framework for Memory Hierarchy, Using a Finite State Machine to Control a Simple Cache, Parallelism and Memory Hierarchies: Cache Coherence, Parallelism and Memory Hierarchy: Redundant Arrays of Inexpensive Disks and
• Implementing Cache Controllers. (9L,2T)
• Input/Output Unit: access of I/O devices, I/O ports, I/O control mechanisms – Program Controlled I/O. Interrupt controlled I/O and DMA controlled I/O; I/O interfaces – Serial port, parallel port, USB port, SCSI bus, PCI bus; I/O peripherals – Keyboard, display, secondary storage devices. (8L,2T)</td></tr>
<tr><td>Essential Reading</td><td colspan="9">1. Patterson and Hennessy, “Computer Organization and Design,” Morgan Kaufmann, 5 th Edition, ISBN-13 : 978-8131222744, 2013.
2. C. Hamacher, Z. Vranesic, and S. Zaky, “Computer Organization,” Tata McGraw Hill, 5 th Edition, ISBN-9789339212131, 2002.</td></tr>
<tr><td>Supplementary Reading</td><td colspan="9">1. J. P. Hayes, “Computer Architecture and Organization,” Tata McGraw Hill, ISBN-13 : 978-1259028564, 2017.
2. M. J. Murdocca, V. P. Heuring, “Computer Architecture and Organization - An Integrated Approach,” John Wiley & Sons Inc., ISBN-13:978-0471733881, 2007.
3. A. S. Tanenbaum, “Structured Computer Organization,” Prentice Hall, 5th Edition, ISBN-13 : 978-0132916523, 2006.</td></tr>
</table>

<table>
<tr><td>Course Title</td><td>Database Systems</td><td>Course No</td><td colspan="6"></td></tr>
<tr><td rowspan="2">Department/ Specialization</td><td rowspan="2">Computer Science and Engineering</td><td rowspan="2">Credits</td><td>L</td><td>T</td><td>P</td><td colspan="2">C</td></tr>
<tr><td>3</td><td>1</td><td>0</td><td colspan="2">4</td></tr>
<tr><td>Faculty proposing the course</td><td>Faculty, Department of CSE</td><td>Status</td><td>Core</td><td>■</td><td>Elective</td><td colspan="2">□</td></tr>
<tr><td>Offered for</td><td>B.Tech</td><td>Type</td><td>New</td><td>■</td><td>Revision</td><td colspan="2">□</td></tr>
<tr><td>To take effect from</td><td>July 2021</td><td rowspan="2">Submitted for approval</td><td colspan="5" rowspan="2">44th Senate</td></tr>
<tr><td>Prerequisite</td><td>Nil</td></tr>
<tr><td>Learning Objectives</td><td colspan="7">Objective of the course is to equip students with skillsets required for database design and implementation. Various concepts such as ER modeling, Schema Mapping, Normalization, Lossless Join etc. would be explored to help in efficient an and effective databases.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="7">● To appreciate the systematic design and principles involved in any database development.
● To understand the Importance of canonical normal forms and its design in large scale database systems
● To design and implement Database with formal analysis and design thinking</td></tr>
<tr><td>Course Contents (with approximate breakup of hours for lecture/ tutorial/practice)</td><td colspan="7">Introduction to Database Systems, Database System Architecture, Schema, Database Models, Relational Model, ER Modelling and case studies. (7L,2T)
Expressive power of relational databases, Relational Algebra (5L,2T)
Database Languages, DDL, DML, Structured Query Language (SQL), SQL views, case studies (8L,3T)
Database Design, Normal Forms (First to third normal form), Boyce codd Normal Form, Database decomposition, Functional Dependencies, Loss-less Join decomposition (8L,2T)
Transaction Processing and Concurrency control (4L,1T)
Internal schema Design, Indexing, B-trees, B+ trees (5L,2T)
Introduction to advanced concepts like Data mining, Data warehousing, XML(5L)</td></tr>
<tr><td>Essential Reading</td><td colspan="7">1. R. Elmasri and S. B. Navathe, “Fundamentals of Database Systems,” Pearson, 7th Edition, 2016, ISBN 9789332582705</td></tr>
<tr><td>Supplementary Reading</td><td colspan="7">1. A. Silberschatz, H. F. Korth, and S. Sudharsan, “Database System Concepts,” Tata McGraw Hill, 6th Edition, 2011, ISBN 9332901384.
2. C. J. Date, A. Kannan, and S. Swamynathan, “An Introduction to Database Systems,” Pearson, 8th Edition, 2006, ISBN 978-0321197849</td></tr>
</table>

<table>
<tr><td>Course Title</td><td>Theory of Computation</td><td>Course No</td><td colspan="6"></td></tr>
<tr><td rowspan="2">Department/ Specialization</td><td rowspan="2">Computer Science and Engineering</td><td rowspan="2">Credits</td><td>L</td><td>T</td><td>P</td><td colspan="2">C</td></tr>
<tr><td>3</td><td>1</td><td>0</td><td colspan="2">4</td></tr>
<tr><td>Faculty proposing the course</td><td>Faculty, Department of CSE</td><td>Status</td><td>Core</td><td>■</td><td>Elective</td><td colspan="2">□</td></tr>
<tr><td>Offered for</td><td>B.Tech</td><td>Type</td><td>New</td><td>■</td><td>Revision</td><td colspan="2">□</td></tr>
<tr><td>To take effect from</td><td>July 2021</td><td rowspan="2">Submitted for approval</td><td colspan="5" rowspan="2">44th Senate</td></tr>
<tr><td>Prerequisite</td><td>Nil</td></tr>
<tr><td>Learning Objectives</td><td colspan="8">This course aims to provide fundamentals of computing models such as finite state automata, push down automata, linear bounded automata and Turing machine. Powers and limitations of the models will also be discussed. Solvability and Tractability will be introduced through Turing machine</td></tr>
<tr><td>Learning Outcomes</td><td colspan="8">• To design various computational models useful for solving problems
• To understand the relationship among digital computer, algorithm and Turing machine.
• To verify whether a given problem is solvable or tractable.</td></tr>
<tr><td>Course Contents (with approximate breakup of hours for lecture/ tutorial/practice)</td><td colspan="8">Finite Automata & Regular Languages - (10L,3T)
Languages vs Problems. Finite State Automata, Regular Languages. Closure properties, Limitations, Pumping Lemma, Myhill-Nerode relations, Quotient Construction. Minimization Algorithm.
Non-determinism, Regular Grammar & Regular Expressions - (10L,3T)
Notion of non-determinism. Acceptance condition. Equivalence of NFA and DFA. Regular Grammar and NFA, Pattern matching and regular expressions. Regular Expressions and Regular languages. More closure properties of regular languages.
Push Down Automata & Context-free Languages (CFLs) - (12L,4T)
Grammars and Chomsky Hierarchy, CFLs, Chomsky Normal Form, Pumping Lemma for CFLs, Inherent Ambiguity of Context-Free Languages, Cock-Younger-Kasami Algorithm, Applications to Parsing. Pushdown Automata (PDA), PDA vs CFLs. Non-equivalence of Deterministic and non- deterministic versions of PDA. Deterministic CFLs.
Linear Bounded Automata, Turing Machines & Computability - (12L,4 T)
Introduction to Linear Bounded Automata (LBA), Turing Machines. Context Sensitive Language Vs LBA. Turing Machine vs Phrase Structure Language. Multi-tape Turing machines. Recursive and Recursively enumerable languages. Undecidability of Halting Problem. Reductions. Introduction to Theory of NP-completeness.</td></tr>
<tr><td>Essential Reading</td><td colspan="8">1. Introduction to Automata Theory, Languages and Computation, Hopcroft, Motwani, and Ullman, Pearson Publishers, Third Edition, ISBN: 9780321455369, 2006.</td></tr>
<tr><td>Supplementary Reading</td><td colspan="8">1. Elements of the Theory of Computation, H. R. Lewis and C.H. Papadimitriou, Prentice Hall Publishers, ISBN. 0-13-2624 78-8, 1981
2. Introduction to Languages and the Theory of Computation, John. C. Martin, Tata McGraw-Hill, ISBN 978-00731914612003.</td></tr>
</table>

<table>
<tr><td>Course Title</td><td>Computer Organization and Architecture Practice</td><td>Course No</td><td colspan="6"></td></tr>
<tr><td rowspan="2">Department/ Specialization</td><td rowspan="2">Computer Science and Engineering</td><td rowspan="2">Credits</td><td>L</td><td>T</td><td>P</td><td colspan="2">C</td></tr>
<tr><td>0</td><td>0</td><td>4</td><td colspan="2">2</td></tr>
<tr><td>Faculty proposing the course</td><td>Faculty, Department of CSE</td><td>Status</td><td>Core</td><td>■</td><td>Elective</td><td colspan="2">□</td></tr>
<tr><td>Offered for</td><td>B.Tech</td><td>Type</td><td>New</td><td>■</td><td>Revision</td><td colspan="2">□</td></tr>
<tr><td>To take effect from</td><td>July 2021</td><td rowspan="2">Submitted for approval</td><td colspan="5" rowspan="2">44th Senate</td></tr>
<tr><td>Prerequisite</td><td>Nil</td></tr>
<tr><td>Learning Objectives</td><td colspan="7">Exposure to assembly language programming, instruction set design, and processor design for a given instruction set are given. Assembler macros, interrupt service routines, and simple device driver programs would also be introduced. Computer system design concepts are introduced.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="7">● Assembly Language Instructions and programming
● Machine code based program execution
● Input and output device interfacing and programming
● Programming Interrupt service routines
● Writing device driver program to control and monitor the peripheral device</td></tr>
<tr><td>Course Contents (with approximate breakup of hours for lecture/ tutorial/practice)</td><td colspan="7">Exercises will mainly involve writing the assembly language programs - Execution of assembly language programs: Single–step, break points, Accessing the contents of registers, accessing the contents of memory locations - Implementation of higher level language assignment statements with arithmetic expressions and logical expressions - Implementation of control transfer statements. Macros - Software interrupts - Operating system function calls - Interrupt service routines - Simple device drivers - Assembly language programming in C language. I/O interfacing and programming. Computer System Design.</td></tr>
<tr><td>Essential Reading</td><td colspan="7">1. Patterson and Hennessy, “Computer Organization and Design,” Morgan Kaufmann, 5 th Edition, ISBN-13 : 978-8131222744, 2013.</td></tr>
<tr><td>Supplementary Reading</td><td colspan="7">1. C. Hamacher, Z. Vranesic, and S. Zaky, “Computer Organization,” Tata McGraw Hill, ISBN-9789339212131, 2002.</td></tr>
</table>

<table>
<tr><td>Course Title</td><td>Database Systems Practice</td><td>Course No</td><td colspan="5"></td></tr>
<tr><td rowspan="2">Department/ Specialization</td><td rowspan="2">Computer Science and Engineering</td><td rowspan="2">Credits</td><td>L</td><td>T</td><td>P</td><td colspan="2">C</td></tr>
<tr><td>0</td><td>0</td><td>4</td><td colspan="2">2</td></tr>
<tr><td>Faculty proposing the course</td><td>Faculty, Department of CSE</td><td>Status</td><td>Core</td><td>■</td><td>Elective</td><td colspan="2">□</td></tr>
<tr><td>Offered for</td><td>B.Tech</td><td>Type</td><td>New</td><td>■</td><td>Revision</td><td colspan="2">□</td></tr>
<tr><td>To take effect from</td><td>July 2021</td><td rowspan="2">Submitted for approval</td><td colspan="5" rowspan="2">44th Senate</td></tr>
<tr><td>Prerequisite</td><td>Nil</td></tr>
<tr><td>Learning Objectives</td><td colspan="7">The focus of this course is on database design, architecture, and relational models. Normal forms, internal schema design would also be explored. This course introduces SQL programming. Database design preserving functional dependencies and loss-less decomposition properties would be addressed.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="7">● Conceptual design using ER diagrams, programming using structured query language, Ability to Design and Implement Database based on formal guidelines
● Students would also be equipped with skills required for basic application development involving database connectivity.</td></tr>
<tr><td>Course Contents (with approximate breakup of hours for lecture/ tutorial/practice)</td><td colspan="7">Introduction to SQL. Schema, table creation using SQL, Data definition and data manipulation using SQL. Implementation of set theoretic operations on databases. Views using SQL. Implementation of algorithms related to functional dependencies and loss-less decomposition. Indexing using B-trees and B+ trees (creation, insertion, deletion).
Assignment/Mini project-based application design and development involving database</td></tr>
<tr><td>Essential Reading</td><td colspan="7">1. R. Elmasri and S. B. Navathe, “Fundamentals of Database Systems,” Pearson, 7th Edition, 2016, ISBN 9789332582705</td></tr>
<tr><td>Supplementary Reading</td><td colspan="7">1. A. Silberschatz, H. F. Korth, and S. Sudharsan, “Database System Concepts,” Tata McGraw Hill, 6th Edition, 2011, 978-0321197849
2. C. J. Date, A. Kannan, and S. Swamynathan, “An Introduction to Database Systems,” Pearson, 8th Edition, 2006, ISBN 978-0321197849</td></tr>
</table>

<table>
<tr><td>Course Title</td><td>Data Science –An Applied Perspective</td><td>Course No</td><td colspan="6"></td></tr>
<tr><td rowspan="2">Department/ Specialization</td><td rowspan="2">Computer Science and Engineering</td><td rowspan="2">Credits</td><td>L</td><td>T</td><td>P</td><td colspan="2">C</td></tr>
<tr><td>3</td><td>0</td><td>2</td><td colspan="2">4</td></tr>
<tr><td>Faculty proposing the course</td><td>Faculty, Dept. of CSE</td><td>Status</td><td>Core</td><td>■</td><td>Elective</td><td colspan="2">□</td></tr>
<tr><td>Offered for</td><td>B.Tech</td><td>Type</td><td>New</td><td>■</td><td>Revision</td><td colspan="2">□</td></tr>
<tr><td>To take effect from</td><td>July 2021</td><td rowspan="2">Submitted for approval</td><td colspan="5" rowspan="2">44th Senate</td></tr>
<tr><td>Prerequisite</td><td>Nil</td></tr>
<tr><td>Learning Objectives</td><td colspan="8">This course covers the basic concepts of Data Science to help the student to learn, understand and practice data analytics encompassing concepts from descriptive, inferential statistics and predictive techniques and big data concepts.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="8">● Ability to identify the characteristics of datasets ; Ability to select and implement machine learning techniques suitable for the respective application ;
● Ability to solve problems associated with big data characteristics such as high dimensionality;
● Ability to integrate machine learning libraries and mathematical and statistical tools</td></tr>
<tr><td>Course Contents (with approximate breakup of hours for lecture/ tutorial/practice)</td><td colspan="8">Introduction to relevant industry applications and analytics – Descriptive Statistics – Data Visualization & Interpretation -Measures of Central Tendency & Dispersion - Basic and advanced plots such as Stem-Leaf Plots, Histograms, Pie charts, Box Plots, Violin Plots etc. – Merits of Demerits & Interpretation (10)
Inferential Statistics – Hypothesis Testing - Tests of Significance – Analysis of Variance - Regression – Linear and Logistic (8)
Predictive Analytics – Supervised and Unsupervised – Association Rules, Classification, Clustering, Outlier Analysis, Time Series Modeling (14)
Big Data Characteristics – Map Reduce – Deduplication, Distributed Storage, Implementation using Hadoop / Pyspark platforms (8)
Practice Component: Concepts from Descriptive Statistics, Inferential and Predictive Analytics would be test driven using platforms such as Python, R etc. ML support in these platforms for rule mining and application, classification & clustering algorithms etc. would also be test driven as part of the practice exercises. Modern technologies for big data handling such as Pyspark – support for Map reduce would also be test driven. Applications relevant to the students stream of specialization would be explored for exercises / course project as case studies. (14 sessions – weekly exercises)</td></tr>
<tr><td>Essential Reading</td><td colspan="8">1. J Han, M Kamber, Data Mining Concepts & Techniques, Elsevier, 3rd Edition, 2007, ISBN 9780123814791</td></tr>
<tr><td>Supplementary Reading</td><td colspan="8">1. Joel Grus, Data Science from Scratch, Orielly, 2nd Edn, 2019, ISBN 9781492041139
2. Leskovec, Anand Rajaraman,, Ullmann, Mining of Massive Data Sets, Cambridge University Press, Open Source free version , ISBN 9781107015357
3. P Bruce, Practical Statistics for Data Scientists, O’Reilly, 2017, iSBN 9789352135653</td></tr>
</table>

<table>
<tr><td>Course Title</td><td>Operating Systems</td><td>Course No</td><td colspan="6"></td></tr>
<tr><td rowspan="2">Department/ Specialization</td><td rowspan="2">Computer Science and Engineering</td><td rowspan="2">Credits</td><td>L</td><td>T</td><td>P</td><td colspan="2">C</td></tr>
<tr><td>3</td><td>1</td><td>0</td><td colspan="2">4</td></tr>
<tr><td>Faculty proposing the course</td><td>Faculty, Department of CSE</td><td>Status</td><td>Core</td><td>■</td><td>Elective</td><td colspan="2">□</td></tr>
<tr><td>Offered for</td><td>B.Tech</td><td>Type</td><td>New</td><td>■</td><td>Revision</td><td colspan="2">□</td></tr>
<tr><td>To take effect from</td><td>July 2021</td><td rowspan="2">Submitted for approval</td><td colspan="5" rowspan="2">44th Senate</td></tr>
<tr><td>Prerequisite</td><td>Nil</td></tr>
<tr><td>Learning Objectives</td><td colspan="7">This first level course focuses on exposing students to the purpose, structure and functions of an operating system. Operating systems abstraction, mechanisms and their implementation support for concurrency (threads) and synchronization, resource management, scheduling strategies, etc. are explored.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="7">● Sound understanding of basic concepts relating to the design and implementation of an operating system.
● Specifics relating to scheduling, multithreading, synchronization, etc. to understand the structure of the operating system (Linux), at the concept and the source code level.
● Ability to use Kernel API support to implement various features to be supported by an OS</td></tr>
<tr><td>Course Contents (with approximate breakup of hours for lecture/ tutorial/practice)</td><td colspan="7">Functionalities & Services of an Operating System – System Calls & Types - Process Concept – Process Control Block – Linux System calls for Process creation, Inter Process Communication using Shared memory / Message passing. (10L,2T) Concurrency – Multithreaded programming – benefits, challenges, models, Pthreads library in Linux – thread creation, cancellation, thread specific data, Thread pools, Signal handling , Scheduling – Preemptive, Non preemptive algorithms FCFS, SJF, SRT, RR – Thread scheduling – contention scope, pthread support for scheduling. (11L,3T) Synchronization – Race condition – Critical Section Problem, Solution, Mutex Locks and Semaphores – Priority Inversion, Pthreads synchronization - Producer Consumer problem (multi threaded) example Deadlock characterization – Resource graph – Avoidance & Prevention – Safe state – Bankers algorithm – recovery schemes. (10L,3T)
Memory management – logical v/s physical address space – Segmentation, Paging, Page table structures , Virtual memory, Page replacement strategies, File Systems – file operations, types, access methods, Directory structure, Mounting file systems. (11L,3T) Introduction to operating systems for hand held devices - RTOS, Free RTOS</td></tr>
<tr><td>Essential Reading</td><td colspan="7">1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Concepts, John Wiley, 9 th Edn, 2015, ISBN 978-0471694663</td></tr>
<tr><td>Supplementary Reading</td><td colspan="7">1. Andrew S Tanenbaum, Modern Operating Systems, Prentice Hall, 2009, ISBN 9788120339040
2. Stallings. W, Operating System: Internals and Design Principles, Prentice Hall, 2011, ISBN 9332518807
3. Gary Nut, Operating Systems: A Modern Perspective, Addison Wesley, 2003, ISBN 978-0201773446</td></tr>
</table>

<table>
<tr><td>Course Title</td><td>Computer Networking</td><td>Course No</td><td colspan="6"></td></tr>
<tr><td rowspan="2">Department/ Specialization</td><td rowspan="2">Computer Science and Engineering</td><td rowspan="2">Credits</td><td>L</td><td>T</td><td>P</td><td colspan="2">C</td></tr>
<tr><td>3</td><td>1</td><td>0</td><td colspan="2">4</td></tr>
<tr><td>Faculty proposing the course</td><td>Faculty, Department of CSE</td><td>Status</td><td>Core</td><td>■</td><td>Elective</td><td>□</td></tr>
<tr><td>Offered for</td><td>B.Tech</td><td>Type</td><td>New</td><td>■</td><td>Revision</td><td>□</td></tr>
<tr><td>To take effect from</td><td>July 2021</td><td rowspan="2">Submitted for approval</td><td colspan="4" rowspan="2">44th Senate</td></tr>
<tr><td>Prerequisite</td><td>Nil</td></tr>
<tr><td>Learning Objectives</td><td colspan="7">To introduce the basics of computer networking, error detection and correction techniques, and flow control techniques. Also an exposure to IP addressing and routing and its associated protocols would be given. A highlight of various application layer protocols and its relevance in modern networking world would be discussed.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="7">● To design a local area network and analyze the network using performance metrics.
● To appreciate the importance of subnetting, masking, and nuances involved in setting up a campus network.</td></tr>
<tr><td>Course Contents (with approximate breakup of hours for lecture/ tutorial/practice)</td><td colspan="7">Evolution of computer networks, creating a small network, Data transfer between nodes, encoding of bits in physical layer, NRZ, Manchester, Differential Manchester, Performance evaluation of a network: propagation delay, transmission delay, RTT, effective bandwidth. (10L,3T)
Error detection techniques in Data link layer (LRC, CRC, Two dimensional parity check), Hamming Error correcting codes. Data transfer between nodes using stop and wait protocol, sliding window protocol (Go-back-n and selective reject), performance analysis of stop and wait and sliding window protocols. Flow control at data link layer. Introduction to layer-2 devices (switches, bridges) and addressing scheme at Layer-2 (MAC addresses). (10L,3T)
Creating a small network using Ethernet (IEEE 802.3) Token Ring (IEEE 802.5), Performance evaluation of IEEE 802.3 and 802.5 networks. Introduction to Layer-3 devices, IP addresses, IPv4,IPv6, Error detection at layer-3 using Checksum. IP addressing schemes, subnetting, CIDR (10L,3T)
Introduction to TCP/IP, IP routing, RIP, OSPF, Circuit and Packet switching, ICMP, Introduction to networking commands: Ping, Traceroute, IPconfig, UDP, congestion control and avoidance. (10L,3T)
Introduction to DHCP, FTP, HTTP(s) and other application layer protocols, Introduction to Network security. (5L)</td></tr>
<tr><td>Essential Reading</td><td colspan="7">1. Larry L.Peterson and Bruce S Davie, Computer Networks: A systems Approach,Morgan, 5th Edn, 2011. ISBN: 9780123850591
2. William Stallings, Data and Computer Communications, 10th Edn, Pearson, 2017. ISBN: 9780133506488</td></tr>
<tr><td>Supplementary Reading</td><td colspan="7">1. Andrew S. Tanenbaum, Computer Networks, 5th Edn, 2014. ISBN: 9788131770221
2. Behrouz Forouzan, TCP/IP protocol suite, Tata McGraw Hill, 4th Edn, 2010. ISBN: 9780070706521</td></tr>
</table>

<table>
<tr><td>Course Title</td><td>Compiler Design</td><td>Course No</td><td colspan="6"></td></tr>
<tr><td rowspan="2">Department/ Specialization</td><td rowspan="2">Computer Science and Engineering</td><td rowspan="2">Credits</td><td>L</td><td>T</td><td>P</td><td colspan="2">C</td></tr>
<tr><td>3</td><td>1</td><td>0</td><td colspan="2">4</td></tr>
<tr><td>Faculty proposing the course</td><td>Faculty, Department of CSE</td><td>Status</td><td>Core</td><td>■</td><td>Elective</td><td colspan="2">□</td></tr>
<tr><td>Offered for</td><td>B.Tech</td><td>Type</td><td>New</td><td>■</td><td>Revision</td><td colspan="2">□</td></tr>
<tr><td>To take effect from</td><td>July 2021</td><td rowspan="2">Submitted for approval</td><td colspan="5" rowspan="2">44th Senate</td></tr>
<tr><td>Prerequisite</td><td>Nil</td></tr>
<tr><td>Learning Objectives</td><td colspan="7">The objective of this course is to train students to design various phases of compiler such as Lexical analyzer, syntax analyzer, semantic analyzer, intermediate code generator, code optimizer and code generator. Students are also exposed to design compiler construction tools such as Lexical Analyzer generator and parser generator. Applications of finite state machine and pushdown automation in compiler design are also taught in this course.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="7">● At the end of the course, students will be able to design a programming language and compiler for the same.
● Students will also be able to write large programs.</td></tr>
<tr><td>Course Contents (with approximate breakup of hours for lecture/ tutorial/practice)</td><td colspan="7">Need of compiler-cross compiler-Introduction to phases of compiler –Lexical Analyzer Design using DFAs —regular expression and its application to give syntax of word –Automatic design of Lexical Analyzer from regular expression, Construction of NFA without epsilon moves from regular expression- Efficient Lexical analyzer using Minimization of automata- limitation of recognition capability of Lexical analyzer using Pumping lemma (12L,3T)
Context free grammar & its application to give syntax of program statement – Types of parsing – Top down & bottom up–Recursive descent– Predictive–Shift reduce–Operator precedence–SLR (10L,3T)
Semantic analysis - Intermediate code generation: Declaration – Assignment statements – Boolean expressions– looping and branching statements (7L,2T)
Back patching and procedure calls code generator design issues – Runtime storage management – Code Optimization: Basic blocks – Flow graphs – Next use information – Code generator case study – Directed acyclic graph representation of basic blocks – Peephole optimization technique Introduction to code optimization (10L,3T)
Storage optimization & allocation strategies).Assembly Code Generation: from syntax tree and Directed acyclic graph - from three address code. (5L,1T)</td></tr>
<tr><td>Essential Reading</td><td colspan="7">1. Alfred Aho, Ravi Sethi and Jeffrey D Ullman, Compilers Principles, Techniques and Tools, Pearson Education, 2003. ISBN: 9780321491695</td></tr>
<tr><td>Supplementary Reading</td><td colspan="7">1. Levine J.R, Mason T, Brown D, Lex & Yacc, OReilly Associates, 1992 ISBN: 9781565920002.
2. Allen I. Holub, Compiler Design in C, Prentice Hall, 2003. ISBN: 9780131550452</td></tr>
</table>

<table>
<tr><td>Course Title</td><td>Operating System Practice</td><td>Course No</td><td colspan="6"></td></tr>
<tr><td rowspan="2">Department/ Specialization</td><td rowspan="2">Computer Science and Engineering</td><td rowspan="2">Credits</td><td>L</td><td>T</td><td>P</td><td colspan="2">C</td></tr>
<tr><td>0</td><td>0</td><td>4</td><td colspan="2">2</td></tr>
<tr><td>Faculty proposing the course</td><td>Faculty, Department of CSE</td><td>Status</td><td>Core</td><td>■</td><td>Elective</td><td>□</td></tr>
<tr><td>Offered for</td><td>B.Tech</td><td>Type</td><td>New</td><td>■</td><td>Revision</td><td>□</td></tr>
<tr><td>To take effect from</td><td>July 2021</td><td rowspan="2">Submitted for approval</td><td colspan="4" rowspan="2">44th Senate</td></tr>
<tr><td>Prerequisite</td><td>Nil</td></tr>
<tr><td>Learning Objectives</td><td colspan="7">The course aims to equip the student with implementation level constructs / support in Linux for various concepts such as process management, concurrency, scheduling, deadlock avoidance, etc.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="7">● To relate the operating system concepts listed above to the Linux operating system and support for the same available through various system calls.
● To use LINUX Kernel Support for various features such as multiprocessing multithreading etc.
● To Test Drive various Features of an OS relating to application scenario</td></tr>
<tr><td>Course Contents (with approximate breakup of hours for lecture/ tutorial/practice)</td><td colspan="7">Linux System Calls for process creation, management – Applications such as command prompt simulator using fork – Interprocess Communication using Shared Memory and Pipes – Producer Consumer – Applications using pipes / shm – Concurrency – Multithreading –Pthread support – Applications such as merge sort, min-max-average, etc. in a multi threaded fashion – Scheduling –pthread interfaces setschedpolicy – getschedpolicy based applications – Synchronization – threaded solution for classical problems like dining philosophers, readers writers, etc. using mutex locks and semaphores - Deadlock detection / avoidance algorithms.</td></tr>
<tr><td>Essential Reading</td><td colspan="7">1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Concepts, John Wiley, 9 th Edn, 2015, ISBN 9788120339040</td></tr>
<tr><td>Supplementary Reading</td><td colspan="7">1. Robert Love, Linux Systems Programming, O Reilly Media, 2 nd Edition, 2013, ISBN 9781449339531
2. D Butlar, J Farrell, B Nichols, Pthreads Programming, O Reilly Media, 1996, ISBN 9781565921153</td></tr>
</table>

<table>
<tr><td>Course Title</td><td>Computer Networking Practice</td><td>Course No</td><td colspan="6"></td></tr>
<tr><td rowspan="2">Department/ Specialization</td><td rowspan="2">Computer Science and Engineering</td><td rowspan="2">Credits</td><td>L</td><td>T</td><td>P</td><td colspan="2">C</td></tr>
<tr><td>0</td><td>0</td><td>4</td><td colspan="2">2</td></tr>
<tr><td>Faculty proposing the course</td><td>Faculty, Department of CSE</td><td>Status</td><td>Core</td><td>■</td><td>Elective</td><td colspan="2">□</td></tr>
<tr><td>Offered for</td><td>B.Tech</td><td>Type</td><td>New</td><td>■</td><td>Revision</td><td colspan="2">□</td></tr>
<tr><td>To take effect from</td><td>July 2021</td><td rowspan="2">Submitted for approval</td><td colspan="5" rowspan="2">44th Senate</td></tr>
<tr><td>Prerequisite</td><td>Nil</td></tr>
<tr><td>Learning Objectives</td><td colspan="7">To understand basic networking commands, MAC/IP addressing, file transfer between two systems, etc. Simulation of error control techniques and flow control techniques using well known protocols would be addressed as part of this course.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="7">● To design, test and troubleshoot aspects associated with local area networking.
● To appreciate the importance of error detecting codes and flow control techniques.</td></tr>
<tr><td>Course Contents (with approximate breakup of hours for lecture/ tutorial/practice)</td><td colspan="7">Connecting two nodes using Ethernet cable and study the performance evaluation parameters such as delay, effective bandwidth - Basic Networking commands – Ping, IPConfig, Traceroute, NSlookup - Introduction to Socket Programming. File transfer using TCP. Echo, Chat between two or more clients using socket programming - Simulation of Stop and Wait Protocol - Simulation of Stop and Wait protocol with NACK, Modelling of ACK, NACK drops, etc., -Modelling and simulation of Sliding window protocol - Sliding window protocol with ACK/NACK drops, frame drops etc., - Performance evaluation through simulation of IEEE 802.3/802.5 networks - Implementation of OSPF. Introduction to NS2/OPNET simulator, Case studies.</td></tr>
<tr><td>Essential Reading</td><td colspan="7">1. Larry L.Peterson and Bruce S Davie, Computer Networks: A systems Approach,Morgan, 5th Edn, 2011.ISBN: 9780123850591
2. William Stallings, Data and Computer Communications, 10th Edn, Pearson, 2017.ISBN: 9780133506488</td></tr>
<tr><td>Supplementary Reading</td><td colspan="7">1. Andrew S. Tanenbaum, Computer Networks, 5th Edn, 2014. ISBN: 9788131770221
2. Behrouz Forouzan, TCP/IP protocol suite, Tata McGraw Hill, 4th Edn, 2010. ISBN: 9780070706521</td></tr>
</table>

<table>
<tr><td>Course Title</td><td>Compiler Design Practice</td><td>Course No</td><td colspan="5"></td></tr>
<tr><td rowspan="2">Department/ Specialization</td><td rowspan="2">Computer Science and Engineering</td><td rowspan="2">Credits</td><td>L</td><td>T</td><td>P</td><td colspan="2">C</td></tr>
<tr><td>0</td><td>0</td><td>4</td><td colspan="2">2</td></tr>
<tr><td>Faculty proposing the course</td><td>Faculty, Department of CSE</td><td>Status</td><td>Core</td><td>■</td><td>Elective</td><td colspan="2">□</td></tr>
<tr><td>Offered for</td><td>B.Tech</td><td>Type</td><td>New</td><td>■</td><td>Revision</td><td colspan="2">□</td></tr>
<tr><td>To take effect from</td><td>July 2021</td><td rowspan="2">Submitted for approval</td><td colspan="5" rowspan="2">44th Senate</td></tr>
<tr><td>Prerequisite</td><td>Nil</td></tr>
<tr><td>Learning Objectives</td><td colspan="7">The objective of this course is to train students to design various phases of compiler such as Lexical analyzer, syntax analyzer, semantic analyzer, intermediate code generator, code optimizer and code generator. Students are also exposed to design compiler construction tools such as Lexical Analyzer generator and parser generator. Applications of finite state machine and pushdown automation in compiler design are also taught in this course.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="7">● At the end of the course, students will be able to design a programming language and compiler for the same.
● Students will also be able to write large programs.</td></tr>
<tr><td>Course Contents (with approximate breakup of hours for lecture/ tutorial/practice)</td><td colspan="7">Lexical analyzer implementation in C - Lexical analyser implementation using LEX tool Recursive descent parser implementation in C for an expression grammar - YACC and LEX based implementation for an expressions grammar - YACC implementation of a calculator that takes an expression with digits, + and * and computes and prints its value - Front end implementation of a compiler that generates the three address code for a simple language- Back end implementation of a compiler which takes the three address code (output of previous exercise) and results in assembly language instructions - Implementation of peephole optimization in C.</td></tr>
<tr><td>Essential Reading</td><td colspan="7">1. Alfred Aho, Ravi Sethi and Jeffrey D Ullman, Compilers Principles, Techniques and Tools, Pearson Education, 2003. ISBN: 9780321491695</td></tr>
<tr><td>Supplementary Reading</td><td colspan="7">1. Levine J.R, Mason T, Brown D, Lex & Yacc, OReilly Associates, 1992 ISBN: 9781565920002.
2. Allen I. Holub, Compiler Design in C, Prentice Hall, 2003. ISBN: 9780131550452</td></tr>
</table>

<table>
<tr><td>Course Title</td><td>Professional Communication</td><td>Course No</td><td colspan="4">HS3001</td></tr>
<tr><td rowspan="2">Department/ Specialization</td><td rowspan="2">English</td><td rowspan="2">Credits</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td>1</td><td>0</td><td>2</td><td>2</td></tr>
<tr><td>Faculty proposing the course</td><td>Dr. Parvathy Das
Faculty, Dept. of SH</td><td>Status</td><td colspan="4">Core ■ Elective □</td></tr>
<tr><td>Offered for</td><td>B.Tech.</td><td>Type</td><td colspan="4">New ■ Revision □</td></tr>
<tr><td>To take effect from</td><td>July 2021</td><td rowspan="2">Submitted for approval</td><td colspan="4" rowspan="2">44th Senate</td></tr>
<tr><td>Prerequisite</td><td>Nil</td></tr>
<tr><td>Learning Objectives</td><td colspan="7">• Develop the capability to apply for a job and participate in selection process
• Acquire interview skills
• Gain proficiency in language skills indispensable for a successful professional
• Develop emotional intelligence</td></tr>
<tr><td>Learning Outcomes</td><td colspan="7">• Prepare résumé and cover letter
• Ready to perform at different levels of the interview process
• Able to use interpersonal skills in challenging situations
• Competent to draft various documents for specific purposes</td></tr>
<tr><td>Course Contents (with approximate breakup of hours for lecture/ tutorial/practice)</td><td colspan="7">• Preparing cover letter, résumé, digital profile; video profile; Email etiquette (L2,P4)
• Interview skills, Group discussion and impromptu speech (L2,P6)
• Social communication skills (L4,P6)
○ Conversational English appropriateness, context based speaking in general situations, discussion and associated vocabulary in professional situations)
○ Non-verbal communication – relevance and effective use of paralinguistic features – body language, chronemics, haptics, proxemics
○ Emotional intelligence (EI) and social intelligence at workplace – theoretical perspectives and their application in relevant workplace situations – EI and leadership skills – assessments and best practices in organizations
• Conflict management and communication at workplace (L4,P6)
○ Cross-cultural communication, Argumentation, negotiation, persuasion, decision making, case study of challenging situations
○ Organizing a meeting, working as part of a team, briefing
○ Business presentations – Preparing effective presentations, delivering presentaions and handling questions
• Writing proposals, statement of purpose, research article, agreements, summary Proofreading (L1,P4)
• Training for proficiency assessment (L1,P2)</td></tr>
<tr><td>References</td><td colspan="7">1. Tebeaux, Elizabeth, and Sam Dragga. The Essentials of Technical Communication. OUP, 2018.
2. Sabin, William A. The Gregg Reference Manual: A Manual of Style, Grammar, Usage, and Formatting. McGraw-Hill, 2011, pp 408-421.
3. Raman, Meenakshi and Sangeeta Sharma. Technical Communication: Principles and Practice. OUP, 2015.
4. Caruso, David R. and Peter Salovey. The Emotionally Intelligent Manager: How to Develop and Use the Four Key Emotional Skills of Leadership. John Wiley and Sons, 2004.
5. https://learnenglish.britishcouncil.org/business-english/youre-hired/episode-01
6. https://www.youtube.com/watch?v=HAnw168huqA
7. https://www.youtube.com/watch?v=azrqlQ_SLW8
8. https://owl.purdue.edu/owl/purdue_owl.html
9. Turabian,Kate L. Student's Guide to Writing College Papers. University of Chicago Press, 2010.</td></tr>
<tr><td>Methodology for content delivery</td><td colspan="7">Since students have been introduced to the basics of technical and professional communication in the first semester, this course is designed with the purpose of giving them intense training in professional and academic communication with global competence. Once the concept is introduced, adequate time should be devoted to practice and review.</td></tr>
</table>