BTech Computer Science and Engineering (CSE) Curriculum – 2019 Intake Onward

Department Core Curriculum

Semester 1

Course code	L-T-P-C	Course Name
CS100	1-0-0-1	Introduction to Profession

Total Credits 23

Refer to Institute Core Course List for semester-1 for the remaining courses in semester-1.

Semester 2

Course code	L-T-P-C	Course Name
CS102	2-0-2-3	Software Tools

Total Credits 21

Refer to Institute Core Course List for semester-2 for the remaining courses in semester-2.

Semester 3

Course code	L-T-P-C	Course Name
CS210	3-0-3-4	Digital Systems Design
CS220	3-0-2-4	Data Structures and Algorithms
CS221	3-0-0-3	Discrete Structures
CS230	3-0-0-3	Probability and Statistics for Computer Science
	3-0-0-3	Open Elective

Total Credits 17

Semester 4

Course code	L-T-P-C	Course Name
CS211	3-0-3-4	Computer Architecture
CS212	3-0-3-4	Computer Networks
CS222	3-0-3-4	Algorithm Design
	3-0-0-3	Open Elective

Total Credits 15

Semester 5

Course code	L-T-P-C	Course Name
CS300	3-0-3-4	Programming Language Paradigms
CS310	3-0-3-4	Operating Systems
CS320	3-0-0-3	Logic in Computer Science
CS330	3-0-3-4	Artificial Intelligence
	3-0-0-3	Open Elective

Total Credits 18

Semester 6

Course code	L-T-P-C	Course Name
CS331	3-0-3-4	Machine Learning
CS321	3-0-0-3	Theory of Computation
CS311	3-0-3-4	Compiler Design
	3-0-0-3	Program Elective

Total Credits 14

Semester 7

Course code	L-T-P-C	Course Name
	3-0-0-3	Program Elective
	3-0-0-3	Program Elective
	3-0-0-3	Open Elective
	3-0-0-3	Open Elective

Total Credits 12

Semester 8

Course code	L-T-P-C	Course Name
	3-0-0-3	Program Elective
	3-0-0-3	Program Elective
	3-0-0-3	Open Elective
	3-0-0-3	Open Elective

Total Credits 12

Program Total Credits 132

BTech project/thesis and external internship

Of the 24 credits in Semester 7 & 8, 12 can be acquired via a BTech Project/Thesis spanning both the semesters (6+6 credits) or a semester long internship in Semester 7 (12 credits).

Courses

Course numbering

A course with numbering CSXYZ is a course in the year X of category Y. The final number Z linearly orders courses with the same X and Y components.

Course categories	code
Programming	0
Systems	1
Theory	2
Applications	3

Course Descriptions

CS100

Introduction to Profession 1-0-0-1

Objective: The course provides a bird's-eye view of Computer Science & Engineering and its sub-disciplines.

Prerequisite: None

Contents: Origins and history of computer science, subareas of computer science, careers in computer science, research domains in computer science, computer science in other disciplines.

Suggested Textbooks:
None

Reference Texts:
None

CS101

Introduction to Computing 3-0-2-4

Objective: The student should be able to write reasonably complex programs involving sorting, searching, file operations, etc. in a procedural and object oriented way.

Prerequisite: None

Contents:

- Problem solving step-by-step, notion of algorithm
- Introduction to Python language
- Variables and types, arithmetic and Boolean expressions, comparison operators
- Control structures – if, if else, if elif else, while, while else
- Functions, passing arguments, recursion, default arguments

- Turtle graphics
- Lists, tuples, operations on lists and tuples
- Strings, string operations, split and join on strings
- Sorting, bubble sort, selection sort, binary search
- Higher order list operations – map, filter, reduce, zip
- Files, file operations – read, write, seek, tell
- Dictionaries, dictionary operations
- Classes and objects, object constructors, object and class variables, dunder functions
- Classes case study – matrices, polynomials, complex numbers, linked lists, trees
- Number systems, decimal, octal, hexadecimal, binary, conversions between number systems, adding, subtracting, multiplying, one's complement, two's complement, bit operations
- GUI using tkinter, buttons, labels, entry widgets, layout using place, grid and pack

Suggested Textbooks:

1. How to Think like a Computer Scientist, Allen B. Downey, 2002

Reference Texts:

1. Think Python: An Introduction to Software Design, A. B. Downey, 2012

CS102

Software Tools 2-0-2-3

Objective: The course familiarises the student with the standard tools on a GNU/Linux system and enables one to perform tasks like creating and managing sourcecodes, documents, webpages, slides, images etc.

Prerequisite: None

Contents:

- Basic GNU/Linux commands
- Shell scripting, awk, sed
- Web development tools, HTML, CSS, Javascript, PHP
- Basics of typography, typesetting using TeX, LaTeX, Beamer
- Raster and vector image editing, basics of color theory

Suggested Textbooks:
None

Reference Texts:

1. The Unix Programming Environment, B. Kernighan, R. Pike, 2015
2. The Not So Short Introduction to LATEX2e

CS210

Digital Systems Design 3-0-3-4

Objective: To learn how to design, build and analyze digital circuits and systems.

Prerequisite: None

Contents:

- Digital logic, Medium/Large/Very Large Scale Integration
- Basics of CMOS logic and gates, basic digital blocks, multiplexer, decoder, encoder, arbiter, bus
- Boolean algebra, minimizing boolean expressions, Karnaugh map, Quine-McCluskey etc
- Number representation
- Combinational circuit analysis, basics of circuit timing, pipelining, finite state machines, algorithmic state machines, registers, counters, memory
- Introduction to Verilog hardware description language. Lab sessions on design, implementation and simulation of basic digital systems using Verilog and Zynq board.

Suggested Textbooks:

1. Digital Logic & Computer Design, M. M. Morris, 2016

Reference Texts:
None

CS211

Computer Architecture 3-0-3-4

Objective: To understand the architecture of the microprocessor, to be able to program in assembly language and to design a simple microprocessor in HDL.

Prerequisite: CS210 Digital Systems Design

Contents:

- Computer organization
- Basic building blocks of a processor, MIPS processor architecture, MIPS instruction set architecture (ISA)
- Processor pipeline, branch prediction, instruction/data/task parallelism, data and control hazards
- MIPS assembly language programming
- Memory hierarchy, cache
- CISC vs. RISC.
- Lab sessions include MIPS assembly language programming and processor design and implementation using Verilog.

Suggested Textbooks:

1. Computer Organization and Design: The Hardware/Software Interface, 5th Ed. MIPS, D. A. Patterson

Reference Texts:

1. Computer Organization and Design, Revised Printing, Third Edition: The Hardware/Software Interface, J. L. Hennessy, D. A. Patterson

---

CS212

Computer Networks 3-0-3-4

Objective: To understand how computer networks are structured, how they work and what are the basic principles behind their design (with the Internet as a guiding example). To gain practical experience with tools and techniques for effective use of computer networks (for example, monitoring/tracing tools and socket programming).

Prerequisite: None

Contents:

- Layering abstraction, network architecture, packet switching, performance evaluation of networks
- Application layer, web and HTTP, socket programming, DNS, email protocols, peer-to-peer applications
- Transport layer, protocols for reliable data transfer, UDP and TCP protocols, congestion control and flow control
- Network layer, router architecture, IPv4 and IPv6, routing algorithms, distance vector and link-state algorithms, routing protocols on the internet (RIP, OSPF, BGP)
- Link layer, error detection and correction techniques, multiple access protocols.

Suggested Textbooks:

1. Computer Networking: A Top-Down Approach, J. F. Kurose, K. W. Ross, 2017

Reference Texts:
None

CS220

Data Structures and Algorithms 3-0-2-4

Objective: Given an algorithm using abstract data types the student should be able to implement it efficiently in a programming language like C. Also given a program the student should be able to find its time and space complexity.

Prerequisite: None

- C language basics, programming with pointers
- Time and space complexity of algorithms
- Basic data structures, lists, stacks, queues, binary search trees, heaps, hashing, trees, tree traversals, graphs, self balancing trees
- Graph search, BFS, DFS, minimum spanning tree algorithms, shortest path algorithms.

Suggested Textbooks:

1. The C Programming Language, B. W. Kernighan, D. Ritchie, 2015.
2. Data Structures & Algorithms, A. Aho, 2002

Reference Texts:

1. Introduction to Algorithms, T. H. Cormen, C. E. Leiserson, R. Rivest, C. Stein, 2010.

CS221

Discrete Structures 3-0-0-3

Objective: The students should be able to frame arguments formally about standard discrete structures used in computer science and present them as formal proofs.

Prerequisite: None

Contents:

- Proofs, proof strategies
- Sets, relations, functions, equivalence relations, partial orders, cardinality, countability, uncountability
- Combinatorics, basic counting techniques, pigeonhole principle, principle of inclusion-exclusion, recurrence relations, generating functions
- Notion of groups, Burnside Lemma and Polya's counting theorem
- Basics of graph theory.

Suggested Textbooks:

1. Discrete Mathematics and Its Applications, K. Rosen, 2017.
2. Algebra, by M. Artin, Prentice Hall, 1991.

Reference Texts:

1. Elements of Discrete Mathematics: A Computer Oriented Approach, C Liu, D. Mohapatra, 2017

CS222

Algorithm Design 3-0-3-4

Objective: The course aims to train students in a few standard approaches to the design and analysis of algorithms for a variety of problems arising in computer science and its applications. It also equips students with a suite of programs/algorithms which may be adapted to various situations.

Prerequisite: CS220 Data Structures and Algorithms

Contents:

- Asymptotic analysis of algorithms
- Basic bit-wise computations and their analysis

- Number theoretic algorithms such as the GCD and modulo computations
- Introduction to graph algorithms
- Algorithm design strategies such as Greedy, Divide and Conquer, and Dynamic programming
- Examples of modelling of real-life problems from application areas.

Suggested Textbooks:

1. Algorithms, S. Dasgupta, C. H. Papadimitriou, U. V. Vazirani, 2017.
2. Algorithm Design, J. Kleinberg, E. Tardos, 2013.

Reference Texts:

1. Introduction to Algorithms, T. H. Cormen, C. E. Leiserson, R. Rivest, C. Stein, 2010.

CS230

Probability and Statistics for Computer Science 3-0-0-3

Objective: The course gives an elementary-level introduction to probability and statistics for engineers and scientists. Along with theory and methods, this course focuses on applications in real-life using statistical computing and graphics tools, e.g., R programming language.

Prerequisite: None

Contents:

- Data visualisation tools and techniques
- Discrete experiments, Probability space, Equally-likely outcomes and combinatorial problems, Non-equally likely outcomes
- Conditional probability, Bayes formula, Independent events
- Random variables, Binomial and Poisson distributions, Expectation, Variance, Linearity of expectation
- Markov and Chebyshev inequalities, simple applications
- Joint distributions, joint densities, correlation
- Statistics, sampling, central limit theorem, hypothesis testing

Suggested Textbooks:

1. Introduction to Probability and Statistics for Engineers and Scientists, S. Ross, 2007

Reference Texts:

1. An Introduction to probability theory and its applications. (Vols. 1, 2), W Feller, 3/e

---

CS300

Programming Language Paradigms 3-0-3-4

Objective: To familiarise with different paradigms of programming in particular object-oriented, functional and event-driven.

Prerequisite: None

Contents:

- Programming paradigms, procedural, object oriented, functional, and event-driven
- Aspects of programming languages, type systems, scope, parameter passing, abstract data types, objects, inheritance, polymorphism, templates
- Functional programming, introduction to a functional programming language like Haskell, names, expressions and lists, functions, recursion, higher order and curried functions
- Event-driven programming

Suggested Textbooks:

1. Programming Languages: Concepts & Constructs, R. Sethi, 2006.
2. Programming in Haskell, G. Hutton, 2016.

Reference Texts:

1. Structure and Interpretation of Computer Programs, H. Abelson, G. J. Sussman, J. Sussman, 2005

CS310

Operating Systems 3-0-3-4

Objective: To understand the internals of an operating system and build a basic operating system.

Prerequisite: None

Contents:

- Basic concept of control flow in a computer system
- Introduction to OS for desktop systems
- Processes, inter-process communication, scheduling algorithms and policies
- Physical and virtual memory management, mass storage systems and disk scheduling
- Concurrency and process synchronization
- Introduction to mobile OS, OS for TV and Real Time Operating Systems (RTOS).
- Lab sessions include exercises on process generation, creation, permissions etc. and design and implementation of a basic OS.

Suggested Textbooks:

1. Operating System Principles, A. Silberschatz, P. B. Galvin, G. Gagne, 2006

Reference Texts:

1. Operating Systems, H. M. Deitel, P. J. Deitel, D. R. Choffnes, 2007

CS311

Compiler Design 3-0-3-4

Objective: Using standard tools student should be able to write a compiler for a simple imperative language.

Prerequisite: None

Contents

- Lexical analysis
- Syntax analysis, LL and LR grammars
- Semantic analysis, attributes and computation rules, type safety
- Intermediate code generation, syntax tree, three-address code, Quadruple, Triple, SSA, one pass intermediate code generation method
- Code optimization, basic block, flow graphs, local and global optimizations
- Introduction to data-flow analysis
- Run time environments, activation records, garbage collector algorithms
- Code generation.

Suggested Textbooks:

1. Compilers, A. Aho, J. Ullman, R. Sethi, M. S. Lam, 2013.

Reference Texts:
2. Engineering a Compiler, K. Cooper, L. Torczon, 2008.

CS321

Theory of Computation 3-0-0-3

Objective: The student should be able to appreciate the fundamental models of computation for regular, context-free, recursive and recursively enumerable languages. The student should be able tell apart computable problems from those that are not.

Prerequisite: CS221 Discrete Structures

Contents:

- Notion of a formal language, regular languages, finite state automata, DFA, NFA, regular expressions, rquivalence of all the notions, Myhill-Nerode theorem, pumping lemma, closure and decision properties of regular languages, equivalence and minimization of DFA
- Notion of a grammar, context free grammars and languages, derivation and parsing, PDAs, PDAs and CFGs capture same language class, Chomsky normal form CFGs, pumping lemma, closure and decision properties of CFLs.
- Turing Machine, historical motivation, robustness of the TM, universal Turing machine, recursive and r.e. languages, separation of the two classes, undecidable problems, Rice's theorem.

Suggested Textbooks:

1. Introduction to Automata Theory, Languages, and Computation, J. E. Hopcroft, R. Motwani, J. D. Ullman, 2008
2. Introduction to the Theory of Computation, M. Sipser, 2014

Reference Texts:

1. Automata and Computability, D. Kozen, 2007

CS320

Logic in Computer Science 3-0-0-3

Objective: To enable students to understand the syntax and semantics of propositional logic and first order logic and model specifications as formulas in a suitable vocabulary.

Prerequisite: CS221 Discrete Structures

Contents:

- Propositional logic, syntax, semantics, satisfiability algorithms, constraint satisfaction problems
- Soundness and completeness of propositional logic
- First order logic, syntax, semantics, applications like program verification
- Using SAT solvers

Suggested Textbooks:

1. Logic in Computer Science: Modelling and Reasoning about Systems, M. Huth, M. Ryan, 2004

Reference Texts:

1. Mathematical Logic for Computer Science, M. Ben-Ari, 2001

CS330

Artificial Intelligence 3-0-3-4

Objective: This course gives a broad introduction to Artificial Intelligence (AI). It will help to understand various AI concepts such as heuristic search, learning methods, uncertainty modeling, and explore use-cases and applications of AI in real-life.

Prerequisite: CS220 Data Structures and Algorithms

Contents:

- Basics of problem-solving, problem representation paradigms, state space
- Search techniques, heuristics, uninformed and informed search methods
- Constraint satisfaction problems, local search
- Supervised learning and classification problems, linear classifiers, perceptrons, artificial neural

networks, *k*-nearest neighbour classifiers
- Unsupervised learning and clustering, *k*-mean clustering, agglomerative clustering
- Uncertainty treatment, review of basic probability, formal and empirical approaches, introduction to Bayesian theory, Markov models

Suggested Textbooks:

1. Artificial Intelligence 3e: A Modern Approach, S. Russel and P. Norvig, 2015.

Reference Texts:

1. Machine Learning, T. Mitchell, 2017.

CS331

Machine Learning 3-0-3-4

Objective: To equip students to apply machine learning methods to real-world applications such as recommender systems, computer vision, bioinformatics, and text mining.

Prerequisite: CS230 Probability and Statistics for CS

Contents:

- Data science basics, how to wrangle, visualize, and analyze data, using models to explore your data
- Supervised learning, linear and logistic regression, generative learning, maximum likelihood estimation (MLE), maximum a posteriori (MAP) estimation, support vector machines, artificial neural networks and deep learning.
- Unsupervised learning, mixture of Gaussians, EM algorithm, autoencoders
- Bias-variance tradeoff, regularization and model selection
- Dimensionality reduction, principal component analysis, singular value decomposition

Suggested Textbooks:

1. Machine Learning, T. Mitchell, 2009.

Reference Texts:

1. Elements of Statistical Learning, T. Hastie, R. Tibshirani, and J. Friedman, 2017.
2. Pattern Recognition and Machine Learning, C. Bishop, 2010.
3. Foundations of Data Science, M. Blum, J. Hopcroft, and R. Kannan, 2018.
4. R for Data Science: Import, Tidy, Transform, Visualize, and Model Data, H. Wickham and G. Grolemund, 2016.
