IIT Palakkad

UG Curriculum 2022

BTech Computer Science and Engineering

Applicable to all from 2022-23 Academic Year

21 April, 2022

The UG Curriculum and Sample Template for Computer Science and Engineering

Program : **Bachelor of Technology**

Department : **Computer Science and Engineering**

Year : **2022 Onwards**

The BTech CSE program strives to impart knowledge across the depth and breadth of Computer Science and Engineering. The curriculum is developed with a special emphasis on fundamentals and practical learning. The core courses are intended to prepare and motivate the students for a large pool of electives which covers several deep and state-of-the-art topics. The core courses are listed below.

Level 1	Level 2	Level 3
Introduction to Programming	Discrete Mathematics Foundations of Computing Systems Foundations of Computing Systems Lab Systems Programming Data Structures and Algorithms Data Structures and Algorithms Lab Introduction to Artificial Intelligence	Design and Analysis of Algorithms Theory of Computation Computer Architecture Computer Architecture Lab Operating Systems Operating Systems Lab Compiler Design Compiler Design Lab

A tentative set of program electives include (but not limited to) the following: Logic, Topics in Discrete Mathematics, Graph Theory and Combinatorics, Topics in Graph Theory, Probabilistic Method, Probability and Computing, Parameterized Algorithms, Approximation Algorithms, Combinatorial Optimization, Computational Complexity, Foundations of Data Science and Machine Learning, Coding Theory, Cryptography, Game Theory and Mechanism Design, Paradigms of Programming, Functional Programming, Parallel Programming, Computational Methods and Applications, Computer Networks, Database Management Systems, Compiler Optimizations and Program Analysis, Advanced Computer Architecture, Advanced Computer Architecture Lab, Embedded Systems, Synthesis of Digital Systems, PreSilicon Design Verification using Formal Property Verification, Software Engineering, Real Time Systems, System on Chip Design Lab, Artificial Intelligence for Cyber Security, Responsible Artificial Intelligence, Machine Learning, Big Data Lab, Information Retrieval, Natural Language Processing, Computer Vision.

To guide the students towards arriving at a feasible ordering of courses, a course plan is proposed. Multiple variations of this plan are possible. For instance, a student may take fewer program electives than what is proposed in the plan if the total credit requirement for program electives is satisfied. Also, a student may choose to take two project courses one with 3 credits and the other with 6 credits instead of the proposed three 3-credit project courses.

BTech Computer Science and Engineering 2022 onwards (Sample Template)

Sl No.	Semester	Course Code	Course Title	Category	Credits
1	I	PH1030	Physics	IC	2-1-0-3
2		MA1011	Linear Algebra and Series	IC	3-1-0-4
3		ME1130	Engineering Drawing	IC	1-0-3-3
4		ID1010	Ecology and Environment	IC	2-0-0-2
5		ID1050A	Engineering Design	IC	1-0-3-3
6		ME1150	Mechanical Workshop	IC	0-0-3-2
7		PH1130/ CY1140	Physics/Chemistry Lab	IC	0-0-3-2
			Total		**19**

Sl No.	Semester	Course Code	Course Title	Category	Credits
1	II	MA1021	Multivariable Calculus	IC	3-1-0-4
2		CY1040	Basic Chemistry for Engineers	IC	2-1-0-3
3		HS1010	Technology and Society	IC	2-0-0-2
4		CE1020	Engineering Mechanics	IC	3-1-0-4
5		ID1110	Introduction to Programming	IC	2-0-3-4
6		EE1110	Electrical Workshop	IC	0-0-3-2
7		PH1130/ CY1140	Physics/Chemistry Lab	IC	0-0-3-2
			Total		**21**

Sl No.	Semester	Course Code	Course Title	Category	Credits
1	III	CS2020A	Discrete Mathematics	PMC	3-1-0-4
2		CS2011	Foundations of Computing Systems	PMC	3-0-0-3
3		CS2111	Foundations of Computing Systems Lab	PMC	0-0-3-2
4		CS2013	Systems Programming	PMC	1-0-3-3
5			Science and Mathematics Elective 1	SME	3
6			Humanities and Social Sciences Elective 1	HSE	3
7			Open Elective 1	OE	3
			Total		**21**

Sl No.	Semester	Course Code	Course Title	Category	Credits
1		CS2030	Data Structures and Algorithms	PMC	3-0-0-3

2	IV	CS2130	Data Structures and Algorithms Lab	PMC	0-0-3-2
3		CS3060	Computer Architecture	PMC	3-0-0-3
4		CS3160	Computer Architecture Lab	PMC	0-0-3-2
5		DS2020	Introduction to Artificial Intelligence	PMC	3-0-2-4
6			Science and Mathematics Elective 2	SME	3
7			Humanities and Social Sciences Elective 2	HSE	3
			Total		**20**

Sl No.	Semester	Course Code	Course Title	Category	Credits
1	V	CS3070	Design and Analysis of Algorithms	PMC	3-1-0-4
2		CS3050	Theory of Computation	PMC	3-1-0-4
3		CS3010	Operating Systems	PMC	3-0-0-3
4		CS3110	Operating Systems Lab	PMC	0-0-3-2
5			Program Major Elective 1	PME	3
6			Open Elective 2	OE	3
7			Life Sciences	IC	2
			Total		**21**

Sl No.	Semester	Course Code	Course Title	Category	Credits
1	VI	CS3040	Compiler Design	PMC	3-0-0-3
2		CS3140	Compiler Design Lab	PMC	0-0-3-2
3			Program Major Elective 2	PME	3
4			Program Major Elective 3	PME	3
5			Open Elective 3	OE	3
6			Project	Project	3
7			Program Major Elective* (for Honours)	PME*	3
			Total		**17/20**

Sl No.	Semester	Course Code	Course Title	Category	Credits
1	VII		Project	Project	3
2			Program Major Elective 4	PME	3
3			Program Major Elective 5	PME	3
4			Humanities and Social Sciences Elective 3	HSE	3
5			Open Elective 4	OE	3
6			Program Major Elective* (for Honours)	PME*	3

7			Program Major Elective* (for Honours)	PME*	3
			Total		**15/21**

Sl No.	Semester	Course Code	Course Title	Category	Credits
1	VIII		Project	Project	3
2			Program Major Elective 6	PME	3
3			Program Major Elective 7 [1]	PME	1
4			Open Elective 5	OE	3
5			Program Major Elective* (for Honours)	PME*	3
			Total		**11/14**

Total Credits	
Institute Core (IC)	42
Program Major Core (PMC)	44
Program Major Elective (PME)	19
Humanities and Social Sciences Elective (HSE)	9
Sciences and Mathematics Elective (SME)	6
Open Elective (OE)	15
Project	9

[1] The 7th PME course of 1-credit is mentioned in the course plan assuming that all other PME courses are 3 credits each. In practice, as PME courses may have varied credits, the number of PME courses is not important and it is only required that a student obtains at least 19 credits in total.

Computer Science and Engineering

Core course Syllabi

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Proforma for proposing course (New)

Course Title : Discrete Mathematics[1]

Course Code : CS2020A

Credit : 3-1-0-4 (L-T-P-C)

Category : Core

Target Programme : UG

Target Discipline : CSE

Prerequisite (if any) :

Date of proposal :

Date of approval :

Proposing faculty : Krishnamoorthy Dinesh & Deepak Rajendraprasad

Course Content:

Logic and set theory. Introduction to proofs, axiomatic method, proof patterns. Well ordering principle. Logical formulas - propositional and predicate. Sets, relations, equivalences and partial orders. Induction. Recursive definitions. Infinite Sets: Cantor's theorem, diagonalization argument, halting problem. Logic of sets. [*12 lectures + 4 tutorial sessions*]

Graph Theory. Graphs, degree. Common graphs. Walks, paths, connectivity, cycles, trees, forests. Cliques, Independent sets. Graph Isomorphism, bipartite graphs and matchings. Colouring. Planar graphs: Euler's formula, 6-colouring of planar graphs. Regular polyhedra. [*8 lectures + 3 tutorial sessions*]

Combinatorics. Product rule, division rule, counting subsets, sequences with repetitions - binomial and multinomial theorems. Pigeonhole principle. Inclusion-exclusion. Combinatorial proofs. Twelvefold way. Recurrence relations - Fibonacci numbers and Towers of Hanoi puzzle. [*8 lectures + 3 tutorial sessions*]

Discrete Probability. Events and probability spaces, The four step method, birthday principle. Set theory and probability. Conditional probability, law of total probability. Independence. Probability versus confidence. Random variables: independence. distribution functions, expectations. Linearity of Expectation [*14 lectures + 4 tutorial sessions*]

---

[1] Revision of CS2020 Discrete Mathematics for Computer Science and CS2010 Logic for Computing

Learning Outcomes:

1. Use logical notation to define and reason about fundamental mathematical concepts such as sets, relations, functions, and integers.
2. Evaluate elementary mathematical arguments and identify fallacious reasoning (not just fallacious conclusions).
3. Synthesize new proofs based on standard proof patterns.
4. Apply graph-theoretic models to solve problems like job allocation, scheduling, connectivity etc.
5. Calculate numbers of possible outcomes of elementary combinatorial processes such as permutations and combinations.
6. Calculate probabilities and discrete distributions for simple combinatorial processes; calculate expectations.

Textbook

1. *Mathematics for Computer Science*. Eric Lehman, F Thomson Leighton, Albert R Meyer. This book is available online under the terms of the Creative Commons Attribution ShareAlike 3.0 license. URL: https://courses.csail.mit.edu/6.042/spring18/mcs.pdf Printed version: 12th Media Services. ISBN-13: 978-1680921229.

Reference

1. *Discrete Mathematics and Applications*. Kenneth Rosen (7th Edition, 2012), McGraw-Hill Education (ISBN-13: 978-0073383095)
2. *Invitation to Discrete Mathematics*. Jiří Matoušek and Jaroslav Nešetřil (2nd Edition) Oxford University Press (ISBN-13: 978-0198570424)
3. *Discrete Mathematics: Elementary and Beyond*. László Lovász, József Pelikán, Katalin Vesztergombi, Springer 2003, ISBN-13: 978-0387955858.

Popular Readings

1. *Logicomix: An epic search for truth*. Apostolos Doxiadis and Christos Papadimitriou. Bloomsbury USA. ISBN-13: 978-1596914520
2. *What is the Name of This Book?: The Riddle of Dracula and Other Logical Puzzles*. Raymond M. Smullyan. Dover Publications. ISBN-13: 978-0486481982
3. *Proofs from THE BOOK*. Martin Aigner and Günter M. Ziegler. Springer. ISBN-13 : 978-3642008559
4. *Combinatorics: A Very Short Introduction*. Robin Wilson. Oxford University Press. ISBN-13 : 978-0198723493

Notes

1. This course is modelled along the Mathematics for Computer Science course (6.042J, Spring 2015) at MIT.

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Proforma for proposing course (New)

Course Title: Foundations of Computing Systems

Course Code : CS2011

Credit: 3-0-0-3 (L-T-P-C)

Category: Core

Target Program: UG

Target Discipline: CSE

Prerequisite (if any): Introduction to Programming or Systems Programming, Foundations of Computing Systems Lab (prerequisite/corequisite)

Date of proposal:

Date of approval:

Proposing faculty: Piyush P Kurur and Sandeep Chandran

Course Content:

Boolean logic and its implementation - binary digits (bits), transistors and switching circuits, basic gates and boolean operations using gates, composability of gates and building blocks of digital circuits such as multi-bit gates, and multiplexers/demultiplexers. *(3 lectures)*

Combinational circuits - Binary number system, signed numbers (two's complement), adder and multiplier circuits *(3 lectures)*

Sequential circuits - latches and flip-flops, registers, SRAM and DRAM cells and their characteristics, memory hierarchy *(3 lectures)*

Instruction Set Architecture - program counter, register set, basic operations - arithmetic, branch, and I/O operations, instruction encodings (machine instructions) *(3 lectures)*

CPU design - instruction and data memory, operations within a CPU such as fetch, decode, execute, and writeback *(6 lectures)*

Assembler - Instruction mnemonics, instruction sequences, symbols, working of an assembler - symbol table and two-pass assembler *(6 lectures)*

Stack machines - PUSH and POP instructions, procedures and subroutines, execution stack and argument passing *(6 lectures)*

Compilation - Lexing and parsing, Abstract syntax tree, syntax-directed translation, illustration of these concepts through an expression to postfix conversion *(6 lectures)*

Operating systems - Program loading and execution, program isolation (processes, virtual memory and threads) *(6 lectures)*

Learning Outcomes:

1. Understand the abstractions and interfaces that are the building blocks of modern computer systems
2. Appreciate the interactions and challenges related to the fundamental building blocks of a computer system (Computer Architecture and Organization, Operating Systems, and Compilers)

Text Books:

1. Noam Nisan, and Shimon Schocken, The Elements of Computing Systems: Building a Modern Computer from First Principles, The MIT Press. ISBN-10: 0262640686 ISBN-13: 978-0262640688

References:

1. Harold Abelson, Gerald Jay Sussman, and Julie Sussman, Structure and Interpretation of Computer Programs, Published by The MIT Press, ISBN-10: 8173715270, ISBN-13: 978-8173715273
2. Randal E. Bryant, and David R. O'Hallaron, Computer Systems: A Programmer's Perspective, Published by Pearson Education India, ISBN-10: 9332573905, ISBN-13: 978-9332573901
3. Maurice Bach, Design of the UNIX Operating System, Published by Prentice Hall, ISBN-10: 0132017997, ISBN-13: 978-0132017992

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Proforma for proposing course (New)

Course Title: Foundations of Computing Systems Lab

Course Code : CS2111

Credit: 0-0-3-2 (L-T-P-C)

Category: Core

Target Program: UG

Target Discipline: CSE

Prerequisite (if any): Foundations of Computing Systems (prerequisite/corequisite)

Date of proposal:

Date of approval:

Proposing faculty: Piyush P Kurur and Sandeep Chandran

Course Content:

This is a companion lab to Foundations of Computing Systems that enables the students to build a computer from first principles. A typical offering of this course will illustrate examples of

(i) combinational circuits such as adders and multipliers (1 week)

(ii) sequential circuits for real world tasks such as vending machines (1 week)

(iii) a simple register-based CPU (3 weeks)

(iv) assemblers for the CPU introduced (3 weeks)

(v) abstract stack-based machines (2 weeks)

(vi) simple compilers and translators (2 weeks)

Learning Outcomes:

Understand the implementation of the different components of a computer system (digital hardware, microarchitecture, operating systems and compilers) and their interconnection.

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Proforma for proposing course (New)

Course Title: Systems Programming[1]

Course Code : CS2013

Credit: 1-0-3-3 (L-T-P-C)

Category: Core

Target Program: UG

Target Discipline: CSE

Prerequisite (if any): Introduction to Programming or programming background which will be assessed by an entry test

Date of proposal:

Date of approval:

Proposing faculty: Piyush P Kurur, Sandeep Chandran, and Albert Sunny

Course Content:

Linux Fundamentals - linux file system, absolute and relative file paths, pseudo files, Linux terminal, navigation and simple commands, version control *(1 lecture + 3 hours lab)*

Basic structure of a computer: processor, memory, secondary storage and I/O peripherals; mapping of programming constructs to hardware elements - variables and its addresses, data types and its representation, ranges and overflow *(1 lecture + 3 hours lab)*

C/C++ programming: Arrays, Array indexing, memory view of (multi-dimensional) arrays, functions, pointers, dereferencing and address operators, pass-by-value and pass-by-reference mechanisms in function calls, pointer arithmetic, array manipulation using pointers, user-defined/abstract data types, file operations *(5 lectures + 9 hours lab)*

Debugger, Coding across multiple files, Coding conventions, Stages of compilation (preprocessor, compiler, assembler, linker, loader), Static and Dynamic libraries *(2 lectures + 6 hours lab)*

Makefile, make variables, pattern substitutions and wildcards, Organizing code into multiple folders, order-only prerequisites of Makefiles *(2 lectures + 6 hours lab)*

Shell scripting, I/O redirection, UNIX pipe (|) and command pipelines *(2 lectures + 6 hours lab)*

Multi-threaded programs (using libraries), multi-process programs and inter-process communication (including sockets) *(1 lecture + 3 hours lab)*

---

[1] This course is a mix of topics from the current CS1020 Introduction to Programming and CS5107 Programming Lab (core course for M.CaM and M.SoCD students that is not open to UG students)

Learning Outcomes:

1. Prepare students to write performant C/C++ programs
2. Equip students with tools that improve their software development productivity
3. Give them hands-on experience on advanced programming constructs

Text Books:

1. Harvey M. Deitel, and Paul J. Deitel, C How to Program, Published by Pearson, ISBN: 978-0132261197
2. Richard Blum, Linux Command Line and Shell Scripting Bible. Publisher: Wiley Publishing. ISBN-10: 111898384X, ISBN-13: 978-1118983843

References:

1. The Unix Programming Environment, Authors: Brian W Kernighan and Rob Pike. Publisher: Pearson. ISBN-10: 9332550255, ISBN-13: 978-9332550254.
2. Randal E. Bryant, and David R. O'Hallaron, Computer Systems: A Programmer's Perspective, Published by Pearson Education India, ISBN-10: 9332573905, ISBN-13: 978-9332573901
3. Blogs and other materials available on the internet (as prescribed by the instructors)

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Proforma for proposing course (Revised)

Course Title : Data Structures and Algorithms[1]

Course Code : CS2030

Credit : 3-0-0-3 (L-T-P-C)

Category : Core

Target Programme : UG

Target Discipline : CSE

Prerequisite (if any) : Systems Programming or Introduction to Programming

Date of proposal :

Date of approval :

Proposing faculty : Jasine Babu and Krithika Ramaswamy

Course Content:

Introduction
Computation, algorithms and data structures. Elementary data types, abstract data types and data structures. RAM model of computation. Example of computational problems and algorithms - searching an element in an array using linear search and binary search. Notion of efficiency of algorithms - running time and memory consumption. Notions of best, worst and average case complexity. *[3 lectures]*

Iteration and Recursion
Iterative and recursive algorithms (Towers of Hanoi, Euclid's GCD algorithm), proof of correctness using induction. Sorting arrays using selection sort, insertion sort, quicksort and merge sort (with proof of correctness and efficiency analysis). *[8 lectures]*

Asymptotic Analysis
Asymptotic notation. Estimates of binomials and factorials. Methods of solving recurrences using induction, recursion tree and Master method. Application of asymptotics to analysing efficiency of algorithms. *[8 lectures]*

[1] Minor revision of CS2030 Data Structures and Algorithms course

Linear Data Structures
Arrays and linked lists (single and doubly linked lists). Array and linked list based implementations for Queue and Stack. Applications to infix-postfix conversion and expression evaluation. *[6 lectures]*

Non-linear Data Structures
Trees - rooted trees, traversal of trees, binary trees, expression trees, binary search trees. Heaps and priority queues using min/max heaps. *[8 lectures]*
Graphs - adjacency matrix and adjacency list representations, depth first search, breadth first search. *[6 lectures]*
Hash tables - hash functions, open addressing, chaining. *[3 lectures]*

Learning Outcomes:

1. To understand basic data structures, their implementation and some of their standard applications.
2. To develop the ability to design and analyze basic algorithms and prove their correctness using the appropriate data structure learned in the course.

Text Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein. Introduction to Algorithms, Third Edition, PHI Learning, 2009. ISBN:978-81-203-4007-7.

References:

1. Sanjoy DasGupta, C. H. Papadimitriou, Umesh Vazirani. Algorithms, First Edition, Tata McGraw Hill, 2006. ISBN: 978-0073523408.

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Proforma for proposing course (Revised)

Course Title: Data Structures and Algorithms Lab[1]

Course Code : CS2130

Credit : 0-0-3-2 (L-T-P-C)

Category : Core

Target Programme : UG

Target Discipline : CSE

Prerequisite (if any) : Data Structures and Algorithms (prerequisite/corequisite), Systems Programming

Date of proposal :

Date of approval :

Proposing faculty : Jasine Babu and Krithika Ramaswamy

Course Content:

This is a companion lab of the Data Structures and Algorithms course. In this course, the students will learn to implement basic data structures in C/C++ and to use them for implementing some of the standard algorithms they learned in the theory course. A sample offering is given below.

Weeks 1-2: Iterative and recursive algorithms such as linear search, binary search, Towers of Hanoi and Euclid's GCD algorithm.

Weeks 3-4: Selection sort, insertion sort, quicksort, external merge sort.

Weeks 5-6: Linked lists - single and doubly linked lists, queue and stack using linked lists.

Weeks 7-9: Binary trees, binary search trees, expression trees and infix-postfix conversion.

Weeks 10-11: Heaps and priority queues using min/max heaps. Graphs - representations and traversals.

Learning Outcomes:

1. To be able to implement basic data structures and some of their standard applications.
2. To develop the ability to design and implement simple algorithms using the appropriate data structure learned in the course.

[1] CS2010 Data Structures and Algorithms Lab

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Proforma for proposing course (Revised)

Course Title: Computer Architecture[1]

Course Code : CS3060

Credit: 3-0-0-3

Category: Core

Target Programme: UG

Target Discipline: CSE

Prerequisite (if any): Foundations of Computing Systems, Computer Architecture Lab (prerequisite/corequisite)

Date of proposal:

Date of approval:

Proposing faculty: Vivek Chaturvedi

Course Content:

Introduction: Performance and power characteristics of a computer *(3 lectures)*

Instruction Set Architecture: Representation of Instructions, Machine instructions, Operands, addressing modes, Instruction formats, Instruction sets, Instruction set architectures - CISC and RISC architectures. *(6 lectures)*

Organization of a processor: Registers, ALU and Control unit, Data path in a CPU, Instruction cycle, Organization of a control unit - Operations of a control unit, Hardwired control unit, Microprogrammed control unit, pipelining and pipeline hazards *(9 lectures)*

Memory Subsystem: Semiconductor memories, Memory cells - SRAM and DRAM cells, Internal Organization of a memory chip, Organization of a memory unit, Error correction memories, Interleaved memories, Cache memory unit - Concept of cache memory, Mapping methods, Organization of a cache memory unit, Fetch and write mechanisms, Memory management unit - Concept of virtual memory, Address translation, Hardware support for memory management. *(9 lectures)*

I/O Subsystem and System Interconnection: I/O transfers - Program controlled, Interrupt driven and DMA, Privileged and non-privileged instructions, Software interrupts and exceptions. Programs and processes role of interrupts in process state transitions *(9 lectures)*

Advanced Topics: Parallel architectures, SIMD, MIMD etc., Future trends *(6 lectures)*

[1] Minor revision of CS2600 Computer Organization

Learning Outcomes:

1. State the history and development of modern computer systems.
2. Interpret data representation and computer arithmetic.
3. Determine the key aspects of microarchitecture and instruction set architecture
4. Estimate performance of computer systems and suggest methods for performance enhancement
5. Specify the importance of memory hierarchy for efficient memory design and virtual memory to overcome memory wall.
6. Predict performance of IO subsystems
7. Describe emerging computing trends and some parallel architectures

Text Books:

1. Patterson, J.L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, Morgan Kaufmann, 5th edition, 2013, ISBN-13: 9780124078864
2. Smruti R Sarangi, Computer Organisation and Architecture, McGraw Hill Education, 1st edition, 2017, ISBN-13: 978-9332901834

References:

1. Andrew S. Tanenbaum, Structured Computer Organization, Prentice Hall, 6th edition, 2012, ISBN: 978-0132916523.
2. C. Hamacher, Z. Vranesic and S. Zaky, Computer Organization, McGraw-Hill, 5th edition,2002, ISBN: 0072320869.
3. J.L.Hennessy, D.A.Patterson, Computer Architecture: a quantitative approach, Morgan Kaufmann, 5th edition, 2011, ISBN: 978-1558605961.

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Proforma for proposing course (Revised)

Course Title : Computer Architecture Lab[1]

Course Code : CS3160

Credit : 0-0-3-2

Category : Core

Target Programme : UG

Target Discipline : CSE

Prerequisite (if any) : Computer Architecture (prerequisite/corequisite)

Date of proposal :

Date of approval :

Proposing faculty : Vivek Chaturvedi

Course Content: This is a companion lab to Computer Architecture course. A typical offering will either require students to implement/extend microarchitectural features in a cycle accurate simulator, or implement a simple processor from scratch using a HDL. The assignments may include design of micro-architecture blocks (2 weeks), experiments on instruction set architecture and its data-path (3 weeks), exploration of data hazards in pipelining (2 weeks) and memory hierarchy (3 weeks).

Learning Outcomes: Students are able to appreciate the concepts learnt in the theory course and the challenges in applying them in practice.

[1] Minor revision of CS2610 Computer Organization Lab

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Proforma for proposing course (New/~~Revised/MOOC~~)

Course Title	: Introduction to Artificial Intelligence
Course Code	: DS2020
Credit	: 3-0-2-4
Category	: Program Core
Target Program	: UG
Target Discipline	: Data Science and Engineering
Prerequisites	: Introduction to Programming Discrete Mathematics
Data of proposal	: 10/01/2022
Data of approval	:
Proposing faculty	: Narayanan C Krishnan

Course Content:

1. Introduction to AI - Rational Agents - [2 lectures]
2. Search-(BFS, UCS, A* Search, Heuristics, Local Search) - [4 lectures, 6 lab hours]
3. Adversarial Search [3 lectures, 6 lab hours]
4. Knowledge Representation and logical inference (propositional logic, resolution, predicate logic, ontologies)[9 lectures]
5. Planning [4 lectures]
6. Probabilistic reasoning (probabilistic graphical model, exact and approximate inference)[5 lectures, 4 lab hours]
7. Markov Decision Processes (introduction to mdp) [6 lectures, 6 lab hours]
8. Decision making under uncertainty (introduction to PoMDPS, game theory, mechanism design, [3 lectures]
9. Reinforcement learning (introduction to rl) [6 lectures, 6 lab hours]

Learning Outcomes:

- Learn the fundamentals of field artificial intelligence. The students will be familiar with a broad range of topics in AI, ready to undertake specialized courses.
- Gain hands-on programming experience through the implementation of the AI techniques for various synthetic and real-world applications.

Textbooks:

Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, Pearson, 2020. ISBN 978-0134610993

Reference Books:

Artificial intelligence: Kevin Knight, Elaine Rich, Shivashankar Nair, McGraw Hill, 3rd Edition 2017 ISBN 978-0070087705

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Proforma for proposing course (Revised)

Course Title : Design and Analysis of Algorithms[1]

Course Code : CS3070

Credit : 3-1-0-4 (L-T-P-C)

Category : Core

Target Programme : UG

Target Discipline : CSE

Prerequisite (if any) : Data Structures and Algorithms

Date of proposal :

Date of approval :

Proposing faculty : Jasine Babu and Krithika Ramaswamy

Course Content:

Review of Basic Concepts
Asymptotic analysis of efficiency of algorithms, amortized analysis with an example (binary counter or stack with multipop), lower bounds for searching, finding maximum and sorting *[6 lectures + 1 tutorial hour]*

Graph Search Algorithms: DFS and BFS with applications to connectivity in undirected graphs, topological sorting and strongly connected components in directed graphs, shortest paths in unweighted graphs. *[6 lectures + 3 tutorial hours]*

Algorithm Design Paradigms
Overview of design paradigms - greedy strategy, dynamic programming, divide and conquer, iterative improvement.

Greedy Algorithms - optimal substructure and greedy choice, shortest paths in directed acyclic graphs, Dijkstra's algorithm, minimum spanning trees, cut property and cycle property theorems, Prim's algorithm, Kruskal's algorithm - union-find data structure and amortized analysis of path compression heuristic. *[10 lectures + 3 tutorial hours]*

[1] Minor revision of CS2040 Design and Analysis of Algorithms course

Dynamic Programming - optimal substructure, overlapping subproblems, ordering of subproblems and memoization, revisiting shortest paths in directed acyclic graphs, Bellman-Ford algorithm, all-pairs shortest algorithms and Floyd-Warshall algorithm. *[8 lectures + 3 tutorial hours]*

Iterative Improvement - Networks and flows, Max-flow min-cut theorem, Ford-Fulkerson algorithm, Edmonds-Karp algorithm. Matching in bipartite graphs using flows. *[7 lectures + 1 tutorial hour]*

Computational Complexity
Polynomial time reductions, classes NP, co-NP and P, NP-complete problems. *[7 lectures + 1 tutorial hour]*

Remarks: The tutorials sessions will typically be utilized for discussing problem sets and/or evaluating programming assignments. Course contents overlap with MCaM core course CS5009 Algorithms, however, CS5009 is not an elective for UG.

Learning Outcomes:

1. Develop the ability to analyze the running time and prove the correctness of basic algorithms.
2. To be able to design efficient algorithms for moderately difficult computational problems, using various algorithm design techniques taught in the course.
3. To be able to prove the hardness of NP-Hard problems using simple reductions.

Textbooks

1. Algorithms by Sanjoy Dasgupta, Christos H. Papadimitriou, Umesh Vazirani. McGraw-Hill Education, 2006. ISBN: 978-0073523408.

Reference

1. The Design and Analysis of Algorithms by Dexter C Kozen. Texts and Monographs in Computer Science, Springer, 1992. ISBN:0-387-97687-6.
2. Introduction to Algorithms (3rd Edition) by Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein. PHI Learning, 2009. ISBN:978-81-203-4007-7.
3. Algorithm Design by Jon Kleinberg and Eva Tardos. Pearson, 2015. ISBN:978-93-325-1864.

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Proforma for proposing course (Existing)

Course Title : Theory of Computation[1]

Course Code : CS3050

Credit : 3-1-0-4

Category : Core

Target Programme : UG and MCaM

Target Discipline : CSE

Prerequisite (if any) : Discrete Mathematics

Date of proposal :

Date of approval :

Proposing faculty : Deepak Rajendraprasad

Course Content:

Regular Languages & Finite Automata: Regular Languages and Regular Expressions, Deterministic and Non-deterministic Finite Automata, Kleene's Theorem, Pumping Lemma, Myhill-Nerode Theorem. *[12 lectures]*

Introduction to Context-free Languages & Pushdown Automata: Context-free Languages and Grammars, Ambiguity, Chomsky Normal Form, CYK Algorithm, Pumping Lemma, Introduction to Deterministic and Nondeterministic Pushdown Automata. *[9 lectures]*

Turing Machines & Recursive Languages: Mathematical modelling of computation, Deterministic Turing Machines, Church-Turing Thesis, Chomsky Hierarchy, Universal Turing Machines. Recursive and Recursively Enumerable Languages. Non-recursive Languages and Undecidable Problems, the Halting Problem. *[12 lectures]*

Complexity: Resource-bounded computation. Classes P and NP, PSPACE and EXP. Polynomial time reductions, NP-completeness. *[9 lectures]*

Learning Outcomes: After completion of this course, a student will be able to

1. Give the mathematical definition of various computational models and state and prove their limitations.
2. To explain important notions in computing like nondeterminism, reductions and resource boundedness.

[1] Existing course: Theory of Computation CS3050 (BTech CSE) and CS5017 (MCaM)

Text Books:

1. Introduction to Languages and The Theory of Computation (4th Edition) by John C. Martin, McGraw-Hill Publishers, 2011. ISBN: 9780073191461.
2. Automata and Computability by Dexter C. Kozen. Springer Publishers 2007. ISBN: 0387949070.

References:

1. Elements of Computation Theory by Arindama Singh, Springer-Verlag London, 2009. ISBN: 978-1-4471-6142-4.
2. Introduction to Automata Theory, Languages and Computation by Hopcroft, Motwani, and Ullman. 3rd Edition, Pearson Publishers, 2006. ISBN:0321462254.
3. Elements of the Theory of Computation by H. R. Lewis and C.H. Papadimitriou, Prentice Hall Publishers, 1981. ISBN-13: 978-0132624787.

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Proforma for proposing course (Revised)

Course Title: Operating Systems[1]

Course Code : CS3010

Credit: 3-0-0-3 (L-T-P-C)

Category: Core

Target Program: UG

Target Discipline: CSE

Prerequisite (if any): Foundations of Computing Systems, Systems Programming, Data Structures and Algorithms, Data Structures and Algorithms Lab, Operating Systems Lab (prerequisite/corequisite)

Date of proposal:

Date of approval:

Proposing faculty: Sandeep Chandran

Course Content:

UNIX System Calls, Shell, Privileged Instructions, Interrupt handling *(6 lectures)*

Process Management - Processes and Threads, Scheduling (context-switch, scheduling algorithms, fairness, infinite wait, optimal scheduling, priority inversion) *(9 lectures)*

Inter-process communication, Synchronization Primitives, Deadlocks *(9 lectures)*

Memory Management: Virtual Memory, Demand Paging *(9 lectures)*

File systems: I/O Management, Security. *(9 lectures)*

Learning Outcomes:

1. Understand how an operating system functions as a middle layer between the hardware of a computer and the user programs and the various tasks involved in this.
2. Appreciate design issues and concepts underlying some of the well known operating systems.
3. Compare pros and cons of various design options and issues involved in designing an operating system.

Text Books:

1. Modern Operating Systems, Andrew S. Tanenbaum, Herbert Bos, Pearson Education India; Fourth edition 2016. ISBN-13:978- 9332575776

[1] Minor revision of CS3010 Operating Systems

References:

1. Operating Systems: A Design-Oriented Approach, Charles Crowley, International edition, McGraw-Hill Education (ISE Editions). ISBN-13 978 0071144629
2. Operating Systems Concepts, Abraham Silberschatz, Peter B. Galvin and Greg Gagne, Wiley, 2015. ISBN-13: 978-8126554270
3. Operating Systems: Internals and Design Principles William Stallings, Pearson Education India; 7 edition (2013). ISBN-13: 978-9332518803
4. Linux Kernel Development (Developer's Library), Robert Love, Pearson Education India, 3 edition (2010). ISBN-13: 978-8131758182

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Proforma for proposing course (Revised)

Course Title: Operating Systems Lab[1]

Course Code : CS3110

Credit: 0-0-3-3 (L-T-P-C)

Category: Core

Target Program: UG

Target Discipline: CSE

Prerequisite (if any): Operating Systems (prerequisite/corequisite)

Date of proposal:

Date of approval:

Proposing faculty: Sandeep Chandran

Course Content: This is a companion lab to the Operating Systems course. In a typical offering, students extend features in an existing operating system or implement a tiny operating system which has (primitive) implementation of all the important features like process management (3 weeks), memory management (3 weeks), file systems (3 weeks) and inter-process communication and synchronization (3 weeks).

Learning Outcomes: The student is able to appreciate the various concepts learnt in the theory course and their implementation in a real operating environment.

[1] CS3110 Operating Systems Lab

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Proforma for proposing course (Revised)

Course Title: Compiler Design[1]

Course Code : CS3040

Credit: 3-0-0-3 (L-T-P-C)

Category: Core

Target Program: UG

Target Discipline: CSE

Prerequisite (if any): Foundations of Computing Systems, Data Structures and Algorithms, Data Structures and Algorithms Lab, Theory of Computation, Compiler Design Lab (prerequisite/corequisite)

Date of proposal:

Date of approval:

Proposing faculty: Piyush P Kurur and Unnikrishnan Cheramangalath

Course Content:

Introduction to language translators and overview of the compilation process. *(5 lectures)*

Lexical analysis: specification of tokens, token recognition, conflict resolution. *(5 lectures)*

Parsing: Overview of Context Free Grammars (CFG), Parse trees and derivations, left recursion, left factoring, top-down parsing, shift reduce parsing, conflict resolution. *(10 lectures)*

Syntax directed translation. Semantic analysis, Type checking, intermediate code generation. *(6 lectures)*

Runtime environments: activation records, and heap management. *(6 lectures)*

Code optimization: basic blocks, liveness, register allocation. *(5 lectures)*

Advanced topics: Overview of machine dependent and independent optimizations. *(4 lectures)*

Learning Outcomes: At the end of the course we expect the student to know enough of the theory (Parsing, Code generation, Code optimisation) and tools (lexical analyzers, parser generators, code generators) so that they can build a compiler that converts from a non-trivial high level language to a machine code.

---

[1] CS3040 Compiler Design

Text Books:

1. Compilers: Principles, Techniques, and Tools, Second Edition. Alfred Aho, Monica Lam, Ravi Sethi, Jeffrey D. Ullman, Pearson, January 2013. ISBN-978-9332518667.
2. Modern Compiler Implementation in Java. Andrew W Appel, Jens Paisberg. Cambridge University Press, January 2002. ISBN-978-0521820608
3. Modern Compiler Implementation in ML, Andrew W Appel, Cambridge University Press, December 1997 . ISBN 0 521 58274 1 (hardback) ISBN 0521582741
4. Compiler Construction: Principles and Practice, 1st Edition, Kenneth C. Louden, Cengage Learning; 1 edition (January 24, 1997), ISBN-13: 978-0534939724

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Proforma for proposing course (Revised)

Course Title: Compiler Design Lab[1]

Course Code : CS3140

Credit: 0-0-3-2 (L-T-P-C)

Category: Core

Target Program: UG

Target Discipline: CSE

Prerequisite (if any): Compiler Design (prerequisite/corequisite)

Date of proposal:

Date of approval:

Proposing faculty: Piyush P Kurur and Unnikrishnan Cheramangalath

Course Content:

Week 1: Specification of language tokens using one of the tools for lexical analysis: lex, flex, ml-lex.
Week 2 : Language to evaluate arithmetic expressions.
Weeks 3-4: Specification of language using one of the LALR parsing tools: yacc, bison, ml-yacc
Weeks 5-6: Develop a parser for a language that consists of elementary data types, and arithmetic, relational, and logical operator(s), control statement(s) and loop construct(s), and functions.
Weeks 7-8: Creation of Abstract Syntax Tree (AST) from the language grammar.
Weeks 9-10: Conversion of AST to an intermediate code.
Weeks 11-12: Code generation from the intermediate code for a specific computer architecture.

Learning Outcomes: Develop a compiler covering all phases from lexical analysis to code generation and gain the skill of developing a system software from scratch.

[1] CS3140 Compiler Design Lab