DEPARTMENT OF Computer Science and Engineering

Syllabi for BTech MINOR Course Structure in Computer Science and Engineering

(To be applicable for 2010-batch onwards).

Semester	Course Code	Course Title	L-T-P-C
3rd	CS 205M	Theoretical Foundations of Computer Science	3 - 0 - 0 - 6
4th	CS 206M	Data Structures and Algorithms	3 - 0 - 0 - 6
5th	CS 322M	Digital Logic and Computer Architecture	3 - 0 - 0 - 6
6th	CS 350M	Computer Systems	3 - 0 - 0 - 6
7th	CS 441M	Software Engineering	3 - 0 - 0 - 6
		Total credits	**15 -0 -0 -30**

CS 205M Theoretical Foundations of Computer Science (3 0 0 6)

Functions, relations, partial orders, recurrences, summations, generating functions, asymptotics; Graphs: basic concepts; Elementary Logic and proof techniques.
Alphabets, Languages, Grammars; Finite automata: regular expressions, regular languages; Context free languages: pushdown automata; Turing machines: recursively enumerable languages; Chomsky hierarchy.

Texts:

1. K. H. Rossen, *Discrete Mathematics and its Applications*., McGraw Hill International Edition, 1999.
2. J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Pearson Education, 3rd Edition, 2007.

References:

1. J. P. Tremblay and P. R. Manohar, *Discrete Mathematics with Applications to Computer Science*, McGraw Hill International Edition, 1989.
2. C. L. Liu, *Elements of Discrete Mathematics.* 2/e. McGraw-Hill International Edition, 1986.
3. M. Sipser, *Introduction to the Theory of Computation*, Thomson, 1997.
4. H. R. Lewis and C. H. Papadimitriou, *Elements of the Theory of Computation*, Prentice-Hall of India, New Jersey, 1981.

CS 206M Data Structures and Algorithms (3-0-0-6)

Review of fundamental Data Structures; Models of Computation: random access machines, space and time complexity measures, lower and upper bounds; Design techniques: the greedy method, divide-and-conquer, dynamic programming, backtracking; Sorting and Searching; Graph algorithms; Hashing: separate chaining, linear probing, quadratic probing; Search Trees: binary search trees, AVL trees, B-trees; NP-completeness;

Texts:

1. T H Cormen, C E Leiserson, R L Rivest and C Stein, *Introduction to Algorithms*, 3/e, MIT Press, 2009.
2. Jon Kleinberg and Eva Tardos, *Algorithm Design,* 1/e, Pearson Education, 2006.

References:

1. S Sahni, *Data Structures, Algorithms and Applications in C++*, McGraw-Hill, 2001.
2. M T Goodrich and R Tamassia, *Algorithm Design: Foundations, Analysis and Internet Examples*, John Wiley & Sons, 2001.

CS 322M Digital Logic and Computer Architecture (3-0-0-6)

Boolean Algebra; Minimisation and realisation of switching circuits; Basic building blocks of combinational circuits: Multiplexer, De-multiplexer, Encoder, Decoder, Adder, Subtracter; Design of synchronous sequential circuits: Flip-flops, Registers, Counters, Finite State Machines, State tables and diagrams, Excitation functions of memory elements.

Instruction sets with various addressing modes; Memory organisation: ROM, Cache, Main Memory; CPU design: ALU, Control unit design: hardwired and microprogrammed; I/O transfer: Program controlled, Interrupt controlled and DMA.

Texts:

1. M. Morris Mano, *Digital Design,* 3/e, Pearson Education, 2007.
2. William Stallings, *Computer Organization and Architecture: Designing for Performance*, 8/e, Pearson Education India. 2010.

References:

1. A. P. Malvino, D. K. Leach and G. Saha, *Digital Principles and Applications,* 6/e, McGraw Hill, 2006.
2. V. C. Hamacher, Z. G. Vranesic and S. G. Zaky, *Computer Organization*, 5/e, McGraw Hill, 2002.
3. D. A. Patterson and J. L. Hennessy, *Computer Organization and Design*, 3/e, Morgan Kaufmann, 2006.
4. Barry B. Brey, *The INTEL Microprocessors,* 8/e, Prentice Hall, 2008.

CS 350M COMPUTER SYSTEMS (3 0 0 6)

Pre-requisite: CS 206M, CS 322M

Introduction to structure and organization of computer systems, operating systems, and networks; Processes and threads and their scheduling, synchronization, deadlocks in concurrent processes; Memory management basics, demand paging and virtual memory implementation; File system design and implementation.

Basics of digital communication, digital transmission of data, modulation; Multiplexing; Data link control with sliding window protocols, error control; Local area networks, Ethernet, wireless networks; Concepts of switched networks; Internet addressing and routing algorithms; Transport protocols, UDP, TCP, flow control, congestion control; Application layer protocols such as DNS, SSL, Web.

Texts:

1. A. Silberschatz, P. B. Galvin and G. Gagne, *Operating System Concepts*, 8th Ed, Wiley India, 2009.
2. A. S. Tanenbaum, *Computer Networks*, 4th Ed, Pearson India, 2003.

References:

1. L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach*, 4th Ed, Elsevier India, 2007.
2. W. Stallings, *Data and Computer Communications*, 8th Ed, Pearson India, 2007.
3. W. Stallings, *Operating Systems: Internals and Design Principles*, 6th Ed, Pearson India, 2009.

SEMESTER – 7

CS 441M Software Engineering (3-0-0-6)

Introduction: software engineering principles, life cycle; Requirement specification: styles, operational and descriptive; Design: a brief concept on objects, data abstraction, inheritance, polymorphism, data encapsulation, software design using functional and object oriented approaches, architectural, component-level and user interface design; Brief introduction on database system (specially SQL, MySQL); Verification: testing, validation; Software reuse: design patterns; Software management; Software Modelling: UML

Texts:

1. R S Pressman, *Software Engineering: A Practitioner's Approach*, 7/e, McGraw-Hill, 2010.
2. I Sommerville, *Software Engineering*, 5/e, Addison-Wesley, 2000.

References:

1. T C Lethbridge and Robert Laganière, *Object Oriented Software Engineering*, Tata McGraw-Hill, 2004.
2. Jacobson Ivar, Magnus Christerson, Patrik Jonsson and Gunnar Overgaard, *Object Oriented Software Engineering*, Addison Wesley, 1992.
3. Jacobson Ivar, Grady Booch and James Rumbaugh, *Unified Software Development Process,* Addison Wesley, 1999.
4. S Bennett, S McRobb and R Farmer, *Object Oriented Systems Analysis and Design Using UML,* 2/e, Tata McGraw-Hill, 2004.
5. E Gamma, R Helm, R Johnson and J M Vlissides, *Design Patterns: Elements of Reusable Object Oriented Software*, Addison Wesley, 1994.