Revised Curriculum of M.Tech. in Information Technology

(From Autumn 2019)

Eligibility for M.Tech (IT):

B.E. /B.Tech./AMIE/AMIETE in CSE/IT/ECE/allied subjects or MCA or its equivalent or M.Sc. in Computer Science/IT/Electronics/Mathematics/Statistics with a minimum of 55% marks in aggregate. Candidates selected under GATE must possess a valid GATE score in CS.

Course Structure

Time Duration:

Minimum duration:	2 years (4 semesters)
Maximum duration:	4 years (8 semesters)

Credit Requirements (Revised):

i.	**Minimum Credit requirement**	**: 68**
ii.	**Core Courses**	**: 19**
iii.	**Electives**	**: 18**
iv.	Term Projects	: 24
v.	**Seminar**	**: 02**
vi.	Open Electives	: 06

COURSE STRUCTURE (Revised)

Course Code	Title	Credit Structure L-T-P			Total Credit
1st Semester					
CS 531	Object Oriented Programming & Design	3	1	1	5
CS 601	Design Analysis of Algorithms	3	0	0	3
CS 634	Selected Topics in Computer Networks	3	0	1	4
CS 543	**Advanced Programming Lab I**	**0**	**1**	**2**	**3**
	Elective 1				3
	Open Elective 1				**3**
Total Credit					**21**
2nd Semester					
IT 610	Advanced Database System	3	0	1	4
	Elective 2	3	0	1	3
	Elective 3	3	0	1	3
	Elective 4	3	0	1	3
CS 545	**Seminar**	**0**	**2**	**0**	**2**
	Open Elective 2				3
Total Credit					**18**
3rd Semester					
IT 604	Term Project I	0	0	8	8
	Elective 5				3
	Elective 6				3
Total Credit					**14**
	4th Semester				
IT 605	Term Project II	0	0	0	16
Total Credit					**16**

Elective Courses

Course Code	Title	Credit Structure	Total Credit
CS 509	Data Communication	3-0-1	4
CS 505	Software Engineering	3-0-1	4
CS 525	Artificial Intelligence	3-0-0	3
CS 424	Formal Language and Automata	3-0-0	3
IT 518	Graph Theory	3-0-0	3
CS 529	Embedded Systems	3-0-1	4
CS 621	Mobile Computing	4-0-0	4
CS 625	Web Technology	3-0-1	4
IT 517	Pattern Recognition	4-0-0	4
IT 509	Data Mining & Data Warehousing	3-0-1	4
IT 507	Computer Security & Cryptography	3-0-0	3
CS 606	Computer Architecture and Parallel Processing	3-0-0	3
CO 504	Natural Language Processing	3-0-0	3
CS 610	Bioinformatics	3-0-0	3
IT 611	Distributed Systems	3-0-0	3
CS 725	Knowledge Representation and Reasoning	4-0-0	4
CS 731	Data Mining in Security	3-0-1	4
CS 533	Computational Geometry	3-0-0	3
IT 523	Discrete Mathematics	3-0-0	3
CS 522	Computer Graphics	3-0-2	4
CS 523	Enterprise Resource Planning	3-0-0	3
CS 524	Theory of Computation	3-0-0	3
CS 502	System Software	2-0-2	3
CS 507	Computer Networks	3-0-1	4
CS 508	Database Management Systems	2-1-2	5
IT 503	Multimedia Systems	3-0-2	4
IT 504	E-Commerce	3-0-0	3
IT 506	Logic Programming	3-0-0	3
IT 510	Advanced Operating Systems	3-0-1	4
CS 532	Compiler Design	3-0-1	4
CS 602	Image Processing	3-0-0	3
CS 607	Optimization Technique	3-0-0	3

CS 543 : Advanced Programming Lab I
Credits: 3(L-0 T-1 P-2)

PREREQUISITES

Undergraduate courses in procedural programming and object oriented programming

COURSE OBJECTIVES

- Be competent with use of basic constructs provided by high-level programming languages
- Be able to use computational thinking to design solutions
- Be competent with use of computational approaches to solve problems in science and engineering
- Develop proficiency in implementation of Algorithmic techniques using appropriate data structures
- Be proficient in using advanced data structures
- Develop competency in implementation of advanced algorithms

COURSE CONTENT

This course will involve the students extensively in writing programs for assignments in contemporary programming languages. Topics include:

- Matrix operations like inverse, rank, linear equations, Eigen values and vectors
- Implementing algorithms using recursions, divide-and-conquer, greedy, dynamic programming concepts
- Linear programming using Cplex/LINGO.
- Implement some commonly used graph algorithms
- Implement network flow algorithms/matching
- Implement advanced data structures like B-Tree, dynamic trees, skip lists, suffix tree, red black tree, tries, R-tree, geometric data structures (KD-tree, interval tree, Quad tree, range tree etc.)

COURSE OUTCOMES

After completion of course, students would be able to:

- Master in use of procedural programming language C and object oriented language C++ to implement basic algorithms and use of data structures
- Master in development and implementation of algorithms for different problems in Computer Science
- Master in implementation of graph algorithms for some well known problems
- Master in solving some linear programming problems using Cplex/LINGO
- Master in implementation of advanced data structures for solving problems related to cryptography, networks, machine learning, text processing, internet programming, compiler construction etc.

Books/References:

<u>Text Books:</u>

1. Introduction to Scientific Computation and Programming, Daniel T. Kaplan
2. T. Cormen, C. Leiserson, R. Rivest, and C. Stein, Introduction to Algorithms, 2nd edition, 2001.
3. Fischer and Leblanc, Crafting a Compiler with C

4. Clean Code: A Handbook of Agile Software Craftsmanship by Robert C Martin
5. Programming Pearls by Jon Bently
6. Algorithms + Data Structures = Programs by Niklaus Wirth
7. Thinking in C++ by Bruce Eckel
8. Thinking in Java by Bruce Eckel
9. Effective Java by Joshua Bloch
10. Python in Practice. Mark Summerfield
11. Python Cookbook. David Beazley and Brian K. Jones

CS 545 : Seminar

Credits: L-0-T-2-P-0: 2

PREREQUISITES

Major core courses of M.Tech IT program

COURSE OBJECTIVES

- to create an environment to engage students in delivering and listening to interesting talks that promotes discussion
- to provide students with opportunity to learn new concepts and skills acquired in core courses and further extend these ideas to solve research/industry related problems
- know how to read research papers critically and efficiently
- to learn fundamental principles, generalizations and important theories of Computer Science
- to enable students to find their own field of interest in academia, industry or entrepreneurship
- to help students develop their own learning and teaching styles and communication skills

COURSE CONTENT:

Student presentations:

- Each student will present one paper during the term

Class evaluations:

- Each week each student is asked to write a short evaluation of one of the papers being presented

Class Discussion:

- Discuss the papers – expose the flaws, analyse the writing, what was the impact?

COURSE OUTCOMES

Because the curriculum is about individuals, there are no specific course levels learning outcomes. Nevertheless, after completion of the course, students would be able to:

- Explain factual knowledge (terminology, classifications, methods, trends). of current areas of research.

- State and explain some fundamental principles, generalizations, or theories the student has learned in this course.
- To apply gained knowledge in thinking, problem solving, or decisions making process.
- To achieve specific skills, competencies, and points of view needed by computing professionals.
- to judge the value of different contributions
- to identify promising new directions

Books/References:

A list of works will be posted by mentors/teachers at the start of the course. The students also have the option of choosing works according to his/her own areas of interest.