Curriculum of the iMTech Program for B.Tech. (IT) and M.Tech. (IT) at IIIT-Bangalore for Class of 2017

July 31, 2012,
IIIT-Bangalore.

International Institute of Information Technology, Bangalore.

Contents

1 Dual Degree Program Curriculum Structure

1.1 Learning Tasks, Achievements and Credits

Learning takes place through different types of courses and research projects. Student achievement in different learning tasks is assessed though a process of tests, examinations, term papers, seminar presentations, etc. The institute has a well-defined grading system in place.

The credit system is summarized in Table 1.

1.2 Streams

Streams indicate areas of specialization. Choice of stream enables students interested in particular areas to choose several courses in the stream of their choice.

1.2.1 List of Streams

There will be two streams which will be an umbrella of *focus areas*. Following are the streams with their focus areas:

- CSIS (Computer Science and Information Systems) stream
  - Computer Science,
  - Database and Information Systems,
  - Information Technology and Society,
  - Software Engineering.
- E&CE (Electronics and Communication Engineering) stream
  - Embedded System Design,
  - Information Technology and Society,
  - Networking and Communication Systems,
  - Signal Processing.

The student strength for each stream will be capped – a stream cannot have more than 50% of the students enrolled for the year in it, thus ensuring that both streams are equally loaded.

1.3 Courses: Categories and Levels

Different types of courses in IIIT-B are well-defined in the manual for the M.Tech. program of the institute. However, the iMTech program will have a large undergraduate component. It becomes necessary to identify explicitly courses at the undergraduate and the graduate levels. For this purpose, courses have been classified into levels.

- Level 1 courses are undergraduate level courses.
- Level 2 and Level 3 courses are graduate level courses, at basic and advanced levels, respectively.

It should be noted that the credit system takes into consideration the course levels also. The credit system for *undergraduate level (Level 1)* courses and *graduate level (Level 2 and higher)* courses is as shown in Table 1.

Undergraduate Level (Level 1) Courses			Graduate Level (Level 2 & Higher) Courses		
Type	Interaction Time	#Credits	Type	Interaction Time	#Credits
Lecture	1 hour/week/semester	1	Lecture	3 hours/week/semester	4
Tutorial	1 hour/week/semester	1		2 hours/week/semester	3
Laboratory	2 hours/week/semester	1	Laboratory	2 hours/week/semester	1
			Project Elective / Supervised Reading	per semester	4
			Thesis	two semesters	32 (Maximum 20 per semester)

TABLE 1
Proposed Credit System Based on Course Levels and Types.

Part	Course Type	Course Details	# Courses	# Credits
IT	Core		11	46
	Elective		11	44
	Summer Internship		-	4
	M.Tech. Thesis		-	32
	Total # Credits of the IT Part			**126**
Non-IT	Core	Basic Engineering Sciences / Skills	5	20
		Mathematics	4	16
		Physics	2	8
		Chemistry / Introductory Bioscience	1	4
		Introduction to Profession	1	2
		English	1	2
		Technical Communication	1	2
		Physical Education (Pass/Fail)	2	0
		Total # Credits		51
	Elective	HSS / Management	4	16
		Application of IT to Domains	2	8
		Total # Credits		24
	Total # Credits of the Non-IT Part			**78**
Total # Credits of the Program (≈ 62% IT; ≈ 38% Non-IT)				204 (= 126 + 78)

TABLE 2
Broad Content Structure of the Proposed Curriculum.

1.4 iMTech Program Curriculum Structure: Components and Credits

The following decisions emerged after detailed discussions and study of similar programs in premier institutions/universities in India and abroad:

- The period of the program will be minimum five years.
- The total number of credits required to fulfill requirements of the program is 204.
- The broad content structure of the program is summarized in Table 2, and its overall summary can be found in Table 3. At the top-level, the contents of the program are categorized into the Non-IT and the IT parts.
- The courses with L (lecture),T (tutorial) and P (practical) components (or only L and P components), the T and P components together (or just the P component, as the case may be) will be independently managed as separate courses and students assigned grades for them. The credit values for these parts should be based on the L, T, and P loadings of the courses, as given in Tables 4, 5, and 6, and should comply with the credit system, as given in Table 1.

Category	# Credits	Percentage (%)
IT core	46	22.5
IT electives	44	21.5
IT Thesis	32	15.68
IT Summer Internship	4	1.96
Mathematics + Physics + Chemistry/Introductory Bioscience	28	13.72
HSS / Management	16	7.84
Basic Engineering Sciences / Skills	20	9.8
Application of IT to Domains	8	3.92
Introduction to Profession + Technical Communication + English + Physical Education	6	2.94

TABLE 3
Credit Weightage for Course Categories in the Content Structure of the Proposed Curriculum.

1.5 The Non-IT Part

As shown in Table 2, the Non-IT part carries 78 credits, which is roughly 38% of the total number of credits for the program. The Non-IT part is divided into core and elective components.

1.5.1 The Non-IT Part: Core Component

The core component of the Non-IT part carries 54 credits.The core component can be further categorized into (a) the Basic Engineering Sciences and Skills courses, (b) the Mathematics and Sciences courses, and (b) miscellaneous courses.

1.5.1.1 Basic Engineering Sciences / Skills Courses

The Basic Engineering Sciences/Skills courses carry 20 credits, which is equivalent to five specific courses, as listed in Table 4. These courses are intended to provide foundational skill sets for a graduate of the iMTech program. Table 4 also gives the allocation of lectures, tutorials and practicals components (L:T:P) of the courses.

The broad objective(s) of each course in this category and its recommended course contents are included in Appendix A.

	Course Name	Category (Basic Engg. Science/Skill)	L:T:P	Credits
1	Basics of Electronics, Digital Logic and Circuits	Science	3:1:2	5
2	Signal Processing	Science	3:0:0	3
3	Signals and Systems	Science	3:1:0	4
4	Programming I	Skill	3:0:2	4
5	Programming II	Skill	3:0:2	4

TABLE 4
Credit Distribution for the Basic Engineering Sciences/Skills Courses in the Proposed Curriculum.

1.5.1.2 Mathematics and Sciences Courses

The Mathematics and Sciences courses carry 28 credits, which is equivalent to four Mathematics courses, two Physics courses, and one Chemistry or Introductory Bioscience course. The following are a few salient features of these courses:

- Students may opt for a course in Chemistry or in Introductory Bioscience.
- Table 5 shows the allocation of lectures, tutorials, and practicals (or laboratory) components (L:T:P) for Mathematics, Physics, Chemistry, and Introductory Bioscience courses.
- Mathematics courses will include continuous and discrete mathematics, algebra, graph theory, etc. as are deemed important.

Appendix D gives course descriptions of the Mathematics and Basic Sciences Courses.

Course Type	L:T:P	Credits
Mathematics	3:1:0	4
Physics / Chemistry / Introductory Bioscience	3:0:2	4

TABLE 5
Credit Distribution in the Mathematics and Sciences Courses in the Core Component of the Non-IT Part of the Proposed Curriculum.

1.5.1.3 Miscellaneous Courses

Miscellaneous courses, consisting of Introduction to Profession, English, Technical Communication, and Physical Education, carry six credits. A few salient features of these courses are:

- Physical Education courses, which are required to be done in the first two semesters, are zero-credit pass/fail courses. Students should mandatorily *pass* these courses.
- Introduction to Profession, English, and Technical Communication are 2-credit courses.

1.5.2 The Non-IT Part: Elective Component

The elective component of the Non-IT part carries 24 credits. The elective component can be further categorized into (a) Humanities, Social Sciences, and Management courses, and (b) courses on Application of IT to Domains.

1.5.2.1 Humanities, Social Sciences, and Management Courses

Humanities, Social Sciences and Management courses carry 16 credits, which is equivalent to four courses. These courses are to be selected from a set of courses. Following are the sets of courses to select from:

- Level 1 courses:
  - Society and Technology.
  - Introduction to Epistemology and Approaches to Understanding the World.
  - Economics.
  - Marketing and Strategy.
  - Accounting and Finance.
  - Introductory Cognitive Sciences.

- Level 2 courses:
  - From the Industrial Era to the Information Era.
  - The Economics of Information and ICTs.
  - The Cultural Logic of the Cyberworld.

The following are a few salient features of these courses:

- Students must take at least one Level 2 course.
- This list of courses may be modified from time to time.

1.5.2.2 Courses on Application of IT to Domains

These courses carry eight credits, which is equivalent to two courses. Students have to select these courses from a stipulated list.

This component is for study of deployment of information technology in some important domains and exploration of various aspects of this deployment, including technical, economic, social, manpower, environmental and other relevant aspects. These courses are categorized under the Non-IT part, since the emphasis here would not solely be on technical aspects of information technology per se, but also on non-technical aspects, as mentioned above, in large-scale use of information technology in particular domains.

Important areas such as banking and finance, health, infrastructure, transportation, e-governance, energy, water, etc. would be a few of the interesting domains to develop this category of courses in. These courses will involve several of the following features: lectures, seminars, field work, case studies, mini-projects, etc.

These courses will be undertaken in the seventh and the eighth semesters of the program.

1.6 The IT Part

As shown in Table 2, the IT part carries 126 credits, which is roughly 62% of the total number of credits for the program. The IT part comprises of the core and the elective components, M.Tech. thesis and summer internship.

1.6.1 The IT Part: Core Component

The core component of the IT part is intended to give the students a broad-based knowledge covering the entire spectrum of IT as defined in the vision and objectives of the program. The core component of the IT part carries 46 credits, over 11 courses.

1.6.1.1 Core Courses: Discussions Regarding Choice of Courses

The core courses are mandatory for all students in the iMTech program. These are the courses that are considered to be relevant for all students irrespective of their eventual choice of stream. The IT core courses were drawn from all the computing disciplines such as Computer Science, Information Systems, Computer Engineering, Information Technology and Software Engineering, as defined by ACM/IEEE [1].

The core courses have lecture, tutorial and/or practical (or laboratory) components. Table 6 gives a list of recommended core courses, with the allocation of credits for each course between lectures, tutorials and practicals (L:T:P).

	Course Name	L:T:P	Credits
1	Computer Architecture and Organization	3:1:2	5
2	Computer Networks	3:0:2	4
3	Data Structures and Algorithms	3:1:2	5
4	Database Systems	3:1:2	5
5	Design and Analysis of Algorithms	3:1:0	4
6	Digital Communication	3:0:2	4
7	Discrete Mathematics	3:1:0	4
8	Fundamentals of IT Infrastructure, Security and Management	3:0:2	4
9	Operating Systems	3:0:2	4
10	Programming Languages	3:0:0	3
11	Software Engineering	3:0:2	4

TABLE 6
Credit Distribution in the Core Component of the IT Part of the Proposed Curriculum.

The broad objective(s) of each core course and its recommended course contents are included in Appendix B.

1.6.2 The IT Part: Elective Component

The elective component of the IT part carries 44 credits over 11 courses. This component varies for each student, as it depends on the stream choice of the student.

1.6.2.1 Stream-based Classification of Courses

The recommended list of courses in the focus areas in each of the streams has been included in Appendix C.

1.6.2.2 Constitution of the Elective Component

- Students should take not less than seven *stream elective* courses, which are courses in their chosen stream.
- Students should take at least three *open electives*, which are elective courses chosen from the stream different from the student's choice of stream.

The stream and open electives may include project electives, and supervised reading declared for the respective streams.

Students will be strongly recommended proper counselling for a proper choice of electives. The choice should primarily depend on the research area chosen by a student for his Master's thesis.

1.6.2.3 Stipulations Regarding Choices of Stream, Stream Electives and Open Electives

The following stipulations regarding choices of stream, stream electives and open electives are recommended:

- Students have to declare their stream at the end of the fifth semester of the program.
- Students can register for the IT elective courses from the sixth semester onwards.

1.6.3 The IT Part: M.Tech. Thesis

All iMTech program students have to do a thesis under the supervision of a faculty member of the institute. This is in accordance with:

- The emphasis on the research component in similar programs in leading institutions in India and other countries, and
- The recommendations of the Rama Rao committee in 1995 [2] which reviewed postgraduate engineering education in India, and whose recommendations have been widely adopted.

The distribution of the thesis work in the fifth year of the iMTech program is 12 credits in the ninth semester and 20 credits in the tenth semester.

1.6.4 The IT Part: Summer Internship

At the end of the sixth semester of the iMTech program, all students are required to do a 4-credit summer internship in an organization outside IIIT-B. The summer internship is expected to provide exposure to the industry practices and trends.

1.7 Semester Plan

The semester plan for the iMTech program is included in Appendix E.

2 Student Selection

2.1 Eligibility

Prospective candidates for the program should have done their 10+2 or equivalent level with Mathematics and must have obtained first class or equivalent.

2.2 Admission Process

Admission to the program is on the basis of the rank obtained All-India Engineering Entrance Examination (AIEEE). Selection for offering admission to the program will be made as per the appropriate administrative process(es) as defined by the institute.

References

[1] "Computing Curricula 2005: The Overview Report," pp. 1–62, September 2005. [Online]. Available: http://www.acm.org/education/curric_vols/CC2005-March06Final.pdf

[2] P. Rama Rao (Committee Convener), "Reshaping Post Graduate Education in Engineering and Technology," *Report of the committee for review of Postgraduate Technical Education*, 1995.

Appendix A: Course Descriptions of Basic Engineering Sciences/Skills Courses

Basic Engineering Sciences Courses:

Basics of Electronics, Digital Logic and Circuits

Objectives

This course has two components with separate credits:

(i) Basics of Digital Logic and Circuits (3 credits),
(ii) Basics of Electronics: Analog and Digital Circuits Laboratory (2 credits).

Basics of Digital Logic and Circuits component is theory centric whereas Basics of Electronics component is laboratory centric. The grade awarded to a student may reflect the consolidated assessment of the two components.

The objectives of the course can be broadly stated as the following:

(i) Basics of Digital Logic and Circuits: The objectives of this component include providing the students a basic understanding of the principles of digital electronics and digital design.
(ii) Basics of Electronics: Analog and Digital Circuits laboratory: This component aims to give students an introduction to various electronic components and basic circuits, both analog and digital, through lectures and laboratory experiments. This will enable them to understand the working of systems of which these components and circuits are basic building blocks. The lectures are built around laboratory experiments.

Proposed Course Contents

(i) Basics of Digital Logic and Circuits:
3 lecture hours/week, totalling $\approx$ 42 hours.

- Number systems and codes:
  - Review of number systems; binary arithmetic; binary weighted and non weighted codes; error detecting and error correcting codes.
- Boolean algebra:
  - Postulates and theorems, representation of switching functions; SOP & POS forms; Karnaugh map representation; minimisation using K-map.
- Design of combinational circuits:
  - Tabular minimisation; design of single output and multi output functions; design using conventional AND, OR, NOT, NAND, NOR & EX-OR gates; design using MSI & LSI devices; digital multiplexer/selector, decoder, demultiplexer; design of 4 bit adder, carry look-ahead adder, magnitude comparator; BCD converter, logic implementations using ROM, PAL & PLA.
- Introduction to sequential circuits
  - Combinational *vs.* sequential circuits, asynchronous *vs.* synchronous circuits; state table and state diagram; state assignment; memory elements and their excitation functions; T flip-flop, D flip-flop, R-S flip-flop, JK flip-flop and their excitation requirements; design of synchronous sequenetial circuits like sequence detectors and binary counters.
- Capabilities and minimisation of sequential machines:

  - Melay and Moore machines; capabilities and limitations of finite state machine; state equivalence and machine minimisation.
- Algorithmic state machines:
  - ASM chart, timing considerations, control implementation; design with multiplexers and PLA control; introduction to unate functions and threshold logic

The recommended textbooks for this component of the course are as follows:

- "Switching and Finite Automata Theory," by Kohavi, TMH edition.
- "Digital Logic Design," by Mano.
- "Introduction to Switching Theory and Logic Design," by F.J. Hill and Peterson, John Wiley Publications.
- "Digital Design with Standard MSI and LSI," by Thomas, Blakeslee, Wiley Interscience Publication.

(ii) Basics of Electronics: Analog and Digital Circuits Laboratory:
Includes $\approx$ 20 lecture hours and 8 laboratory experiments.

In addition to the above-stated objectives of this component, upon the successful completion of this course, the student is expected to:

- Understand and experiment with electronic components and circuits in the lab.
- Be familiar with common electronic instruments. He/she would have acquired the basic skills to handle components and operate the instruments with confidence.
- Have developed interest in experimentation and learnt to account for observations and debug well and quickly.
- Be able to design an experiment to verify a concept.
- Be able to design electronic circuits to solve a problem.

A few guidelines for the instructor:

- Each week will include one lecture hour and three lab hours. First and second week may be a lecture and discussion week devoted to introduction, basic familiarity with theoretical concepts and description of the components and instruments.
- Laboratory exercises are expected to help in understanding and appreciate theoretical concepts.
- Home assignments each week may be given to help students to make their home study hours more useful.
- Students should be encouraged to submit lab report in groups (of two) at the end of each lab experiment.
- Each student should design an interesting electronic circuit to solve a problem at the end of the course.
- Students may be encouraged to fabricate their designs.

Topic	#Lecture Hours
Unit 1: Electronic Components	5
Unit 2: Measuring Instruments & Signal Generators	2
Unit 3: Electronic Circuits	3
Unit 4: Analog Computer Design	2
Unit 5: A/D & D/A Conversion	2
Unit 6: Digital Circuits	1
Unit 7: Combinational Circuits	1
Unit 8: Sequential Circuits	1
Unit 9: Microcontrollers	3
Total	20

Brief description of the lecture units is as follows:

- Unit 1: Input/output and I-V characteristics of electronic components: Resistors, inductors, capacitors, transformers, diodes, Bipolar Junction Transistors (BJT), Field Effect Transistors (FET). Introduction to operational amplifiers and timer circuits and their characteristics.
- Unit 2: Measuring instruments and signal generators: Principle of operation of digital multi meters, voltage sources, signal generators, oscilloscopes and probes for measuring. Virtual instrumentation setup. General introduction to concepts of grounding, shielding and isolation. Types, ratings and identification of commonly used components in electronics.
- Unit 3: Electronic circuits using operational amplifiers and timers: Amplifiers, integrators, differentiators, rectifiers and waveform generators.
- Unit 4: Analog computer design using Operational Amplifiers and other electronic components and circuits. Circuits for addition, multiplication, finding logarithms, solving differential equations.
- Unit 5: A/D and D/A conversion: Sample and hold systems. Statement of sampling theorem. Digital to analog converters, analog to digital converters.
- Unit 6: Digital circuits: Boolean algebra, NOT, OR, AND, NAND, NOR and EXOR gates. CMOS logic gates, Propagation delay, BJT inverter and the TTL NAND gate. Analysis and synthesis of combinational circuits. Some applications using OR, AND, NOT, NAND, NOR, EXOR gates and multiplexers.
- Unit 7: Combinational circuits: Arithmetic functions, Multiplexers, Demultiplexers, Programmable Logic arrays.
- Unit 8: Sequential circuits: Flip flops, registers, ripple and synchronous counters.
- Unit 9: Microcontrollers: Early introduction to digital computers and programming. Exposure to assembly code and programming to build complex sequential circuits with combinational logic and registers. Discussion of a simple microcontroller.

Guidelines for Experiments:

Lab instructors are free to set experiments to illustrate the concepts given in the brief description of Lab experiments. Some examples are given for illustration. Depending on the level of understanding of the students experiments can be planned for the lab. The instructor can choose 8 out of the 10 suggested labs. The lab experiment can be altered, provided the spirit of the experiment is maintained.

Brief description of the labs is as follows:

- Lab 1: Getting familiar with instruments and components. Use of digital multi meters, voltage sources, bread boards, circuits, components and precautions to be taken while working in the lab (earthling, shorting and heating problems). Explore the transient and steady-state I-V characteristics of linear passive devices and some simple RC, RL and RLC circuits.
- Lab 2: Getting familiar with instruments and components. Getting familiar with signal generators, oscilloscopes and probes for making measurements. Introduction to virtual instrumentation set up and spectrum analyzer. Explore the I-V characteristics of diodes, BJTs and MOSFETs and simple half wave, full wave rectifier circuits and voltage regulation property of Zener diode.
- Lab 3: Electronic circuits using op-amps: Rectifiers, adders, integrators and differentiators.
- Lab 4: Electronic circuits using op-amps and timers: Amplifiers, oscillators.
- Lab 5: Experiment on A/D and D/A conversion or experiment on AC to DC conversion with op-amp, diodes and filters.
- Lab 6: Digital circuits: Experiments on digital gates using MOSFETs such as inverters, NAND, NOR, and multiplexers.
- Lab 7: Combinational circuits: Adders and other simple combinational circuits.
- Lab 8: Sequential circuits: Experiments with flip flops, registers and counters.
- Lab 9: Microcontroller: A simple experiment to illustrate the idea of programming and computer

design with a circuit consisting of combinational circuit (as a possible CPU to do arithmetic computations) and registers and other memories to read and write data.
- Lab 10: Build your own application.

Lab experiments will be as follows:

- Expt 1: Explore the transient and steady-state V-I characteristics of following linear passive devices and their combinations, and provide your observations: -
  (a) Resistors (R), Inductors (L), Capacitors (C).
  (b) RL, LC, RC and RLC circuits.
- Expt 2: Non-linear Passive component: Diode
  (a) Explore the V-I characteristics of diode in both forward and reverse biased modes.
  (a) Make half-wave, full-wave center-tapped and bridge rectifiers using diode, and provide your observations of the observed output for the applied relevant input.
  (a) Make a voltage regulator using Zener diode, and provide your observations of the observed output for the applied relevant input.
- Expt 3: Explore the V-I characteristics of following active devices and provide your observations:
  (a) NPN and PNP BJTs.
  (a) N-channel and p-channel MOSFETs.
- Expt 4: Make following amplifier circuits using BJTs:
  (a) Common-emitter amplifier using NPN or PNP BJT Observe voltages and currents at the output in the time domain for the applied relevant input voltages and currents. Provide and explain your observations.
- Expt 5: Make following digital gates using MOSFETs:
  (a) Inverter.
  (a) 2-input NAND gate.
  (a) 2-input NOR gate.
  (a) 2×1 multiplexer with 1-bit wide data.
  (a) Tristate buffer (optional).

  Provide observations for voltage at the output for all relevant patterns applied at the input(s).
- Expt 6: Make following digital gates using MOSFETs:
  (a) S-R latch using NAND gates.
  (a) D-type flip-flop using NOR gates.

  Provide observations for voltage at the output for all relevant patterns applied at the input(s).
- Expt 7: Make following digital circuit using discrete digital components (gates, FFs etc):
  (a) 4-bit up and down modulo-16 counters with reset and enable signals.
  (a) 4-bit up saturating counter with reset and enable signals.

  Provide observations for the voltage at all output pins for all relevant patterns applied at the input(s).
- Expt 8: Make following digital circuit using discrete digital components (gates, FFs etc) and provide your observations for all relevant patterns applied at the input:
  (a) One Mealy/Moore FSM chosen from a real-world application in scaled down form, e.g.,
    (a) Traffic light controller with no functionally for left- and right-turn control (only through traffic is controlled).
    (a) Vending machine with limited input currency accepted.
- Optional experiments:
  (a) Lab experience on operational amplifier and its applications to solve differential equations.
  (a) Lab experience on sensors and their characteristics.

The recommended reference books for this component of the course are as follows:

- ”Microelectronics,” by Grabel Millman, McGraw-Hill. International Edition, 1987.
- ”The Art of Electronics,” by Horowitz and Hill, 2nd Edition, 1989/1990 Cambridge University Press.
- ”A Symbolic Analysis of Relay and Switching Circuits,” by Claude E. Shannon, Transactions of American Institute of Electrical Engineers, Volume 57, 1938.

Signal Processing

Objectives

This course presents an introduction of the basic analysis tools and techniques for analog and digital processing of signals. In particular, Fourier and Z-Transform based Linear System theory will be covered in details.

Proposed Course Contents

- Discrete-time signals and systems.
- The Z-Transform and its applications.
- Frequency domain analysis.
- Discrete Fourier Transform.
- Fast Fourier Transform algorithms.
- Digital structures.
- Design of digital filters

Signals and Systems

Objectives

This course provides students with an exposure to the concepts of signals and systems.

Proposed Course Contents

- Dynamic representation of systems, systems attributes, causality, linearity, stability, time-invariance; special signals, complex exponentials, singularity functions (impulse and step functions); Linear Time-Invariant Systems, differential equation representation, convolution integral; discrete form of special functions; discrete convolution and its properties; realization of LTI systems (differential and difference equations).
- Fourier analysis of continuous time signals and systems, Fourier series, Fourier Transform and properties, Parsevals theorem, frequency response of LTI systems; sampling theorem.
- Fourier analysis of discrete time signals & systems, Discrete-time Fourier series, Discrete-time Fourier Transform (including DFT) and properties; frequency response of discrete time LTI systems.
- Laplace Transform and its inverse, definitions, existence conditions, region of convergence and properties, applications of Laplace Transform for the analysis of continuous time LTI system (stability etc.), significance of poles and zeros.
- Z-Transform and its inverse, definitions, existence, region of convergence and properties, applications of Z-Transform for the analysis of discrete time LTI systems, significance of poles and zeros.
- Random signals, introduction to probability, Bayes Theorem, concept of random variable, probability density and distribution functions, function of a random variable; moments, independence of a random variable; introduction to random processes; auto- and cross- correlations; wide-sense stationarity, power spectral density, white noise, random processes through LTI systems.

The recommended textbooks for this course are:

- "Signals and Systems," by Alan V. Oppenheim and Alan S. Willsky, Pearson Edn.
- "Communication Systems," by Haykin Simon, John Wiley.
- "Signals and Systems," by I. J. Nagrarth, Tata Mc Graw Hill.
- "Signals and Systems," by Farooq Husain, Umesh Pub.
- "Adaptive Signal Processing," by W. Bernad, Pearson Edn.

BASIC ENGINEERING SKILLS COURSES:

Programming I

Objectives

This course is first of the two programming courses. This knowledge area includes those skills and concepts that are essential to programming practice independent of the underlying specialization. As a result, this area includes units on fundamental programming concepts, basic data structures, algorithmic processes, and basic security. These units, however, by no means cover the full range of programming knowledge that a IT undergraduate must know. It is expected that a second programming course is taught that reinforces these concepts.

Proposed Course Contents

- Introduction to computer problem-solving.
- Fundamental data structures (Data types, representation of numeric data, strings, etc.).
- Fundamental algorithms.
- Factoring methods.
- Array techniques.
- Merging, sorting and searching.
- Text processing and pattern searching.
- Dynamics data structure algorithms.
- Recursive algorithms.

These topics are to be covered at a fundamental level with focus on practice rather than on theory.

The recommended textbook is "How to Solve it by Computers," by Dromey. This course is accompanied by C and Python programming laboratory session.

Programming I Lab

INTRODUCTION
It is important to start with C programming language purely from a skills perspective. Since C language is viewed as a mandatory skill by the industry in India, it is better to expose the students to C early on in the program. On the other hand, it is widely believed that one of higher-level programming languages is better suited to expose the students to programming concepts. In order to have a good mix of the two, the both C as well as Python programming languages for the labs of the Programming I course are recommended.

LAB SESSIONS & ENVIRONMENT
There will be a 2 hour lab session every week. The Programming I Lab will be based on Linux programming environment. In addition to learning C and Python programming languages, the students are expected to gain good user-level experience and skills of the Unix operating system.

(i) C Programming Lab

STRUCTURE
All the sessions of the C Programming Lab will end with the description of a stretch exercise that students can work on outside of the lab hours. The C Programming Labs are structured based on specific themes for each lab session. Each lab session is divided into multiple lab exercises.

Contents

- Lab 1: Preliminaries.
  - Objective: The objective of this lab is to familiarize the students with the C programming environment.
  - Exercises:
    * Introduction to Unix.
    * Basic I/O program 1.
    * Basic I/O program 2.
    * Basic I/O program 3.
    * Basic I/O program 4.
  - Comment: Lab 1 is intentionally kept light because the basic objective is to familize the student with the programming environment, which includes Unix operating system, editor, compilation, execution, etc.
- Lab 2: Data Types and Expressions.
  - Objective: The objective of this lab is to start using variables of various primary data types in the C language and use them as part of various expressions.
  - Exercises:
    * Variables and data types.
    * Type casting and data type conversion.
    * Expression evaluation.
- Lab 3: Control Flow.
  - Objective: The objective of this lab is to provide an introduction to control structures in C language.
  - Exercises:
    * Control: if statement.
    * Control: if-else statement.
    * Control: switch-case statement.
    * Iterative: for loop.
    * Iterative: while loop.
    * Iterative: do-while loop.
- Lab 4: Functions.
  - Objective: The objective of this lab is to introduce modular software development using functions.
  - Exercises:
    * Function exercise #1 (prototypes, void return and void parameters).
    * Function exercise #2 (parameters and return values).
    * Function exercise #3 (global variables).
    * Function exercise #4 (static variables).
    * Function exercise #5 (multi-file programming).
    * Introduction to built-in libraries (math.h, string.h, etc.).
- Lab 5: Recursion.
  - Objective: The objective of this lab is to understand recursion in C programming language.
  - Exercises:
    * Recursion exercise #1.
    * Recursion exercise #2.
- Lab 6: Arrays.
  - Objective: The objective of this lab is to introduce the students to arrays in C programming language.
  - Exercises:

    * 1-d array exercise #1.
    * 1-d array exercise #2.
    * 2-d array exercise #3.
    * n-d array exercise #4.
- Lab 7: Pointers.
  - Objective: The objective of this lab is learn about pointers in C language.
  - Exercises:
    * Pointers and addresses.
    * Pointers and function arguments.
    * Pointers and arrays.
    * Address arithmetic.
    * Character pointers and functions.
- Lab 8: More on Pointers.
  - Objective: The objective of this lab is learn about advanced concepts about pointers in C language.
  - Exercises:
    * Pointer arrays.
    * Pointers to pointers.
    * Pointers to functions.
- Lab 9: Structures.
  - Objective: The objective of this lab is to learn about structures in C programming language.
  - Exercises:
    * Basics of structures.
    * Structures and functions.
    * Arrays of structures.
- Lab 9: Advanced Structures and Unions.
  - Objective: The objective of this lab is to learn about advanced concepts in structures and unions in C programming language.
  - Exercises:
    * Pointers to structures.
    * Self-referential structures.
    * Unions.
    * Bit-fields.
- Lab 10: File I/O.
  - Objective: The objective of this lab is to learn how to do File I/O using C programming language.
  - Exercises:
    * Text I/O sequential access.
    * Binary I/O sequential access.
    * Binary I/O random access.
- Lab 11,12: C Programming Project.
  - Objective: The objective of the last two lab sessions is to do a non-trivial programming project that tries to make use a majority of the C programming language constructs and paradigmns. The project can be a group project with 3 members each. The size of the project should be such that completion of the project should be possible in about 4 hours of collective programming (about 10 person hours).

(ii) Python Programming Lab

OBJECTIVES

The Python part of the Programming I lab course will aim to get the students to get off the block as quickly as possible and start building programs for reasonably complex problems using the rich collection of constructs and built-in and other readily available libraries in Python. The focus will be on problem solving using Python as a tool. So the course is structured around a set of problems that are designed to introduce the students to language features in chunks till they are equipped to build a fairly non-trivial piece of code themselves.

The Python programming lab is intended to complement the C programming lab in several ways, e.g. in terms of richness of the programming constructs, use of a feature-rich IDE, and introduction to GUI programming. The idea is to give the students an opportunity to get hands-on experience with building projects that will make learning programming a fun-filled exercise.

CONTENTS

- Lab 1: Preliminaries.
  - Objective: Understand the interpreted nature of Python (as opposed to C). Appreciate the fact that to a large extent Python allows a "natural" style of programming. Carry out simple tasks using the Python interpreter command line.
  - Constructs Introduced: basic data types (string, int, float etc.), large integers in Python, collections (lists) and associative lists and operations on these; variables, assignment, operators, expressions; basic I/O; numerical computations using the Python math library. Creating and running Python source files (.py).
  - Class Exercises: Basic exercises on all the above.
  - Take Home Exercises: Output 3-letter month name given the month number using strings; Convert a date in the d/m/y format (d, m and y are day month and year respectively as numbers) to a given (fixed) format; Take the principal amount and the term of a loan and print the EMI.
- Lab 2: Control Structures (Loops and Conditionals) and Functions.
  - Objective: Use control structures to direct the "flow" of computation. Get a basic understanding of modularization using functions and its role in dealing with complexity, maintainability and readability of programs.
  - Constructs Introduced: if-then-else; while- and for-loops; Iterators on lists. Conditional expressions; Functions and their arguments. Basic object oriented dot(.) notation.
  - Class Exercises: Pictorial numbers. Convert a number in words to numeric. Random Number Generator.
  - Take Home Exercises: Binary Search; Simulate a queue; Find the average of all the input numbers until a prompt; Invert a string; Find the square root of a number using Newtons method where the iterative formula is given; Generalization of the pictorial numbers exercise; Convert roman numerals to decimal and vice versa; Answer simple questions with a fixed structure (e.g. Is the dolphin a mammal?) using an associative list as a "database" of animals with their classification.
- Labs 3,4: More Exercises on Loops & Functions. Recursion.
  - Objective: Get comfortable with the idea that functions can call themselves. More involved exercises using loops, functions and recursion.
  - Constructs Introduced: Use of random.py module. Command-line arguments.
  - Class Exercises: Quicksort. Miller-Rabin Primality Test.
  - Take Home Exercises: Complete the Quicksort and Miller-Rabin Primality Test; Solve the Koenigsburg Problem on graphs; Write a decoder for a text that has been encrypted using a Caesar cypher.
- Labs 5,6: Functional Style Programming.
  - Objective: Exposure to functional style of programming lazy evaluations, functions as arguments, etc..
  - Constructs Introduced: Eval function. Lambda construct. Yield construct. Map-Filter-Reduce functions.

  - Class Exercises: Creating a Unix-style find with parameters acting as filters on the file attributes. Symbolic algebra.
  - Take Home Exercises: Extend symbolic algebra to say symbolic differentiation etc.; Simple transliteration of an Indian language text into another Indian language; Simple Natural Language Interface to the Unix file system.
- Labs 7,8: File I/O and Byte-level File Manipulation in Python.
  - Objective: Work with bytes and low-level I/O in files. Use of external modules and libraries in Python.
  - Constructs Introduced: Byte-Array, Hexadecimals and Octals. File loops. Modules such as heapq.py.
  - Class Exercises: Compression using Huffman Code. RSA.
  - Take Home Exercises: Complete the compression and extraction programs based on Huffman code. Implement a naïve RSA encryption-decryption.
- Labs 9,10: Object Orientation & GUI Using Python.
  - Objective: An elementary familiarity with OO notions. Ability to create and work with simple GUIs and graphics.
  - Constructs Introduced: Classes, wxPython library and some graphics library like VPython.
  - Class Exercises: Geometric shapes and some simple primitives. Convex hull.
  - Take Home Exercises: Operations on sparse matrices.
- Labs 10+: Mini Projects.
  - Objective: Creating reasonably sized working projects using libraries and some glue Python code.
  - Exercises:
    * Create a mashup that looks at the IIITB Weblogs, find out the geographical location of the IP addresses where the hits are coming from (use one of the freely available IP locator APIs/Websevices), and mark them on Google Maps bigger marker for the locations that generate more hits.
    * Make a bouncing balls program draws a set of "walls" of arbitrary sizes and orientations and creates a few (1-3) balls with randomly chosen initial velocities and let them bounce around in a manner consistent with the physical laws of motion.
    * ...

Programming II

Objectives

This is a second course on programming that is expected to reinforces the concepts taught in Programming I and the Data Structures course. This course mainly introduces the students to object-oriented programming and the basics of event driven programming.

Proposed Course Contents

- Object-oriented design.
- Encapsulation and information-hiding.
- Separation of behavior and implementation.
- Classes and subclasses.
- Inheritance (overriding, dynamic dispatch).
- Polymorphism (subtype polymorphism *vs.* inheritance).
- Event-handling methods.
- Event propagation.
- Exception handling.

This course is accompanied by C++ and Java programming laboratory session.

APPENDIX B: COURSE DESCRIPTIONS OF MATHEMATICS AND BASIC SCIENCES COURSES

On consultation with various subject matter experts, the course descriptions for the Mathematics and Basic Sciences courses have been derived. The course contents of the Introductory Biosciences course has been finalized after consultation with the Institute of Bioinformatics and Applied Biotechnology (IBAB) and the National College.

MATHEMATICS COURSES

Mathematics I: Basic Calculus

Topic	#Hours
Principles of Inference: Negation, Disjunction, Implication, Equivalence, Truth tables and tautologies.	4
Real numbers, Sequences, Series, Limit, Continuity, Differentiability, Mean value theorems and applications, (with less stress on ϵ, δ) Linear Approximation.	8
Power series, Taylor's theorem (one variable), Approximation by polynomials, Critical points, Convexity.	6
Riemann integral, fundamental theorems of integral calculus, Improper integrals.	7
Curve tracing, Graphs of polar equations (optional)	
Space co-ordinates, lines and planes, Polar coordinates, Cylinders,	9
Quadric surfaces, volume, area, length, Continuity, Differentiability of vector functions, arc length, Curvature, torsion, Serret-Frenet formulas, Functions of two or more variables, partial derivatives, Statement only of Taylor's theorem and criteria for maxima/minima/saddle points.	
Double, triple integrals, Jacobians, Surfaces, integrals, Vector Calculus, Green, Gauss, Stokes Theorems.	8
Total	42

Recommended books:

- ″Calculus and Analytic Geometry,″ by Thomas and Finney, 9th edition, Pearson Education India.
- ″Introduction to Logic,″ by Patrick Suppes, Dover edition, 1999. (For elementary logic portion).
- Proper web notes.

Additional references:

- ″Mathematical Analysis,″ by Tom M. Apostal, Addison-Wesley, 1974.
- ″Theory of Computation,″ by Harry Lewis and Christos H. Papadimitriou, Prentice-Hall, 2nd edition, 1997. (For logic).
- ″Real and Complex Analysis,″ by Walter Rudin, McGraw - Hill, 2006.

Mathematics II: ODE & PDE

Topic	#Hours
Introduction and Motivation to Differential Equations, First Order ODE, Geometrical interpretation of solution, Equations reducible to separable form, Exact Equations, integrating factor, Linear Equations, Orthogonal trajectories, Picard's Theorem for IVP (without proof) and Picard's iteration method, Euler's Method, Improved Euler's Method.	6
Second Order Linear differential equations, fundamental system of solutions and general solution of homogeneous equation, use of known solution to find another, Existence and uniqueness of solution of IVP, Wronskian and general solution of nonhomogeneous equations, Euler-Cauchy Equation, extensions of the results to higher order linear equations.	12
Power Series Method - application to Legendre Equations, Legendre Polynomials, Frobenius Method, Bessel equations, Properties of Bessel functions, Sturm comparison Theorem, Sturm Liouvile BVP, Orthogonal functions, Fourier Series and Integrals.	9
Laplace and Fourier Transforms (with less stress on theoretical aspects)	7
Introduction to PDE, basic concepts, Linear and quasi-linear first order PDE, second order PDE and classification of second order linear PDE (Canonical forms), D'Alemberts formula and Duhamel's principle for one dimensional wave equation, Laplce and Poisson's equation, maximum principle with application in two and three dimensions, Fourier Method for IBV problem for wave and heat equation in rectangular regions.	14
Total	48

Recommended books:

- "Advanced Engineering Mathematics," by Erwin Kreyszig, 8th edition, Wiley, ISBN-10: 0470458364, ISBN-13: 978-0470458365.
- "Partial Differential Equations," by Fritz John, 4th edition, 1981. (For basic concepts, Linear and quasi-linear first order PDE).
- Proper web notes (NPTEL notes are available).

Additional references:

- "Differential Equations with Applications and Historical Notes," by George F. Simmons, McGraw-Hill Science/Engineering/Mathematics, 2nd edition, 1991
- "Introduction to Ordinary Differential Equations," by Shepley L. Ross, 4th edition, Wiley, 1989.
- "Elements of Partial Differential Equations," by Ian Sneddon.
- "An Elementary Course in Partial Differential Equations," by Amaranath, Alpha Science Intl. Ltd., 1997.
- "Advanced Theory of Statistics," by Kendall and Stuart, all volumes.

Mathematics III: (2+2 credits)

Summary:
Mathematics III is a composition of two modules, each running for half a semester. We may use the nomenclature “Mathematics IIIA” and “Mathematics IIIB” for the 2 modules.

Topic	#Hours
Mathematics IIIA: Complex Analysis	
Complex Numbers, geometric representation, powers and roots of complex numbers, Functions of a complex variable, Analytic functions, Cauchy-Riemann equations; elementary functions, Conformal mapping (for linear transformation).	8
Contours and contour integration, Cauchy’s theorem, Cauchy integral formula, Power Series, term by term differentiation, Taylor series, Laurent series, Zeros, Singularities, poles, essential singularities, Residue theorem, Evaluation of real integrals and improper integrals.	12
	20
Mathematics IIIB: Statistics and Probability (Adopted from : Chap. 22 and 23 Kreyszig- Part G):	
Data representation, average, probability, permutations and combinations, random variables, probabilistic distributions, mean & variance, binomial, Poisson, hypergeometric, Normal distributions, distributions of several random variables.	14
Mathematical Statistics: random sampling, confidence intervals, testing of hypothyses, decisions, goodnes of fit, χ^2 test, linear regression.	7-8
	21-22
Total	41-42

Recommended books:

- Mathematics IIIA:
  - ”Advanced Engineering Mathematics,” by Erwin Kreyszig, 8th edition, Wiley, ISBN-10: 0470458364, ISBN-13: 978-0470458365.
  - Proper web notes (NPTEL notes are available).
- Mathematics IIIB:
  - ”Advanced Engineering Mathematics,” by Erwin Kreyszig, 8th edition, Wiley, ISBN-10: 0470458364, ISBN-13: 978-0470458365.
  - Proper web notes.

Additional references:

- Mathematics IIIA:
  - ”Complex Analysis,” by Ahlfors, McGraw Hill, 1979.
  - ”Complex Variables and Applications,” by James Brown and Ruel Churchill, McGraw Hill, 2008.
- Mathematics IIIB:
  - ”Introduction to Mathematical Statistics,” by Hogg and Craig, 3rd edition, Macmillan, 1971.
  - ”The Advanced Theory of Statistics,” by Kendell and Stuart, Volume 3, Griffin, 1976.

Mathematics IV: Linear Algebra & Algebra

Topic	#Hours
Matrices, Matrix Operations (Addition, Scalar Multiplication, Multiplication, Transpose, Adjoint) and their properties; Special types of matrices (null, Identity, Diagonal, Triangular, Symmetric, Skew-Symmetric, Hermitian, Skew-Hermitin, Orthogonal, Unitary, Normal), Solution of the matrix equation Ax=b, Row-reduced Echelon form, Determinants and their properties (without proof).	8
Vector Space, Subspaces, Linear Dependence/Independence, Basis, Standard Basis of dimension, Coordinates with respect to a basis, Complementary Subspaces, Standard inner product, Norm, Gram-Schmidt Orthogonalisation Process, Generalisation to the vector space Linear Transformation form (motivation X-AX), Image of a basis identifies the linear transformation, Range Space and Rank, Null Space and Nullity, Matrix Representation of a linear transformation, Structure of the solutions of the matrix equation Ax=b, Linear Operators and their representation by matrices, Similar Matrices and linear operators, Invertible linear operators, Inverse of a non-singular matrix, Cramer's method to solve the matrix equation Ax=b, Eigenvalues and eigenvectors of a linear operator, Characteristic Equation, Bounds on eigenvalues, Diagonalisability of a linear operator, Canonical forms.	12
Standard Inner product, Norm, Gram-Schmidt Orthogonalisation Process, Self-Adjoint, Normal and Unit array operators, Properties of eigenvalues and eigenvectors, Spectral theorem Self-Adjoint and Normal, Quadratic form $X^T AX$, Positive and Semi-Positive Definite Matrices.	10
Introduction to Abstract Algebra: Groups, Rings, Modules, Ideals, Fields and examples of finite fields.	10
Total	40

Recommended books:

- ”Linear Algebra,” by K. Hoffman and R. Kunz, Prentice-Hall, 1971.
- Relevant portions from the books written by Artin, Galien, Herstain.
- Proper web notes.

Additional references:

- ”Linear Algebra and its Applications,” by Gilbert Strang, Nelson Engineering, 2007.
- ”Finite Dimensional Vector Spaces,” by P. R. Halmos, Princeton University Press.
- ”Linear Algebra,” by Helson, Holden-day, 1990.
- ”Lectures on Abstract Algebra,” volumes by N. Jacobson, Springer.

Chemistry Course

Theory Component

Unit #	Topic
I	Schrödinger equation; interpretation of wave function; hydrogen atom. Structure and bonding; atomic and molecular orbitals; VSEPR; energy levels in molecules and solids. Trends in the Periodic table.
II	Chemical kinetics; steady state approximation; collision theory. Catalysis, chemical potential; fugacities, activities; equilibrium constants and free energy; relationship between G and emf; standard potentials.
III	Transition elements and their uses ; catalysis; semiconducting and super conducting materials, zeolites and spinels. Coordination Chemistry; Transition metal ions and complexes; 18-electron rule; simple ligands such as CO, ethylene, triphenylphosphine. Organometallic chemistry; homogeneous catalysis; magnetochemistry; role of metal ions in biological processes.
IV	Metallurgy; basic principles; extraction and refining of metals and applications.
V	Reaction mechanism; nucleophilic, electrophilic substitutions; free radical addition; additions and substitutions to aromatic systems; addition to compounds containing carbonyl groups. Linear and cyclic conjugation; benzene; aromaticity and properties of conjugated systems.
VI	Structure of organic molecules; conformations of alkanes and cycloalkanes; glucose and fructose; E and Z; Anomeric effect. R and S; molecular chirality; optical and geometrical isomerism; importance of optical activity in drug synthesis and biological activity.
VII	Synthesis of organic molecules; photochemistry of organic and bio molecules; chemistry of life processes; molecular systems of technological and biological importance; biotechnology and biomedical applications.

Laboratory Component

Experiments related to general, organic, physical, inorganic and bio- chemistry.

- 1. Estimation of iron in heamatite ore.
- 2. Estimation of hardness in water.
- 3. Estimation of available chlorine in bleaching powder.
- 4. Redox titration by potentiometry.
- 5. Dissociation constant of polybasic acid using pH titration.
- 6. CMC of soap solution by conductivity measurements.
- 7. Iron-orthophenanthroline by colorimetry.
- 8. Estimation of glucose by DNS method.
- 9. Estimation of inorganic phosphate by Fiske-Subbaraw method.
- 10. Preparation of Aspirin; meta dinitrobenzene.
- 11. & 12. Functional groups analysis in organic compounds: hydrocarbons, alcohols, aldehydes, ketones, carboxylic acids, amines, amides, phenols, nitro compounds.

INTRODUCTORY BIOSCIENCES COURSE

Objectives

Students who successfully complete this course will be able to:

- Identify major kingdoms, a few important phyla and their main characteristics.
- List the taxonomical features of organisms that are commonly used as research models and discuss the probable relevance of data from such models to various other species.
- List the major pathogens affecting lives of humans, and important plants and animals of importance to human needs, directly or indirectly; also explain the important taxonomical characters, as well as brief their life cycles.
- Relate the diversity to the evolution concepts.
- Explain the interactions within populations and the surrounding abiotic/biotic factors.
- Understand the relationship between population dynamics and evolution (genetic drift and related concepts).
- Describe the significance of balance of variousinteractions and recycling of chemical constituents within a system.
- Identify major animal and plant physiological systems, their constituents (organs and tissues) and the mechanisms of operation.
- Discuss a few important examples of common disorders affecting various systems in humans and plants.
- Explain the peculiarities of germ cell production in the context of genetic diversity, and general physiology.
- Explain concepts in genetics and developmental biology.
- Establish the relationship between reproduction, cell divisions, genetics, development and differentiation.
- Describe the basic principles of inheritance and their discovery; explain genetic linkage and its applications.
- List the major human genetic disorders and explain their characters.

The following textbooks are recommended for this course:

- "Typologies and Taxonomies: An Introduction to Classification Techniques (Quantitative Applications in the Social Sciences)," by K. D. Bailey, Sage Publications, 1994.
- "Evolution," by M.W. Strickberger, Jones and Barlett publishers Inc., London, 1996.
- "Organizing Knowledge: Taxonomies, Knowledge and Organizational Effectiveness," by P. Lambem, Chandos Publishers, 2007.
- "Ecology, Concepts and Applications", by M. C. Molles, McGraw-Hill Higher Education, 2nd edition, 2002.
- "Ecology: from Individuals to Ecosystems," by M. Begon, C. R. Townsend and J. L. Harper, Blackwell Publishing, 4th edition, 2005.
- "Plant Physiology," by L. Taiz and E. Zeiger, Palgrave Academic Publisher, 4th edition, 2006.
- "Principles of Anatomy and Physiology," by G. J. Tortora and B. H. Derrickson, Wiley Higher Education, 11th edition, 2006.
- "Endocrinology: An Integrated Approach," by S. S. Nussey, and S. A. Whitehead, London:Taylor & Francis, 2001.
- "Genes IX," by B. Lewin, Jones & Bartlett Publishers, Inc., 9th edition, 2008.
- "An Introduction to Genetic Analysis," by A. J. F. Griffiths, J. H. Miller, D. T. Suzuki, R. C. Lewontin, and W. M. Gelbart, W. H. Freeman & Co., 8th edition, 2000.
- "Developmental Biology," by S. F. Gilbert, Sunderland (MA): Sinauer Associates, Inc., 2000.

Proposed Course Contents

Theory Component

Unit #	Topic
I	Taxonomy and Biodiversity: Broad basis of classification of major kingdoms: protista, plants, animals; reference to NCBI sources; research models from different major taxa; taxonomy and brief life cycles of examples (major ones only) pests in plants, animals and humans.
II	Ecology: Introduction to ecosystem concepts; brief description of abiotic and biotic factors influencing dynamics of ecosystem; population ecology; urban ecosystems and impact of modern human lifestyles on various ecosystems.
III	Plant and Animal Physiology: Basics of concepts in cellular vs. general physiology; an overview of plant physiology: plant tissues and functions, water balance, solute transport, translocation (phloem), plant growth, respiration and photosynthesis; human physiology: digestive system, circulation, excretion, muscle structure and functions, and neural tissues and functions; examples of common disorders affecting different systems in humans and plants.
IV	Genetics and Evolution: Mandelian laws of inheritance, examples of multiple alleles governing one phenotype; overview of cytogenetics and genetic linkage; brief overview of molecular genetics; types of mutations and their role in evolution; major human genetic disorders; Darwin's theory of evolution, origin of life (current theory), natural selection and concept of speciation, brief history of species, phylogeny, cladistic systematics, molecular basis of evlolution.
V	Reproduction and Developmental Biology Significance of meiosis in sexual reproduction, differences between male and female gamete generation, fertilization, embryogenesis, early development (– plants and humans), role of hormones in germ cell production and early development and puberty; brief discussion on diversity in these processes across other important species, including plants.

Laboratory Component

(10 sessions of 3 hours or 15 sessions of 2 hours each).

- 1. Microscopy:
  - a) Introduction (Self prepared PPT – from Leeuwenhoeks Simple Microscope to High Voltage Electron Microscope/SEM, etc.) followed by the usage of simple, monocular, binocular, oil immersion microscopes.
  - b) Mounting of algae, fungal hyphae, spores, macerated plant tissue, ...
  - c) Stem /root sections – TS, VS, free hand/paraffin slides, animal histology slides.
  - d) Significance of staining, preservation, permanent slides, cytological preparations.
- 2. Biodiversity:
  - a) Plantae: Bryophyta, Pteridophyta, Gymnospermae and Angiospermae.
  - b) Animalia: Protozoa tro Mamalia
  - c) Microbes as models in research (E.Coli, Petite Yeast): Observation of Lactobacilli.
  - d) Plant pathogens: Rust disease, Smut, Mildews.
  - e) Angiosperm taxonomy: Concept of the "Art & Science" of observation, identification and classification of plants (Michelia, Rosa, Datura, Orchid).
- 3. Experiments in plant physiology:
  - a) Photosynthesis – pigments – spectroscopic analysis, Emerson effect.
  - b) Biological oxidation – respiratory quotient, fermentation.
  - c) Enzymes: catalase activity, ...

  - d) Preliminary analysis to identify natural products from plant source (raw drugs).
  - e) Extraction of DNA.
- 4. Experiments in animal physiology:
  - a) Carbohydrate metabolism – reducing sugars – test for sugars in urine, blood.
  - b) Proteins, lipids.
  - c) Sodium intake / loss using aquatic models, muscle twitch,
  - d) Virtual dissection – virtual tour of the human body.
- 5. Genetics:
  - a) Analysis of visual traits - tongue folding, rolling, dimple cheek, ...
  - b) PTC test, blood group – Rh Factor.
  - c) Brain teasers – genetic problems.
- 6. Reproductive biology:
  - a) Observation of incubated chick material for embryogeny.
  - b) Slides on Meiosis.
  - c) Observation of plant pollen, seed – embryo.
- 7. Ecology:
  - a) Field visit to study varied ecosystems – Hydrosere – Xerosere.
  - b) Ecological adaptations in plants – arid plants, salt tolerance, aquatic adaptations, Haustoria.
  - c) Quadrat study.
- 8. Applied ecology
  - a) Water depletion, water shed development, pollution, surface and ground water status.
  - b) Air quality, pollutants.
  - c) Solid waste management.
  - d) Urban Environmental Problems – social, cultural & economic. (Assignment suggested).
- 9. Computational biology:
  - a) Introduction to bioinformatics.
  - b) Assignments /field report writing & analysis /Herbaria /Plastnation technique.
  - c) Tour report /taxonomic indexing /use of Web media in systematics.
- 10. Round up:
  - a) Ideas to share & care for a "better world".
  - b) Group discussion on environmental issues.
  - c) Resolutions, feedback.

Physics Courses

Physics I: Classical Physics

Topic	#Hours
Kinematics (1-, 2-, and 3-dimensions; polar and spherical polar coordinates)	3
Dynamics (Newton's laws, many particles, rigid body, conservation laws including moment of inertia, angular momentum, conservation of angular momentum and precession)	8
Elasticity, stress strain curve, elastic and plastic regions and breakdown stress	1
Central force	2
Simple harmonic motion	1
	15
Electrostatics (Gauss' theorem, simple applications, conductors, electric dipole, capacitance, energy density, dielectrics)	4
Magnetostatics (Biot-Savarts' law, simple applications, magnetic moment, magnetic materials)	4
Faraday's law	1
Maxwell's equations, EM waves, EM waves in media, concept of polarization, coherence, lasers	5
	14
Thermodynamics, internal energy, first law of thermodynamics, reversible and irreversible operations, second law of thermodynamics, entropy	6
Heat transfer - conduction, convection and radiation	
Stefan's law	
Emissivity	
	6
Sound waves, water waves	2
Group velocity, standing waves	1
Interference	2
	5
Phase space and chaos	2
Total	42

Recommended books:

- ”An Introduction to Mechanics,” by Daniel Klepner and Robert Kolenkow, Tata McGraw Hill, 2007.
- ”Concepts in Physics,” by H.C. Verma, Vols I and II, Bharati Bhawan, 2011.
- ”Physics,” by Resnick, Halliday and Krane, Vols I and II, John Wiley India, 2009.
- ”The Feynman Lectures on Physics,” by Feyman, Vols I, II and III, Leighton and Sands, Narosa, 2008.
- ”Electricity and Magnetism,” by E.M. Purcell, Berkeley Physics Course, Vol 2, Tata McGraw Hill.

Physics II: Modern Physics

Topic	#Hours
Quantum Mechanics:	
Need for QM, photoelectric effect, D'Broglie relation	2
Schröndinger's equation, elementary solution (particle in 1-, 2-dimensional boxes)	4
Operators, expectation value, simple harmonic oscillator, angular momentum, hydrogen atom	10
	16
Postulates of statistical mechanics (concept of phase space), $s = k \ln W$	3
Classical, quantum statistics	1
Applications: (a) Maxwell-Boltzmann	2
Applications: (b) Bose-Einstein	2
Applications: (c) Fermi-Dirac	2
(Specific heat of dielectric solids, metals, Planck's radiation formula)	
	10
Crystal structure, origin of bands, electrical properties of solids	2
Semiconductors	3
Lasers	3
Special theory of relativity	3
Nuclear physics (properties of nuclei)	1
Nuclear physics (nuclear energy)	2
	14
Total	40

Recommended books:

- "A Textbook of Quantum Mechanics," by Mathews and Venkatesan, 2/e, Tata McGraw Hill.
- "Quantum Mechanics," by L. I. Shiff, 3/e, McGraw Hill Education, 2010.
- "Concepts af Modern Physics," by Beiser, Tata McGraw Hill.
- "Introduction to Modern Physics," by H. S. Mani and G. K. Mehta, Affiliated East-West Press, India.
- "Statistical Physics (Berkeley Physics Course, Vol.5)," by F. Reif, McGraw Hill, 1967.

(Remark: Hydrogen molecule (bonding) should be discussed in the Chemistry courses.)

Appendix C: Course Descriptions of IT Core Courses

Computer Architecture and Organisation

Objectives

The objectives of this course are to provide the student with an understanding of the internal organisation of a modern computer. The student will know of the principal blocks that constitute a processor, its organisation and instruction set.

Proposed Course Contents

- Von Neumann Machine, computer components, functions, bus inter connection, PCI, ALU, integer arithmetic, addition, subtraction, multiplication and division, floating point arithmetic.
- Machine instruction set, types of operands, types of operations, addressing modes, instruction formats, processor organization, register organization, instruction cycle, instruction pipelining, pentium processor, power-PC processor, RISC.
- 8085 Microprocessor organization, assembly language programming of 8085, processor control unit, operation, micro-operations, hardwired control, micro program control, horizontal and vertical micro instructions, micro instruction sequencing and execution, nanoprogramming, Applications of microprogramming.
- Internal memory, semiconductor main memory, cache memory, DRAM organization, associate memory organization, magnetic disk, CDROM, magnetic tape, memory management, memory hierarchy, partitioning, paging, virtual memory, demand paging scheme, segmentation.
- Input/Output, external devices, I/O modules, I/O addressing, programmed I/O, Interrupt driven I/O, priority, arbitration, DMA, I/O channel, I/O processor.

The recommended textbooks for the course are as follows:

- "Computer Organization and Architecture," by William Stalling, 4th Edition, PHI.
- "Computer Architecture and Organization," by Hayes, MH.
- "Introduction to Microprocessors," by Mathur.

Computer Networks

Objectives

This is a course on fundamentals of computer networking systems. The student will learn about how different kinds of networks (internet and cellular) are interconnected and the various types of applications that run over them from one part of the globe to the others efficiently. It will cover Internet and Telecom architecture and major building blocks. Hence the course deals with application, transport, network, data link layers protocols/algorithms.

Proposed Course Contents

- Logical and physical topologies; the need for various topologies.
- Client, Server, Connection oriented and connectionless services, layered architecture, Internet Protocol (IP) layer, Circuit-Packet-Message switching.
- Need for services by application layer protocols, HTTP, FTP, SMTP, SMTP *vs.* HTTP, MOME, DNS.
- Socket programming for TCP and UDP.

- Relationship of transport layer with application and network layers, multiplexing and demultiplexing, UDP.
- GBN, SR, TCP: connection, segment structure.
- TCP reliability, flow control, and congestion control algorithms.
- Link-state routing algorithm, distance-vector routing algorithm.
- Intra-autonomous system routing: RIP, OSPF, Inter-autonomous system routing: BGP.
- IPv4 and IPv6 packet format and basic differences and alignments.
- Mobility at network layer.
- Error detection and correction techniques; multiple access protocols in LAN: channel portioning, random access, taking turn.
- Address resolution protocol.
- Wired and Wireless LAN medium access Control Protocol Pure/Slotted ALOHA, CSMA, CSMA/CD: Ethernet, IEEE 802.11: CSMA/CA.
- Public Switched Telecom Network (PSTN) and PLMN network architecture; each major building blocks functionality.

Recommended Reading:

- "Computer Networking," by Kurose and Ross.
- "Local Area Network," by G. Keiser.
- "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," by G. Bianchi, IEEE Journal of Selected Areas in Communications, Vol. 18, No. 3, March 2000.

Data Structures and Algorithms

Objectives

The aim of the course is to provide students with a grasp of the principles of the many data structures used in modern software.

Proposed Course Contents

- Arrays, stacks, queues and linked lists .
- Dictionaries:
  - Skip-lists, hashing, analysis of collision resolution techniques.
- Trees:
  - Traversals, binary search trees, balanced binary search trees.
- Priority queues and binary heaps.
- Object oriented implementation and building libraries.
- Applications to discrete event simulation.
- Sorting:
  - Merge, quick, radix, selection and heap sort.
- Graphs:
  - Breadth-first search and connected components, Depth-first search in directed and undirected graphs.
- Union-find data structure and applications.
- Directed acyclic graphs and topological sort.

The recommended textbook for the course is "Introduction to Algorithms," by Thomas H. Cormen, Charles E. Leiserson and Ronald L. Rivest.

Database Systems

Objectives

The ACM document on Computer Science covers Database Systems area under the title "Information Management." Information Management (IM) plays a critical role in almost all areas where computers are used. This area includes the capture, digitization, representation, organization, transformation, and presentation of information; algorithms for efficient and effectiveaccess and updating of stored information, data modeling and abstraction, and physical file storage techniques. It also encompasses information security, privacy, integrity, and protection in a shared environment. The student needs to be able to develop conceptual and physical data models, determine what IM methods and techniques are appropriate for a given problem, and be able to select and implement an appropriate IM solution that reflects all suitable constraints, including scalability and usability.

Proposed Course Contents

- Introduction to database systems:
  - History and motivation for database systems; components of database systems; DBMS functions; database architecture and data independence.
- Database design:
  - Data modeling; conceptual models (such as entity-relationship or UML); object-oriented model;
  - Relational data model:
    - Functional dependency; decomposition of a schema; lossless-join and dependency-preservation properties of a decomposition; candidate keys, superkeys, and closure of a set of attributes; normal forms (1NF, 2NF, 3NF, BCNF); relational algebra;
  - Mapping conceptual schema to a relational schema.
- DBMS:
  - Components of a DBMS; indexing; transaction processing; concurrency control; query processing; failure and recovery.
- Database programming:
  - SQL, other procedural access methods (JDBC, stored procedures, frameworks, etc.).
- Other topics (at an introductory level):
  - Data warehousing, specialized databases.

This course is to be accompanied by a Database laboratory session.

Design and Analysis of Algorithms

Objectives

This course will provide a comprehensive introduction to the study of computer algorithms. The course will cover design of many algorithms in considerable depth. In addition to design, the course also will cover analysis of algorithms to understand their time and space complexity. The course will also involve project work to understand the implementation aspects of some of the algorithms

Proposed Course Contents

It is assumed that the required mathematical foundations will be covered by another course. The contents for this course are:

- Introduction to algorithms, their design and analysis.
- Divide and conquer techniques.
- Dynamic programming.
- Greedy algorithms.
- Amortized analysis.
- Number-theoretic algorithms:
  - Greatest common divisor, modular arithmetic, solving modular linear equations, the Chinese remainder theorem.
- String matching algorithms.
- Complexity:
  - Polynomial-time, NP-completeness, reductions.

The recommended textbook for the course is "Introduction to Algorithms," by Thomas H. Cormen, Charles E. Leiserson and Ronald L. Rivest.

Digital Communications

Objectives

Digital communication systems are basic workhorses behind the information age. Examples include the wireless systems such as cellular system (3G/4G) and WiFi , wire-line systems including DSL, cable modems. This course aims to provide an introduction to the basic principles behind the design of these broad-band systems and get an insight into the underlying principles behind the design and analysis of digital communication systems.

Proposed Course Contents

- Overview:
  - Digital vs. analog communication, block diagram of a digital communication system.
- Source coding and compression:
  - Huffman and Lempel-Ziv compression algorithms; source entropy; quantization; sampling.
- Communication over erasure channels:
  - Notion of reliable communication; feedback *vs.* forward error correction coding; linear codes and iterative decoding algorithms; example: fountain codes for reliable packet delivery over the Internet.
- Communication over noisy Gaussian channels:
  - Maximum likelihood detector.
- Communication over band-limited channels:
  - Pulse design for band-limited channels, power/bandwidth trade-off, inter-symbol interference, equalization: maximum likelihood sequence detection via the Viterbi algorithm; orthogonal frequency division multiplexing (OFDM).
- Communication over wireless channels:

  - Complex base-band representation of pass-band channels; modelling of multi-path wireless channels; key parameters: delay spread, coherence bandwidth, coherence time, Doppler spread; channel fading.
- Time, frequency and antenna diversity techniques:
  - Introduction to MIMO, flavours of OFDM.

Discrete Mathematics

Objectives

This course covers elementary discrete mathematics that is required for a computer science, engineering or information technology degree.

Proposed Course Contents

- Basic logic: Propositional logic: logical connectives; truth tables; normal forms (conjunctive and disjunctive); validity; predicate logic; limitations of predicate logic, universal and existential quantifcation; modus ponens and modus tollens.
- Proof techniques: Notions of implication, converse, inverse, contrapositive, negation, and contradiction; the structure of formal proofs; direct proofs; proof by counter example; proof by contraposition; proof by contradiction; mathematical induction; strong induction; recursive mathematical definitions.
- Set Theory: Definition of set, Relations, Equivalence relations and equivalence classes, Principle of mathematical induction, Posets, Chains and well-ordered sets. cardinality and countability.
- Combinatorics: Principles of addition and multiplication, Arrangements, Permutation and combinations, Multinomial theorem, Partitions and allocations, Pigeonhole principle, Inclusion-exclusion principle, Generating functions, Recurrence relations.

The following textbooks and study materials are recommended for this course:

- ”Elements of Discrete Mathematics,” by C. L. Liu, second edition 1985, McGraw-Hill Book Company. Reprinted 2000.
- ”Discrete Mathematics and Applications,” by K. H. Rosen, fifth edition 2003, TataMcGraw Hill Publishing Company.
- Proper web notes (NPTEL notes are available on discrete mathematics).

The following can be used as additional references:

- ”Discrete Math for Computer Science Students” by K. Bogart, S. Drysdale, C. Stein (freely available online).
- ”Discrete Mathematics,” by Laszlo Lovasz, Jozsef Pelikan, Katalin L. Vesztergombi, Springer 2003.

Fundamentals of IT Infrastructure, Security and Management

Objectives

IT professionals will encounter a variety of computing platforms in their careers. One of the roles of the IT professional is to select, deploy, integrate and administer platforms or components to support the organizations IT infrastructure. This knowledge area includes the fundamentals of hardware and software and how they integrate to form essential components of IT systems.

Since IT systems are increasingly under attack, knowledge of Information Assurance and Security (IAS) is of paramount importance to the profession of IT. The IT professional must understand, apply, and manage information assurance and security in computing, communication, and organizational systems. It is also important for the IT professional to provide users with a framework to be sufficiently security aware to be an asset to the organization rather than a liability. IAS includes operational issues, policies and procedures, attacks and defense mechanisms, risk analyses, recovery, and information security.

Proposed Course Contents

The course contents are drawn from broadly four knowledge areas included in ACM IT document:

- Platform Technologies (PT),
- Information Assurance and Security (IAS),
- Information Managment (IM), and
- System Administration (SA)

The course contents are as follows:

- PT. Computing infrastructures.
- PT. Enterprise deployment software.
- IAS. Fundamental aspects.
- IAS. Security mechanisms (countermeasures).
- IAS. Operational issues.
- IAS. Policy.
- IAS. Attacks.
- IAS. Vulnerabilities.
- IM. Managing the database environment.
- SA. Applications.
- SA. Administrative activities.
- SA. Administrative domains.

This course is to be accompanied by a Linux System Administration laboratory session.

Operating Systems

Objectives

A joint ACM and IEEE Computer Society curriculum recommendation for undergraduate degree in computer science from the report in 2008 proposes that a course in Operating Systems should explain the issues that influence the design of contemporary operating systems. Since operating systems is the software layer that abstracts the hardware to enable programmers to control it, the students will require a laboratory component to experiment with operating systems.

Proposed Course Contents

- Overview of operating systems:
  - Role and purpose of operating systems; evolution of operating systems; functionality and purpose of a typical operating system; client-server models; design issues - efficiency, robustness, flexibility, portability, security, compatibility; interactions with computer architecture.
- Operating systems principles:

  - Structuring methods - monolithic, layered, modular, micro-kernel, virtual machine, exokernel; abstractions, processes, resources; device organization; interrupts; user *vs.* kernel modes, transition of modes.
- Concurrency:
  - States and state diagrams; structures; dispatching and context switching; concurrent execution; mutual exclusion; deadlocks - causes, conditions, prevention; models and mechanisms - semaphores, monitors, condition variables, rendezvous; producer-consumer problems and synchronization; multiprocessor issues - spin-locks, reentrancy .
- Scheduling and dispatching:
  - Preemptive *vs.* non-preemptive scheduling; schedulers and policies; processes and threads; interprocess communication; classical IPC problems - Dining Philosophers, Readers and Writers.
- Memory management:
  - Physical memory and memory management hardware; paging, virtual memory, segmentation; working sets, thrashing; caching.
- Security and protection:
  - Overview - threats, intruders; cryptography; protection mechanisms; authentication; insider attacks; malware; defenses - protection, access control; backups.
- File Systems:
  - Files - data, metadata, organization, buffering; directories - contents, structure; file systems - partitioning, mount/unmount, virtual file system; implementation techniques; memory-mapped files; special purpose file systems; management and optimization - naming, searching, access, backups.

This course will involve lectures, and tutorials, and will be accompanied by an Operating Systems laboratory session.

Programming Languages

Objectives

This course will involve teaching of concepts involved in programming instead of programming languages. Programming languages will be used as examples to illustrate the concepts. The course will involve extensive programming to understand the concepts.

Proposed Course Contents

- Programming language foundations.
- Denotational semantics.
- Functional and imperative languages.
- Procedures, types, memory management.
- Modularity, abstraction and Object-Oriented Programming.

The following are the textbooks recommended for this course:

- “Concepts in Programming Languages,” by John Mitchell, Cambridge University Press, 2002.
- “Programming Languages,” by Ravi Sethi, second edition, Pearson Education, 2007.
- “Programming Language Pragmatics,” by Michael Scott, Morgan Kaufman, 2000.

Software Engineering

Objectives

This course is intended to provide foundational knowledge in the area of Software Engineering and help them to understand critical concepts encountered while dealing with complex software projects. The course will cover both process and technical aspects of software engineering and will form the basis for further specilized courses in this area.

Proposed Course Contents

- Managing software development:
  - Lifecycle models, quality control, configuration management, project management, personal and team software process, software metrics.
- Software development:
  - Requirements management, design and analysis techniques, architectures of software systems, software testing.

The following textbooks are recommended for this course:

- "Software Architecture in Practice," Second Edition, by Len Bass, Paul Clements, and Rick Kazman, Addison-Wesley 2003.
- "Models of Software Systems," by D. Garlan, J. Wing, O. Celiku, and D. Kroening. Draft book.
- "Essentials of Software Engineering," by Tsui and Karam. (T&K), 2007, Jones and Barrlet Publishing, ISBN-13: 978-07637-3537-1.
- "Software Engineering: A Practitioner's Approach," by Roger S. Pressman, sixth Edition, 2005, " McGraw Hill, ISBN 0-07-301933-X.
- Additional reading material specific to topics being handled.

Appendix D: Focus area-wise Classification of IT Elective Courses

The following focus area-wise lists of elective courses in the IT-part of the iMTech curriculum are recommended.

The lists of elective courses suggested under the focus areas are large. However the actual lists of courses made available to students will be much smaller. The lists given below are only indicative of what has been recommended as important and relevant areas and topics for each focus area. The actual lists will be lists of elective courses developed by the faculty for the focus areas from time to time and made available to students.

The levels of the elective courses, namely, Levels 1, 2 or 3, have been recommended in the following list.

This list excludes the supervised reading, and project electives which are possible elective courses for each stream.

- COMPUTER SCIENCE
  - Level 1 Courses:
    - Artificial Intelligence
    - Compiler Design
    - Computer Graphics
    - Distributed and Parallel Computing
    - Graph Theory and Applications
    - Mathematical Programming/optimization
    - Numerical and Scientific Computation
    - Scientific Visualization
    - Theory of Computation
  - Level 2 Courses:
    - Advanced Algorithms
    - Advanced Data Structures
    - Advanced Operating Systems
    - Algorithms and Complexity
    - Applied Cryptography
    - Approximation Algorithms
    - Complexity Theory
    - Computational Geometry
    - Foundations of Automatic Verification
    - Logic in Computer Science
    - Machine Learning
    - Network Science
    - Program Analysis
    - Randomized Algorithms
    - Real-time Systems
    - Semantics of Programming Languages
- DATABASE AND INFORMATION SYSTEMS
  - Level 1 Courses:
    - Data Modeling
    - Geographic Information Systems
  - Level 2 Courses:
    - Advanced Database Management Systems

    * Conceptual Modelling
    * Data Analytics
    * Information Networks
    * Web Information Retrieval
    * Semantic Web
    * Multi-Agent Systems

- EMBEDDED SYSTEM DESIGN
  - Level 1 Courses:
    * Analog VLSI Design
    * Digital System Design Using FPGAs
    * Microwaves
    * Optical Communication
    * Software Aspects of Embedded Systems
  - Level 2 Courses:
    * Advanced Algorithms and Optimization Methods
    * Advanced Computer Architecture
    * Advanced Control Systems
    * Advanced Electronic Devices
    * Analog/Mixed Signal VLSI Design II
    * CAD Tools I
    * CAD Tools II
    * Design for Testability
    * Digital VLSI Design I
    * Digital VLSI Design II
    * Embedded System Design I
    * Embedded System Design II
    * Fault-tolerant Computer System Design
    * Interconnection Networks
    * Hardware-software Co-design
    * Mixed Signal VLSI Design I
    * Real-time Systems
    * Reconfigurable Computing
    * Ubiquitous Computing

- INFORMATION TECHNOLOGY AND SOCIETY
  - Level 2 Courses:
    * Dynamics of the Information Technology Industry
    * Economic and Social Impacts of Information Technology
    * Intellectual Property Rights

- NETWORKING AND COMMUNICATION SYSTEMS
  - Level 1 Courses:
    * Discrete Event Systems Modeling and Simulation
    * Informatics in Industrial Process Automation
  - Level 2 Courses:
    * Advanced Digital Communication
    * Communication in Industrial Process Automation
    * Information Theory and Coding
    * Mobile Computing
    * Network Security

    * Wireless Access Networks
    * Wireless Communications
    * Wireless Sensor Networks

- SIGNAL PROCESSING
  - Level 1 Courses:
    * Audio Coding
    * Biometrics
    * Digital Signal Compression
    * Digital Image Processing
    * Multimedia Systems
    * Speech Processing
  - Level 2 Courses:
    * Advanced Digital Signal Processing
    * Computer Vision and Pattern Recognition
    * Digital Photography
    * Medical Image Processing
    * Medical Imaging
    * Medical Signal Processing
    * Statistical Signal Processing
    * Time-frequency Analysis
    * Video Analytics
    * Video Processing

- SOFTWARE ENGINEERING
  - Level 1 Courses:
    * OOAD, UML
    * Usability Engineering
  - Level 2 Courses:
    * Design Patterns and Software Architecture
    * Hardware and Software Reliability
    * Formal Specification
    * Fundamentals of Performance and Reliable IT Infrastructures
    * Performance, Reliability Analysis and Optimization of Contemporary Infrastructures
    * Software Testing

APPENDIX E: SEMESTER PLAN

INTEGRATED M.TECH. SEMESTER PLAN

Following is semester-wise plan for the iMTech program.

Course Code	Course Name	#Hours	#Credits	Course Category
	SEMESTER I – Credits: 16; Lecture Courses: 5; Practical Courses: 3			
BS002-L	Chemistry / Biology	3	3	Basic Sciences
BS002-P	Chemistry / Biology Lab	2	1	Basic Sciences
BS004-L	Mathematics I	4	4	Basic Sciences
ES002-L	Programming I	3	3	Basic Engineering Sciences/Skills
ES002-P	Programming I Lab	2	1	Basic Engineering Sciences/Skills
OT001-P	Physical Education I	2	0	Miscellaneous
OT003-L	English	2	2	Miscellaneous
OT004-L	Introduction to Profession	2	2	Miscellaneous
	SEMESTER II – Credits: 22; Lecture Courses: 5; Practical Courses: 3			
BS005-L	Mathematics II	4	4	Basic Sciences
ES001-L	Basic Electronics, Digital Logic and Circuits	4	4	Basic Engineering Sciences/Skills
ES001-P	Basic Electronics Lab	2	1	Basic Engineering Sciences/Skills
CC004-L	Data Structures and Algorithms	4	4	IT Core
CC004-P	Data Structures Lab	2	1	IT Core
ES004-L	Signals and Systems	4	4	Basic Engineering Sciences/Skills
HSS001-L	HSS/M I	4	4	HSS/M
OT002-P	Physical Education II	2	0	Miscellaneous
	SEMESTER III – Credits: 21; Lecture Courses: 5; Practical Courses: 2			
BS006-L	Mathematics III	4	4	Basic Sciences
CC002-L	Computer Architecture and Organization	4	4	IT Core
CC002-P	Computer Architecture Lab	2	1	IT Core
CC012-L	Discrete Mathematics	4	4	IT Core
BS007-L	Physics I	3	3	Basic Sciences
BS007-P	Physics I Lab	2	1	Basic Sciences
HSS002-L	HSS/M II	4	4	HSS/M
	SEMESTER IV – Credits: 20; Lecture Courses: 5; Practical Courses: 2			
BS009-L	Mathematics IV	4	4	Basic Sciences
CC002-L	Design and Analysis of Algorithms	4	4	IT Core
BS008-L	Physics II	3	3	Basic Sciences
BS008-P	Physics II Lab	2	1	Basic Sciences
ES003-L	Programming II	3	3	Basic Engineering Sciences/Skills
ES003-P	Programming II Lab	2	1	Basic Engineering Sciences/Skills
HSS003-L	HSS/M III	4	4	HSS/M
	SEMESTER V – Credits: 20; Lecture Courses: 4; Practical Courses: 4			
CC003-L	Computer Networks	3	3	IT Core
CC003-P	Computer Networks Lab	2	1	IT Core
CC006-L	Digital Communications	3	3	IT Core
CC006-P	Digital Communications Lab	2	1	IT Core
CC009-L	Operating Systems	3	3	IT Core
CC009-P	Operating Systems Lab	2	1	IT Core
ES005-L	Signal Processing	3	3	Basic Engineering Sciences/Skills
CC005-L	Database Systems	4	4	IT Core
CC005-P	Database Lab	2	1	IT Core

Course Code	Course Name	#Hours	#Credits	Course Category
	SEMESTER VI – Credits: 21; Lecture Courses: 6; Practical Courses: 2			
CC008-L	Fundamentals of IT Infrastructure, Security and Management	3	3	IT Core
CC008-P	Fundamentals of IT Infrastructure Lab	2	1	IT Core
CC010-L	Programming Languages	3	3	IT Core
CC011-L	Software Engineering	3	3	IT Core
CC011-P	Software Engineering Lab	2	1	IT Core
OT005-L	Technical Communication	2	2	Miscellaneous
TBD-L	IT Elective I	4	4	IT Elective
TBD-L	IT Elective II	4	4	IT Elective
	SUMMER TERM – Credits: 4; Lecture Courses: 0; Practical Courses: 0			
INT001-P	Summer Internship	0	4	Internship
	SEMESTER VII – Credits: 20; Lecture Courses: 5; Practical Courses: 0			
HSS004-L	HSS/M IV	4	4	HSS/M
ITD001-L	IT in Domains I	4	4	IT in Domains
TBD-L	IT Elective III	4	4	IT Elective
TBD-L	IT Elective IV	4	4	IT Elective
TBD-L	IT Elective V	4	4	IT Elective
	SEMESTER VIII – Credits: 20; Lecture Courses: 5; Practical Courses: 0			
ITD002-L	IT in Domains II	4	4	IT in Domains
TBD-L	IT Elective VI	4	4	IT Elective
TBD-L	IT Elective VII	4	4	IT Elective
TBD-L	IT Elective VIII	4	4	IT Elective
TBD-L	IT Elective IX	4	4	IT Elective
	SEMESTER IX – Credits: 20; Lecture Courses: 2; Practical Courses: 0			
IMT299	Research/Thesis		12	Research/Thesis
TBD-L	IT Elective X	4	4	IT Elective
TBD-L	IT Elective XI	4	4	IT Elective
	SEMESTER X – Credits: 20; Lecture Courses: 0; Practical Courses: 0			
IMT299	Research/Thesis		20	Research/Thesis