SYLLABUS

(5-Year Integrated Master of Computer Application)

Department of Computer Science and Technology

School of Media, Technology and Information Science

(As passed in the Board of Studies and Academic Council)

This page is left blank intentionally

Table of Contents

Abbreviations

AEC	Ability Enhancement Course
CO	Course Outcomes
DSE	Department Specific Elective Course
MCA	Master of Computer Application
MDC	Multidisciplinary Course
MIL	Modern Indian Language
NEP	National Education Policy
PC	Programme Outcomes
PO	Programme Objectives
SC	Scheduled Caste
SEC	Skill Enhancement Course
ST	Scheduled Tribe (Hills & Plains)
UG	Under-Graduate
VAC	Value Added Course

About the Department:

Integrated The Department of Computer Science and Technology was established in 2014 after the Department of Information and Computer Science was retitled. The Department of Computer Science & Technology is a well-equipped unit of Assam Women's University to deliver technical skills to the students making them fit for both corporate world and research. The department also emphasizes on building communication skills by organizing presentations periodically. Local industry is also consulted for quality content in the curricula of offered programmes. The laboratory of the department has sufficient resources for teaching software and basic electronics skills.

About the Programme:

Integrated Master of Computer Application (MCA) is a 5-Year programme introduced from academic session 2024-25 under National Education Policy (NEP) in the Department of Computer Science and Technology at Assam Women's University. This programme will have multiple entry and exit points.

1. **Exit at first year: UG Certificate in Computer Application**

   After 1-year programme of study if a student wants to exit, then she will get a UG certificate in Computer Application provided she has earned minimum of 44 credits and also complete a mandatory 4 credit work based vocational course or internship/apprenticeship offered during summer term.

2. **Exit at second year: UG Diploma in Computer Application**

   After 2-year programme of study if a student wants to exit, then she will get a UG Diploma in Computer Application provided she has earned a minimum of 88 credits and also complete a mandatory 4 credit work based vocational course or internship/apprenticeship offered during 1st year or second year summer term.

3. **Exit at third year: Bachelor in Computer Application**

   After 3-year programme of study, student will get Bachelor degree in Computer Application provided she has secured minimum 120 credits.

4. **Exit at fourth year:**

   a. **Bachelor in Computer Application (Honours with Research)**

      After completion of 4-year programme of study, student will be awarded Bachelor degree in Computer Application (Honours with Research) if the student completes a rigorous research project of 12 credits in their major area(s) of study under the guidance of a faculty member. Student has to complete the research project in the 8th semester.

   b. **Bachelor of Computer Application (Honours)**

      After completion of 4-year programme of study, student will be awarded Bachelor degree in Computer Application (Honours) on securing minimum 160 credits by taking 12 credits advanced level course in lieu of Research Project.

5. **Exit at fifth year: Master of Computer Application**
   After completion of 5-year programme of study, student will be awarded Integrated Master of Computer application on securing minimum requisite credits.

Eligibility Criteria:

The candidates must have passed 10+2 or 12th or equivalent examination with a minimum aggregate mark of 50% (45% for SC/ST candidates) from any recognized board.

Programme Objectives:

The objectives of this programme are as follows:

PO1: To produce computer professionals with good ethical values and knowledge of computer applications, who can successfully design feasible and technically sound innovative solutions to real life problems.

PO2: To equip students with all the skills required to develop application software and information system in diverse area where computers are used.

PO3: To expose the students with latest software tools and new technologies.

PO4: The programme delves to inculcate an inquisitive mind to promote research-based thinking and encourages students to write research articles to step into the domain of Research and Development. It prepares graduates who will contribute to societal growth through research in their chosen field.

PO5: To prepare graduates who will perform both as an individual and in a team through good analytical, designing, implementation skills and with professional ethics.

Programme Outcomes:

On completion of integrated MCA programme, the graduates will be able to:

PC1: Apply the knowledge of computing and mathematics relevant to Information Technology to various real-life applications for any given requirement.

PC2: Identify, analyze, formulate and solve complex Computing problems reaching substantiated conclusions using fundamental principles of Mathematics, Computing sciences, and relevant domain disciplines.

PC3: Design and develop application software for any desired needs with appropriate considerations for any specific requirement on societal and environmental aspects.

PC4: Understanding and application of modern tool and technique: Create, select, adapt and apply appropriate techniques, resources and modern computing tools to computing activities with an understanding of the limitations.

PC5: Create systems through software development to solve problems in Industry domain areas.

PC6: Understand and commit to professional ethics, cyber regulations, responsibilities and norms of professional computing practice.

PC7: Involve in perennial learning for a continued career development and progress as a computer professional.

PC8: Create a culture that focus on innovation and Entrepreneurship.

PC9: Function effectively as an individual and as a member or leader in diverse teams and multidisciplinary environments, communicate effectively and present technical information in oral and written reports.

Courses Offered:

The courses offered in this programme are classified into the following:

1. **Major Course:** Major courses are discipline specific courses.

2. **Minor Course:** Minor courses are supporting discipline specific courses. Students can choose minor courses from other disciplines also.

3. **Multidisciplinary course (MDC):** Multidisciplinary courses have to be chosen from other disciplines.

4. **Ability Enhancement Course (AEC):** These courses are about Modern Indian Language (MIL) or regional language & English language focused on language and communication skill.

5. **Skill Enhancement Course (SEC):** Theses courses focus on hands-on training, soft skill etc. Student has to choose from a basket of Skill Enhancement Courses offered by various departments of the university.

6. **Value added course (VAC):** These courses are about understanding India/ Env. Sc./ Env. Ed/ Digital and Tech solutions/ Health and Wellness, Yoga Education, Sports and fitness, Community engagement etc. Student has to choose from a basket of value-added course offered by various departments of the university.

7. **Summer Internship:** Internship can be carried out during summer term. A summer term is for eight weeks / two months from 01st June to 31st July.

8. **Research Project:** If a student wants to be awarded with BCA (honours with research), then she has to complete a rigorous research project of 12 credits in their major area(s) of study under the guidance of a faculty member. Student has to complete the research project in the 8th semester.

9. **Pre-requisite Course:** Theses courses is required to undertake as an introductory course which will be pass or fail course.

Course Distribution (with credits):

Year	Semester	Course Title	Type of Course	Course Code	Credit per course	Total Credit (Minimum)
1st	I	Computer Fundamentals	Major	IPCTMJ1401	4	22
		Mathematics-I	Prerequisite	IPCTPQ1201	2	
		-	Minor	-	4	
		-	MDC	-	4	
		-	AEC	-	3	
		-	SEC	-	3	
		-	VAC	-	3	
	II	Introductory Computing using C	Major	IPCTMJ2401	4	22
		Mathematics-II	Prerequisite	IPCTPQ2201	2	
		-	Minor	-	4	
		-	MDC	-	4	
		-	AEC	-	3	
		-	SEC	-	3	
		-	VAC	-	3	
2nd	III	Digital Logic and Design	Major	IPCTMJ3401	4	22
		Data Structure	Major	IPCTMJ3402	4	
		-	Minor	-	4	
		Mathematics-III	Prerequisite	IPCTPQ3201	2	
		-	MDC	-	4	
		-	AEC	-	3	
		-	SEC	-	3	
		-	VAC	-	3	
	IV	Database Management System	Major	IPCTMJ4401	4	20
		Computer Organization and Architecture	Major	IPCTMJ4402	4	
		Formal Language and Automata	Major	IPCTMJ4403	4	
		Data Communication	Major	IPCTMJ4404	4	
		-	Minor	-	4	
3rd	V	Operating System	Major	IPCTMJ5401	4	22
		Computer Network	Major	IPCTMJ5402	4	
		Software Engineering	Major	IPCTMJ5403	4	
		Object Oriented Programming using Java	Major	IPCTMJ5404	4	
		-	Internship	IPCTIN5201	2	
		-	Minor	-	4	
	VI	Computer Graphics	Major	IPCTMJ6401	4	24
		Web Technology	Major	IPCTMJ6402	4	
		Python Programming	Major	IPCTMJ6403	4	
		Data Mining	Major	IPCTMJ6404	4	
		-	Minor	-	4	
		-	Project	IPCTPR6401	4	

Back to Contents

Year	Semester	Course Title	Type of Course	Course Code	Credit per course	Total Credit (Minimum)
4th	VII	Big Data Concepts	Major	IPCTMJ7401	4	20
		Computer Security and Cryptography	Major	IPCTMJ7402	4	
		Artificial Intelligence	Major	IPCTMJ7403	4	
		Research Methodology	Major	IPCTMJ7404	4	
		Numerical Method	Prerequisite	IPCTPQ7201	2	
		-	Minor	-	4	
		-	Project (internship)	IPCTPR7401	4	
	VIII	Machine learning	Major	IPCTMJ8401	4	22
		Dissertation/Research Project	Major	IPCTPR8801	8	
		Image Processing	Major (in lieu of research project)	IPCTMJ8402	4	
		Block Chain Architecture		IPCTMJ8403	4	
		Embedded System		IPCTMJ8404	4	
		-	Minor	-	4	
		Academic Writing		IPCTAW8201	2	
5th	IX		Major		4	
			Major		4	
			Major		4	
			DSE		4	
			DSE		4	
			Internship		2	

MAJOR COURSES

Computer Fundamentals

(IPCTMJ1401)

1. About the Course: This is a major course and is aimed at presenting foundation concepts on Computer system, its peripherals and various components like Registers, Arithmetic & Logic Unit, Control Unit and Memory etc. The course is organized as a series of lectures, hands-on exercises using Laboratory sessions on various free and open-source software.

2. Course Description:

- Target Audience: First semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1+ 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
  - ➢ Lectures: 30 Hours (15 Weeks X 2 Hours)
  - ➢ Tutorial: 15 hours (15 Weeks x 1 Hour)
  - ➢ Practical: 30 hours (15 weeks x 2 hours)

3. Prerequisites and Dependencies: Since this course is a first semester course, there is no particular prerequisite. However, the students are expected to have elementary knowledge of basic mathematics and logic.

4. Objective: The major objective of this course is to provide students with understandings of how a computer works, what are its various components, different types of peripherals used, concept of hardware and software, types of memories in a computer. Basic usage of some free and open-source software is also a major part of the course.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1 Understanding foundation concepts of information and information processing in computer systems.

CO2 Understanding of the basic components of a computer: ALU, CU, Memory, etc.

CO3 Understanding various computer terminologies.

CO4 Differentiation among Hardware, Firmware and Software.

6. Topics:

Module I: Introduction (9 hours)

What is computer, Computer characteristics, Computer evolutions and generations, Types of computers, Basic components of computer --Control unit, ALU, Input/output device and memory.

Module II: CPU and Memory Unit (12 hours)

CPU architecture-components of CPU, instruction set, register set, types of processor, Memory architecture-storage criteria, primary vs secondary storage, main memory, Cache memory, Secondary storage, Magnetic tape, Magnetic disks , Optical disks, Flash drive.

Module III: Input and Output Devices (6 hours)

Input devices-Keyboard, Scanner, Digitizer, Touch screen etc , Output devices-Monitors, Printers, Plotters.

Module IV: Software and Operating System (9 hours)

Software-concepts and needs, Types of software-system and application software, Algorithm, Flowchart, Pseudo code, Programming Language-Machine language, Assembly language and High-level language, Assembler, Compiler and Interpreter.

Module V: Computer Network (9 hours)

Data communication, Components of Data Communication System, Network topology, LAN, WAN, Internet, World Wide Web, Introduction to network security

7. Laboratory Sessions: The practical component of this course is of one credit which amounts to 2 hours of Laboratory class per week.

Section 1: Office Package (LibreOffice)

- Introduction to Windows 10 OS Environment, Command Prompt, Navigation using GUI based File Explorer (This PC), Concept of Desktop, Icons, Folders and Files, Introduction to Command Prompt. Hands-on various CMD Prompt commands, Environment Variables in Windows.
- Introduction to Linux (Lubuntu) Environment, Concept of Terminal and Terminal commands.
- Introduction to Spreadsheet software: LibreOffice Calc – Layout, Formula Bar, Cell Address, Font design and formatting, Arithmetic Operators (+, –, * and /), Aggregate functions (Avg, Sum, Max, Min and Truncate, etc.), Relational Operators (>, <, >=, <=, =, !=) Introduction to If .. else statement. Nested If .. else statement, Logical Operators (&, | and !).
- Count If, Introduction to 2D and 3D Graphs – Bar, Pie, Line, Vector, XY Labelling etc.
- Absolute and Reference Cell Addressing, Pivot Table.
- Introduction to Documentation software: LibreOffice Writer – Page Layout, Page Orientation, Page Columns, Font and Paragraph design, Introduction to Numbered and Bullet List and Sub-list, Table design.
- Insertion of Images and shapes, Formatting an Image, Page wrap, Alignment, Insertion of text box. Header and Footer, Page Number, Page Break, Template Design.
- Cover Page design and concept of Mail Merge (using LibreOffice Calc).
- Introduction to Presentation software: LibreOffice Impress – Slide Layout, Master Slide Design, Font and Paragraph Design, Inserting Image and graphs.
- Slide and Custom element Animation. Properties of animation (speed, Event of occurrence, Duration, etc.)

Section 2: Multimedia (GIMP, Audacity & OpenShot)

Back to Contents

- Introduction to Image editing software: GNU Image Manipulation Program (GIMP). Environment of GIMP. Concept of Layering, Introduction to various Tools.
- Class exercise on GIMP.
- Introduction to Audio editing software: Audacity, Concept of Monophonic and Stereophonic sound, Concept of channels (L-R), Fading, Combining different Audio tracks.
- Class exercise on Audacity.
- Introduction to Video editing software: OpenShot, Intro to its environment, Concept of tracks, Trimming and Fade effects (Audio and Visual).

8. <u>Referential Sources</u>:

Useful Books and Papers

1. Rajaraman V. “Fundamentals of Computer”. PHI Publishing. Sinha P.K. “Foundation of Computing”. PHI Publishing.
2. Byron S Gottfried, “Programming With C”. McGraw Hill
3. Brian W. Kernighan, Dennis Ritchie, “The C Programming Language”. Pearson Education India

----

Back to Contents

Introductory Computing using C (IPCTMJ2401)

1. About the Course: This is a Major Course containing advanced concepts of programming and software code writing within the framework of structural and procedural programming paradigms. The course is organized as a series of lectures and hands-on laboratory sessions using C programming languages and focusing on discussing how to write a program of moderate complexity by using C language.

2. Course Description:

- Target Audience: Second semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1+ 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
  - ➢ Lectures: 30 Hours (15 Weeks X 2 Hours)
  - ➢ Tutorial: 15 hours (15 Weeks x 1 Hour)
  - ➢ Practical: 30 hours (15 weeks x 2 hours)

3. Prerequisites and Dependencies: There is no prerequisite to this course

4. Objective: The course is oriented to those who want to advance their structured and procedural programming understating and to improve C programming skills. The major objective is to provide students with understanding of code organization and functional hierarchical decomposition with using complex data types.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Understanding a functional hierarchical code organization.
CO2	Ability to work with textual information, characters and strings.
CO3	Ability to work with arrays of complex objects.
CO4	Understanding a concept of object thinking within the framework of functional model.
CO5	Ability to handle possible errors during program execution.

6. Topics:

Module I: Introduction to computer and programming (6 hours)

Computer, block diagram of computer, hardware, software, program, types of software, operating system, compiler, programming languages, types of programming languages, algorithm, pseudocode, flowchart, desirable characteristics of a program, C programming language, history of C, structure of a C program, C character set, identifiers and keywords, writing, compiling and executing a c program, datatypes, constants, escape sequences, string constants, variables and arrays, declarations, expressions, statements, symbolic constants

Module II: Operators and I/O (5 hours)

Arithmetic operators, unary operators, relational and logical operators, assignment operators, conditional operator, data input output, single character input, single character output, data input from user: scanf function, writing output data: printf function, display formatting using printf function, the gets and puts function, error and debugging techniques

Module III: Control statements (6 hours)

Control statements, branching statement, looping statements: for, while, do-while, switch statement, break statement, continue statement, goto statement

Module IV: Functions and arrays (6 hours)

C functions, defining a function, calling a function, function prototypes. Passing arguments to function, recursion, defining an array, processing an array, passing array to a function, multidimensional array

Module V: Storage classes, strings and pointers (6 hours)
Storage classes, automatic variables, register variables, external variables, static variables, defining and initializing a string, null character, reading and writing a string, library functions for strings, pointers, pointer declaration, passing pointer to function, pointers and one dimensional array, dynamic memory allocation, operations on pointers, pointers and multidimensional array, arrays of pointers, passing functions to other functions

Module VI: Structures and unions (5 hours)

Structure, defining a structure, processing structure, user defined data types, structure and pointers, passing structures to functions, self-referential structures, union, defining a union, processing union

Module VII: Handling file (5 hours)

Files, opening and closing files, reading and writing a file, processing a file, unformatted files, binary files, random access of files

Module VIII: Bitwise operators and bitfield (6 hours)

Bitwise operators, one's complement operator, logical bitwise operators, masking, shift operators, bitwise assignment operators, bitfields, defining bitfields, processing bitfields

7. Laboratory Sessions: The practical component of this course is of one credit which amounts to 2 hours of Laboratory class per week.

- Introduction to the concept of Keywords, Datatypes, Operators, Variables, Constants in C. Branching statements
- Control statements using loops and goto command
- Introduction to Functions, Arrays, Pointers in C, Strings
- Structures and Unions
- File Handling

8. Referential Sources:

1. Byron Gottfried, "Programming with C". McGraw Hill Education
2. S.K. Srivastava, "C in Depth". BPB Publications

Back to Contents

Digital Logic and Design (IPCTMJ3401)

1. About the Course: This is a Major Course. Digital logic is the basis of electronic systems. This course mainly addresses the concepts behind digital logic, its principles and technique involved in designing digital circuit. The course is organized as a series of lectures with both theory and laboratory sessions.

2. Course Description:

- Target Audience: Third semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1+ 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
    - ➢ Lectures: 30 Hours (15 Weeks X 2 Hours)
    - ➢ Tutorial: 15 hours (15 Weeks x 1 Hour)
    - ➢ Practical: 30 hours (15 weeks x 2 hours)

3. Prerequisites and Dependencies: Students are expected to have good logical and reasoning skills.

4. Objective: The course mainly deals with the concepts and principles involved in designing digital circuits. The course addresses basic concepts of a digital system including topics from number system to sequential circuits. This course will help a student in understanding how different circuits such as Flip-flop, RAM and Multiplexer of a digital system work and how to design these circuits

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	To work with binary, octal and hexadecimal number system.
CO2	To understand the concepts behind digital system and why those systems are called so.
CO3	To understand the concept of Boolean Algebra and its application in designing digital circuits.
CO4	To design different combinational circuits using minimum number of digital gates.
CO5	To understand the concepts behind operations of flip-flops and to design sequential circuit.

6. Topics:

Module I: Introduction (6 hours)

Introduction to digital systems, Number systems like Decimal numbers, Binary numbers, Octal and Hexadecimal numbers, Number base conversions, Complements: r's complement, (r-1)'s complement, Arithmetic operations on binary numbers, Subtraction with r's and (r-1)'s complement, Binary fixed point representation, Representation of positive and negative number, overflow.

Module II: Encoding (2 hours)

Information representation by code, Binary codes, BCD, Excess: 3, Alphanumeric codes: ASCII, Unicode.

Module III: Boolean algebra and logic gates (9 hours)

Definition of Boolean algebra: basic and axiomatic definition, Theories and properties of Boolean algebra, Boolean function:it's complement, canonical and standard form, Minterms and Maxterms, SOP, POS, Digital logic gates: AND, OR, NOT, NAND, NOR, Exclusive OR, Exclusive NOR, Truth table, Implementation of Boolean function using gates, Universal gates: NAND and NOR implementation, Simplification of Boolean function: the Map Method, Don't care condition, the tabulation method

Module IV: Combinational Circuits (14 hours)

Combinational logic design procedure, Adder:half adder and full adder, Subtractor, Code conversion, Parity generator and parity checker, Binary parallel adder, Decimal adder, BCD adder, Magnitude comparator, Decoder, Demultiplexer, Multiplexer, Boolean function implementation, , Programmable Logic Array(PLA), Read Only Memory

Module V: Sequential Circuits (14 hours)

Sequential logic, flip:flop:RS flip:flop, JK flip:flop, D flip:flop, T flip:flop, Triggering of flip:flop, State table, State diagram, State equation, Flip:flop excitation table, Design procedure of sequential circuit, Design of counter,Synchronous and asynchronous counter, Ripple counter, BCD counter, Binary counter, Timing sequence, Johnson counter, Register, Shift register

7. Laboratory Sessions: The practical component of this course is of one credit of 2 hours class per week. The laboratory work consists of designing digital circuits using logic simulator tool and digital trainer kit. Following are the components that will be covered in practical sessions.

- Introduction to Integrated Circuit and IC digital logic families.
- Study the operation of different logic gates and ICs available for gates.
- Designing of different combinational circuits (half adder, full adder, parallel adder, magnitude comparator, decoder, encoder, MUX, de-MUX, parity generator etc.).
- Construction of flip-flops. Design sequential circuit.
- Implementation of counters (asynchronous and synchronous)

8. Referential Sources:

1. M. Morris Mano: Digital Logic and Computer Design, PHI (EEE)
2. M. Morris Mano: Computer System Architecture, PHI (EEE)

Back to Contents

Data Structures
(IPCTMJ3402)

1. About the Course: This is a Major Course and is aimed at teaching efficient storage mechanism of data for an organized and easy access.

2. Course Description:

- Target Audience: Third semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1+ 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
    - ➢ Lectures: 30 Hours (15 Weeks X 2 Hours)
    - ➢ Tutorial: 15 hours (15 Weeks x 1 Hour)
    - ➢ Practical: 30 hours (15 weeks x 2 hours)

3. Prerequisites and Dependencies: Students must have a basic understanding of the C programming language, especially the concept of arrays, functions and pointers in C.

4. Objective: The major objective of this course is to provide students with the understanding of data organization and efficient formation of complex data structures on a computer.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Understanding efficient storage mechanisms of data for easy access.
CO2	Designing and implementing various basic and advanced data structures.
CO3	Making decisions for representation of the data in the real world.

6. Topics:

Module I: Basics and Linear Data Structure (6 hours)

Algorithm, Time and space analysis of algorithms-average, best and worst case analysis, Data type-Abstract Data Type, Linear and Non linear data structure, Array, Representation of array, Stack-definition and concept, primitive operations on stack, stack as an abstract data type, implementation of push and pop operations, infix expression, prefix expression and postfix expressions, evaluating a postfix expression, converting an infix expression to postfix form, queue, primitive operations on queue, queue as ADT, implementation of queue operations, priority queue, Linked list- definition and concept, primitive operations on list, list implementation of stack, list implementation of queue, list as a data structure, list implementation of priority queue, array implementation of lists, circular list, stack as circular list, queue as circular list, Josephus problem, doubly linked list. Module

Module II: Non-linear Data structure (10 hours)

Tree- definition and concept, Binary tree, operations on binary tree, application of binary tree, node representation of binary tree, internal and external nodes, array representation of binary tree, binary tree traversals, threaded binary tree, the Huffman algorithm, representing lists as binary tree, trees, tree traversals, general expressions as trees, evaluating an expression tree, constructing a tree.

Module III: Sorting and searching (15 hours)
Sorting, bubble sort, quick sort, selection sort, binary tree sort, heap sort, insertion sort, shell sort, merge sort, radix sort, basic search techniques, dictionary as an abstract data type, sequential search, indexed sequential search, binary search, interpolation search, binary search tree operations, efficiency of BST operations, multiway search tree, operations on MST, B tree, operations on B tree, B+ tree, digital search tree, hashing, Collision-resolution technique

7. Laboratory Sessions: The practical component of this course is of one credit which amounts to 2 hours of Laboratory class per week.

- Link List, Operations on a Linked List (Singly).
- Stack, Queue, Tree
- Binary search tree, Multiway Search Tree.
- Linear and Binary search using Array. Sorting – Bubble sort.
- Quick sort, selection sort
- Insertion sort, Merge sort, Heap sort

8. Referential Sources:

Useful Books-

1. Tanenbaum A. “Data Structures using C and C++”. 3rd ed., Que Publishing, 2013.
2. Lipschutz S. “Data Structures with C”. 4th ed., PHI Publishing, 2015.

Useful Web Sources-

1. https://nptel.ac.in/courses/106102064/ : NP-TEL content for Data Structure.

Back to Contents

Database Management System
(IPCTMJ4401)

1. About the Course: This is a Major Course that deals with designing, maintenance and transaction of database systems. This course is organized as a series of lectures with both theory and laboratory sessions. This course covers basic database concepts, data models, database architecture, relational database languages, SQL, functional dependencies and normalization, and database transactions.

2. Course Description:

- Target Audience: Fourth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1+ 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
  - ➢ Lectures: 30 Hours (15 Weeks X 2 Hours)
  - ➢ Tutorial: 15 hours (15 Weeks x 1 Hour)
  - ➢ Practical: 30 hours (15 weeks x 2 hours)

3. Prerequisites and Dependencies: This is a Major course of Integrated MCA programme offered in Fourth semester. Students must have fundamental knowledge of computer including software, basic programming and discrete mathematics.

4. Objective: The main objective of the course is to equip students with the skills of database design. The principles and techniques involved in designing a productive and good database from conceptual level to implementation level are covered in this course. The course also addresses issues of database transaction and error recovery.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Will have a broad understanding of database concept and DBMS software.
CO2	Will be able to design a database for an application software, at conceptual level using ER modelling tool and to convert that database into implementation level using Relational model.
CO3	Will be able to model a good normalized database to remove redundant data.
CO4	Will be able to write SQL commands to work with any database.
CO5	Will have an understanding on issues involved in database transaction and error recovery.

6. Topics:

Module I: Introduction to Database (6 hours)

Database, characteristics of database approach, advantages of DBMS, database models, database architecture and data independence, database languages, classification of DBMSs

Module II: Entity Relationship Model (8 hours)

Database design and ER Model: overview, ER Model, Constraints, ER Diagrams, ERD Issues, weak entity sets, subclasses, superclasses, and inheritance, specialization and generalization

Module III: Relational Data Models and SQL (9 hours)

Relational model concept, relational model constraints, relational database schemas, Codd’s rules, ER to relational model mapping, SQL data definition and data types, specifying constraints in SQL, retrieval queries in SQL, insert, delete and update statements in SQL, assertions, triggers, views, schema change statements.

Module IV: Relational Algebra and Calculus (7 hours)

Unary relational operations: SELECT and PROJECT, relational algebra operations from set theory, binary relational operations: JOIN and DIVISION, tuple relational calculus, domain relational calculus

Module V: Dependencies and Normal Forms (10 hours)

Importance of a good schema design, motivation for normal forms, dependency theory functional dependencies, Armstrong's axioms for FD's, closure of a set of FD's, minimal covers, definitions of 1NF, 2NF, 3NF and BCNF, decompositions and desirable properties of them, algorithms for 3NF and BCNF normalization, multi-valued dependencies and 4NF, join dependencies and definition of 5NF.

Module VI: Transaction Processing and Error Recovery (8 hours)

concepts of transaction processing, ACID properties, concurrency control, locking based protocols for CC, error recovery and logging, undo, redo, undo-redo logging and recovery methods.

7. Laboratory Sessions: The practical component of this course is of one credit, i.e., 2 hours of lab classes per week. The laboratory work consists of the following.

- Introduction to DBMS software and SQL.
- Introduction to Oracle software
- DDL to create, change schema of database and relation and to grant access right.
- SQL data type.
- Insert, update, delete and retrieval queries in SQL.
- Specifying primary and foreign key and other integrity constraints.
- Nested SQL queries and joining of tables.
- Aggregate functions.
- SQL to create views.
- PL/SQL subprograms.
- Writing triggers.

8. Referential Sources:

1. Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems.
2. Avi Silberschatz, Henry Korth, S. Sudarshan, Database System Concepts.

Back to Contents

Computer Organization and Architecture (IPCTMJ4402)

1. About the Course: This is a Major course which deals with the structure of different parts of a computer system and how these parts function together to form the whole system. Low level programming of a system for a specific architecture is also addressed in this course.

2. Course Description:

- Target Audience: Fourth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1+ 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
    - Lectures: 30 Hours (15 Weeks X 2 Hours)
    - Tutorial: 15 hours (15 Weeks x 1 Hour)
    - Practical: 30 hours (15 weeks x 2 hours)

3. Prerequisites and Dependencies: There is no prerequisite for this course. Students are expected to have fundamental knowledge on computer hardware, software and programming.

4. Objective: The objective of the course is to equip students with the knowledge of structure and function of different units of a computer system and how these units coordinate with each other to perform a task. The students are also taught about Instruction set Architecture and assembly language programming.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1 To understand the internal structure of a computer system.

CO2 To describe structure of different units of a computer system and how they are organized and co-ordinate with each other.

CO3 To understand Instruction Set Architecture and machine language. CO4. To write assembly language program.

6. Topics:

Module I: Introduction to Digital System (6 hours)

Digital vs Analogue System, Binary number system, 1's and 2's complement, Information representation using code, ASCII code, Digital logic gates, Combinational circuit-adder, subtractor, Decoder, Encoder, MUX, demultiplexer, Sequential Circuit-flipflop, registers

Module II: Basic structure of computer (3 hours)

Block diagram of a Computer System, Functional Units of computer, Basic Operational Concept, Bus Structure, Performance-processor clock, performance equation, pipelining and Superscalar Operation, Instruction set: CISC and RISC, Multiprocessor and Multicomputer, Computer generations

Module III: Machine Instructions and Programs (10 hours)

Number, arithmetic operations and characters, Memory locations and addresses-Byte addressability, Big-Indian and Little-Indian assignment, Word Assignment, Memory operations, Instructions Register transfer notation, Assembly language notation, Basic instruction type, Instruction execution, Branching, Addressing modes, Assembly language, assembler directive, Basic Input-Output Operation, Stacks, Subroutines

Module IV: The Memory System (9 hours)

Some basic concepts, Interfacing memory with processor, Semiconductor RAM Memories, Internal Organization of Memory Chips, Static and Dynamic memory System, Memory Controller, Read Only Memories Speed, Size and Cost of Memory, Cache MemoriesMapping function and replacement algorithm, Hit Rate and Miss Penalty, Virtual memoriesaddress translation

Module V: Input / Output Organization (9 hours)

Accessing input/output device, interrupts-interrupt hardware, enabling and disabling interrupt, handling multiple devices, controlling device request, Exceptions, Direct Memory Accesses, Buses-Synchronous and asynchronous buses, Interface circuits-parallel and serial port, Standard I/O interface-PCI bus, SCSI bus, USB bus.

Module VI: Pipelining (9 hours)

Pipeline performance, data hazards-operand forwarding, handling data hazards in software, Instruction hazards- unconditional branches, Conditional branches and branch prediction, Influence on instruction set, Design Issue.

7. Laboratory Sessions: The practical component of this course is of one credit, i.e., 2 hours of classes per week. The laboratory sessions mainly deals with the following topics-

- Introduction to 8086- Instruction Set Architecture and the simulator to be used for programming.
- Assembly language programming and assembler.
- Different types of registers.
- Memory addressing.
- Variable, array, constant etc.
- Data transfer operation.
- Interrupt and I/O operations
- Arithmetic and logic instructions
- Program flow control- branching, looping etc

8. Referential Sources:

1. Carl Hamacher, Zvonko Vranesic, Zaky, “Computer Organization”, McGraw Hill
2. William Stallings,” Computer Organization and architecture: Designing for performance”, Pearson Education India
3. Mano M Morris, “Computer System Architecture”, Pearson Education India

----

Back to Contents

Formal Language and Automata (IPCTMJ4403)

1. **About the Course**: This is a Major Course that deals with the study of abstract computing devices and the computational problems that can be solved using them. This course is organized as a series of lectures with theory and tutorial sessions.

2. **Course Description**:

- Target Audience: Fourth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (3 + 1+ 0)
- Total Credit Hours: 5
- Total Contact Hours: 60 hours (15 Weeks X 3 hours + 15 weeks x 1 hours)
    - Lectures: 45 Hours (15 Weeks X 3 Hours)
    - Tutorial: 15 hours (15 Weeks x 1 Hour)
    - Practical: Nil

3. **Prerequisites and Dependencies**: This is a major course offered for fourth semester students. The students are expected to have good reasoning skill and mathematical background.

4. **Objective**: The main objective of the course is to give students a broad understanding of automata theory and to introduce computability theory. Automata is mainly an abstract computing device and is a useful model for many kinds of software and hardware components, used in many area of computer science such as artificial intelligence, embedded system, compiler design etc.

5. **Course Outcomes**: After course completion, following are the course outcomes.

CO1	Acquire a fundamental understanding of the core concepts in automata theory, formal language and grammar.
CO2	Identify formal language classes and their relationship.
CO3	Determine the decidability and intractability of computational problems.
CO4	Be able to design grammars and automata for different languages.

6. **Topics**:

Module I: Finite automata (6 hours)

Finite automata, alphabets, strings, languages, deterministic finite automata, how a DFA processes a string, transition functions, the languages of a DFA, nondeterministic finite automata, the language of an NFA, Equivalence between NFA and DFA, application of NFA and DFA in text search, finite automata with epsilon transitions, uses of epsilon transitions, epsilon closure, transition function for epsilon NFA, languages of epsilon NFA, eliminating epsilon transitions

Module II: Regular expressions and languages (12 hours)

Regular expression, operators of regular expressions, building regular expressions, precedence of regular expression operators, finite automata and regular expression, converting DFA to regular expression, converting regular expressions to automata, applications of regular expressions: lexical analysis, finding patterns in text, algebraic laws for regular expressions, pumping lemma, closure and decision properties of regular expressions, equivalence of regular languages, minimization of DFA

Module III: Context Free Grammar and Push Down Automata (12 hours)
Context free grammar, derivation using a grammar, leftmost and rightmost derivations, the language of a grammar, sentential forms, parses trees, inference, derivation and parse tree, ambiguous grammar, removing ambiguity from grammars, PDA, graphical notation of PDA, instantaneous description, acceptance by final state, acceptance by empty stack, conversion between empty stack and final state, equivalence between PDA and CFG, conversion from grammar to PDA and PDA to grammar, deterministic PDA and it's relationship with regular language, CFG and ambiguous grammar, Chomsky normal form, pumping lemma for CFL, closure and decision properties of CFL, Context sensitive language, linear bounded automata, Chomsky hierarchy.

Module IV: Turing machine (12 hours)

Turing machine, notation for Turing machine, instantaneous description for Turing machine, transition diagram for Turing machine, the language of a Turing machine, Turing machines and halting, multitrack Turing machine, nondeterministic Turing machine, simulating a Turing machine by computer and simulating a computer by Turing machine.

Module V: Undecidability and intractability (12 hours)

Languages not recursively enumerable, enumerating binary strings, codes for Turing machine, the diagonalization language, recursive languages, universal language, classes P and NP, examples of P and NP problems, polynomial time reduction.

7. Referential Sources:

1. Introduction to automata theory, languages and computation, John Hopcroft, Rajeev Motwani and Jeffrey Ullman
2. Theory of automata, formal language and automata, S. P. Eugene Xavier
3. An introduction to formal language and automata, Peter Linz

----

Data Communication
(IPCTMJ4404)

1. About the Course: This is a Major Course and is aimed at making a student comfortable with the basic concepts of networking. The course covers lowest two layers of TCP/IP model and thus provides a base for Computer Network course which contains rest of the TCP/IP layers.

2. Course Description:

- Target Audience: Fourth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (3 + 1+ 0)
- Total Credit Hours: 5
- Total Contact Hours: 60 hours (15 Weeks X 3 hours + 15 weeks x 1 hours)
  - Lectures: 45 Hours (15 Weeks X 3 Hours)
  - Tutorial: 15 hours (15 Weeks x 1 Hour)
  - Practical: Nil

3. Prerequisites and Dependencies: Data Communication delves partially (which are completely covered in computer network course) into the issues that arise in the course of communication between two processes (instances of programs) in two nodes over a computer network. To understand this scenario, a student should have basic knowledge of what is a program and how is it written.

4. Objective: The major objective of this course is to provide students with understanding the issues that are handled in two lowest layers of TCP/IP model, viz., Physical layer and Data link layer.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Understanding TCP/IP and OSI model
CO2	Understanding signal, transmission, transmission media and switching
CO3	Understanding the Data Link layer in TCP/IP model.

6. Topics:

Module I: Introduction (6 hours)

Data communications, components, data representation, data flow, networks, physical structure of network, network models, categories of networks, Internetwork, Internet, protocols, OSI model, layers in the OSI model: physical layer, data link layer, network layer, transport layer, session layer, presentation layer, application layer, TCP/IP protocol suite, comparison between OSI and TCP/IP model, physical addresses, logical addresses, port addresses

Module II: Signals and transmission (15 hours)

Analog and digital data, analog and digital signals, periodic and non-periodic signals, sine wave, phase, wavelength, time and frequency domains, composite signals, bandwidth, digital signal, bit rate, bit length, digital to digital conversion, line coding, block coding, analog to digital conversion, pulse code modulation, delta modulation, parallel and serial transmission, digital to analog conversion, amplitude shift keying, frequency shift keying, phase shift keying, quadrature amplitude modulation, analog to analog conversion, amplitude modulation, frequency modulation, phase modulation

Module III: Transmission media and switching (15 hours)
Guided media, twisted pair cable, co-axial cable, fiber-optic cable, unguided media, radio waves, micro waves, infrared, switching-packet, message and circuit switching, circuit switched networks, datagram networks, virtual circuit network, telephone network, dial up modems, DSL

Module IV: Data link layer (15 hours)

Data link layer and its functionalities, types of errors, redundancy, detection versus correction, coding, polynomial code, block coding, hamming distance, linear block codes, cyclic codes, CRC, checksum, framing, flow control, error control, protocols, simplest protocol, stop and wait protocol, go back n protocol, selective repeat protocol

7. Referential Sources:

1. Forouzan Behrouz A., “Data Communications and Networking”. Mcgraw Hill
2. Stallings William. “Data and Computer Communications”. Pearson Education India

----

Back to Contents

Operating System (IPCTMJ5401)

1. About the Course: This is a Major Course and is aimed to make students understand the concepts of the most important system software called operating system. The course covers various issues of operating system that one needs to know to understand the structure of an operating system.

2. Course Description:

- Target Audience: Fifth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1+ 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
    - ➢ Lectures: 30 Hours (15 Weeks X 2 Hours)
    - ➢ Tutorial: 15 hours (15 Weeks x 1 Hour)
    - ➢ Practical: 30 hours (15 Weeks x 2 Hour)

3. Prerequisites and Dependencies: Operating system course has a dependency on "Introductory Computing using C" course. A student needs to be also aware of basic concepts like hardware components, software and types of software.

4. Objective: The major objective of this course is to provide students with understanding of the overall structure of an operating system.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	To understand the services provided by operating system and design of an operating system
CO2	Understanding Process Management module of operating system
CO3	Understanding Memory Management module of operating system.
CO4	Understanding File System Management module of operating system.
CO5	Understanding I/O management module of operating system.
CO6	Understanding OS Security: Threats and Security Controls

6. Topics:

Module I: Basics (4 hours)

Operating System Functionalities, Types of Operating Systems, Structure of Operating System, Distributed Systems, Services, System Calls, Virtual Machines, System Boot.

Module II: Process Management (10 hours)

Process Scheduling - Uniprocessor scheduling algorithms, Multiprocessor and Real-time scheduling algorithms, Process Synchronization - Peterson's Solution, Bakery Algorithm, Hardware Support to Process Synchronization, Semaphores, Critical Regions, Monitors - Deadlock prevention, deadlock avoidance and Deadlock Detection and Recovery - Bankers Algorithm, Threads.

Module III: Memory Management (10 hours)

Swapping, Contiguous Memory Allocation, Paging Structure of the Page Table Segmentation Example: The Intel Pentium, Virtual memory, Demand Paging, Page Replacement, Thrashing, Memory Mapped Files, Allocation of Kernel Memory.

Module IV: File Systems (7 hours)

Contiguous, Sequential and Indexed Allocation, File system interface, File System implementation, Case study of Unix File system - Buffer Cache, Inodes, The system calls - ialloc, ifree, namei, alloc and free, Mounting and Unmounting files systems, Network File systems.

Module V: I/O System (7 hours)
I/O Hardware, Application I/O Interface, Kernel I/O Subsystem, Device drivers - block and character devices, streams, Character and Block device switch tables.

Module VI: Protection and Security (7 hours)

Goals of Protection, Principles of Protection, Domain Protection, Access Matrix, Access Control, Security Problem, Program Threats, System and Network Threats, User Authentication, Firewall.

7. Laboratory Sessions: This course contains 2 hours of practical classes per week. Following topics will be covered in the laboratory classes.

- Introduction to various process, memory and file management commands in windows operating system.
- Introduction to various process, memory and file management commands in linux operating system.
- Introduction to system calls in linux operating systems.
- Introduction to system calls in windows operating systems.

8. Referential Sources:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, “Operating System Concepts”. Wiley India Pvt. Ltd
2. Maurice Bach, “Design of the Unix Operating System”. Prentice Hall India Learning Private Limited

----

Computer Network
(IPCTMJ5402)

1. **About the Course**: This is a Major Course and is aimed to make the students understand the concepts of computer network and make them capable of configuring networks and handling issues that arise in a computer network.

2. **Course Description**:

- Target Audience: Fifth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1+ 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
  - ➢ Lectures: 30 Hours (15 Weeks X 2 Hours)
  - ➢ Tutorial: 15 hours (15 Weeks x 1 Hour)
  - ➢ Practical: 30 hours (15 Weeks x 2 Hour)

3. **Prerequisites and Dependencies**: Computer Network has no dependency however, a student needs to have knowledge about C programming as the practical classes of the course covers an advanced networking module of C programming that is called Socket programming.

4. **Objective**: The major objective of this course is to provide students with understanding of all the aspects of a communication over a network and all the standards related to such communication. This course presents understanding of entire communication process and standards between a pair of programs under execution (processes). The course also includes lab component that covers the programming aspect to develop such programs.

5. **Course Outcomes**: After course completion, following are the course outcomes.

CO1	Understanding the Network layer in TCP/IP model.
CO2	Understanding the Transport layer in TCP/IP model.
CO3	Understanding application layer.

6. **Topics**:

Module I: Network Layer (15 hours)

Network layer and its functionalities, IPv4 addresses, address space, notation of IP addresses, classful addressing, classless addressing, IPv6 addresses, address mapping, ARP, RARP, DHCP, forwarding techniques, forwarding process, Adaptive and nonadaptive routing, unicast routing protocols, distance vector routing, link state routing, path vector routing, unicast, multicast and broadcast routing

Module II: Transport layer (15 hours)

Transport layer and its functionalities, Process to process delivery, client/server paradigm, multiplexing and demultiplexing, connectionless versus connection-oriented service, TCP handshaking, connection establishment and connection release, data traffic, congestion, congestion control, QoS, integrated services, differentiated services

Module III: Application layer (15 hours)

Domain name system, namespace, flat namespace, hierarchical name space, domain name space, Internet, DNS in Internet, name resolution, mapping between names and addresses, telnet, electronic mail, FTP, WWW, browser, web server, URL, cookies, Application layer protocol-SMTP, MIME, a brief introduction to cryptography, symmetric vs asymmetric cryptography

7. Laboratory Sessions: This course contains 2 hours of practical classes per week. Following topics will be covered in the laboratory classes.

- Introduction to switches, routers, cables.
- Crimping of cat 6 cables.
- Various networking commands in windows operating systems.
- Various networking commands in linux operating systems.
- Configuring a LAN using switches and cat 6 cable.
- Socket programming in C.

8. Referential Sources:

1. Forouzan Behrouz A., "Data Communications and Networking". Mcgraw Hill
2. Stallings William. "Data and Computer Communications". Pearson Education India

----

Back to Contents

Software Engineering
(IPCTMJ5403)

1. **About the Course**: This is a Major Course and it comprises of the core principles in efficient and consistent software development and maintenance. An introduction to object-oriented software development process and design has also been included in the course.

2. **Course Description**:

- Target Audience: Fifth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1+ 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
  - Lectures: 30 Hours (15 Weeks X 2 Hours)
  - Tutorial: 15 hours (15 Weeks x 1 Hour)
  - Practical: 30 hours (15 Weeks x 2 Hour)

3. **Prerequisites and Dependencies**: The students are expected to have knowledge on programming and software.

4. **Objective**: The major objective of a software engineering course is to equip students with the principles, methodologies, and tools necessary to design, develop, test, and maintain high-quality software systems. This includes teaching problem-solving skills, software design patterns, development processes (such as Agile and DevOps), project management techniques, and best practices for ensuring software reliability, efficiency, scalability, and security. The course aims to prepare students to work effectively in software development teams, manage software projects, and adapt to evolving technologies and industry demands.

5. **Course Outcomes**: After course completion, following are the course outcomes.

CO1	Define various software application domains and applying different process model used in software development
CO2	Explain needs for software specifications.
CO3	Convert the requirements model into the design model and demonstrate use of software and user interface design principles.
CO4	Can classify different testing strategies and tactics
CO5	Justify role of SDLC in Software Project Development
CO6	Generate project schedule, design and develop network diagram for different type of Projects.

6. **Topics**:

Module I: Introduction (5 hours)

Why study Software Engineering, Evolution and Impact of Software Engineering, Software Development Projects, Programs vs Products, Emergence of Software Engineering, Early Computer Programming, High-Level Language Programming, Control Flow- Based Design, Data-Structure-Oriented Design, Data Flow-Oriented Design, Object-Oriented Design

Module II: Software Life Cycle Models (6 hours)

Use of Life Cycle Model, Classical Waterfall Model, Iterative Waterfall Model, Prototyping Waterfall Model, Evolutionary Model, Spiral Model , Comparisons of different Life Cycle Models

Module III: Software Project Management (8 hours)

Responsibility of a Software Project Manager, Project Planning, Matrices for Project Size Estimation: LOC, Function Point Metric, Project Estimation Technique-Empirical Estimation Technique, Heuristic Technique, Analytical Estimation Technique, COCOMO Model, Scheduling Work Breakdown Structure, Activity Networks and Critical Path Model, Gantt Charts, PERT charts, Project Monitoring and Control, Organization Structure and Team Structure, Staffing, Risk Management- Risk Identification, Risk Assessment, Risk Containment, Software Configuration Management.

Module IV: Requirement Analysis and Specification (6 hours)
Requirement Gathering and Analysis, Software Requirement Specification, Characteristics of a good SRS Document, Functional Requirement, Traceability, Algebraic Specification

Module V: Software Design (8 hours)

Classification of Design Activities and Design Methodologies, Analysis versus Design, Cohesion and coupling, Functional Independence, Approaches to Software Design Function Oriented Design, Object-oriented Design, Function Oriented Design: Structured Analysis, Data Flow Diagrams (DFD's), Case Scenarios using DFD, Object-oriented Design: Object Modeling using UML, UML Diagrams, User Interface Design: Characteristics of a good user interface

Module VI: Coding and Testing (6 hours)

Coding Standards and Guideline, Code Review-Code Walkthrough, Code Inspection, Clean Room Testing, Software Documentation, Testing-designing test cases, Testing in the Large vs Testing in the small, Unit Testing-Driver and Stub Modules, Black-Box Testing, White Box Testing, McCabes Cyclomatic Complexity Metric, Data Flow-based Testing, Mutation Testing, Debugging, Program Analysis Tool, Integration Testing, System Testing-Performance Testing, Error Seeding ,Regression Testing

Module VII: Software Maintenance and Software Reuse (6 hours)

Characteristics of Software Maintenance, Software Reverse Engineering, estimation of Maintenance Cost, Software Reuse-basic issues in any Reuse Program, A Reuse Approach Domain analysis, Component classification, Searching, Repository Maintenance, Reuse without Modification, Reuse at Organization level

7. Laboratory Sessions: The practical component of this course is of one credit which amounts to 2 hours of Laboratory class per week.

- Distinguish between Manual and Automation testing.
- Introduction of Commercial and open-source automation tools.
- Understanding test automation process.
- Working with testing framework, preferably Selenium / QTP.

Back to Contents

- Setup/Installation/configuration of testing environment, Preparation, maintenance, debugging of test-scripts, Exception handling and reporting.

8. Referential Sources:

1. Fundamentals of Software Engineering by Rajib Mall
2. Test Automation Using selenium webdriver with java by Navneesh Garg

Useful Web Sources: https://www.toolsqa.com

Back to Contents

Object Oriented Programming using Java (IPCTMJ5404)

1. **About the Course**: This is a Major Course and is aimed at making a student comfortable with object-oriented Programming and Design using Java and its features. The course is organized as a series of lectures with both theory and laboratory sessions.

2. **Course Description**:

- Target Audience: Fifth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1+ 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
  - Lectures: 30 Hours (15 Weeks X 2 Hours)
  - Tutorial: 15 hours (15 Weeks x 1 Hour)
  - Practical: 30 hours (15 Weeks x 2 Hour)

3. **Prerequisites and Dependencies**: Students are expected to be comfortable in procedure oriented (C language). So “Introductory Computing using C” is prerequisites for this course.

4. **Objective**: The major objective of this course is to equip students with programming skills to design high-end GUI based applications using Java APIs.

5. **Course Outcomes**: After course completion, following are the course outcomes.

CO1	Understanding the approach to solve a problem in Java.
CO2	Writing an efficient Java program with appropriate language constructs to solve a problem.
CO3	GUI based application development.

6. **Topics**:

Module I: Introduction to Object Oriented Programming (2 hours)

Programming paradigm, What is Object Oriented Programming, Object oriented programming vs procedure oriented programming, Basic concepts of OOP-Encapsulation, Polymorphism and Inheritance, Object oriented language

Module II: Core Java Programming (8 hours)

Java Overview: Genesis, Java Philosophy, Java & Internet, Object-Oriented Programming features, Java Applet and Application, Java Environment and Java Development Kit (JDK) & Java Standard Library (JSL), Java language fundamentals, The scope and lifetime of variable, Type conversion and casting, Control statements, Arrays, classes and objects: The this keyword, Garbage collection, Overloading constructor, Using object as parameters, Argument passing, Returning objects, Recursion, Introducing Access control (public, private and protected), static, final, nested classes, String class, Command-line argument.

Module III: Inheritance, Exception handling (5 hours)

Inheritance: Member access and inheritance, method overriding, dynamic method dispatch, using abstract classes, using final with inheritance, the Object class; Packages, Interface, classpath, Exception handling: Fundamentals, Exception types, Java's built-in exceptions, user defined exceptions.

Module IV: Multithreading and I/O Basics (5 hours)

Multithreaded Programming: The Java thread model (thread priorities, synchronization and inter-thread communication); Deadlock, ThreadGroup, I/O Basics : (Streams, The stream classes, The predefined streams, Reading console input, writing console output, The transient and volatile modifiers, using instance of native methods

Module V: String handling, Utility classes, java.lang and java.io (8 hours)
String handling: String constructors, methods for character extraction, string searching & comparison, data conversion using valueof (), StringBuffer, Exploring java.lang: Simple type wrappers, System class, class Class, Math functions, The utility classes: Vector, Stack, HashTable, StringTokenizer, Bitset, Date, Calendar, GregorianCalendar, Random, Observable, Input/Output-Exploring java.io: The java.io classes and interface, File class and methods for creating, renaming, listing and deleting files and directories, I/O stream classes (FileInputSream, FileOutputStream, BufferedInputStream, BufferedOutputStream, PushBackInputStream, InputStreamReader, BufferedReader, BufferedWriter, PrintStream, RandomAccessFile)

Module VI: Networking, Images (5 hours)

Networking: Socket overview, Stream Sockets, Datagram sockets, Manipulating URLs, Establishing a simple Server/Client using Stream Sockets, Connectionless Client/Server Interaction with Datagrams, Images: File formats, image fundamentals, creating, loading and displaying images, ImageObserver, MediaTracker

Module VII: Applet class and Swing (7 hours)

The Applet class: applet architecture, passing parameters to applets, getDocumentBase, getCodeBase, and showDocument, AppletContext and AudioClip interfaces, Graphics class and methods for drawing lines, rectangles, polygons and ovals, Swing: Component and Container classes, Layout managers (FlowLayout, GridLayout, BorderLayout), Handling events, Adapter classes, Anonymous inner classes Swing GUI components (JLabel, JTextField, JTextArea, JButton, JCheckBox, JRadioButton, JList, JComboBox, JScrollBar, JScrollPane, JToolTip, JPanel, JFrame), Menus: JmenuBar, JMenu, JMenuItem, JSeparator

Module VIII: JDBC (2 hours)

Java database connectivity (JDBC): Introduction to JDBC, type of JDBC connectivity, establishing database connections, Accessing relational database from Java programs

Module IX: Java Beans, Java Servlets (3 hours)

Java Beans: Introducing JavaBeans Concepts and Bean Development kit (BDK), Using the Bean Box, Writing a simple Bean, Bean Properties (simple properties), Manipulating events in the Bean Box

Back to Contents

7. Laboratory Sessions: The practical component of this course is of one credit which amounts to 2 hours of Laboratory classes per week. Following are the components that will be covered in practical sessions. Students has to submit a minor Project using Java language at the end of this course.

- Introduction to editor, compiler, debugger and IDE.
- Compilation process of a Java program.
- Basic java programs with input and output statements.
- Java programs to handle data types and variables.
- Java programs with branching statements.
- Java programs with looping statements.
- Java programs with array.
- Handling strings in a Java program.
- Multi-dimensional array in Java program.
- Functions in java program.
- Java programs to implement object-oriented concepts.
- Multithreading concepts.
- GUI Programming (AWT/Swing).
- Working with database (JDBC/ ODBC connection).

8. Referential Sources:

1. Schildt Herbert, "The Complete Reference Java". McGraw Hill.
2. Balagurusamy E., "Programming with Java: A Primer". McGraw Hill.

Useful Web Sources: https://nptel.ac.in/courses/106105191/: NP-TEL content.

Back to Contents

Computer Graphics
(IPCTMJ6401)

1. About the Course: This is a Major Course that mainly addresses different methodologies and algorithms used in generating picture on display device. This course is organized in a series of lectures with theory, tutorial and practical sessions.

2. Course Description:

- Target Audience: Sixth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1+ 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
  - Lectures: 30 Hours (15 Weeks X 2 Hours)
  - Tutorial: 15 hours (15 Weeks x 1 Hour)
  - Practical: 30 hours (15 Weeks x 2 Hour)

3. Prerequisites and Dependencies: Students are expected to have basic knowledge on computers including algorithm and programming. Courses on "Introductory Computing using C" and "Data Structure" are prerequisite for this course.

4. Objective: The main objective of the course is to provide students a broad understanding on how a 2D or 3D image is generated on a display device. Different algorithms for generating pictures starting from a line to a 3D object is included in the course. Students will be taught to design 2D/3D graphics and apply different operations such as transformations and clipping over an image.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Will have a broad idea on different algorithms to generate a 2D/3D image on a display device.
CO2	Can perform different transformations such as translation, rotation, scaling, reflection and shearing on images.
CO3	Can design 2D/3D graphics, animation and can apply clipping operations.
CO4	Can apply color models

6. Topics:

Module I: Introduction (4 hours)

Overview of Graphics System-Video display devices, Raster Scan System, Random Scan System, Graphics Monitor, Hard copy devices, Graphics Software.

Module II: Output Primitives and Attributes (9 hours)
Points and lines, Line Drawing Algorithm, Circle and Ellipse Generation Algorithm, Scan line polygon fill algorithm, Boundary fill and flood fill algorithm, Antialiasing

Module III: Two Dimensional Geometric Transformations and Viewing (11 hours)
Translation, Scaling, Rotation, Reflection and Shear Transformation: Matrix representation and Homogeneous Coordinates, Composite Transformations, Transformations between coordinate system, viewing pipeline, Viewing coordinate reference frame, Window to view port

coordinate transformation, Two dimensional viewing function, line and polygon clipping algorithm

Module IV: Three Dimensional Concepts (16 hours)

Three-dimensional display method, Three dimensional graphics package, Three dimensional object representation-polygon surface, Curved line and surface, Quadric surface, Blobby object, Spline representation, Cubic spline interpretation method, Beizer curves and surfaces, B-spline curves and surfaces, Octrees, Three dimensional geometric modeling and transformations, Three dimensional viewing, visible surface detection, Surface rendering methods

Module V: Color Models and Color Application (5 hours)

Standard primaries and Chromaticity diagram, RGB color model, YIQ color model, CMY color model, HSV color model, HLS color model, Color selection and application

7. Laboratory Sessions: The practical component of this course is of one credit, i.e., 2 hours of classes per week. The practical component mainly contains programming the algorithms that are discussed in theory classes in any high-level language.

- Introduction to graphics package.
- How to draw pixel, line, circle etc. using graphics function and different parameters of the functions.
- Program to implement DDA algorithm.
- Program to implement Bresenham's line drawing algorithm.
- Program to implement Mid-point circle generating algorithm.
- Program to implement Mid-point ellipse drawing algorithm.
- Program to implement Scan-line polygon fill algorithm.
- Program to implement Flood fill and boundary fill algorithm.
- Draw 2D image using graphics function.
- Program for 2D geometric transformation (translation, rotation, scaling, reflection, shearing etc.).
- Program for point clipping, line clipping and polygon clipping.
- Design simple 2D animation.

8. Referential Sources:

1. Hearn and Baker, "Computer Graphics".
2. Edward Angel, "Interactive Computer Graphics: A Top-Down Approach with OpenGL". Pearson, 5th Edition, 2009.

Back to Contents

Web Technology
(IPCTMJ6402)

1. About the Course: This is a Major Course and is aimed at teaching skills to design interactive and dynamic web sites. The course is designed to deliver key technology components like descriptive languages and server-side program elements. In addition, the course gives specific contents that are beneficial for developing web-based solutions like communication with a relational database, data security principles and approaches. The focus of this course would be on advanced topics in emerging Web technologies. These include extensions of Web standards, combination of different Web technologies, Web toolkits and development environments, current backend Web frameworks.

2. Course Description:

- Target Audience: Sixth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1+ 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
  - ➢ Lectures: 30 Hours (15 Weeks X 2 Hours)
  - ➢ Tutorial: 15 hours (15 Weeks x 1 Hour)
  - ➢ Practical: 30 hours (15 Weeks x 2 Hour)

3. Prerequisites and Dependencies: The course is built up on the knowledge of previous courses such as the "Data Structures" and "Introductory Computing using C".

4. Objective: The main objective of this course is to provide knowledge on web architecture, web services, server-side scripting technologies to focus on the development of web-based information systems and web services.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Define the fundamental ideas and standards underlying Web Service Technology.
CO2	Define the fundamental principles for cloud applications.
CO3	Discuss concepts at the frontier of industrial practice and emerging standards.
CO4	Differentiate the major frameworks allowing to develop web services and cloud applications and assess their suitability for specific usage scenarios.

6. Topics:

Module I: Introduction (3 hours)

Computer and network, Intranet vs Internet, Client-Server Computing, IP address, Internet services, Hyper Text Transfer Protocol(HTTP), HTTP transaction-persistent vs nonpersistent , Ports and sockets, Proxy Server.

Module II: World Wide Web (4 hours)
Architecture-client, server, Uniform Resource Locator(URL), Domain Name Service(DNS), Address resolution, Name resolution, Web documents-static document, dynamic document and

active documents, Cookies, Virtual hosting, Browser, Browser architecture, HTTP request and response.

Module III: Markup Language (4 hours)
Markup language, SGML, HTML, HTML tags and attributes, Cascading Style Sheet (CSS).

Module IV: Web Programming (3 hours)
Scripting language, Client Side Scripting Language and Server Side Scripting Language, Writing JavaScript.

Module V: Introduction to PHP (7 hours)
PHP Functionalities, Datatypes, Variables, Constants, Arrays, Functions, Strings, System Calls, Explode-Implode and other native functions.

Module VI: Core PHP Concepts (8 hours)
Handling Html Form with PHP, Working with file and Directories, Sessions and Cookies, Database connectivity using MySQL and MySQLi, Concept of PDO, Performing basic database operation using DML commands (such as Insert, Delete, Update, Select), Setting query parameter, Executing query Joins (Cross joins, Inner joins, Outer Joins, Self joins.), Exception Handling. Introduction to Wordpress CMS.

Module VII: Introduction to Laravel (8 hours)
Installation, Artisan CLI, Laravel directory structure, Basic routing, Call a controller method from a route, Template inheritance, Blade conditional statements, Blade Loops, Executing PHP functions in blade, Introduction to Migrations, Migration structure, Creating a basic controller, Creating a route using a closure, Eloquent ORM Models, Eloquent ORM INSERT, READ, UPDATE, DELETE, Using models in controllers.

Module VIII: Introduction to NodeJS and ExpressJS (8 hours)
Installation procedures, NodeJS console, REPL, TLS/SSL, Debugger, Process, Child Process, Buffers, Streams, File System, Path, Query String, Assertions, Callbacks, Events, TTY, Webmodules, Database connectivity, Intro to ExpressJS, Express.js fundamental concepts like Routing and HTTP Methods, Middleware, Cookies, REST API, Scaffolding, Templating and Error Handling.

7. Laboratory Sessions: This course contains 2 hours of practical classes per week. Following topics will be covered in the laboratory classes.

- Introduction to Server-side scripting technology using PHP.
- Core PHP concepts.
- Introduction to Laravel framework.
- Concept on CMS using Wordpress/Drupal.
- Introduction to NodeJS.
- Introduction to ExpressJS.

8. Referential Sources:

1. Papazoglou, “Web Services: Principles and Technology (2nd edition)”; ISBN: 978-027373216-7, Prentice Hall, 2012
2. Cerami, “Web Services Essentials”; ISBN: 0596002246, O’Reilly,2002

Useful Web Sources: https://www.php.net/docs.php: PHP Documentation

Python Programming
(IPCTMJ6403)

1. About the Course: This is a Major Course that focuses on how to practice and culture the art of programming with Python as a language. The course is designed to explore Python's powerful features, making it ideal for writing effective programs.

2. Course Description:

- Target Audience: Sixth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1+ 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
  - Lectures: 30 Hours (15 Weeks X 2 Hours)
  - Tutorial: 15 hours (15 Weeks x 1 Hour)
  - Practical: 30 hours (15 Weeks x 2 Hour)

3. Prerequisites and Dependencies: The course is intended to those having experience with high level languages from both procedure and object-oriented paradigms. Prior knowledge of a scripting languages (Perl, PHP, UNIX/Linux shells) is helpful but not mandatory.

4. Objective: This course leads the students from the basics of writing and running Python scripts to more advanced features such as file operations, working with binary data and using the extensive functionality of Python modules. Extra emphasis is placed on features unique to Python, such as tuples, array slices, and output formatting.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Able to develop Python Scripts.
CO2	Implementing object-oriented concepts.
CO3	Build and package Python modules for reusability.
CO4	File I/O.
CO5	Implementing database and GUI applications.

6. Topics:

Module I: Introduction to Python (3 hours)

Introduction to Python and IDLE to develop programs. Working with datatypes and variables, operators and expressions, working with numeric data, working with string data, Python functions, Boolean expressions, selection structure, iteration structure.

Module II: Working with Data (7 hours)
Mutable and Immutable data type, Lists, indexing and slicing, tuples, dictionaries, references and copies.

Module III: Program Structure (8 hours)

Assignment statements, blocks and syntax rules, expression statements, branching, multiway branching, looping, decisions, continue and break, nested loops control flow.

Module IV: Functions (8 hours)

Defining functions, scope rules, global statements, pass by value vs reference, variable length arguments, argument matching, passing arguments.

Module V: Modules (8 hours)

Imports and attributes, creating modules, namespaces, reloading, module packages, handling files - file I/O, file scanners, files and directories, file positioning, renaming and deleting files.

Module VI: Classes and Objects (7 hours)

An introduction to classes and objects, defining a class, working with object composition, encapsulation, inheritance, exception handling.

7. Laboratory Sessions: The practical component of this course is of one credit which amounts to 2 hours of Laboratory classes per week. Following are the components that will be covered in practical sessions.

- Introduction to editor, translator, IDLE.
- Writing Python scripts.
- Functions in Python.
- File I/O in Python.
- Data analysis modules in Python.
- GUI based application development in Python.

8. Referential Sources:

1. Brown Martin C., “Python: The Complete Reference”. McGraw Hill.
2. Lutz Mark, “Programming Python”. O’Reilly.
3. Allen B.Downey,” Think Python: How to Think Like a Computer Scientist”,O’REILLY

Useful Web Sources: https://nptel.ac.in/courses/106106145/ : NP-TEL content

----

Data Mining
(IPCTMJ6404)

1. **About the Course**: This is a Major Course and is aimed to make students understand the concepts of data mining for extracting useful patterns, information from huge amount of data.

2. **Course Description**:

- Target Audience: Sixth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1+ 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
  - ➢ Lectures: 30 Hours (15 Weeks X 2 Hours)
  - ➢ Tutorial: 15 hours (15 Weeks x 1 Hour)
  - ➢ Practical: 30 hours (15 Weeks x 2 Hour)

3. **Prerequisites and Dependencies**: Data Mining requires "Introductory Computing using C" and "Python Programming" as prerequisite as the algorithms involved in data processing are to be implemented using C, Python programming.

4. **Objective**: The major objective of this course is to provide students with understanding of Data mining and various techniques that are used to extract useful patterns from data.

5. **Course Outcomes**: After course completion, following are the course outcomes.

CO1	Understanding Data mining.
CO2	Understanding Preprocessing tasks in Data analysis.
CO3	Understanding Classification techniques.
CO4	Understanding Clustering techniques.
CO5	Understanding Association Rule Mining Techniques.

6. **Topics**:

Module I: Introduction (8 hours)
What is data mining, data mining tasks, types of data- attribute, measurement, data quality, data preprocessing, measure of similarity and dissimalarity, Introduction to Data Warehouse

Module II: Mining Frequent Patterns, Associations, and Correlations (12 hours)
Market Basket Analysis, Frequent Itemsets, Closed Itemsets, and Association Rules, Frequent Pattern Mining, The Apriori Algorithm: Finding Frequent Itemsets Using Candidate Generation, Generating Association Rules from Frequent Itemsets, Mining Frequent Itemsets without Candidate Generation, Mining Multilevel Association Rules, Mining Multidimensional Association Rules.

Module III: Supervised Learning (15 hours)
Classification & Prediction: Decision Tree Techniques, Decision Tree Induction, Attribute Selection Measures, Tree Pruning, Scalability and Decision Tree Induction, Bayes' Theorem, Naïve Bayesian Classification, Bayesian Belief Networks, Training Bayesian Belief Networks, Rule Extraction from a Decision Tree, A Multilayer Feed-Forward Neural Network, Defining a Network Topology, Back-Propagation Method, k-Nearest-Neighbor Classifiers, Genetic

Algorithms, Regression: Linear Regression, Nonlinear Regression, Classifier Accuracy Measures, Holdout Method and Random Subsampling, Cross-validation, Bootstrap, Ensemble Methods, Bagging, Boosting.

Module IV: Unsupervised Learning (10 hours)
Clustering, Types of Data, Partitioning Methods, Hierarchical Methods, Density Based Methods, Grid Based Methods, Model Based Methods, Clustering High-Dimensional Data, Constraint Based Methods, Outlier Analysis.

7. Laboratory Sessions: This course contains 2 hours of practical classes per week. Following topics will be covered in the laboratory classes.

- Implementation of data preprocessing techniques
- Implementation of Data Characterization measures
- Implementation of Proximity Measures
- Implementation of classification techniques
- Implementation of Clustering Techniques
- Using of Weka software for different data mining tasks

8. Referential Sources:

1. Jiawei Han and Micheline Kamber, "Data Mining: Concepts and Techniques". Morgan Kaufmann India.
2. Pang-Ning Tan, Steinbach, Karpatne, Vipin Kumar, "Introduction to Data Mining", Pearson
3. Ian H. Witten, Eibe Frank, Mark A. Hall, "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann

----

Big Data Concepts
(IPCTMJ7401)

1. About the Course: This is a Major Course. This course covers various topics in big data tools, architectures, and systems that constitute big-data computing solutions in high-performance networks.

2. Course Description:

- Target Audience: Seventh semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (3 + 1 + 0)
- Total Credit Hours: 4
- Total Contact Hours: 60 hours (15 Weeks X 3 hours + 15 weeks x 1 hours)
    - ➢ Lectures: 45 Hours (15 Weeks X 3 Hours)
    - ➢ Tutorial: 15 hours (15 Weeks x 1 Hour)
    - ➢ Practical: Nil

3. Prerequisites and Dependencies: Students are expected to have knowledge on 'Data Structure', 'Design and Analysis of Algorithm', 'Data Base Management System' and Data Mining.

4. Objective: The major objective of the Big Data Concepts page is to equip readers with a comprehensive understanding of the Big Data Platform, including an overview of Apache Hadoop, HDFS (Hadoop Distributed File System) concepts, and the process of interfacing with HDFS. Additionally, the page aims to provide a solid foundation in understanding and executing MapReduce jobs, enabling users to effectively leverage these core components in managing and analyzing large-scale data.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Identify Big Data and its Business Implications.
CO2	List the components of Hadoop and Hadoop Eco-System.
CO3	Access and Process Data on Distributed File System.
CO4	Manage Job Execution in Hadoop Environment.
CO5	Develop Big Data Solutions using Hadoop Eco System.

6. Topics:

Module I: Introduction (15 hours)
Trends of Computing for Big Data-High-performance Computing (Supercomputers and Clusters), Grid Computing, Cloud Computing, Mobile Computing, Big Data Overview-Drivers of Big Data, Big Data Attributes , Data Structures , Big Data Ecosystem , Examples of Data Analytics

Module II: Big Data Tools, Techniques, and Systems (15 hours)
History of Hadoop, Apache Hadoop, Analysing Data with Hadoop, Hadoop Streaming, Hadoop Echo System, IBM Big Data Strategy, Introduction to Infosphere BigInsights and Big Sheets ,Analysing Data with Hadoop, Hadoop Streaming, Hadoop Echo System, Exascale Computing, HDFS, HBase, and NoSQL (Document Store, Graph DB, etc.), bigSQL MapReduce, Spark, Oozie, Tez, Hive, Pig, etc

Back to Contents

Module III: HDFS(Hadoop Distributed File System) (15 hours)
The Design of HDFS, HDFS Concepts, Command Line Interface, Hadoop file system interfaces, Data flow, Data Ingest with Flume and Scoop and Hadoop archives, Hadoop I/O: Compression, Serialization, Avro and File-Based Data structures.

Module IV: Map Reduce (15 hours)
Anatomy of a Map Reduce Job Run, Failures, Job Scheduling, Shuffle and Sort, Task Execution, Map Reduce Types and Formats, Map Reduce Features

7. Referential Sources:

1. Tom White " Hadoop: The Definitive Guide" Third Edit on, O'reily Media, 2012.
2. Seema Acharya, Subhasini Chellappan, "Big Data Analytics" Wiley 2015. References:
3. Michael Berthold, David J. Hand, "Intelligent Data Analysis", Springer, 2007.
4. Jay Liebowitz, "Big Data and Business Analytics" Auerbach Publications, CRC press(2013)
5. Tom Plunkett, Mark Hornick, "Using R to Unlock the Value of Big Data: Big Data Analytics with Oracle R Enterprise and Oracle R Connector for Hadoop", McGrawHill/Osborne Media (2013), Oracle press.
6. Anand Rajaraman and Jef rey David Ulman, "Mining of Massive Datasets", CambridgeUniversity Press, 2012.
7. Bill Franks, "Taming the Big Data Tidal Wave: Finding Opportunities in Huge DataStreams with Advanced Analytics", John Wiley & sons, 2012. Glen J. Myat, "Making Sense of Data", John Wiley & Sons, 2007 Pete Warden, "Big Data Glossary", O'Reily, 2011.
8. Michael Mineli, Michele Chambers, Ambiga Dhiraj, "Big Data, Big Analytics: EmergingBusiness Intelligence and Analytic Trends for Today's Businesses", Wiley Publications, 2013.

Back to Contents

Computer Security and Cryptography (IPCTMJ7402)

1. **About the Course**: This is a Major Course and is aimed at making a student aware of various security issues encountered while accessing software and websites. The course is organized as a series of lectures with theory, tutorial and practical sessions.

2. **Course Description**:

- Target Audience: Seventh semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
    - ➢ Lectures: 30 Hours (15 Weeks X 2 Hours)
    - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
    - ➢ Practical: 30 Hours (15 Weeks X 2 Hours)

3. **Prerequisites and Dependencies**: There is no prerequisite for this course. However, knowledge on mathematics and computer networks will be helpful.

4. **Objective**: The major objective of this course is to equip students with awareness of various threats that are encountered while accessing software and web services.

5. **Course Outcomes**: After course completion, following are the course outcomes.

CO1	Explain the fundamentals of cryptography, such as encryption, digital signatures and secure hashes.
CO2	Select appropriate techniques and apply them to solve a given problem.
CO3	Design and evaluate security protocols appropriate for a given situation.
CO4	Demonstrate an understanding of the mathematical underpinning of cryptography.
CO5	Demonstrate an understanding of some legal and socio-ethical issues surrounding cryptography.
CO6	To understand various protocols for network security to protect against the threats in the networks.

6. **Topics**:

Module I: Introduction to Cryptography and Block Ciphers (7 hours)
Introduction to security attacks - services and mechanism - introduction to cryptography Conventional Encryption: Conventional encryption model - classical encryption techniques substitution ciphers and transposition ciphers – cryptanalysis – steganography - stream and blockciphers - Modern Block Ciphers: Block ciphers principals - Shannon's theory of confusion and diffusion - fiestal structure - data encryption standard (DES) - strength of DES - differential and linearcrypt analysis of DES - block cipher modes of operations - triple DES and AES.

Module II: Confidentiality and Modular Arithmetic (8 hours)
Confidentiality using conventional encryption - traffic confidentiality - key distribution random number generation - Introduction to graph - ring and field - prime and relative prime numbers

- modular arithmetic - Fermat's and Euler's theorem - primality testing Euclid's Algorithm - Chinese Remainder theorem - discrete algorithms.

Module III: Public key cryptography and Authentication requirements (8 hours)
Principles of public key crypto systems - RSA algorithm - security of RSA - key management Diffle-Hellman key exchange algorithm - introductory idea of Elliptic curve cryptography – Elgamel encryption - Message Authentication and Hash Function: Authentication requirements - authentication functions - message authentication code - hash functions birthday attacks – security of hash functions and MACS.

Module IV: Integrity checks and Authentication algorithms (8 hours)
MD5 message digest algorithm - Secure hash algorithm (SHA) Digital Signatures: Digital Signatures - authentication protocols - digital signature standards (DSS) - proof of digital signature algorithm - Authentication Applications: Kerberos and X.509 - directory authentication service - electronic mail security-pretty good privacy (PGP) - S/MIME.

Module V: IP Security and Key Management (7 hours)
IP Security: Architecture - Authentication header - Encapsulating security payloads combining security associations - key management.

Module VI: Web and System Security (7 hours)
Web Security: Secure socket layer and transport layer security - secure electronic transaction (SET) - System Security: Intruders - Viruses and related threads - firewall design principals – trusted systems.

7. Laboratory Session: The practical component of this course is of one credit which amounts to 2 hours of Laboratory classes per week. Following are the components that will be covered in the practical sessions.

- Atbash Cipher, ROT13 Cipher, Caesar Cipher
- Affine Cipher, Running Key Cipher, Vigenère and Gronsfeld Cipher
- Homophonic Substitution Cipher, Four-Square Cipher, Hill Cipher, Playfair Cipher
- ADFGVX Cipher, ADFGX Cipher, Base64 Cipher, Fractionated Morse Cipher
- CRC Hash, MD5 Hash, RIPEMD160 Hash, SHA-1 Hash, SHA-256 Hash, RSA Algorithm

8. Referential Sources:

1. William Stallings, "Cryptography and Network security Principles and Practices", Pearson/PHI.
2. Wade Trappe, Lawrence C Washington, "Introduction to Cryptography with coding theory", Pearson.
3. W. Mao, "Modern Cryptography – Theory and Practice", Pearson Education.

Useful Web Sources:

1. https://nptel.ac.in/courses/106105031 : Cryptography and Network Security
2. http://practicalcryptography.com/ : Practical Cryptography
3. https://cacr.uwaterloo.ca/ : A Cryptographic Compendium containing technical reports

----

Back to Contents

Artificial Intelligence
(IPCTMJ7403)

1. About the Course: This is a Major Course. At present, Artificial Intelligence is one of the most dominant areas that studies how to realize the intelligent human behaviors on a computer. This course is organized in a series of lectures which includes both theory and tutorial sessions.

2. Course Description:

- Target Audience: Seventh semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (3 + 1 + 0)
- Total Credit Hours: 4
- Total Contact Hours: 60 hours (15 Weeks X 3 hours + 15 weeks x 1 hours)
  - ➢ Lectures: 45 Hours (15 Weeks X 3 Hours)
  - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
  - ➢ Practical: Nil

3. Prerequisites and Dependencies: Students are expected to have good knowledge in algorithm, programming, graph theory for this course.

4. Objective: The main objective of this course is to provide the fundamental knowledge to the students so that they can understand what AI is. Basic principles, techniques and application of Artificial Intelligence are introduced in this course. After completion of the course, students who become interested in AI can go for further advanced study and research.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Gain a historical perspective of AI.
CO2	Become familiar with basic principle of AI towards problem solving, Knowledge representation and planning.
CO3	Will understand different search strategies used for problem solving CO4. Becoming familiar with learning algorithm.

6. Topics:

Module I: Overview (8 hours)
What is Artificial Intelligence? Turing test, history of AI, Intelligent agents, Agent programs, task environment, PEAS, Types of intelligent agent

Module II: Problem solving by searching (12 hours)
problem and goal formation, well defined problem and solution, searching for solution, uninformed search strategies- Breadth First Search , Uniform Cost Search, Depth First Search, Bidirectional Search, Informed search strategies-greedy best first search, A* Search, heuristic function, hill climbing search, genetic algorithm, AND-OR search tree, Adversarial search-the minimax algorithm, alpha-beta pruning, constraint satisfaction problem

Module III: Knowledge Representation (10 hours)
Knowledge based agents, Symbolic Logics- Introduction, Propositional Logics, Syntax and semantics of FOPL, Properties of Well Formed Formulas, Inference Rules, Knowledge representation using rules ,dealing with inconsistence- Default Reasoning and the closed world

assumptions, Predicate Completion, Modal and temporal logics, Fuzzy logic and natural language computations, Probabilistic reasoning- Bayesian probabilistic reasoning, Heuristic reasoning method, Structured knowledge- Associative networks, Frame structure

Module IV: learning (10 hours)
Forms of learning, supervised learning, learning decision tree, unsupervised learning, artificial neural network, support vector machine, reinforcement learning

Module V: Expert System Architecture (10 hours)
Rule-Based System Architecture, Nonproduction System Architecture, Dealing with uncertainty, Knowledge Acquisition and Validation, Knowledge System Building tools

Module VI: Natural Language Processing (10 hours)
Overview of Linguistics, Grammar-level of knowledge used in language understanding, Grammar and languages, Parsing Techniques, Semantic analysis and Representation structures, Natural language generation, Natural language systems

7. Referential Sources:

1. Dan W. Patterson," Artificial Intelligence and Expert System", Prentice Hall E. Rich & K. Knight, "Artificial Intelligence", Tata McGraw Hill.
2. N.J. Nilson, "Principles of Artificial Intelligence", Narosa Pub. House.
3. Sturat J. Russell, Peter Norvig, "Artificial Intelligence, a modern approach", Prentice Hall

----

Research Methodology
(IPCTMJ7403)

1. About the Course: This is a Major Course. This course will help students to gain insight how scientific research is conducted. This course is organized in a series of lectures which includes both theory and tutorial sessions.

2. Course Description:

- Target Audience: Seventh semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (3 + 1 + 0)
- Total Credit Hours: 4
- Total Contact Hours: 60 hours (15 Weeks X 3 hours + 15 weeks x 1 hours)
  - ➢ Lectures: 45 Hours (15 Weeks X 3 Hours)
  - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
  - ➢ Practical: Nil

3. Prerequisites and Dependencies: This course does not have any prerequisite course.

4. Objective: The main objective of this course is to aware students how scientific research is carried out. This course will help students in critical review of literature and assessing the research trends, quality and potential of research and equip students to undertake research.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1 Students will get to know about research methodology and technique of defining a research problem

CO2 Students will know the functions of the literature review in research and carrying out a literature search, its review, developing theoretical and conceptual frameworks and writing a review.

CO3 Students will know about various research designs and their characteristics.

CO4 Students will know the details of sampling designs, measurement and scaling techniques and also different methods of data collections.

CO5 Students will know several parametric tests of hypotheses and Chi-square test.

6. Topics:

Module I: Introduction (10 hours)

Meaning of Research, Objectives of Research, Motivation in Research, Types of Research, Research Approaches, Significance of Research, Research Methods versus Methodology, Research and Scientific Method, Importance of Knowing How Research is Done, Research Process, Criteria of Good Research, Defining research problem and technique involved, Meaning and need for research design, Feature of a good design, Different research designs Basic Principles of Experimental Designs , Reviewing the literature: Place of the literature review in research, Bringing clarity and focus to research problem, Improving research methodology, Broadening knowledge base in research area, Enabling contextual findings, Review of the literature, searching the existing literature, reviewing the selected literature, Developing a theoretical framework, Developing a conceptual framework, Writing about the literature reviewed.

Module II: Data collection (10 hours)
Census and Sample Survey, Implications of a Sample Design, Steps in Sampling Design Criteria of Selecting a Sampling Procedure, Characteristics of a Good Sample Design, Different Types of Sample Designs, How to Select a Random Sample?, Random Sample from an Infinite Universe ,Complex Random Sampling Designs, Measurement in Research, Measurement Scales, Sources of Error in Measurement, Technique of Developing Measurement Tools , Scaling, Meaning of Scaling, Scale Classification Bases, Important Scaling Techniques, Scale Construction Techniques , Methods of Data Collection, Collection of Primary Data, Observation Method, Interview Method, Collection of Data through Questionnaires, Collection of Data through Schedules , Difference between Questionnaires and Schedules , Some Other Methods of Data Collection , Collection of Secondary Data 11

Module III: Processing and analysis of data (10 hours)
Processing Operations, Some Problems in Processing, Elements/Types of Analysis , Statistics in Research, Measures of Central Tendency, Measures of Dispersion, Measures of Asymmetry (Skewness), Measures of Relationship, Simple Regression Analysis, Multiple Correlation and Regression, Partial Correlation, Association in Case of Attributes, Other Measures

Module IV: Hypothesis testing (10 hours)
Testing of Hypotheses: Hypothesis, Basic Concepts Concerning Testing of Hypotheses, Testing of Hypothesis, Test Statistics and Critical Region, Critical Value and Decision Rule, Procedure for Hypothesis Testing, Hypothesis Testing for Mean, Proportion, Variance, for Difference of Two Mean, for Difference of Two Proportions, for Difference of Two Variances, P-Value approach, Power of Test, Limitations of the Tests of Hypothesis. Chi-square Test: Test of Difference of more than Two Proportions, Test of Independence of Attributes, Test of Goodness of Fit, Cautions in Using Chi Square Tests, ANOVA, Multivariate Analysis Technique

Module V: Interpretation and Report Writing (5 hours)
Meaning of Interpretation, Technique of Interpretation, Precaution in Interpretation, Significance of Report Writing, Different Steps in Writing Report, Layout of the Research Report 348 Types of Reports, Oral Presentation, Mechanics of Writing a Research Report, Precautions for Writing Research Reports, Ethics and plagaiarism, How to write a Research Paper

7. Referential Source:

1. C R Kothari, Gaurav Garg, “Research Methodology Methods and Techniques”, New Age International Publishers.
2. Trochim, “Research Methods: the concise knowledge base”, Atomic Dog Publishing 2005
3. Fink, “Conducting Research Literature Reviews: From the Internet to Paper” Sage Publications 2009

Back to Contents

Machine Learning
(IPCTMJ8401)

1. About the Course: This is a Major Course. It is an introductory level PG course and is aimed to make students familiar with the key algorithms and theory that form the core of machine learning.

2. Course Description:

- Target Audience: Eighth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
  - Lectures: 30 Hours (15 Weeks X 2 Hours)
  - Tutorial: 15 hours (15 Weeks X 1 Hour)
  - Practical: 30 Hours (15 Weeks X 2 Hours)

3. Prerequisites and Dependencies: Students are expected to have basic knowledge on probability, statistics, algorithm and computer programming for registration in this course. Therefore all "Mathematics" courses, "Introductory computing using C" are considered to be prerequisite courses. For practical session, students are expected to have knowledge on python programming.

4. Objective: The primary objective of this course is to give students a basic understanding to machine learning and to study and construct of computer algorithms that improves automatically through experience.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Understand a wide variety of learning algorithms.
CO2	Understand how to evaluate model generated from data.
CO3	Understand the difference between supervised and unsupervised learning.
CO4	To develop skills of using recent machine learning software tools to evaluate learning algorithms and model selection for solving practical problems.

6. Topics:

Module I: Introduction (4 hours)
Well-Posed Learning Problems, Definition of learning system, designing a learning system training data, concept representation, function approximation, Perspective and issues in machine learning, supervised and unsupervised learning

Module II: Concept Learning (6 hours)
Concept learning task, Concept learning as search-general to specific ordering of hypothesis, Finding a Maximally Specific Hypothesis, Version space and the candidate -elimination algorithm, Inductive bias

Module III: Decision Tree Learning (9 hours)
Introduction, Decision tree representation, Decision tree learning algorithm, Hypothesis Space

Search in Decision Tree learning, Inductive bias in decision tree learning, Issues in decision tree learning, Entropy Based Node selection, ID3 Algorithm, Random Forest

Module IV: Artificial Neural Network (8 hours)
Introduction, Neural Network representation, perceptrons, Multilayer and backpropagation algorithms, Convolutional network, Recurrent network

Module V: Bayesian Learning (9 hours)
Introduction, Bayes Theorem, Bayes theorem and concept learning, Maximum likelihood and least -square error hypothesis, MAP Hypothesis, Minimum Description Length (MDL) principle, Bayesian Classifiers Bayes optimal classifier, Naive Bayes classifier, Bayes optimal classifier

Module VI: Linear model and Support Vector Machine (9 hours)
Linear models for classification, Discriminant Functions, Probabilistic Generative Classifiers, Probabilistic Discriminative Classifiers, Linear models for Regression ,Linear basis function models , Bayesian linear regression , Bias-variance decomposition, Theory of SVM, VC dimension, Linearly separable data , Non-linearly separable data

7. Laboratory Sessions: This course contains 2 hours of practical classes per week. Following topics will be covered in the laboratory classes.

- Logistic Regression classifier
- Linear Regression and Gradient Descent
- Decision Tree
- Single layer Backpropagation
- SVM

8. Referential Sources:

1. Tom M. Mitchell,”Machine learning”,McGraw Hill
2. O Theobald, “Machine Learning for Absolute Beginners: A Plain English Introduction”, Scatterplot Press
3. D. Barber,” Bayseian Reasoning and machine learning”, 2012
4. S. Rogers and M. Girolami, “A first course in Machine Learning”, CRC Press, 2011

Image Processing
(IPCTMJ8402)

1. About the Course: This is a Major Course and is aimed at presenting foundation and advanced concepts of image processing.

2. Course Description:

- Target Audience: Eighth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (3 + 1 + 0)
- Total Credit Hours: 4
- Total Contact Hours: 60 hours (15 Weeks X 3 hours + 15 weeks x 1 hours)
  - ➢ Lectures: 45 Hours (15 Weeks X 3 Hours)
  - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
  - ➢ Practical: Nil

3. Prerequisites and Dependencies: “Computer Graphics” and “Data Structure” courses are prerequisites. The students are expected to have elementary knowledge of basic mathematics.

4. Objective: The course aims to introduce the fundamental and analytical techniques to be employed in image processing, as well as the concept of image processing.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1 Understand the need for image transformation and its properties for different types of image transformation.

CO2 Development of image processing applications.

CO3 Understand the rapid advances in image processing.

CO4 Learn about the various techniques used to enhance images.

CO5 Learn various causes of image degradation and give an overview of image restoration techniques

CO6 Understands need for image compression and to learn spatial and frequency domain techniques of image compression

CO7 Learn about various feature extraction techniques for image analysis and recognition

6. Topics:

Module I: Introduction (8 hours)
Examples of fields that use digital image processing, fundamental steps in digital image processing, components of image processing system., A simple image formation model, image sampling and quantization, basic relationships between pixels.

Module II: Image enhancement in the spatial domain (9 hours)
Basic gray-level transformation, histogram processing, enhancement using arithmetic and logic operators, basic spatial filtering, smoothing and sharpening spatial filters, combining the spatial enhancement methods.

Module III: Image restoration (9 hours)
A model of the image degradation/restoration process, noise models, restoration in the presence

of noise only spatial filtering, Weiner filtering, constrained least squares filtering, geometric transforms; Introduction to the Fourier transform and the frequency domain, estimating the degradation function.

Module IV: Color Image Processing (9 hours)
Color fundamentals, color models, pseudo color image processing, basics of full color image processing, color transforms, smoothing and sharpening, color segmentation

Module V: Image Compression (9 hours)
Fundamentals, image compression models, error-free compression, lossy predictive coding, image compression standards

Module VI: Morphological Image Processing (8 hours)
Preliminaries, dilation, erosion, open and closing, hit or miss transformation, basic morphologic algorithms

Module VII: Object Recognition (8 hours)
Patterns and patterns classes, recognition based on decision theoretic methods, matching, optimum statistical classifiers, neural networks, structural methods matching shape numbers, string matching

7. Referential Sources:

1. Rafael C. Gonzalez, “Digital Image Processing”.

Useful Web Sources: https://nptel.ac.in/courses/117105079/ : NP-TEL content.

Block Chain Architecture
(IPCTMJ8403)

1. About the Course: This is a Major Course and is targeted to cover both the conceptual as well as application aspects of Blockchain. This includes the fundamental design and architectural primitives of Blockchain, the system and the security aspects, along with various use cases from different application domains.

2. Course Description:

- Target Audience: Eighth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (3 + 1 + 0)
- Total Credit Hours: 4
- Total Contact Hours: 60 hours (15 Weeks X 3 hours + 15 weeks x 1 hours)
  - ➢ Lectures: 45 Hours (15 Weeks X 3 Hours)
  - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
  - ➢ Practical: Nil

3. Prerequisites and Dependencies: The course has dependency with computer networks and distributed systems, Familiarity with cryptography concepts (e.g., encryption, hashing) and knowledge of data structures and algorithms will be necessary. Proficiency in programming languages (e.g., Java, Python) and understanding of database concepts and management systems are added bonus.

4. Objective: The major objective of studying Blockchain Architecture is to develop a comprehensive understanding of the underlying structure and components of blockchain systems, including the consensus mechanisms, cryptographic techniques, and different layers that constitute these networks. This knowledge will enable a deep exploration of smart contracts, decentralized applications (DApps), and various blockchain platforms, while also addressing critical security and privacy considerations.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Proficiency in creating and Knowledge of blockchain technology, its principles, and features
CO2	Understanding of the architecture and components of a blockchain system.
CO3	Ability to design and develop smart contracts on blockchain platforms.
CO4	Skills to evaluate and compare different blockchain platforms
CO5	Familiarity with consensus mechanisms and their implications
CO6	Understanding of blockchain scalability and performance considerations
CO7	Ability to analyze and address security and privacy challenges in blockchain
CO8	Skills to communicate and present blockchain concepts effectively

6. Topics:

Module I: Introduction to Blockchain (12 hours)
Basics, History, Architecture, Conceptualization, Basic Crypto Primitives, eWallet Service and Personal Cryptosecurity, Merchant Acceptance of Bitcoin

Back to Contents

Module II: Contracts (12 hours)
Financial services, crowdfunding, Bitcoin prediction markets, smart property, smart contracts, Blockchain 2.0 protocol, wallet development projects, Blockchain Development platforms and APIs, Blockchain Ecosystem: decentralized storage, communication, and computation, Ethereum: Turing-Complete Virtual Machine, Dapps, DAOs, DACs, DASs

Module III: Justice Applications (12 hours)
Extensibility of blockchain technologyconcepts, fundamental economic principles, distributed censorship-resistant organizational models, NameCoin: decentralized domain name system, challenges in other decentralized DNS services, digital identity verification, blockchain neutrality, digital divide of Bitcoin, Digital Art, Blockchain Attestation Services, Hashing plus time stamping, proof of existence, virtual notary, Bitnotar, and Chronobit monograph, digital asset proof as an automated feature, blockchain government, decentralized governance services, precedentCoin, random sample elections.

Module IV: Efficiency and Coordination Applications (12 hours)
Blockchain Science: Gridcoin, Foldingcoin, Community Supercomputing, Blockchain Genomics 2.0, EMRs on Blockchain, Bitcoin MOOCs and Smart Contract Literacy, Centralization-Decentralization Tension and Equilibrium

Module V: Advanced Concepts (12 hours)
Currency, Token, Tokenizing, Currency Multiplicity, Demurrage currencies, Technical challenges, Business Model challenges, Government regulation, privacy challenges, Consensus Scalability, Bitcoin-NG, Collective Signing, Byzcoin, Algorand,Cross Fault Tolerance, Secured Multi-Party Computation, Blockchain for Bigdata, Blockchain and AI

7. Referential Sources:

1. Melanie Swan, “Blockchain: Blueprint for a new economy”. O’Reilly Publishing
2. Andreas Antonopoulos, “Mastering Bitcoin”. O’Reilly Publishing
3. Blockchain Architecture Design and Use Cases - By Prof. Sandip Chakraborty & Dr. Praveen Jayachandran, IIT Kharagpur and IBM

Useful Web link: NPTEL Link: https://onlinecourses.nptel.ac.in/noc19_cs63/preview

----

Embedded Systems
(IPCTMJ8404)

1. **About the Course**: This is a Major Course and is aimed at introducing students with concepts of Embedded programming, Microcontroller, Microprocessor, KEIL language for 8051 devices.

2. **Course Description**:

- Target Audience: Eighth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
  - ➢ Lectures: 30 Hours (15 Weeks X 2 Hours)
  - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
  - ➢ Practical: 30 Hours (15 Weeks X 2 Hours)

3. **Prerequisites and Dependencies**: Students are expected to have basic knowledge on computers including algorithm and programming. Courses on "Introductory Computing using C" and "Data Structures", "Computer Organization and Architecture" are prerequisites for this course.

4. **Objective**: This course is designed for students to educate in various Embedded Development strategies and to introduce Bus Communication in processors, Input/output interfacing by imparting knowledge on various processor scheduling algorithms.

5. **Course Outcomes**: After course completion, following are the course outcomes.

CO1	The students will be specialized in Embedded System Design.
CO2	Students will learn open-source computer hardware/software platform for building digital interactive devices.

6. **Topics**:

Module I: Introduction to Embedded System Design (4 hours)
Definition of Embedded System, Embedded System Vs General Computing Systems, History, Classification based on generation, complexity etc. Major application areas. Purposes/specific features, recent trends.

Module II: Embedded System Architecture Design (15 hours)
Hardware architecture with block diagram, its different components with functionality. Different types of processors used their trade-offs features Examples of Domain specific embedded systems with examples e.g. working of washing machine, automobile etc. Networking concept in embedded system Different buses used I2C PCI CANetc. Software architecture, Embedded operating system architecture categories of embedded operating system, Application software options with high level and assembly level language and different tools used for software development. Process of creation of ROM image/firmware design Study of some microcontroller/processor 8051 / PIC /AVR /ARM/DSP study of Embedded readymade boards like Arduino, Raspberry Pi, implementing small projects.

Module III: Design (13 hours)
Process of Embedded System Development, Different models, waterfall model, requirement analysis, design tradeoffs, hardware software co design different hardware platforms - single board PC add on cards custom made hardware platforms. communication interfaces RS232 RS422, USB, Infrared, IEEE 1394 firmware Ethernet, IEEE 802.11 Bluetooth Embedded firmware design, creation of ROM image.

Module IV: Programming (6 hours)
Different programming options Assembly High level for Embedded systems. Requirement of Embedded real time Operating Systems its features implementation

Module V: Development and Testing (6 hours)
Testing of Embedded systems, Embedded product development life cycle EDLC and its importance, Latest trends in Embedded industry, Fundamental concept in RT Linux and Navigation Systems.

7. Laboratory Session: The practical component of this course is of one credit, i.e., 2 hours of classes per week. The practical component mainly contains circuit design using Proteus PCB Simulator or any other simulating software and writing C programs with Keil language.

- Introduction to Software/Simulation/Compiler/IDE Intro to C Programming with Keil language

8. Referential Sources:

1. K.V. Shibu, “Introduction to Embedded Systems”, Mc Graw Hill Education
2. Raj Kamal, “Embedded Systems Architecture programming and design”, Tata Mc Graw Hill
3. K.V.K. Prasad, “Embedded Real Time Systems concept design and programming”, Dreamtech
4. Mazidi and Mazidi, “8051 microcontrollers and embedded Systems”.

MINOR COURSES

Basics of Computer Application
(IPCTMN1401)

1. **About the Course**: This is a minor course and is aimed at presenting foundation concepts of a computer, its peripherals and various components like Registers, Arithmetic & Logic Unit, Control Unit and Memory etc. The course is organized as a series of lectures, hands-on exercises using Laboratory sessions on various free and open-source software.

2. **Course Description**:

- Target Audience: First semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
    - ➢ Lectures: 30 Hours (15 Weeks X 2 Hours)
    - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
    - ➢ Practical: 30 Hours (15 Weeks X 2 Hours)

3. **Prerequisites and Dependencies**: Since this course is a first semester course, there is no particular prerequisite. However, the students are expected to have elementary knowledge of basic mathematics and logic.

4. **Objective**: The major objective of this course is to provide students with understandings of how a computer works, what are its various components, different types of peripherals used, concept of hardware and software, types of memories in a computer. Basic usage of some free and open-source software is also a major part of the course.

5. **Course Outcomes**: After course completion, following are the course outcomes.

CO1	Understanding foundation concepts of information and information processing in computer systems.
CO2	Understanding of the basic components of a computer: ALU, CU, Memory, etc.
CO3	Understanding various computer terminologies.
CO4	Differentiation among Hardware, Firmware and Software.

6. **Topics**:

Module I: Introduction (9 hours)
What is computer, Computer characteristics, Computer evolutions and generations, Types of computers, Basic components of computer --Control unit, ALU, Input/output device and memory.

Module II: CPU and Memory Unit (9 hours)
CPU architecture-components of CPU, instruction set, register set, types of processor, Memory architecture-storage criteria, primary vs secondary storage, main memory, Cache memory, Secondary storage, Magnetic tape, Magnetic disks , Optical disks, Flash drive.

Module III: Input and Output Devices (9 hours)

Input devices-Keyboard, Scanner, Digitizer, Touch screen etc , Output devices-Monitors, Printers, Plotters.

Module IV: Software and Operating System (9 hours)
Software-concepts and needs, Types of software-system and application software, Algorithm, Flowchart, Pseudo code, Programming Language-Machine language, Assembly language and High-level language, Assembler, Compiler and Interpreter.

Module V: Computer Network (9 hours)
Data communication, Components of Data Communication System, Network topology, LAN, WAN, Internet, World Wide Web, Introduction to network security

7. Laboratory Sessions: The practical component of this course is of one credit which amounts to 2 hours of Laboratory class per week.

Section 1: Office Package (LibreOffice)

- Introduction to Windows 10 OS Environment, Command Prompt, Navigation using GUI based File Explorer (This PC), Concept of Desktop, Icons, Folders and Files, Introduction to Command Prompt. Hands-on various CMD Prompt commands, Environment Variables in Windows.
- Introduction to Linux (Lubuntu) Environment, Concept of Terminal and Terminal commands.
- Introduction to Spreadsheet software: LibreOffice Calc – Layout, Formula Bar, Cell Address, Font design and formatting, Arithmetic Operators (+, –, * and /), Aggregate functions (Avg, Sum, Max, Min and Truncate, etc.), Relational Operators (>, <, >=, <=, =, !=) Introduction to If .. else statement. Nested If .. else statement, Logical Operators (&, | and !).
- Count If, Introduction to 2D and 3D Graphs – Bar, Pie, Line, Vector, XY Labelling etc.
- Absolute and Reference Cell Addressing, Pivot Table.
- Introduction to Documentation software: LibreOffice Writer – Page Layout, Page Orientation, Page Columns, Font and Paragraph design, Introduction to Numbered and Bullet List and Sub-list, Table design.
- Insertion of Images and shapes, Formatting an Image, Page wrap, Alignment, Insertion of text box. Header and Footer, Page Number, Page Break, Template Design.
- Cover Page design and concept of Mail Merge (using LibreOffice Calc).
- Introduction to Presentation software: LibreOffice Impress – Slide Layout, Master Slide Design, Font and Paragraph Design, Inserting Image and graphs.
- Slide and Custom element Animation. Properties of animation (speed, Event of occurrence, Duration, etc.)

Section 2: Multimedia (GIMP, Audacity & OpenShot)

- Introduction to Image editing software: GNU Image Manipulation Program (GIMP). Environment of GIMP. Concept of Layering, Introduction to various Tools.
- Class exercise on GIMP.
- Introduction to Audio editing software: Audacity, Concept of Monophonic and Stereophonic sound, Concept of channels (L-R), Fading, Combining different Audio tracks.

- Class exercise on Audacity.
- Introduction to Video editing software: OpenShot, Intro to its environment, Concept of tracks, Trimming and Fade effects (Audio and Visual).

8. <u>Referential Sources</u>:

Useful Books and Papers

1. Rajaraman V. "Fundamentals of Computer". PHI Publishing. Sinha P.K. "Foundation of Computing". PHI Publishing.
2. Byron S Gottfried, "Programming With C". McGraw Hill
3. Brian W. Kernighan, Dennis Ritchie, "The C Programming Language". Pearson Education India

----

Back to Contents

Programming in C
(IPCTMN2401)

1. **About the Course**: This is a minor course and is aimed at advancing concepts of programming and software code organization within the framework of structural and procedural programming paradigms. The course is organized as a series of lectures and hands-on laboratory sessions using C programming languages and focusing on discussing how to write a program of moderate complexity by using C language.

2. **Course Description**:

- Target Audience: Second semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
  - ➢ Lectures: 30 Hours (15 Weeks X 2 Hours)
  - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
  - ➢ Practical: 30 Hours (15 Weeks X 2 Hours)

3. **Prerequisites and Dependencies**: There is no prerequisite to this course.

4. **Objective**: The course is oriented to those who want to advance structured and procedural programming understating and to improve C programming skills. The major objective is to provide students with understanding of code organization and functional hierarchical decomposition with using complex data types.

5. **Course Outcomes**: After course completion, following are the course outcomes.

CO1	Understanding a functional hierarchical code organization.
CO2	Ability to work with textual information, characters and strings
CO3	Ability to work with arrays of complex objects
CO4	Understanding a concept of object thinking within the framework of functional model.
CO5	Ability to handle possible errors during program execution.

6. **Topics**:

Module I: Introduction to computer and programming (6 hours)
Computer, block diagram of computer, hardware, software, program, types of software, operating system, compiler, programming languages, types of programming languages, algorithm, pseudocode, flowchart, desirable characteristics of a program, C programming language, history of C, structure of a C program, C character set, identifiers and keywords, writing, compiling and executing a c program, datatypes, constants, escape sequences, string constants, variables and arrays, declarations, expressions, statements, symbolic constants

Module II: Operators and I/O (7 hours)
Arithmetic operators, unary operators, relational and logical operators, assignment operators, conditional operator, data input output, single character input, single character output, data input from user: scanf function, writing output data: printf function, display formatting using printf function, the gets and puts function, error and debugging techniques

Back to Contents

Module III: Control statements (7 hours)
Control statements, branching statement, looping statements: for, while, do-while, switch statement, break statement, continue statement, goto statement

Module IV: Functions and arrays (7 hours)
C functions, defining a function, calling a function, function prototypes. Passing arguments to function, recursion, defining an array, processing an array, passing array to a function, multidimensional array

Module V: Storage classes, strings and pointers (7 hours)
Storage classes, automatic variables, register variables, external variables, static variables, defining and initializing a string, null character, reading and writing a string, library functions for strings, pointers, pointer declaration, passing pointer to function, pointers and one-dimensional array, dynamic memory allocation, operations on pointers, pointers and multidimensional array, arrays of pointers, passing functions to other functions

Module VI: Structures and unions (6 hours)
Structure, defining a structure, processing structure, user defined data types, structure and pointers, passing structures to functions, self-referential structures, union, defining a union, processing union

Module VII: Handling file (5 hours)
Files, opening and closing files, reading and writing a file, processing a file, unformatted files, binary files, random access of files

7. Laboratory Sessions: The practical component of this course is of one credit which amounts to 2 hours of Laboratory class per week.

- Introduction to the concept of Keywords, Datatypes, Operators, Variables, Constants in C, Branching statements.
- Control statements using loops and goto command
- Introduction to Functions, Introduction to Arrays, Introduction to Pointers in C.
- Strings
- Structures and Unions
- File Handling

8. Referential Sources:

1. Byron Gottfried, “Programming with C”. McGraw Hill Education
2. S.K. Srivastava, “C in Depth”. BPB Publications

Useful Web Sources: http://www.cprogramming.com/ : C and C++ Programming

----

Design and Analysis of Algorithms (IPCTMN3401)

1. About the Course: This is a minor course and is aimed to introduce various types of approaches used to write an algorithm as a solution to a problem. This course is designed to enable students to analyze time requirement of an algorithm.

2. Course Description:

- Target Audience: Third semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (3 + 1 + 0)
- Total Credit Hours: 4
- Total Contact Hours: 60 hours (15 Weeks X 3 hours + 15 weeks x 1 hours)
  - ➢ Lectures: 45 Hours (15 Weeks X 3 Hours)
  - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
  - ➢ Practical: Nil

3. Prerequisites and Dependencies: Since this course analyzes an algorithm, the students must be comfortable with programming and algorithm writing.

4. Objective: The major objective of this course is to classify algorithms with an intent to distinguish various approaches to solve a problem. The course also lays out a standard procedure to analyze time requirement of various types of algorithms.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Ability to analyze time requirement of an algorithm using a standard way.
CO2	Understanding various classes of algorithms that can be employed to solve a problem.
CO3	The ability to apply knowledge of computing and mathematics to algorithm design.

6. Topics:

Module I: Introduction (10 hours)

The role of algorithms in computing, Algorithm as a technology, Growth of functionsAsymptotic notation and properties, Recurrence-the substitution method, the recursiontree method, the master method, Randomized algorithm, Stacks and queues, Linked lists, Implementing pointers and objects, Representing rooted trees, Hash tables, Hash functions, Binary Search Trees

Module II: Some Important Algorithms (20 hours)

Sorting algorithm- Bubble sort, Selection sort, Insertion sort, Merge sort, Quick sort, Heap sort, Sorting in linear time- Radix sort, Bucket sort, Graph algorithms- Representation of Graphs, Breadth-first Search, Depth-first Search, Topological sort, Strongly connected components, Minimum spanning tree-Kruskal and Prim algorithm, Single-Source Shortest Paths- The Bellman-Ford algorithm, Dijkstra's algorithm, All-pairs Shortest Paths- The FloydWarshall algorithm, Number- Theoretic Algorithm- greatest common divisor, Modular arithmetic, The Chinese-remainder theorem, The RSA public-key cryptosystem

Module III: Advanced Data Structure (10 hours)

B-Tree, Binomial Heaps, Fibonacci Heaps, Data structure For Disjoint sets

Module IV: Advanced Design and Analysis Techniques (20 hours)
Dynamic Programming- Matrix-chain multiplication, Elements of dynamic programming, Longest common subsequence, Optimal binary search trees, Greedy algorithm- An activity selection problem, Elements of the greedy strategy, Huffman codes, Amortized analysis-The potential method, Dynamic table, Linear Programming- formulating problem as linear problem, the simplex algorithm, Representations of polynomials, The DFT and FET, NPcompleteness problems, Approximation Algorithms-The vertex-cover problem, The travelling-salesman problem, The set-covering problem, The subset-sum problem

7. Referential Sources:

1. C Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, “Introduction to Algorithms”. McGraw Hill.
2. A. Aho, J. Hopcroft, J. Ullman, “The Design and Analysis of Algorithms”. Addison Wesley.

Database Management System (IPCTMN4401)

1. About the Course: This is a minor course that deals with designing, maintenance and transaction of database systems. This course is organized as a series of lectures with both theory and laboratory sessions. This course covers basic database concepts, data models, database architecture, relational database languages, SQL, functional dependencies and normalization, and database transactions.

2. Course Description:

- Target Audience: Fourth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
  - ➢ Lectures: 30 Hours (15 Weeks X 2 Hours)
  - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
  - ➢ Practical: 30 Hours (15 Weeks X 2 Hours)

3. Prerequisites and Dependencies: Students must have fundamental knowledge of computer including software, basic programming and discrete mathematics.

4. Objective: The main objective of the course is to equip students with the skills of database design. The principles and techniques involved in designing a productive and good database from conceptual level to implementation level are covered in this course. The course also addresses issues of database transaction and error recovery.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Will have a broad understanding of database concept and DBMS software.
CO2	Will be able to design a database for an application software, at conceptual level using ER modelling tool and to convert that database into implementation level using Relational model.
CO3	Will be able to model a good normalized database to remove redundant data.
CO4	Will be able to write SQL commands to work with any database
CO5	Will have an understanding on issues involved in database transaction and error recovery.

6. Topics:

Module I: Introduction to Databases (10 hours)
Database, characteristics of database approach, advantages of DBMS, database models, database architecture and data independence, database languages, classification of DBMSs

Module II: Entity Relationship Model (7 hours)
Database design and ER Model: overview, ER Model, Constraints, ER Diagrams, ERD Issues, weak entity sets, subclasses, superclasses, and inheritance, specialization and generalization

Module III: Relational Data Models and SQL (9 hours)
Relational model concept, relational model constraints, relational database schemas, Codd's rules, ER to relational model mapping, SQL data definition and data types, specifying

constraints in SQL, retrieval queries in SQL, insert, delete and update statements in SQL, assertions, triggers, views, schema change statements.

Module IV: Relational Algebra and Calculus (8 hours)
Unary relational operations: SELECT and PROJECT, relational algebra operations from set theory, binary relational operations: JOIN and DIVISION, tuple relational calculus, domain relational calculus

Module V: Dependencies and Normal Forms (10 hours)
Importance of a good schema design, motivation for normal forms, dependency theory functional dependencies, Armstrong's axioms for FD's, closure of a set of FD's, minimal covers, definitions of 1NF, 2NF, 3NF and BCNF, decompositions and desirable properties of them, algorithms for 3NF and BCNF normalization, multi-valued dependencies and 4NF, join dependencies and definition of 5NF.

Module VI: Transaction Processing and Error Recovery (8 hours)
Concepts of transaction processing, ACID properties, concurrency control, locking based protocols for CC, error recovery and logging, undo, redo, undo-redo logging and recovery methods.

7. Laboratory Sessions: The practical component of this course is of one credit, i.e., 2 hours of lab classes per week. The laboratory work consists of the following.

- Introduction to DBMS software and SQL.
- Introduction to Oracle software
- DDL to create, change schema of database and relation and to grant access right.
- SQL data type.
- Insert, update, delete and retrieval queries in SQL.
- Specifying primary and foreign key and other integrity constraints.
- Nested SQL queries and joining of tables.
- Aggregate functions.
- SQL to create views.
- PL/SQL subprograms.
- Writing triggers.

8. Referential Sources:

1. Ramez Elmasri, Shamkant B. Navathe, “Fundamentals of Database Systems”.
2. Avi Silberschatz, Henry Korth, S. Sudarshan, “Database System Concepts”.

R Programming
(IPCTMN4402)

1. **About the Course**: This is a minor course that delivers the skills to write an efficient program using R language.

2. **Course Description**:

- Target Audience: Fourth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks x 2 hours)
  - ➢ Lectures: 30 Hours (15 Weeks X 2 Hours)
  - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
  - ➢ Practical: 30 Hours (15 Weeks X 2 Hours)

3. **Prerequisites and Dependencies**: The course does not have any prerequisite however, prior knowledge of basic programming concepts (variables, data types, control structures), Fundamental knowledge of statistics and mathematics would help.

4. **Objective**: The major objective of R Programming is to develop a comprehensive understanding of the R programming language, enabling learners to proficiently manipulate and analyze data, apply statistical techniques, create insightful visualizations, and build predictive models using R's powerful tools and libraries.

5. **Course Outcomes**: After course completion, following are the course outcomes.

CO1	Ability to write R programs to solve complex problems.
CO2	Competence in data manipulation, cleaning, and transformation using R.
CO3	Proficiency in statistical analysis and hypothesis testing with R.
CO4	Skill in creating informative and visually appealing data visualizations
CO5	Understanding of machine learning concepts and their implementation in R.
CO6	Practical experience in applying R to real-world data analysis projects
CO7	Ability to write custom functions and packages in R

6. **Topics**:

Module I: Introduction (10 hours)
A brief introduction to R, Attributes, A very brief introduction to OOP in R, Some special values, Types of objects, Sequence generating and vector sub-setting, Types of functions, Data structures, Atomic vectors, Numerical computing, Factors, Lists, environments and data frames, Operators, Vector and matrix subsetting, Replacement functions, Functional programming, Writing functions, Flow control, Exception handling,Function evaluation, Indirect function invocation, Evaluation on exit, Namespaces, Function, Graphics

Module II: Object-Oriented Programming in R (6 hours)
The basics of OOP, Inheritance, Dispatch, Abstract data types, Self-describing data, Implicit classes, S3 generic functions and methods, S3 replacement methods, Classes and its types, Class unions, Accessor functions, Using S3 classes with S4 classes, Using classes and methods in packages, Documentation, Debugging, Mixing S3 and S4 methods

Module III: Input and Output in R (6 hours)
Basic file handling, Viewing files, File manipulation, Working with R's binary format, Connections, Text connections, Inter-process communications, Seek, File input and output, Reading rectangular data, Writing data, Debian Control Format (DCF), FASTA Format, Source and sink: capturing R output

Module IV: Working with Character Data (6 hours)
Built-in capabilities: Modifying text, Sorting and comparing, Matching a set of alternatives, Formatting text and numbers, Special characters and escaping, Parsing and de-parsing, Plotting with text, Locale and font encoding, Regular expressions, Prefixes, suffixes and substrings, Matching patterns

Module V: Foreign Language Interfaces (7 hours)
Calling C and FORTRAN from R,Using .Call and .External, Writing C code to interface with R, Registering routines, Dealing with special values, Single precision, Matrices and arrays, Allowing interrupts, R internals, Using the R API, Loading libraries, Inspecting DLLs

Module VI: R Packages (7 hours)
The search path, Package information, Data and demos, Vignettes, biocViews, Managing Libraries, The DESCRIPTION file, Documentation, Namespaces, Event hooks

Module VII: Data Technologies (7 hours)
Using R for data manipulation, Aggregation and creating tables, Apply functions, Database technologies, DBI, SQLite, Simple XPath, The XML package, Handlers, DOM parsing, XML event parsing, Parsing HTML

7. Laboratory Sessions: The practical component of this course is of one credit which amounts to 2 hours of Laboratory classes per week. Following are the components that will be covered in the practical sessions.

- Introduction to R Studio and other IDEs
- Basic R programs with input and output statements
- R programs to handle variables, assignments and lists
- R programs with branching statements
- R programs with looping statements
- R programs with array
- Writing & Calling functions, conditions and loops in R
- Exceptions, Timingsand Visibility in R programs
- Basic Data Visualization

8. Referential Sources:

1. Robert Gentleman, "R Programming for Bioinformatics". CRC Press
2. Tilman M. Davies, "The Book of R". No Starch Press
3. Pierre Lafaye de Micheaux, Rémy Drouilhet, Benoit Liquet,"The R Software: Fundamentals of Programming and Statistical Analysis". Springer

NPTEL Link https://onlinecourses.nptel.ac.in/noc19_ma33/preview Introduction to R Software - By Prof. Shalabh, IIT Kanpur

----

Graph Theory
(IPCTMN5401)

1. About the Course: This is a minor course. Graph theory is the study of graphs, which are mathematical structures used to represent connections between different pair of objects in a universe of discourse. Graph theory has wide range of applications in many areas including computer science. This course is organized as a series of lectures with both theory and tutorial sessions.

2. Course Description:

- Target Audience: Fifth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (3 + 1 + 0)
- Total Credit Hours: 4
- Total Contact Hours: 60 hours (15 Weeks X 3 hours + 15 weeks X 1 hours)
  - ➢ Lectures: 45 Hours (15 Weeks X 3 Hours)
  - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
  - ➢ Practical: Nil

3. Prerequisites and Dependencies: Students with good mathematical and reasoning skill can opt for the course.

4. Objective: The main objective of the course is to provide students with a broad understanding of different graph theoretic notion and its problem-solving nature. Graph theoretical concepts are widely used in computer science to study and solve different problems. Different graph theoretic algorithms such as finding shortest path between pair of nodes, minimum spanning tress and graph traversal are also included in the syllabus to demonstrate students the importance of graph theory.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Students will understand different graph theoretic notion.
CO2	Students will be able to model and solve problem using graph theory.
CO3	Students will be able to apply different graph theoretic algorithms in applications.

6. Topics:

Module I: Basic concepts of Graph (10 hours)
Definition of Graph, Application of Graph, Incidence, Degree, Isolated and Pendant Vertex, Finite and Infinite Graphs, Null Graph, Isomorphism, Subgraphs, Walk, Paths, Circuits, Connected and Disconnected Graph, Components, Euler Graphs, Operation on Graphs, Hamiltonian Paths and Circuits, The Travelling Salesman Problem, Cut-Sets and their properties, Cut-Vertices, Fundamental circuits, Connectivity, 1-Isomorphism and 2Isomorphism

Module II: Trees (9 hours)
Definition of Trees, Properties of Trees, Distance and Centers, Rooted and Binary Trees, On Counting Trees, Spanning Trees, Finding all Spanning Trees of a Graph, Spanning Trees in a Weighted Graph

Module III: Planar and Dual Graphs (8 hours)
Planar Graphs, Kuratowski's Graph, Different Representation of a Planar Graph, Detection of Planarity, Geometric Dual, Combinational Dual

Module IV: Graph Representation (8 hours)
Matrix representation of Graphs, Incidence Matrix, Circuit Matrix, Fundamental Circuit Matrix and Rank, Cut-Set Matrix, Path matrix, Adjacency Matrix, Relationship between Fundamental Circuit Matrix, Fundamental Incidence Matrix and Fundamental Circuit Matrix

Module V: Coloring, Covering and Partitioning (8 hours)
Chromatic number, Chromatic Partitioning, Chromatic Polynomial, Matching, Covering, The Four-Color Problem

Module VI: Directed Graph (8 hours)
Directed Graph, Types of Digraph, Directed Paths and Connectedness, Euler Digraphs, Trees with Directed Edge, Fundamental Circuits in Digraphs, Adjacency Matrix of a Digraph, Paired Comparisons and Tournaments, Acyclic Digraphs and Decyclization

Module VII: Graph Theoretic Algorithm (8 hours)
Computer Representation of Graphs, Some Basic Algorithm: Connectedness and Components, A Spanning Tree, A set of Fundamental Circuits, Cut Vertices, Shortest Path Algorithm, Depth-First Search on a Graph, Graph-Theoretic Computer Language

7. Referential Sources:

1. Narsingh Deo, "Graph Theory with Application to Engineering and Computer Science". PHI Publishing.
2. Richard J. Trudeau, "Introduction to Graph Theory". Dover Publications.

Web Technology
(IPCTMN6401)

1. About the Course: This is a minor course and is aimed at teaching skills to design interactive and dynamic web sites. The course is designed to deliver key technology components like descriptive languages and server-side program elements. In addition, the course gives specific contents that are beneficial for developing web-based solutions like communication with a relational database, data security principles and approaches. The focus of this course would be on advanced topics in emerging Web technologies. These include extensions of Web standards, combination of different Web technologies, Web toolkits and development environments, current backend Web frameworks.

2. Course Description:

- Target Audience: Sixth semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks X 2 hours)
  - ➢ Lectures: 30 Hours (15 Weeks X 2 Hours)
  - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
  - ➢ Practical: 30 Hours (15 Weeks X 2 Hours)

3. Prerequisites and Dependencies: The course is built up on the knowledge of previous courses such as the “Introductory Computing using C”.

4. Objective: The main objective of this course is to provide knowledge on web architecture, web services, server-side scripting technologies to focus on the development of web-based information systems and web services.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1 Define the fundamental ideas and standards underlying Web Service Technology

CO2 Define the fundamental principles for cloud applications

CO3 Discuss concepts at the frontier of industrial practice and emerging standards.

CO4 Differentiate the major frameworks allowing to develop web services and cloud applications and assess their suitability for specific usage scenarios.

6. Topics:

Module I: Introduction (3 hours)
Computer and network, Intranet vs Internet, Client-Server Computing, IP address, Internet services, Hyper Text Transfer Protocol (HTTP), HTTP transaction-persistent vs nonpersistent, Ports and sockets, Proxy Server.

Module II: World Wide Web (4 hours)
Architecture-client server model, Thin client, Fat Client, Uniform Resource Locator(URL), Domain Name Service(DNS), Address resolution, Name resolution, Web documents-static

document, dynamic document and active documents, Cookies, Virtual hosting, Browser, Browser architecture, HTTP request and response.

Module III: Markup Language (4 hours)
Markup language, SGML, HTML, HTML tags and attributes, Cascading Style Sheet (CSS).

Module IV: Web Programming (3 hours)
Scripting language, Client Side Scripting Language and Server Side Scripting Language, Writing JavaScript.

Module V: Introduction to PHP (7 hours)
PHP Functionalities, Datatypes, Variables, Constants, Arrays, Functions, Strings, System Calls, Explode-Implode and other native functions.

Module VI: Core PHP Concepts (8 hours)
Handling Html Form with PHP, Working with file and Directories, Sessions and Cookies, Database connectivity using MySQL and MySQLi, Concept of PDO, Performing basic database Operation using DML commands (such as Insert, Delete, Update, Select), Setting query parameter, Executing query Joins (Cross joins, Inner joins, Outer Joins, Self joins.), Exception Handling. Introduction to Wordpress CMS.

Module VII: Introduction to Laravel (8 hours)
Installation, Artisan CLI, Laravel directory structure, Basic routing, Call a controller method from a route, Template inheritance, Blade conditional statements, Blade Loops, Executing PHP functions in blade, Introduction to Migrations, Migration structure, Creating a basic controller, Creating a route using a closure, Eloquent ORM Models, Eloquent ORM INSERT, READ, UPDATE, DELETE, Using models in controllers.

Module VII: Introduction to NodeJS and ExpressJS (8 hours)
Installation procedures, NodeJS console, REPL, TLS/SSL, Debugger, Process, Child Process, Buffers, Streams, File System, Path, Query String, Assertions, Callbacks, Events, TTY, Webmodules, Database connectivity, Intro to ExpressJS, Express.js fundamental concepts like Routing and HTTP Methods, Middleware, Cookies, REST API, Scaffolding, Templating and Error Handling.

7. Laboratory Sessions: This course contains 2 hours of practical classes per week. Following topics will be covered in the laboratory classes.

- Introduction to Server-side scripting technology using PHP.
- Core PHP concepts.
- Introduction to Laravel framework.
- Concept on CMS using Wordpress/Drupal.
- Introduction to NodeJS.
- Introduction to ExpressJS.

8. Referential Sources:

1. Papazoglou, “Web Services: Principles and Technology (2nd edition)”; ISBN: 978-027373216-7, Prentice Hall, 2012
2. Cerami, “Web Services Essentials”; ISBN: 0596002246, O’Reilly,2002

Useful Web Sources: https://www.php.net/docs.php : PHP Documentation

Artificial Intelligence
(IPCTMN7401)

1. **About the Course**: This is a minor course. At present, Artificial Intelligence is one of the most dominant areas that studies how to realize the intelligent human behaviors on a computer. This course is organized in a series of lectures which includes both theory and tutorial sessions.

2. **Course Description**:

- Target Audience: Seventh semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (3 + 1 + 0)
- Total Credit Hours: 4
- Total Contact Hours: 60 hours (15 Weeks X 3 hours + 15 weeks X 1 hours)
    - ➢ Lectures: 45 Hours (15 Weeks X 3 Hours)
    - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
    - ➢ Practical: Nil

3. **Prerequisites and Dependencies**: Students are expected to have good knowledge in algorithm for this course.

4. **Objective**: The main objective of this course is to provide the fundamental knowledge to the students so that they can understand what AI is. Basic principles, techniques and application of Artificial Intelligence are introduced in this course. After completion of the course, students who become interested in AI can go for further advanced study and research.

5. **Course Outcomes**: After course completion, following are the course outcomes.

CO1	Gain a historical perspective of AI.
CO2	Become familiar with basic principle of AI towards problem solving, Knowledge representation and planning.
CO3	Will understand different search strategies used for problem solving CO4. Becoming familiar with learning algorithm.

6. **Topics**:

Module I: Overview (12 hours)
What is Artificial Intelligence? Turing test, history of AI, Intelligent agents, Agent programs, task environment, PEAS, Types of intelligent agent

Module II: Problem solving by searching (16 hours)
problem and goal formation, well defined problem and solution, searching for solution, uninformed search strategies- Breadth First Search , Uniform Cost Search, Depth First Search, Bidirectional Search, Informed search strategies-greedy best first search, A* Search, heuristic function, hill climbing search, genetic algorithm, AND-OR search tree, Adversarial search- the minimax algorithm, alpha-beta pruning, constraint satisfaction problem

Module III: Knowledge Representation (16 hours)
Knowledge based agents, Symbolic Logics- Introduction, Propositional Logics, Syntax and semantics of FOPL, Properties of Well Formed Formulas, Inference Rules, Knowledge

representation using rules ,dealing with inconsistence- Default Reasoning and the closed world assumptions, Predicate Completion, Modal and temporal logics, Fuzzy logic and natural language computations, Probabilistic reasoning- Bayesian probabilistic reasoning, Heuristic reasoning method, Structured knowledge- Associative networks, Frame structure

Module IV: learning (16 hours)
Forms of learning, supervised learning, learning decision tree, unsupervised learning, artificial neural network, support vector machine, reinforcement learning

7. Referential Sources:

1. Dan W. Patterson," Artificial Intelligence and Expert System", Prentice Hall.
2. E. Rich & K. Knight, "Artificial Intelligence", Tata McGraw Hill.
3. N.J. Nilson, "Principles of Artificial Intelligence", Narosa Pub. House.
4. Sturat J. Russell, Peter Norvig, "Artificial Intelligence, a modern approach", Prentice Hall.

Block Chain Architecture
(IPCTMN8401)

1. About the Course: This is a minor course and is aimed at students who wants to learn a new and emerging technology in the field of computer science. Blockchain saw tremendous growth in the past few years, but there is still a gap present in Enterprise adoption of Blockchain. This gap arises due to differences between the Blockchain solution from the traditional software solution. This course has been designed in such a way that one can get a thorough understanding of how to design Enterprise Architecture with regards to different business and technology considerations.

2. Course Description:

- Target Audience: Seventh semester students of Integrated MCA programme
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (3 + 1 + 0)
- Total Credit Hours: 4
- Total Contact Hours: 60 hours (15 Weeks X 3 hours + 15 weeks X 1 hours)
    - ➢ Lectures: 45 Hours (15 Weeks X 3 Hours)
    - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
    - ➢ Practical: Nil

3. Prerequisites and Dependencies: This course does not have any prerequisite course.

4. Objective: The objective of this course is to provide conceptual understanding of block chain technology and how it can be used in Industry. The course covers the technological underpinning of block Chain operations in both theoretical and practical implementation of solutions using Ethereum.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Understanding block chain technology.
CO2	Understand Cryptocurrency.
CO3	Understand Smart contract.
CO4	Use of Remix iDE
CO5	Develop block-chain based solutions and write smart contract using Ethereum Framework
CO6	Deploy Decentralized Application

6. Topics:

Module I: Introduction (10 hours)
Introduction: Overview of Block chain, History of Blockchain, Peer to Peer Network, Smart Contract, Wallet, Digital Currency, Ledgers, Types of Blockchain Platform.

Module II: Consensus Mechanism (10 hours)
Permissioned Blockchain, Permissionless Blockchain , Different Consensus Mechanism, Proof of Work, Proof of Stake, Proof of Activity, Proof of Burn, Proof of Elapsed Time, Proof of Authority, Proof of Importance.

Module III: Cryptocurrency and Wallet (10 hours)
Types of Wallet, Desktop Wallet, App based Wallet, Browser based wallet, Metamask, Creating an account in Metamask, Use of faucet to fund wallet, transfer of cryptocurrency in metamask.

Module IV: Smart Contract and Ethereum (10 hours)
Overview of Ethereum, Writing Smart Contract in Solidity, Remix IDE , Different networks of ethereum, understanding blocks practically at blockhcain.com, how to compile and deploy smart contract in remix.

Module V: Understanding Hyperledger Fabric (10 hours)
Overview of Open source Hyperledger project, Hyperledger Fabric- Architecture, Identities and Policies, Membership and Access Control, Channels, Transaction Validation, Writing smart contract using Hyperledger Fabric.

Module VI: Use Cases (10 hours)
Enterprise application of Block chain: Cross border payments, Know Your Customer (KYC), Food Security, Block chain enabled Trade, We Trade – Trade Finance Network, Supply Chain Financing, Identity on Block chain, Blockchain in energy sector, Blockchain in governance.

7. Referential Sources:

1. Melanie Swan, “Blockchain: Blueprint for a New Economy”.
2. Imran Bashier, “Mastering Blockchain: Deeper insights into decentralization, cryptography, Bitcoin, and popular Blockchain frameworks”
3. Andreas M. Antonopoulos, “Mastering Ethereum: Building Smart Contracts and DApps”. O’Reilly Publications.

Back to Contents

MULTIDISCIPLINARY COURSES

Introduction to R Programming (IPCTMD1401)

1. **About the Course**: This is a multidisciplinary course that delivers the skills to write an efficient program using R language.

2. **Course Description**:
   - Target Audience: First semester students of any Integrated programme only
   - Course Period: One semester (15 Weeks)
   - Total Credit (L + T + P): 4 (2 + 1 + 1)
   - Total Credit Hours: 5
   - Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks X 2 hours)
     - ➢ Lectures: 30 Hours (15 Weeks X 2 Hours)
     - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
     - ➢ Practical: 30 Hours (15 Weeks X 2 Hours)

3. **Prerequisites and Dependencies**: The course does not have any prerequisite however, prior knowledge of basic programming concepts (variables, data types, control structures), Fundamental knowledge of statistics and mathematics would help.

4. **Objective**: The major objective of R Programming is to develop a comprehensive understanding of the R programming language, enabling learners to proficiently manipulate and analyze data, apply statistical techniques, create insightful visualizations, and build predictive models using R's powerful tools and libraries.

5. **Course Outcomes**: After course completion, following are the course outcomes.

CO1	Ability to write R programs to solve complex problems.
CO2	Competence in data manipulation, cleaning, and transformation using R.
CO3	Proficiency in statistical analysis and hypothesis testing with R.
CO4	Skill in creating informative and visually appealing data visualizations
CO5	Understanding of machine learning concepts and their implementation in R.
CO6	Practical experience in applying R to real-world data analysis projects
CO7	Ability to write custom functions and packages in R

6. **Topics**:

Module I: Introduction (10 hours)
A brief introduction to R, Attributes, A very brief introduction to OOP in R, Some special values, Types of objects, Sequence generating and vector sub-setting, Types of functions, Data structures, Atomic vectors, Numerical computing, Factors, Lists, environments and data frames, Operators, Vector and matrix subsetting, Replacement functions, Functional programming, Writing functions, Flow control, Exception handling,Function evaluation, Indirect function invocation, Evaluation on exit, Namespaces, Function, Graphics

Module II: Object-Oriented Programming in R (6 hours)
The basics of OOP, Inheritance, Dispatch, Abstract data types, Self-describing data, Implicit classes, S3 generic functions and methods, S3 replacement methods, Classes and its types,

Class unions, Accessor functions, Using S3 classes with S4 classes, Using classes and methods in packages, Documentation, Debugging, Mixing S3 and S4 methods

Module III: Input and Output in R (6 hours)
Basic file handling, Viewing files, File manipulation, Working with R's binary format, Connections, Text connections, Inter-process communications, Seek, File input and output, Reading rectangular data, Writing data, Debian Control Format (DCF), FASTA Format, Source and sink: capturing R output

Module IV: Working with Character Data (6 hours)
Built-in capabilities: Modifying text, Sorting and comparing, Matching a set of alternatives, Formatting text and numbers, Special characters and escaping, Parsing and de-parsing, Plotting with text, Locale and font encoding, Regular expressions, Prefixes, suffixes and substrings, Matching patterns

Module V: Foreign Language Interfaces (7 hours)
Calling C and FORTRAN from R,Using .Call and .External, Writing C code to interface with R, Registering routines, Dealing with special values, Single precision, Matrices and arrays, Allowing interrupts, R internals, Using the R API, Loading libraries, Inspecting DLLs

Module VI: R Packages (7 hours)
The search path, Package information, Data and demos, Vignettes, biocViews, Managing Libraries, The DESCRIPTION file, Documentation, Namespaces, Event hooks

Module VII: Data Technologies (7 hours)
Using R for data manipulation, Aggregation and creating tables, Apply functions, Database technologies, DBI, SQLite, Simple XPath, The XML package, Handlers, DOM parsing, XML event parsing, Parsing HTML

7. Laboratory Sessions: The practical component of this course is of one credit which amounts to 2 hours of Laboratory classes per week. Following are the components that will be covered in the practical sessions.

- Introduction to R Studio and other IDEs
- Basic R programs with input and output statements
- R programs to handle variables, assignments and lists
- R programs with branching statements
- R programs with looping statements, array
- Writing & Calling functions, conditions and loops in R
- Exceptions, Timingsand Visibility in R programs
- Basic Data Visualization

8. Referential Sources:

4. Robert Gentleman, "R Programming for Bioinformatics". CRC Press
5. Tilman M. Davies, "The Book of R". No Starch Press
6. Pierre Lafaye de Micheaux, Rémy Drouilhet, Benoit Liquet,"The R Software: Fundamentals of Programming and Statistical Analysis". Springer

NPTEL Link https://onlinecourses.nptel.ac.in/noc19_ma33/preview Introduction to R Software - By Prof. Shalabh, IIT Kanpur

----

Introduction to Python Programming (IPCTMD2401)

1. **About the Course**: This is a multidisciplinary course that focuses on how to practice and culture the art of programming with Python as a language. The course is designed to explore Python's powerful features, making it ideal for writing effective programs.

2. **Course Description**:

- Target Audience: First semester students of any Integrated programme only
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks X 2 hours)
  - Lectures: 30 Hours (15 Weeks X 2 Hours)
  - Tutorial: 15 hours (15 Weeks X 1 Hour)
  - Practical: 30 Hours (15 Weeks X 2 Hours)

3. **Prerequisites and Dependencies**: The course does not have any prerequisite course.

4. **Objective**: This course leads the students from the basics of writing and running Python scripts to more advanced features such as file operations, working with binary data and using the extensive functionality of Python modules. Extra emphasis is placed on features unique to Python, such as tuples, array slices, and output formatting.

5. **Course Outcomes**: After course completion, following are the course outcomes.

CO1	Able to develop Python Scripts.
CO2	Implementing object-oriented concepts.
CO3	Build and package Python modules for reusability.
CO4	File I/O.
CO5	Implementing database and GUI applications.

6. **Topics**:

Module I: Introduction to computer and programming (6 hours)
Computer, block diagram of computer, hardware, software, program, types of software, operating system, compiler, programming languages, types of programming languages, algorithm, pseudocode, flowchart, desirable characteristics of a program

Module II: Introduction to Python (5 hours)
Introduction to Python and IDLE to develop programs. Working with datatypes and variables, operators and expressions, working with numeric data, working with string data, Python functions, Boolean expressions, selection structure, iteration structure.

Module III: Working with Data (5 hours)
Mutable and Immutable data type, Lists, indexing and slicing, tuples, dictionaries, references and copies.

Module IV: Program Structure (5 hours)
Assignment statements, blocks and syntax rules, expression statements, branching, multiway branching, looping, decisions, continue and break, nested loops control flow.

Module V: Functions (5 hours)
Defining functions, scope rules, global statements, pass by value vs reference, variable length arguments, argument matching, passing arguments.

Module VI: Modules (5 hours)
Imports and attributes, creating modules, namespaces, reloading, module packages, handling files - file I/O, file scanners, files and directories, file positioning, renaming and deleting files.

Module VII: Classes and Objects (5 hours)
An introduction to classes and objects, defining a class, working with object composition, encapsulation, inheritance, exception handling.

7. Laboratory Sessions: The practical component of this course is of one credit which amounts to 2 hours of Laboratory classes per week. Following are the components that will be covered in practical sessions.

- Introduction to editor, translator, IDLE.
- Writing Python scripts.
- Functions in Python.
- File I/O in Python.
- Data analysis modules in Python.
- GUI based application development in Python.

8. Referential Sources:

1. Brown Martin C., "Python: The Complete Reference". McGraw Hill.
2. Lutz Mark, "Programming Python". O'Reilly.
3. Allen B.Downey," Think Python: How to Think Like a Computer Scientist",O'REILLY

Useful Web Sources: https://nptel.ac.in/courses/106106145/ : NP-TEL content

----

Data Analysis using Python
(IPCTMD3401)

1. **About the Course**: This is a multidisciplinary course and is aimed to teach students how to analyze data using Python language.

2. **Course Description**:
- Target Audience: First semester students of any Integrated programme only
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5
- Total Contact Hours: 75 hours (15 Weeks X 3 hours + 15 weeks X 2 hours)
  - Lectures: 30 Hours (15 Weeks X 2 Hours)
  - Tutorial: 15 hours (15 Weeks X 1 Hour)
  - Practical: 30 Hours (15 Weeks X 2 Hours)

3. **Prerequisites and Dependencies**: "Introduction to Python Programming" is the prerequisite course.

4. **Objective**: The main objective of this course is to teach students how to prepare data for analysis, perform analysis, visualize data and extract some meaning information from data or predict future trends etc.

5. **Course Outcomes**: After course completion, following are the course outcomes.

CO1	How to import Dataset and clean the data.
CO2	How to manipulate dataframe.
CO3	Summarize the data.
CO4	Visualize the data.
CO5	How to build and evaluate model.

6. **Topics**:

Module I: Python Fundamentals (6 hours)
Data structure, control statements, functions, object and classes, exception handling, file handling, Python Package for Data Analysis, Working with Numpy and Pandas

Module II: Data preprocessing (9 hours)
Dataset and different type of attributes, Data quality, Noise, understanding domain and dataset, basic insights from data set, Importance of data preprocessing, Structured and unstructured data, Importing and exporting data, cleaning and preparing of data, Identify and handling of missing value, data formatting and data normalization

Module III: Processing, Summarizing, Visualization (8 hours)
Descriptive statistics, grouping, ANOVA, Correlation, Basic of data visualization and data visualization tools, Seaborn creating and plotting maps,

Module IV: Model Development and evaluation (8 hours)
Linear regression, model evaluation using visualization, R-Squared and MSE for In-Sample evaluation, Prediction and decision making, over-fitting, under-fitting and model selection

7. Laboratory Sessions: The practical component of this course is of two credit which amounts to 2 hours of Laboratory classes per week.

- Writing program in python
- Working with Numpy and Panda Package
- Importing and exporting data set
- Cleaning dataset and handling with missing value
- Visualizing data using Matplotlib
- Scikit-learn installation and working with it

8. Referential Sources:

1. Wes McKinney, “Python for Data Analysis- data wrangling with Pandas, NumPy and Ipython”, O’REILLY
2. Fabio Nelli, “Python Data Analytics- with Pandas, NumPy and Matplotlib”, Apress

----

Back to Contents

SKILL ENHANCEMENT COURSES

Introduction to LaTeX

(IPCTSE1301)

1. About the Course: This is a skill enhancement course about LaTeX typesetting system, its history and development. It is widely used for publishing in many scientific fields like mathematics, statistics, computer science, etc. It is a powerful and open-source system that provides numerous facilities for automating typesetting of the document: i.e. structuring page layout, listing and auto-numbering of sections, tables, figures, generating a table of contents, managing cross-referencing, citing, and indexing. This course introduces the basic concepts of LaTeX and is designed for beginners.

2. Course Description:

- Target Audience: First semester students of any Integrated programme only
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 3 (1 + 1 + 1)
- Total Credit Hours: 4
- Total Contact Hours: 60 hours (15 Weeks X 2 hours + 15 weeks X 2 hours)
    - ➢ Lectures: 15 Hours (15 Weeks X 1 Hour)
    - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
    - ➢ Practical: 30 Hours (15 Weeks X 2 Hours)

3. Prerequisites and Dependencies: The course does not have any prerequisite however; students are expected to have basic computer literacy. Additionally, Knowledge of basic formatting concepts (e.g., fonts, styles), Understanding of mathematical and scientific notation and familiarity with text editors and document processing would help.

4. Objective: The major objective of the "Introduction to LaTeX" course is to equip students with a comprehensive understanding of the LaTeX typesetting system, enabling them to create professional-quality documents, reports, and presentations. Through the course, students will gain in-depth knowledge of LaTeX syntax and structure, learn to effectively typeset mathematical equations and scientific formulas, and become familiar with essential LaTeX packages and their functionalities. Additionally, the course emphasizes best practices for producing well-structured and properly formatted documents, preparing students to leverage LaTeX for various academic and professional purposes.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1 Proficiency in creating and formatting documents using LaTeX

CO2 Skills to typeset mathematical equations and scientific formulas

CO3 Knowledge of using LaTeX for generating tables, figures, and graphics

CO4 Ability to create professional-looking reports, articles, and presentations

CO5 Understanding of using LaTeX for bibliographies and citations

CO6 Experience in working with templates and customizing document layouts

CO7 Familiarity with collaborative writing and version control using LaTeX

CO8 Skills to troubleshoot common issues and errors in LaTeX documents

CO9 Ability to convert LaTeX documents into different formats (e.g., PDF, HTML)

Back to Contents

6. Topics:

Module I: Fundamentals of LaTeX (6 hours)

Document structure, formatting commands, creating: tables, mathematical formulas, theorem, propositions etc., symbols, Graphics in Latex, Pstricks package and figuregraphicdrawing, Creation of table of contents, resource and indexing, Creation of directory of resources with Bibtex.

Module II: Formatting (11 hours)

Understanding logical formatting, Titling document, Exploring the document structure, Understanding LaTeX commands, Understanding LaTeX inputs, Trying out the effect of spaces, line breaks, and empty lines, Commenting source text, Printing out special symbols, Writing special characters, Formatting text – fonts, shapes, and styles, creating own commands

Module III: Designing Pages (5 hours)

Specifying margins, changing line spacing, creating table of contents, understanding page styles, customizing header and footer, inserting page breaks, footnotes

Module IV: Creating Lists (11 hours)

Listing latex packages, bulleted list, numbered list, suspending and continuing lists, compact list

Module V: Creating Tables and Inserting pictures (14 hours)

Lining up font commands, typesetting tables, drawing lines in tables, increasing the row height, beautifying tables, spanning entries over multiple columns, inserting code column wise, spanning entries over multiple rows, using array packages, merging cells using the multirow package, adding a caption to our font table, coloring tables, aligning columns at the decimal point, handling narrow columns, inserting pictures, scaling pictures, choosing the optimal file type, including home package pages, putting images behind the text, understanding float placement options, limiting floating avoiding floating at all, breaking figures and tables into pieces

Module VI: Cross Referencing (14 hours)

Assigning a key, referring to a key, referring to a page, introducing variable references, referring to page ranges, combining cleveref and varioref, referring to labels in other documents

Module VII: Listing Content and References (12 hours)

Adjusting the depth of the TOC, shortening entries, adding entries manually, creating and customizing list of figures, creating a list of diagrams, creating a list of tables, using packages for customization, generating an index, specifying page ranges, using symbols and macros in the index, fine tuning page numbers, designing the index layout, creating a bibliography, using bibliography database with Bibtex, looking at Bibtex entry fields, understanding bibtex entry types, listing references without citing,

Module VIII: Typing Math formulas (12 hours)

Typing math formulas, embedding math expressions, within text numbering equations, adding subscripts and super scripts, Greek letters, writing fractions, extracting roots, changing the font style and size, comparing in line formulas to display formulas, using operators, exploring the wealth of math symbols like arrows, harpoons, variable sized operators, binary operation, symbols, variable size delimiters, etc., writing units, building math structures, creating arrays, writing binomial coefficients, typesetting matrices, putting a symbol above another etc.

7. Referential Sources:

1. Stefan Kottwitz, "Latex Beginner's Guide". PACKT Publishing

SWAYAM Link

- https://onlinecourses.swayam2.ac.in/aic20_sp17/preview : LaTeX & XFig typesetting software - By Prof Kannan Moudgalya, Principal Investigator of Spoken Tutorial Project, IIT Bombay

----

Linux Administration
(IPCTSE2301)

1. About the Course: This is a skill enhancement course and introduces the UNIX/Linux operating system, including: task scheduling and management, memory management, input/output processing, internal and external commands, shell configuration, and shell customization. Explores the use of operating system utilities such as text editors, electronic mail, file management, scripting, and C/C++ compilers.

2. Course Description:

- Target Audience: Second semester students of any Integrated programme only
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 3 (1 + 1 + 1)
- Total Credit Hours: 4
- Total Contact Hours: 60 hours (15 Weeks X 2 hours + 15 weeks X 2 hours)
  - ➢ Lectures: 15 Hours (15 Weeks X 1 Hour)
  - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
  - ➢ Practical: 30 Hours (15 Weeks X 2 Hours)

3. Prerequisites and Dependencies: This course does not have any prerequisite course.

4. Objective: The main objective of this course is to provide knowledge on UNIX/Linux operating system.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1 Students will be able to identify and use UNIX/Linux utilities to create and manage simple file processing operations, organize directory structures with appropriate security, and develop shell scripts to perform more complex tasks.

CO2 Effectively use the UNIX/Linux system to accomplish typical personal, office, technical works.

CO3 Monitor system performance and network activities.

6. Topics:

Module I: Introduction (3 hours)

Basic nature of the operating system functions and components: Process Control, Memory Management, Input/output control, Security, Popular OS's.

Module II: Terminologies (7 hours)

Command line user interface – CUI, Graphical user interface – GUI, Internal and External commands, Utility programs, Pathing, File transfers, Command processor / command interpreter, File protection, File Encryption, Directory structure, root directory, Programming language translators, Server, Daemon(s), Unix file naming conventions: File names, File extensions, Wild carding (globbing), Absolute and relative paths.

Module III: File Types and Editors (5 hours)

Text vs. binary files, Executable vs. non-executable, Character device files, Block device files, Pipes, Sockets, Directories, Text file editing and formatting using: vi, emacs / pico, ed, nl, pr and X window-based WYSIWYG text editors: gedit, leafpad

Module IV: File Processing Operations (5 hours)
Create a file, Access a file using the relative pathname, Access a file using the absolute pathname, Erase or delete a file, Copy a file, Move a file, Cut columns of data from a file, Paste / concatenate files, Rename a file, Create a directory, Display the contents of a directory, Display the user initialization files, Change the working directory, Return to the home directory, Remove a directory, Display the file and/or directory information, Change file/directory permissions, Utilities such as sed and awk, Search files, Search for files by attributes, Sort files

Module V: Shell and C/C++ Programming (5 hours)
Shell initialization files, Aliases, Functions, History mechanism(s) sh, ksh, and bash, Shell variables, Script writing, Script debugging, Script usage, Entering C/C++ programs, Finding syntax errors, Compiling, Source files, Object files

Module VI: Client and Server model in Unix (5 hours)
tcp/ip, udp, Electronic mail services, finger, who / w / users, write / chat, telnet, rlogin, ftp, Web browsers, network administration

7. Laboratory Sessions: The practical component of this course is of one credit which amounts to 2 hours of Laboratory classes per week.

- Introduction to Linux environment.
- Introduction to Terminal and commands.
- Introduction to text based editors like: vi, nl, emacs etc and x-window based editors, gedit, leafpad, etc.
- Shell Programming
- C/C++ Programming
- Network administration in Linux environment

8. Referential Sources:

1. Wale Soyinka, “Linux Administration: A Beginner's Guide”. McGraw Hill.
2. Patrick H. Wood and Stephen G. Kochan, “Unix Shell Programming”. Sams.

Web Sources:

- https://nptel.ac.in/courses/117106113 : NP-TEL content on Linux Basics from IIT Madras.

Back to Contents

Basic Web Technology
(IPCTSE3301)

1. About the Course: This is a skill enhancement course and is aimed at teaching skills to design interactive websites. The course is designed to deliver key technology components like descriptive language and client-side program elements.

2. Course Description:

- Target Audience: Third semester students of any Integrated programme only
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 3 (1 + 1 + 1)
- Total Credit Hours: 4
- Total Contact Hours: 60 hours (15 Weeks X 2 hours + 15 weeks X 2 hours)
    - ➢ Lectures: 15 Hours (15 Weeks X 1 Hour)
    - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
    - ➢ Practical: 30 Hours (15 Weeks X 2 Hours)

3. Prerequisites and Dependencies: This course does not have any prerequisite course.

4. Objective: The main objective of this course is to provide knowledge on web architecture, web services and client-side technologies to focus on the development of web sites.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1 Understanding fundamental web concepts (HTTP, URIs, Web browsers, etc.).

CO2 Acquiring skills to write Client-side scripts.

CO3 Students will acquire knowledge on HTML, CSS, XML and JavaScript.

6. Topics:

Module I: Introduction (3 hours)

Computer and network, Intranet vs Internet, Client-Server Computing, IP address, Internet services, Hyper Text Transfer Protocol(HTTP), HTTP transaction-persistent vs nonpersistent , Ports and sockets, Proxy Server.

Module II: World Wide Web (7 hours)

Architecture-client, server, Uniform Resource Locator(URL), Domain Name Service(DNS), Address resolution, Name resolution, Web documents-static document, dynamic document and active documents, Cookies, Virtual hosting, Browser, Browser architecture, HTTP request and response.

Module III: Markup Language (9 hours)

Markup language, SGML, HTML, HTML tags and attributes, Cascading Style Sheet (CSS).

Module IV: Web Programming (7 hours)

Scripting language, Client Side Scripting Language and Server Side Scripting Language, Writing Java Script.

7. Laboratory Sessions: The practical component of this course is of one credit which amounts to 2 hours of Laboratory classes per week.

- Introduction to HTTP, HTTPS, Web Browsers, URIs.
- Introduction to basic HTML, Class exercise on HTML.
- Introduction to CSS, Internal, Inline and External.
- Client-side scripting language.

8. <u>Referential Sources</u>:

Useful Books and Papers

1. Tanenbaum, “Computer Network”. Pearson India.
2. Powell Thomas A., “HTML & CSS The Complete Reference”. McGraw Hill.

Useful Web Sources: https://nptel.ac.in/courses/106/105/106105084/ : NP-TEL content.

Haskell Programming
(IPCTSE3302)

1. **About the Course**: This is a skill enhancement course that encourages breaking up programming tasks into logical units that can be easily translated into provable correct code. Haskell brings together the best features of functional programming and is increasingly being used in the industry, both for building rapid prototypes and for actual deployment.

2. **Course Description**:

- Target Audience: Third semester students of any Integrated programme only
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 3 (1 + 1 + 1)
- Total Credit Hours: 4
- Total Contact Hours: 60 hours (15 Weeks X 2 hours + 15 weeks X 2 hours)
  - Lectures: 15 Hours (15 Weeks X 1 Hour)
  - Tutorial: 15 hours (15 Weeks X 1 Hour)
  - Practical: 30 Hours (15 Weeks X 2 Hours)

3. **Prerequisites and Dependencies**: The course does not have any prerequisite however, students are expected to have Basic programming concepts and logic, understanding of data types, variables, and control flow. Familiarity with a programming language such as Python or Java will be helpful.

4. **Objective**: The major objective of the Haskell Programming course is to equip students with a comprehensive understanding of the Haskell programming language and functional programming principles. By the end of the course, students will have mastered Haskell syntax and programming constructs, recognized the advantages and benefits of functional programming, become familiar with essential Haskell libraries, and developed the skills to apply efficient programming techniques and best coding practices in Haskell.

5. **Course Outcomes**: After course completion, following are the course outcomes.

CO1	Proficiency in writing Haskell code to solve programming problems
CO2	Skills to design and implement functional programs in Haskell
CO3	Knowledge of type systems and type inference in Haskell
CO4	Ability to use higher-order functions and lambda expressions
CO5	Understanding of list comprehension and lazy evaluation in Haskell
CO6	Experience in working with algebraic data types and pattern matching
CO7	Familiarity with monads and their use in handling side effects
CO8	Ability to create reusable and modular Haskell code

6. **Topics**:

Module I: Introduction to Haskell (5 hours)

About pure functional programming, the Haskell ecosystem, history of Haskell, first steps with GHCi, Declaring the data model, characters, numbers, lists, and strings, creating project with Cabal, creating project with Stack, understanding modules, Defining simple functions. Creating simple functions, specifying the function’s type, returning more than one value, working with data types, pattern matching, records.

Module II: Increasing Code Reuse (4 hours)
Parametric Polymorphism, Functions as parameters, higher-order functions, anonymous functions, module imports, smart constructor and views, folds, lists and predicates, lists containing tuples, list comprehensions.

Module III: Using Containers and Type Classes (5 hours)
Using packages, managing dependencies, building packages, containers: maps, sets, trees, graphs, Ad Hoc polymorphism: Type Classes, Declaring Classes and Instances, Simple Binary Trees, Polymorphic Binary Trees, Binary Trees with Monoidal Cache.

Module IV: Laziness and Infinite Structures (4 hours)
Lazy evaluation model, problems with laziness, pattern matching and laziness, profiling with GHC, strictness annotations.

Module V: More Monads (4 hours)
Returning more than one value, the list monad, a new view over Monads, failures and alternatives, association rules learning, the apriori algorithm, search problems, paths in a graph, the logic monad, monads and Lists Redux, Monad Comprehensions, monad Transformers, monad classes

Module VI: Parallel Haskell: Working in several Cores (4 hours)
Parallelism, Concurrency and distribution, software transactional memory, parallelism the apriori algorithm, producer-consumer queues, atomic transactions, rolling back transactions, concurrent use of resources, single process queues, message queues using AMQP, AMQP in Haskell, The Eval Monad

Module VII: Resource Handling (4 hours)
Dealing with files IO and Conduit, randomness, working with files, reading and writing, handling files, error handling, pure errors, catching exceptions, throwing exceptions, streaming data with Conduit, problems with lazy input/output, introducing conduits, accessing files via conduit, basic networking, binary serialization, comma separated values, Parsing with attoparsec, New Type Classes: Functors, Applicatives, Monads, Alternative, and Traversable, Using JSON.

7. Referential Sources:

1. Simon Marlow, “Parallel and Concurrent Programming in Haskell”. O’Reilly Publishing
2. Alejandro Serrano Mena, “Practical Haskell”. Apress Publications

Useful Web Link

- NPTEL Link: https://onlinecourses.nptel.ac.in/noc19_cs80/preview Introduction to Haskell Programming - By Prof. S.P. Suresh & Prof. Madhavan Mukund, Chennai Mathematical Institute

----

Back to Contents

VALUE ADDED COURSES

Introduction to Artificial Intelligence
(IPCTVA1301)

1. About the Course: This is a value-added Course. At present, Artificial Intelligence is one of the most dominant areas that studies how to realize the intelligent human behaviors on a computer. This course is organized in a series of lectures which includes both theory and tutorial sessions.

2. Course Description:

- Target Audience: First semester students of any Integrated programme only
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 3 (2 + 1 + 0)
- Total Credit Hours: 3
- Total Contact Hours: 45 hours (15 Weeks X 2 hours + 15 weeks X 1 hour)
    - Lectures: 30 Hours (15 Weeks X 2 Hours)
    - Tutorial: 15 hours (15 Weeks X 1 Hour)
    - Practical: Nil

3. Prerequisites and Dependencies: The course does not have any prerequisite.

4. Objective: The main objective of this course is to provide the fundamental knowledge to the students so that they can understand what AI is. Basic principles, techniques and application of Artificial Intelligence are introduced in this course. After completion of the course, students who become interested in AI can go for further advanced study and research.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Gain a historical perspective of AI
CO2	Become familiar with basic principle of AI towards problem solving, Knowledge representation and planning
CO3	Will understand different search strategies used for problem solving CO4. Becoming familiar with learning algorithm

6. Topics:

Module I: Overview (11 hours)

What is Artificial Intelligence? Turing test, history of AI, Intelligent agents, Agent programs, task environment, PEAS, Types of intelligent agent

Module II: Problem solving by searching (12 hours)

problem and goal formation, well defined problem and solution, searching for solution, uninformed search strategies- Breadth First Search, Uniform Cost Search, Depth First Search, Bidirectional Search, Informed search strategies-greedy best first search, A* Search, heuristic function, hill climbing search, genetic algorithm, AND-OR search tree, Adversarial search-the minimax algorithm, alpha-beta pruning, constraint satisfaction problem

Module III: Knowledge Representation (11 hours)

Knowledge based agents, Symbolic Logics- Introduction, Propositional Logics, Syntax and semantics of FOPL, Properties of Well Formed Formulas, Inference Rules, Knowledge representation using rules

Module IV: Learning (11 hours)
Forms of learning, supervised learning, learning decision tree, unsupervised learning, artificial neural network, support vector machine, reinforcement learning

7. Referential Sources:

1. Dan W. Patterson,” Artificial Intelligence and Expert System”, Prentice Hall
2. E. Rich & K. Knight, “Artificial Intelligence”, Tata McGraw Hill.
3. N.J. Nilson, “Principles of Artificial Intelligence”, Narosa Pub. House.
4. Sturat J. Russell, Peter Norvig, “Artificial Intelligence, a modern approach”, Prentice Hall

----

Introduction to Data Mining
(IPCTVA2301)

1. About the Course: This is a value-added Course and is aimed to make students understand the concepts of data mining for extracting useful patterns, information from huge amount of data.

2. Course Description:

- Target Audience: Second semester students of any Integrated programme only
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 3 (2 + 1 + 0)
- Total Credit Hours: 3
- Total Contact Hours: 45 hours (15 Weeks X 2 hours + 15 weeks X 1 hour)
  - ➢ Lectures: 30 Hours (15 Weeks X 2 Hours)
  - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
  - ➢ Practical: Nil

3. Prerequisites and Dependencies: Any students from undergraduate programme without any prerequisite course can take this course.

4. Objective: The major objective of this course is to provide students with understanding of Data mining and various techniques that are used to extract useful patterns from data.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Understanding Data mining
CO2	Understanding Preprocessing tasks in Data analysis.
CO3	Understanding Classification techniques.
CO4	Understanding Clustering techniques.
CO5	Understanding Association Rule Mining Techniques.

6. Topics:

Module I: Introduction (10 hours)
What is data mining, data mining tasks, types of data- attribute, measurement, data quality, data preprocessing, measure of similarity and dissimilarity

Module II: Mining Frequent Patterns, Associations, and Correlations (20 hours)
Market Basket Analysis, Frequent Itemsets, Closed Itemsets, and Association Rules, Frequent Pattern Mining, The Apriori Algorithm: Finding Frequent Itemsets Using Candidate Generation, Generating Association Rules from Frequent Itemsets, Mining Frequent Itemsets without Candidate Generation, Mining Multilevel Association Rules, Mining Multidimensional Association Rules.

Module III: Supervised Learning (20 hours)
Classification & Prediction: Decision Tree Techniques, Decision Tree Induction, Attribute Selection Measures, Tree Pruning, Scalability and Decision Tree Induction, Bayes' Theorem, Naïve Bayesian Classification, Bayesian Belief Networks, Training Bayesian Belief Networks, Rule Extraction from a Decision Tree, A Multilayer Feed-Forward Neural Network, Defining a Network Topology, Back-Propagation Method, k-Nearest-Neighbor Classifiers, Genetic

Algorithms, Regression: Linear Regression, Nonlinear Regression, Classifier Accuracy Measures, Holdout Method and Random Subsampling, Cross-validation, Bootstrap, Ensemble Methods, Bagging, Boosting.

Module IV: Unsupervised Learning (10hours)
Clustering, Types of Data, Partitioning Methods, Hierarchical Methods, Density Based Methods, Grid Based Methods, Model Based Methods, Clustering High-Dimensional Data, Constraint Based Methods, Outlier Analysis.

7. Referential Sources:

1. Jiawei Han and Micheline Kamber, “Data Mining: Concepts and Techniques”. Morgan Kaufmann India.
2. Pang-Ning Tan, Steinbach, Karpatne, Vipin Kumar, “Introduction to Data Mining”, Pearson
3. Ian H. Witten, Eibe Frank, Mark A. Hall, “Data Mining: Practical Machine Learning Tools and Techniques”, Morgan Kaufmann.

Back to Contents

Introduction to Machine Learning
(IPCTVA3301)

1. About the Course: This is a value-added Course. It is an introductory level UG course and is aimed to make students familiar with the key algorithms and theory that form the core of machine learning.

2. Course Description:

- Target Audience: Third semester students of any Integrated programme only
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 3 (2 + 1 + 0)
- Total Credit Hours: 3
- Total Contact Hours: 45 hours (15 Weeks X 2 hours + 15 weeks X 1 hour)
    - ➢ Lectures: 30 Hours (15 Weeks X 2 Hours)
    - ➢ Tutorial: 15 hours (15 Weeks X 1 Hour)
    - ➢ Practical: Nil

3. Prerequisites and Dependencies: Students are expected to have basic knowledge on probability, statistics, algorithm and computer programming for registration in this course.

4. Objective: The primary objective of this course is to give students a basic understanding to machine learning and to study and construct of computer algorithms that improves automatically through experience.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Understand a wide variety of learning algorithms
CO2	Understand how to evaluate model generated from data.
CO3	Understand the difference between supervised and unsupervised learning.
CO4	To develop skills of using recent machine learning software tools to evaluate learning algorithms and model selection for solving practical problems.

6. Topics:

Module I: Introduction (7 hours)

Well-Posed Learning Problems, Definition of learning system, designing a learning system training data, concept representation, function approximation, Perspective and issues in machine learning, supervised and unsupervised learning

Module II: Concept Learning (8 hours)

Concept learning task, Concept learning as search-general to specific ordering of hypothesis, Finding a Maximally Specific Hypothesis, Version space and the candidate -elimination algorithm, Inductive bias

Module III: Decision Tree Learning (8 hours)

Introduction, Decision tree representation, Decision tree learning algorithm, Hypothesis Space Search in Decision Tree learning, Inductive bias in decision tree learning, Issues in decision tree learning, Entropy Based Node selection, ID3 Algorithm, Random Forest

Module IV: Artificial Neural Network (7 hours)

Introduction, Neural Network representation, perceptron’s, Multilayer and backpropagation algorithms, Convolutional network, Recurrent network

7. Referential Sources:

1. Tom M. Mitchell, “Machine learning”, McGraw Hill
2. O Theobald, “Machine Learning for Absolute Beginners: A Plain English Introduction”, Scatterplot Press
3. D. Barber, “Bayseian Reasoning and machine learning”, 2012
4. S. Rogers and M. Girolami, “A first course in Machine Learning”, CRC Press, 2011

PREREQUISITE COURSES

Mathematics-I

(IPCTPQ1201)

1. **About the Course**: This is a prerequisite course and is aimed to improve mathematical skill of students. The course is organized as a series of lectures with both theory and tutorial sessions.

2. **Course Description**:

- Target Audience: First semester students of any Integrated MCA programme only
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 2 (1 + 1 + 0)
- Total Credit Hours: 2
- Total Contact Hours: 30 Hours (15 Weeks X 1 Hour + 15 weeks X 1 Hour)
  - ➢ Lectures: 15 Hours (15 Weeks X 1 Hour)
  - ➢ Tutorial: 15 Hours (15 Weeks X 1 Hour)
  - ➢ Practical: Nil

3. **Prerequisites and Dependencies**: This course doesn't have any prerequisite course.

4. **Objective**: This course is mainly designed for those students who have not studied any 'Mathematics' course in 10+2 level. But other students are also advised to take this course to refresh their understanding. Different mathematical theories are largely applied in the field of Computer Science. The main objective of this course to introduce students about functions, differential calculus, integral calculus, differential equation and vector and coordinate geometry.

5. **Course Outcomes**: After course completion, following are the course outcomes.

CO1	Students will get to know about different type of function
CO2	Students will understand the basic concept of Differential Calculus; limit and continuity. Derivative. Rules of differentiation. Tangent to a curve. Taylor, Maxima and minima.
CO3	Students will understand the basic concept of Integral Calculus; Integrals of elementary functions. Substitution and partial fractions. Definite integral as a limit of sum. Properties of definite integrals
CO4	Students will be able to solve differential equation.
CO5	Student will understand the various concept of vectors and coordinate geometry

6. **Topics**:

Module I: Functions (6 hours)

Functions, domains and range of a function, different type of functions, One-one, onto mappings. Inverse and composite mappings

Module II: Differential Calculus (6 hours)

Limits and continuity of function, Partial Differentiation, Chain rule, Total Derivative; Maxima, Minima and Saddle points; Method of Lagrange's multipliers, Taylor's series for two or more variables

Module III: Integral Calculus (6 hours)
Fundamental theorem of calculus (statement only), Integrals of elementary functions. Substitution and partial fractions. Definite integral as a limit of sum. Properties of definite integrals

Module IV: Differential Equation (6 hours)
Basics of first order Differential Equations, Second and Higher order differential equations with constant coefficients. Second order linear differential equations with variable coefficients, method of variation of parameters, Introduction to Partial Differential Equations.

Module V: Vectors and Coordinate Geometry (6 hours)
Vectors and their algebra. Unit vectors. Components of a vector. Position vector. Direction cosines and direction ratios. Dot and cross products. Projection of a vector on another. Distance between two points. Equations of a line, plane and sphere. Intersections. Shortest distance between lines and planes.

7. Referential Sources:

1. NCERT. Mathematics Textbook for class XI and XII.
2. R.D. Sharma, Mathematics, Dhanpat Rai Publications, New Delhi.
3. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.

Mathematics-II
(IPCTPQ2201)

1. About the Course: This is a prerequisite course. Concepts and notations from discrete mathematics are useful in studying and describing problems in all branches of computer science, such as design and analysis of algorithms, programming languages, automata theory, Artificial Intelligence, Data Science, software development etc. Conversely, computer implementations are tremendously significant in applying ideas from discrete mathematics to real-world applications, such as operations research.

2. Course Description:

- Target Audience: Second semester students of any Integrated MCA programme only
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 2 (1 + 1 + 0)
- Total Credit Hours: 2
- Total Contact Hours: 30 Hours (15 Weeks X 1 Hour + 15 weeks X 1 Hour)
  - Lectures: 15 Hours (15 Weeks X 1 Hour)
  - Tutorial: 15 Hours (15 Weeks X 1 Hour)
  - Practical: Nil

3. Prerequisites and Dependencies: There is no particular prerequisite for the course. However, the students are expected to have elementary knowledge of general mathematics and logical reasoning.

4. Objective: The objective of this course is to explain the basic theory of discrete mathematics applied in different fields of Computer Science. Concepts earned in this course will be used in subsequent courses such as "Design and Analysis of Algorithms", "Formal Language and Automata", "Software Engineering", "Introduction to Machine Learning" , "Introduction to Artificial Intelligence", "Data Mining" etc.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1 Students completing this course will get understanding of the concepts of set, function, relation, different algebraic structures, determinant and matrices

CO2 Students completing this course will be able to apply the rules of inference and methods of proof including direct and indirect proof forms, proof by contradiction, and mathematical induction.

CO3 Students completing this course will be able to evaluate Boolean functions and simplify expressions using the properties of Boolean algebra

CO4 Students will be able to solve problems involving recurrence relations and generating functions.

6. Topics:

Module I: Basic Structures (7 hours)

Sets, Set Operations, Functions, Sequence and Summation, Cardinality of sets, Relations and their properties, Representing relations, Closure of relations, Equivalence relations, Partial ordering, Algebraic structure-group, ring, field

Module II: Logic and Proofs (7 hours)
Propositional logic and its application, Propositional equivalence, Predicates and quantifiers, Nested quantifiers, Rules of inference, Boolean algebra, Introduction to proofs, Proof methods and strategy, Mathematical induction, Recursive definition and structural induction.

Module III: Number Theory (7 hours)
Divisibility and modular arithmetic, Primes and Greatest Common Divisors, Solving congruence, Application of congruence.

Module IV: Counting (9 hours)
Basics of counting, The Pigeonhole principle, Permutation and combination, Application of recurrence relation, solving linear recurrence relation, Generating functions.

7. Referential Sources:

1. Rosen K.H., "Discrete Mathematics and it's applications". McGraw Hill
2. Seymour Lipschutz, Marc Lars Lipson, "Discrete Mathematics". Schaum's outlines
3. Narsingh Deo, "Graph Theory with Application to Engineering and Computer Science". PHI Publishing.
4. Jay L. Devore," Probability and Statistics for Engineering and the Sciences". Cengage India Private Limited

----

Back to Contents

Mathematics-III
(IPCTPQ3201)

1. About the Course: This is a prerequisite course. Concepts and notations from discrete mathematics are useful in studying and describing problems in all branches of computer science, such as design and analysis of algorithms, programming languages, automata theory, Artificial Intelligence, Data Science, software development etc. Conversely, computer implementations are tremendously significant in applying ideas from discrete mathematics to real-world applications, such as operations research.

2. Course Description:

- Target Audience: Third semester students of any Integrated MCA programme only
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 2 (1 + 1 + 0)
- Total Credit Hours: 2
- Total Contact Hours: 30 Hours (15 Weeks X 1 Hour + 15 weeks X 1 Hour)
  - Lectures: 15 Hours (15 Weeks X 1 Hour)
  - Tutorial: 15 Hours (15 Weeks X 1 Hour)
  - Practical: Nil

3. Prerequisites and Dependencies: There is no particular prerequisite for the course. However, the students are expected to have elementary knowledge of general mathematics and logical reasoning.

4. Objective: The objective of this course is to explain the basic theory of discrete mathematics applied in different fields of Computer Science. Concepts earned in this course will be used in subsequent courses such as "Design and Analysis of Algorithms", "Formal Language and Automata", "Software Engineering", "Introduction to Machine Learning" , "Introduction to Artificial Intelligence", "Data Mining" etc.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Students completing this course will be able to use tree and graph algorithms to solve problems
CO2	Students completing this course will be able to apply the knowledge of discrete probability and statistics in different domain of computer science.
CO3	Students will be able to solve system of linear equation
CO4	Students will be able to solve problem related to graph and tree.

6. Topics:

Module I: Probability and Statistical Concepts (10 hours)

An introduction to discrete probability, Probability theory, Conditional Probability, Baye's theorem, Mean, median, mode, Random variable and distribution, Expected values and variance.

Module II: Graphs and Trees (10 hours)

Graphs and graphs models, Graph terminology, Special types of graphs Graph isomorphism, Walk, Path, Cycle, Connectivity, Hamiltonian path, shortest path algorithm, Introduction to trees, Tree traversal, Spanning tree, Minimum spanning tree.

Back to Contents

Module III: Matrices and solution of linear equations (10 hours)
Determinant and matrices, matrix inversion, Algebra of matrices, Row Echelon form, Inverse and Rank of a matrix, Symmetric, Skew- symmetric and Orthogonal matrices; Determinants; Linear Independence and Dependence of vectors. Eigen values and Eigenvectors; CayleyHamilton Theorem, Diagonalization of matrices and Orthogonal transformation system of linear equation, solution of linear equation (Gauss's elimination, Rank method)

7. Referential Sources:

1. Rosen K.H., "Discrete Mathematics and it's applications". McGraw Hill
2. Seymour Lipschutz, Marc Lars Lipson, "Discrete Mathematics". SCHAUM'S ouT lines
3. Narsingh Deo, "Graph Theory with Application to Engineering and Computer Science". PHI Publishing.
4. Jay L.Devore," Probability and Statistics for Engineering and the Sciences".Cengage India Private Limited

----

Numerical Methods
(IPCTPQ7201)

1. About the Course: This is a prerequisite course and is aimed at making a student comfortable on computational mathematics, by using basic algorithms underpinning computer predictions in modern systems science. The course is organized as a series of lectures with both theory and tutorial sessions.

2. Course Description:

- Target Audience: Seventh semester students of any Integrated MCA programme only
- Course Period: One semester (15 Weeks)
- Total Credit (L + T + P): 2 (1 + 1 + 0)
- Total Credit Hours: 2
- Total Contact Hours: 30 Hours (15 Weeks X 1 Hour + 15 weeks X 1 Hour)
    - ➢ Lectures: 15 Hours (15 Weeks X 1 Hour)
    - ➢ Tutorial: 15 Hours (15 Weeks X 1 Hour)
    - ➢ Practical: Nil

3. Prerequisites and Dependencies: The course does not have any prerequisite course.

4. Objective: The major objective of this course is to provide the numerical methods of solving the nonlinear equations, interpolation, differentiation, and integration and to improve the student's skills in numerical methods by using the numerical analysis software and computer facilities.

5. Course Outcomes: After course completion, following are the course outcomes.

CO1	Demonstrate understanding of common numerical methods and how they are used to obtain approximate solutions to otherwise intractable mathematical problems
CO2	Derive numerical methods for various mathematical operations and tasks, such as interpolation, differentiation, integration, the solution of linear and nonlinear equations, and the solution of differential equations.
CO3	Analyze and evaluate the accuracy of common numerical methods
CO4	Write efficient, well-documented C code and present numerical results in an informative way.

6. Topics:

Module I: Solution of Equation with One variable (5 hours)
Bisection Method, Fixed-Point Iteration, Newton’s Method, Error Analysis for Iterative Methods, Accelerating Convergence, Muller’s Method

Module II: Interpolation (5 hours)
Interpolation and Lagrange Polynomial, Data Approximation and Neiville’s Method, Divided Differences, Hermite Interpolation, Cubic Spline Interpolation, Parametric Curves.

Module III: Numerical Differentiation and Integration (5 hours)

Numerical Differentiation, Richardson's Extrapolation, Elements of Numerical Integration, Composite Numerical Integration, Romberg Integration, Adaptive Quadrature Methods, Gaussian Quadrature, Multiple Integration, Improper Integrals.

Module IV: Initial Value Problems for Ordinary Differential Equations (5 hours)
The Elementary Theory of Initial-Value-Problems, Euler's Method, Runge Kutta Method, Extrapolation Methods, Higher Order Equations and Systems of Differential Equations.

Module V: Direct Methods of Solving Linear Systems (5 hours)
Linear Systems of Equations, Pivoting Strategies, Linear Algebra and Matrix Inversion, Determinant of a Matrix, Matrix Factorization, Special types of Matrices.

Module VI: Solutions of Nonlinear Systems of Equations (5 hours)
Fixed Points for Functions of Several Variables, Newton's Method, Quasi-Newton's Method, Steepest Descent Method, Homotomy and Continuation Methods.

7. Referential Sources:

1. Richard L. Burden, "Numerical Analysis". Brooks/Cole CENGAGE Learning
2. Peter Linz, Richard L. C. Wang, "Exploring Numerical Methods". Jones and Bartlett Publishers, Inc.

Useful Web Sources

- https://nptel.ac.in/courses/111106101 : NP-TEL content
