School of Computer & Information Sciences

Master of Computer Applications

SCHOOL OF COMPUTER & INFORMATION SCIENCES

Vision Statement:

- To invent, create and bring computing technology solutions to the common man, to the privileged and underprivileged sections of India, to bridge the digital divide and eradication of computer ignorance and digital illiteracy and to build a prosperous and technologically advanced nation.

Mission Statements:

MS-1: To pursue academic and research excellence, nationally and internationally

MS-2: To provide training, advisory, and consultancy to all the stakeholders.

MS-3: To lead the efforts in creative and newer modes of instruction delivery & supervision

School of Computer and Information Sciences

Name of the Academic Program: Master of Computer Applications

Program Educational Objectives (PEOs)

PEO-1: To train the graduates to acquire in depth knowledge of fundamental concepts and programming skills for holistic development

PEO-2: To prepare the graduates for productive careers in software industry, corporate sector and Government Organizations.

PEO-3: To prepare graduates to acquire excellent computing ability so that they can analyze, design and create Solutions for real time problems.

PEO-4: To apply the current tools and techniques to create systems for solving Industry oriented problems.

PEO-5: To prepare graduates to gain multidisciplinary knowledge through real time case studies, projects and industry internship to meet the industry needs.

Mapping Program Educational Objectives (PEOs) with Mission Statements (MS)

	MS-1	MS-2	MS-3
PEO-1	1	3	2
PEO-2	3	2	1
PEO-3	1	2	3
PEO-4	3	2	1
PEO-5	1	2	3

Write '3' in the box for 'high-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low-level' mapping.

School of Computer & Information Sciences

Name of the Academic Program: Master of Computer Applications

Program Outcomes (POs)

PO-1: Computational Knowledge	Apply knowledge of computing fundamentals, computing specialization, mathematics, and domain knowledge to conceptualize computing models
PO-2: Problem Analysis	Identify, formulate, research literature, and solve complex computing problems reaching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines.
PO-3: Design/Development of Solutions	Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.
PO-4: Conduct Investigation of Complex Computing Problems	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO-5: Model Tool Usage	Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations.
PO-6: Professional Ethics	Understand and commit to professional ethics and cyber regulations, responsibilities, and norms of professional computing practice.
PO-7: Life Long Learning	Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional.
PO-8: Project Management and Finance	Demonstrate knowledge and understanding of the computing and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO-9: Communication Efficacy	Communicate effectively with the computing community, and with society at large, about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.
PO-10: Societal and Environmental Concern	Understand and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practice.
PO-11: Individual Team Work	Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments.
PO-12: Innovation and Entrepreneurship	Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

Mapping of Program Outcomes (POs) and Program Specific Outcomes (PSOs) with Program Educational Objectives (PEOs)

	PEO-1	PEO-2	PEO-3	PEO-4	PEO-5
PO-1	1	2	3	2	1
PO-2	1	3	1	1	2
PO-3	2	2	1	1	3
PO-4	3	3	1	2	1
PO-5	3	2	2	1	1
PO-6	1	2	1	2	3
PO-7	2	3	2	1	1
PO-8	3	1	1	1	2
PO-9	1	1	2	2	3
PO-10	1	2	3	2	1
PO-11	2	2	1	1	3
PO-12	3	2	2	1	1

Mapping of Program Specific Outcomes (PSOs) where applicable.

Write '3' in the box for 'high-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low-level' mapping.

School of Computer & Information Sciences

Master of Computer Applications

2 Year Program (w. e. f: July 2022)

Scheme

I-Semester			II-Semester		
Code	**Course Title**	**Credits**	**Code**	**Course Title**	**Credits**
CA411	Theoretical Foundations of Computer Science	4	CA461	Computer Networks	4
CA412	Programming Methodology	3	CA462	Operating Systems	4
CA413	Computer Organization and Architecture	3	CA463	Data Structures	3
CA414	Computer Based Optimization Techniques	3	CA464	Object Oriented Programming	3
	UoH Mandatory Course	4	CA514	Database Management Systems	3
CA415	PM Lab	1.5	CA465	Data Structures Lab	1.5
CA416	Internet Technologies Lab	1.5	CA466	OOP Lab	1.5
			CA516	DBMS Lab	1.5
				Mini Project*	
		20			**21.5**
III-Semester			**IV-Semester**		
CA511	Algorithms	4	CA561	Internship	**6**
CA512	Software Engineering	3			
CA513	Software Project Management	3			
	Elective-I	3/4			
	Elective-II	3/4			
CA515	Software Engineering Lab	1.5			
	Mini Project	6			
CA517	Communication Skills & Technical Writing	3			
		26.5/28.5			

- *** Mini Project evaluation will be held in Semester III**

School of Computer & Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA-I)

Course Code : CA411 Title of the Course : Theoretical Foundations of Computer Science

L-T-P :. 4-0-0 Credits : 4

Prerequisite Course / Knowledge (If any): It is expected that the students must have done a mathematics course at 10+2 level

Course Outcomes (COs) (5 to 8)

After completion of this course successfully, the students will be able to……

- CO1: Apply predicate and propositional logic to represent and solve problems. (Apply)
- CO2: Discuss various ways of simplification and apply the same on minimizing logical circuits, (Understand)
- CO3: Using principle of recursion, be able to frame a real-world situation as a recurrence relation and solve. (Apply).
- CO4: Describe counting principles (Understand)
- CO5: Apply counting principles in real world scenarios. (Apply)
- CO6: Describe graphs and trees techniques (Understand)
- CO7: Apply the graphs and trees techniques to solve the real time problems (Apply)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	2	3	2							
CO2			1		2		2					3
CO3			1	1	2			3				
CO4	1		2						3			
CO5			3		1					2		
CO6	1	3		2								
CO7			1		2					3	3	

Detailed Syllabus:

UNIT-I: Sets, Relations and Functions

Sets, relations and functions; Methods of proof; Equivalence relations; Cardinality; Countable and uncountable sets

UNIT-II: Introductory Logic

Fundamentals of Logic; Logic operators such as AND, OR etc., Truth tables; Logical inferences; Methods of proofs of an implication; First order logic; Predicate calculus Predicates and Quantifiers; Rules of inference for quantified propositions

UNIT-III: Recurrence Relations:

Recursion, Forming and solving recurrence relations by substitution method and generating functions; Method of characteristic roots; solving inhomogeneous recurrence relations

UNIT-IV: Boolean Algebra:

Partial order relations; Lattices; Boolean algebra; Combinatorial circuits; Minimization of Boolean functions using Karnaugh maps

UNIT-V: Theory of Graphs

Graphs, subgraphs, isomorphism, proofs; Types of graphs; paths and cycles; Adjacency matrices; Transitive closure; Connectivity; Directed acyclic graphs; Planar graphs and Euler's formula; Dual of a graph; Hamiltonian and Eulerian graphs; Applications like matching and colouring graphs; Graph traversals (BFS and DFS); Trees; Spanning trees.

Reference Books:

1. Kenneth H Rosen (2012), "*Discrete Mathematics and Its Applications*",7th Edition, McGraw Hill, NY
2. Ralph P Girimaldi(2003), "*Discrete and Combinatorial Mathematics –An Applied Introduction*", 5th Edition, Pearson Addison Wesley, Indian Edition
3. J.R Mott, A Kandel, T.P Baker (2015), "*Discrete Mathematics for Computer Scientists and Mathematicians*", Pearson
4. Ronald L Graham, Donald E Knuth, Oren Patashnik(1994), "*Concrete Mathematics- A Foundation of Computer Science*", 2nd Edition, Addison Wesley .
5. Susanna S. Epp(2010), "*Discrete Mathematics with Applications*", 4th Edition, Brooks/Cole Cengage Learning.

School of Computer & Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA-I)

Course Code : CA412/CA415 Title of the Course : Programming Methodology & PM Lab

L-T-P : 3-0-3 Credits : 3+1.5 (4.5)

Prerequisite Course / Knowledge (If any): Nil

Course Outcomes (COs)

After completion of this course successfully, the students will be able to……

CO-1: Create specification from problem requirements by asking questions to disambiguate the requirement Statement. (Create)

CO-2: Design the solution from specification of a problem and write pseudo code of the algorithm. (Create)

CO-3: Analyze algorithms by tracing algorithms with test cases. (Analyze)

CO-4: Develop C programs using all supported features and compile them using Makefile. (Create)

CO-5: Analyze programs using debugging tools. (Analyze)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1		2		3							
CO2		2	1		3			2				
CO3	1	2	3									
CO4				2	1							3
CO5		2	1		3							

.

Detailed Syllabus:

UNIT-I**:** Introduction to problem solving:
Problems and problem instances; Informal approach to program design: generalisation, special cases, and algorithms, breaking down a problem into functions, input and output.

UNIT-II**:** Introduction to the 'C' programming language:
Program structure; main () function; unnamed and named blocks; basic data types, variables, declaration and definition; initialisation and assignment; arithmetic operators and precedence; implicit and explicit type conversions; arrays; boolean variables and logical operators.

UNIT-III**:** Control structures:
Branching and iteration; functions and parameters; break (), return () and exit () functions; local and global variables; function prototypes.

UNIT-IV: Pointer variables and dynamic structures:
Static and dynamic (run-time) memory structures; static variables; breaking a program across multiple files; creating and linking libraries.

UNIT-V: Detecting and correcting common errors:
Debugging and debuggers; documenting programs; good programming practices; programming exercise (writing a program of at least 200 lines split across multiple files).

Reference Books:

1. Brian W. Kernighan, Dennis M. Ritchie. "The C Programming Language, 2nd Edition", Prentice-Hall India.
2. G. Michael Schneider. "Introduction to Programming and Problem Solving with PASCAL", John Wiley and Sons.
3. Paul Deitel and Harvey Deitel . "C How to Program", Pearson Education India.
4. Stephen Kochan. "Programming in C", Pearson Education India.
5. Brian W. Kernighan and R. Pike. "The Unix Programming Environment", Prentice-Hall India.
6. Chakravarthy Bhagvati. "How to Program (An Informal Guide)", https://scis.uohyd.ac.in/~chakcs/howtoprogram.pdf

School of Computer & Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA-I)

Course Code : CA413 Title of the Course : Computer Organization & Architecture

L-T-P :. 3-0-0 Credits : 3

Prerequisite Course / Knowledge (If any): It is expected that the students must have done a programming course at any level

Course Outcomes (COs)

After completion of this course successfully, the students will be able to……

- CO1: Demonstrate arithmetic operations and assess their performance (Apply).
- CO2: Describe basic Instruction Set Architecture (ISA) (Understand)
- CO3: Explain the basic pipelining of instructions (Understand)
- CO4: Examine how the memory hierarchy has impact on performance of software. (Analyze)
- CO5: Describe Interrupt handling and DMA access for performing I/O. (Understand)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	3	1					1				
CO2	1		2		3							
CO3	1	2	3									
CO4	3			1	2							
CO5	1		3		2							

Detailed Syllabus:

UNIT - I: Computer Evolution & Arithmetic:

A Brief History of computers, Designing for Performance, Von Neumann Architecture, Hardware architecture, Computer Components, Interconnection Structures, Bus Interconnection, Scalar Data Types, Fixed and Floating-point numbers, signed numbers, Integer Arithmetic, 2's Complement method for multiplication, Booths Algorithm, Hardware Implementation, Division, Restoring and Non-Restoring algorithms, Floating point representations, IEEE standards, Floating point arithmetic

UNIT - II: The Central Processing Unit:

Machine Instruction characteristics, types of operands, types of operations, addressing modes, Instruction formats, Instruction types, Processor organization, Processor as running example, Programmers model of, max/min mode, Register Organization, Instruction cycles, Read Write cycles, assembly instruction examples to explain addressing modes

UNIT – III: The Control Unit:

Single Bus Organization, Control Unit Operations: Instruction sequencing, Micro operations and Register Transfer. Hardwired Control: Design methods – State table and classical method, Design Examples - Multiplier CU. Micro-programmed Control: Basic concepts, Microinstructions and micro- program sequencing

UNIT - IV**:**

Memory Organization: Characteristics of memory systems, Internal and External Memory, Types of memories: ROM: PROM, EPROM, EEPROM, RAM: SRAM, DRAM, SDRAM, RDRAM, High-Speed Memories: Cache Memory, Organization and Mapping Techniques, Replacement Algorithms, Cache Coherence, Virtual Memory: Main Memory allocation, Segmentation, Paging, Address Translation Virtual to Physical. Secondary Storage: Magnetic Disk, Tape, DAT, RAID, Optical memory, CDROM, DVD

UNIT - V: I/O Organization:

Input/ Output Systems (features and principles), Programmed I/O, Interrupt Driven I/O, Interrupt structure, Direct Memory Access (DMA), features Buses and standard Interfaces: Synchronous, Asynchronous, Parallel I/O features, Serial I/O features, PCI, SCSI, USB Ports Working mechanisms of Peripherals: Keyboard, Mouse, Scanners, Video Displays, Touch Screen panel, Dot Matrix, Desk-jet and Laser Printers.

UNIT - VI:

Case Studies: Concepts RISC: Instruction execution characteristics, RISC architecture and pipelining. RISC Vs CISC. ARM and Embedded Systems PowerPC, Intel X86 Evolution from 32bit to 64bit architectures. AMD Opteron

Reference Books

1. Patterson D.A. & Hennesy J.L., *Computer Organisation & Design: The Hardware/Software Interface*.
2. Computer Organization and Architecture, 10/E William Stallings ISBN-10: 0134101618 • ISBN-13: 9780134101613- See more at: http://www.pearsonhighered.com/pearsonhigheredus/educator/product/products_detail.page?isbn=9780134101613&forced_logout=forced_logged_out#sthash.WVVJbZUb.dpuf

School of Computer & Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA-I)

Course Code : CA414 Title of the Course : Computer Based Optimization Techniques

L-T-P : 3-0-0 Credits : 3

Prerequisite Course / Knowledge (If any): Mathematics at 10+2 level

Course Outcomes (COs)

After completion of this course successfully, the students will be able to……

- CO-1: Analyze given data by carrying out discrete and inferential statistics (Analyze)
- CO-2: Solve problems on Probability Theory (Apply)
- CO-3: Apply probability computation of an event and compute four moments for a given distribution function (Apply)
- CO-4: Formulate optimization problems as Linear Programming Problem (Create)
- CO-5: Solve Linear Programming Problem using graphical method. (Apply)
- CO-6: Solve LP using Simplex method and its variants (Apply)
- CO-7: Solve Integer programming problems using cutting plane, branch and bound and other methods (Apply).
- CO-8: Solve problems using Dynamic programming technique (Apply)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1		2	1	3								
CO2	3		2	1								
CO3	1		3	2								
CO4			1							2		3
CO5			1	2				3				
CO6			2	1				3				
CO7			1	2				3				
CO8			1	2				3				

Detailed Syllabus:

UNIT-I: Descriptive Statistics, and Hypothesis Testing

Descriptive Statistics- Measures of central tendency**,** Simple regression and correlation - Estimation using regression line, correlation analysis,

Hypothesis Testing - Two sample tests - testing for difference between proportions, probe values Chi-square and analysis of variance - chi-square as test of independence, chi-square as a test of goodness of fit, analysis of variance, t-distribution

UNIT-I: Probability - Basic terminology, types, Probability rules, Statistical independence, statistical dependency, Bayes' theorem Probability distributions - random variables, expected values, binomial distribution, Poisson distribution, normal distribution, choosing correct distribution Sampling and sampling distributions - Random sampling, design of experiments, sampling distributions, operational considerations in sampling

UNIT-III: Introduction to optimization technique: History, objective, decision making: decision making environment (certain and uncertain), decision making criteria (probabilistic and non-probabilistic), Linear Programming: Mathematical formulation of Linear Programming problem, problem, Solution by Graphical and Simplex methods, Revised Simplex method, Duality, Primal –Dual relationship, Dual Simplex method.

UNIT-IV: Special Types of Linear Programming Problem: Integer Programming: Branch and bound method and Gomory cutting plane method. Transportation problem, Northwest corner method, Stepping stone method, Unbalanced transportation problem,

UNIT-V: Assignment problem, The Hungarian method, Machine scheduling: n-jobs on two machine, n-jobs on three machines, two jobs through n machine, and n jobs through m machines

Suggested Reading

1. Douglas C. *Montgomery &* George C. Runger " Applied Statistics and Probability for Engineers " (6e) Wiley ISV Paperback – 1 January 2016

2. F S Hillier and G J Lieberman, Introduction to Operations Research, 7th edition, McGraw Hill, 2000

3. H A Taha, Operations Research –An Introduction, 8th Edition, Pearson Prentice Hall, 2007

School of Computer & Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA-I)

Course Code : CA416 Title of the Course : Internet Technologies Lab (IT Lab)

L-T-P : 0-0-3 Credits : 1.5

Prerequisite Course / Knowledge (If any): Nill

Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

- CO-1: Design web pages using scripting languages, cascading styles sheets and identify its elements and attributes. (Create)
- CO-2: Develop web pages using client-side technologies and perform event handling and validation procedures. (Create)
- CO- 3: Create schemas and documents using markup languages, design and develop lightweight data-interchange formats for exchange of data between client and server applications. (Create)
- CO-4: Apply JavaScript libraries to create dynamic web page, access and use web services for interactive web contents. (Apply)
- CO-5: Develop applications using server-side technologies, implement session management, database connectivity, and create dynamic HTML content with PHP. (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PSO1	PSO2	PSO3
CO1			3	2	1				
CO2			3	2	1				
CO3			2	3	1				
CO4			2	3	1				
CO5		1		2	3				

Detailed Syllabus:

UNIT- I: HTML, Forms & CSS

Introduction To HTML5, WWW, W3C, web publishing, Introduction to Style sheet, types of style sheets- Inline, External, Embedded CSS, text formatting properties, CSS Box Model, CSS Border, margin properties, Positioning Use of classes in CSS, color properties, use of &, Layout Design using CSS.

UNIT- II: JavaScript

Intro to script, types, intro of JavaScript, JavaScript identifiers, operators, control & Looping structure, Intro of Array, Array with methods, Math, String, Date Objects with methods User defined & Predefined functions, DOM objects, Window Navigator, History, Location, Event handling, Validations on Forms.

UNIT- III: Representing Web Data: XML, JSON

XML, XML Schema and DTD document definitions, XSLT transformations and programming, XPath, XQuery, Introduction to JSON.

UNIT- IV: jQuery& AJAX

Introduction to jQuery, Syntax Overview, Anatomy of a jQuery Script, Creating first jQuery script, Traversing the DOM, Selecting Elements with jQuery, Refining & Filtering Selections, Selecting Form Elements, Working with Selections, Chaining, Getters & Setters, CSS, Styling, & Dimensions, Manipulating Elements, Getting and Setting Information about Elements, Moving, Copying, and Removing Elements, Creating New Elements, Manipulating Attributes, Utility Methods, Events, Connecting Event to Elements, Namespacing Events, Event handling, Triggering Event handlers, Event Delegation, Animating effects, animate(), click(), hover(), toggle(), Plugins , Create a basic plugin, Finding &Evaluating Plugins, Writing Plugins, Tabs, Panels and Panes examples, jQuery UI and Forms, AJAX Overview, jQuery's AJAX related methods, Ajax and Forms, Ajax Events

UNIT- V: PHP based Server-Side Programming

Introduction to PHP Programming Language, Basic Programming Features: Expressions and Control flow and Arrays, Form Handling, Database Management using PHP and MySQL, Stateful Server Programming with Cookies and Sessions

Reference Books:

1. Internet and World Wide Web How to Program, P.J. Deitel, H.M. Deitel
2. Jeffrey C. Jackson, "Web Technologies—A Computer Science Perspective", Pearson Education, 2006.
3. Complete reference HTML.
4. JavaScript Bible
5. HTML, DHTML, JavaScript, Perl & CGI Ivan Bayross
6. XML: How to program Deitel&Deitel.
7. Learning PHP, MySQL & Java Script, Robin Nixon, O'Reilly Publications, 2018.

Suggested Exercises

1. **Develop an Ajax application so that it uses any of the file (JSON or XML) as input and displays the read data without changing the front end of the application. (Assume appropriate members and data for the design and development of the required application).**
2. **Design and develop a graphical user interactive application with various components (elements). Also perform client-side validation using JavaScript.**
3. **Write JS code that reads XML file or JSON file and print the details as tabular data. (Assume appropriate members and data for the design and development of the required application).**
4. **Using PHP technologies develop an application to perform the following operations: Login/Register, make a topic, leave replies, edit content, delete content. Create different permissions for different users – simple users should only be allowed to edit (not delete) the topics and replies that were created by them. Admins should be able to delete and edit anything.**
5. **Develop an application using HTML, CSS and Java Script such that access to JSON data from URL parameters will display the data based on the search keywords mentioned in the textbox. (Assume appropriate members and data for the design and development of the required application).**
6. **Using PHP technologies develop an interactive application to support selection, addition, deletion and searching operations.**
7. **Create a basic plugin for the operations using jQuery UI and Forms.**
8. **Create a document that reads and stores cookies containing a user name and number of times, he or she has visited your website. Whenever the user visits the site, the system displays the cookies in alert dialogue box, increments the counter cookie by 1 and then resets the counter's expiration date to one year from the current date.**
9. **Using Web Technologies to develop Visual Aids.**
10. **Practice exercise of developing web base application with the help of PHP and databases.**
11. **Design and develop GUI to perform the event handling operations and triggering the events.**
12. **Develop an application for exemplifying the use of Unobtrusive jQuery at Client Side.**

School of Computer and Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA-II)

Course Code : CA 461 Title of the Course : Computer Networks

L-T-P : 3-1-0 Credits : 4

Prerequisite Course / Knowledge (If any): C Programming, Operating Systems

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO-1: Distinguish between multiplexing techniques (Understand)
- CO-2: Evaluate the different types of switched networks (Analyze)
- CO-3: Explain the functionalities media access for data-link and network protocols. (Understand)
- CO-4: Apply IP addressing and routing algorithms to design networks by subnetting/supernetting (Apply)
- CO-5: Describe the essential principles such as reliable data transfer, flow control, congestion control of a transport layer protocol (Understand)
- CO-6: Predict the topology given the routing protocol messages (Apply)
- CO-7: Analyze and capture network traffic using simulation tools. (Analyze)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	3		2								
CO2		3	2	1								
CO3	1		2		3							
CO4	2			3	1							
CO5	1	2	3									
CO6		3		1	2							
CO7	2			3	1							

Detailed Syllabus:

UNIT - I: Physical Layer: Modulation Techniques: Amplitude, Frequency and Phase, ADSL, Multiplexing Techniques: Frequency division multiplexing, Time division multiplexing, wave length division multiplexing, Differential PCM, Switching Techniques: Circuit, message and packet switching.

UNIT - II: Data Link Layer: PPP, PPPoE, MAC Layer: Ethernet (incl. manchester encoding), Switched Ethernet, VLANs, Spanning Tree Protocol.

UNIT - III: Network Layer: Data Plane: Internet Protocol Addressing: CIDR, Internet Protocol Datagram (including fragmentation and reassembly, routing options), IP Forwarding Algorithm, ARP, ICMP (including ICMP Redirect, ICMP Path MTU discovery, ICMP Destination Unreachable options).

UNIT - IV: Transport Layer: UDP, TCP sliding window protocol, TCP connection establishment, TCP reliability including cumulative and delayed acknowledgements, Nagle algorithm, Karn's algorithm for RTT and RTO estimation, TCP AIMD Congestion Control Algorithm, TCP half-close connections including TCP keepalive timer and probe timer, TCP Fast Retransmit and Fast Recovery.

UNIT - V: Network Layer: Control Plane: Distance Vector Algorithm and Routing Information Protocols V1 and V2, Link State Algorithm and Open Shortest Path First Protocol (OSPF).

UNIT - VI: Application Layer: Domain Naming System (DNS) and Dynamic Host Configuration Protocol (DHCP), Network Management using SNMP.

Reference Books:

1. James F. Kurose and Keith W. Ross. Computer Networking: A top-down approach, 6th edition, Pearson Education.
2. Douglas Comer. Computer Networks and Internets Sixth Edition, 2014. ISBN 0133587932/9780133587937, Pearson Education.
3. Douglas Comer. Internetworking With TCP/IP Volume 1: Principles Protocols, and Architecture, 6th edition, 2013. ISBN-10: 0-13-608530-X ISBN-13: 9780136085300, Pearson Education.
4. Kevin R. Fall and W.Richard Stevens. TCP/IP Illustrated, Volume 1: The Protocols, 2/E, 2012, ISBN-10: 0321336313 ISBN-13: 9780321336316,Pearson Education.
5. Radia Perlman. Interconnections: Bridges, Routers, Switches, and Internetworking Protocols, 2/E, 2000, ISBN-10: 0201634481 ISBN-13: 9780201634488.Pearson Education.

School of Computer and Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA-II)

Course Code: CA462 Title of the Course: Operating Systems
L-T-P : 4-0-0 Credits : 4

Prerequisite Course / Knowledge (If any): -- None

Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

- CO1: Discuss the ways system calls work. (Understand)
- CO2: Develop basic process management tasks such as scheduling, deadlock avoidance algorithms. (Create)
- CO3: Develop paging algorithm. (Create)
- CO4: Construct simple device drivers. (Create)
- CO5: Describe different file systems in existence and learn the pros and cons of the various systems. (Understand)
- CO6: Examine real world OS scheduling algorithms such as those used in Linux and Windows. (Analyze)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1		2	3								
CO2	1	2	3									
CO3			1	3	2							
CO4			3	1	2							
CO5	1	3	2									
CO6				1						2		3

Detailed Syllabus:

UNIT - I: Introduction and Operating System Structures

Operating Systems Functionality, Computer Organization and Architecture, OS Operations, Kernel Data Structures, OS Services, User interfaces to OS, Programmer interfaces to OS, OS Structure, System Boot.

UNIT - II: Process and Thread Management

Process Concept, Process operations, Process Scheduling, Extended Process State Diagram, Process Context Switch in detail; Inter process Communication: Pipes, Named Pipes, Shared Memory; Process Synchronization: Signals, Mutexes, Semaphores, Monitors; Thread Management: thread creation, thread scheduling, thread synchronization; Deadlocks: Resource Allocation Graphs, deadlock detection, prevention and avoidance, recovery from deadlock.

UNIT - III: Memory Management

Memory allocation techniques: paging and segmentation, Swapping, structure of the page table; Virtual memory: demand paging, copy-on-write, Page replacement, allocation of frames, kernel memory allocation, thrashing, memory-mapped files, Translation-Lookaside Buffer (TLB).

UNIT - IV: File System Management

Disk management: formatting, boot block, swap-space management, RAID structure; Disk scheduling algorithms: elevator, C-SCAN; File concept, Access methods, Directory structure, file sharing, protection, file system structure; file system implementation: file system metadata storage structures such as inode, allocation methods, free space management, efficiency and performance including disk cache and recovery from failures.

UNIT - V: I/O Management

I/O devices: polling, interrupt-driven, DMA; Application I/O interface: character and block devices, network devices; clocks and timers, nonblocking and asynchronous I/O, vectored I/O; Kernel I/O interface: I/O scheduling, Buffering, Caching.

Reference Books

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne. Operating System Concepts, 9th edition, Wiley.
2. Charles Crowley. Operating Systems: A Design-Oriented Approach, Prentice-Hall India.
3. W. Richard Stevens,. Advanced Programming in Unix Environment, Pearson Education.
4. W. Richard Stevens. Unix Network Programming, vol. 2, Pearson Education.
5. William Stallings. Operating Systems: Internals and Design Principles, Pearson Education.
6. Maurice J. Bach. The Design of the Unix Operating System, Prentice-Hall India.
7. Robert Love. Linux Kernel Development, Pearson Education.
8. Thomas Anderson and Michael Dahlin. Operating Systems: Principles and Practice, 2nd edition, Recursive Books.

School of Computer and Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA-II)

Course Code: CA463 Title of the Course: Data Structures

L-T-P: 3-0-0 Credits : 3

Prerequisite Course / Knowledge (If any): Must have done a programming course

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO-1 Show and demonstrate which data structures are used for static and dynamic allocations. (L2)
- CO-2 Solve the problem where in elements can be traversed by either direction and select the suitable data structure for this idea using C/Java Programming Language (L3)
- CO-3 Analyze the time taken to solve the problem by using C/Java programming language (L4)
- CO-4 Assess the solution in terms of efficiency, modularity and well-documented programs in C/Java under Linux environment (L5)

Detailed Syllabus

Unit-I

Introduction to data structures and data types: Primitive and Non-Primitive types, Arrays- Sparse matrix, Stacks, Queues, Circular queues, Priority queues, Dequeues, Conversions and Evaluations of expressions, Polynomial representation using arrays, Time complexity analysis of algorithms with respect to data structure operations

Unit-II

Linked Lists: Linked stacks and queues, Circular and Doubly linked lists, Polynomial representation using linked lists.

Unit-III

Trees and Graphs: Binary Trees, Tree Traversal, Binary Search trees and basic operations, Heaps, AVL Trees, height balanced trees. , Graphs – Representation of the graphs, Graph Traversals.

Unit-IV

Sorting and Searching Mechanism: Selection sort, Bubble sort, Insertion sort, Linear Search, Binary Search, Hash Tables.

Reference Books:

1. Horowitz, E., and Sahni.S: Fundamentals of Data structures. Computer Science Press, 1978.
2. Tanenbaum, A.M., and Augenstein, M.J.: Data Structures with Pascal, Prentice - Hall International, 1985.

3. Stubbas, D.: Data Structures with Abstract Data Types and Modula2, Brooks & Cole Pub. Co. 1987.

4. Trembley & Sorenson: An Introduction to Data Structures with Applications; Tata McGraw Hill.

8. Kruse, R. L., Leung, B. P., and Tondo, C. L.: Data Structures and Program Design in C; Prentice - Hall of India, 1999.

9. The *C Programming* Language by Brian W.Kernighan , Dennis M. *Ritchie*

10. Michael J. Folk and Bill Zoellick, "File Structures" (Second Edition).

11. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", 2nd Edition Addison- Wesley, 1997.

12. Schaum‟s Outline Series, "Data Structure", TMH, Special Indian Ed., Seventeenth Reprint, 2009.

13. Mary E. S. Loomes, "Data Management and File Structure", PHI, 2nd Ed., 1989.

School of Computer and Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA-II)

Course Code: CA465 Title of the Course: Data Structures Lab

L-T-P: 0-0-3 Credits : 1.5

Prerequisite Course / Knowledge (If any): Must have done a programming course

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO-1 Solve a problem by choosing appropriate data structures in C programming language (L2)
- CO-2 Select suitable data structure for an idea and propose solution using C Programming Language (L2)
- CO-3 Analyze the time taken to solve the problem by using C programming language (L4)
- CO-4 Assess the solution in terms of efficiency, modularity and well-documented programs in C under Linux environment (L5)

Detailed Syllabus

Unit-I

Implementation of Stacks and different types of Queues data structures using arrays, Conversions and Evaluations of an expressions.

Suggested Assignment

Large integer arithmetic. Arithmetic operations are to be performed on very large integers of N digits (where 0 < N < 20). Write C program that performs the operations of multiplication and division on such large integers.

Two stacks using single array. Implement two stacks using a single array such that neither overflow unless total number of elements in both the stacks is equal to the size of the array.

Deque. Deque is a queue which allows insertions and deletions at both ends. Write a C program that implements deque using both *arrays* and *linked list*. Each implementation should provide support for basic operations such as (i) Insertion of an element at both ends (ii) Deletion of an element from both ends (iii) Displaying all elements of deque.

Evaluation of arithmetic expression. Write a C program that reads an input arithmetic expression in (INFIX, PREFIX, POSTFIX) notations and outputs its result.

Arithmetic expression notation conversion. Write a C program that reads an input arithmetic expression in infix notation (fully parenthesized) and converts it into an output arithmetic expression in postfix notation.

Unit-II

Implementation of basic Linked List operations such as addition, updation, deletion, searching and traversal of all elements of the list

Suggested Assignments

Implement Stack data structure (using linked list) operations. Write C program to perform the basic operations on the stack.

Implement Queue data structure (using linked list) operations. Write C program to perform the basic operations on the queue.

Doubly Linked List operations. Linked list (doubly) is an important data structure for dynamic allocation wherein elements can be traversed by either direction. Write C program to perform the basic operations on the linked list.

Union and Intersection. Given two linked lists of numbers, write a program that finds a resultant linked list which is union of the two input linked lists and another resultant linked list which is intersection of the two input linked lists.

Detecting cyclic linked list. Write a C program that detects whether a given linked list is cyclic or not, if yes, then return the node where the cycle begins.

Unit-III

Implementation of Trees and Graphs of basic operations. Implementation of Adjacency Matrix and List Representation. Breadth and Depth First Search

Suggested Assignments

Tree traversals. Write C program to display a tree using all the methods of traversals: (i) Inorder traversal, (ii) Preorder traversal, (iii) Postorder traversal

Binary Search Tree. Binary Search Tree is an important data structure for dynamic allocation and optimized searching. Write C program to perform the basic operations on binary search tree (BST): (i) Adding, (ii) Updating (iii) Deleting (iv) Search for an element (v) Displaying all elements (in-order).

AVL Trees. Using a C program, perform the following operations on the *AVL tree* data structure: (i) Inserting, (ii) Deleting, (iii) Update, (iv) Searching, (v) Displaying an element

Using Linked List and BST: Construction of a city database using a linked list and binary search tree and the appropriateness of these structures under various demands for the data.

Unit-IV

Sorting and Searching Mechanism: Linear Search, Binary Search, Implementation of Bubble, Insertion, Selection Sort.

Suggested Assignments

Write C programs to perform both linear and binary search on a given random set of integers. The following points should perform by the program

- Take as input an integer, N, which would decide number of integers to be processed and another input an integer, X (0 < X < N+1), which is the key to be searched
- Randomly generate N integers whose values are between 1 to N, multiple entries are allowed
- Output all the indexes (positions) of key in given set of random integers
- Count number of comparisons in the linear and binary searching process, please note comparisons involved in sorting process (in case of binary search) are not to be included
- Output the result in following table:-

Input size (N)	Number of Comparisons	
	Linear Search	Binary Search
10		
30		
50		
70		
100		

Reference Books:

1. Horowitz, E., and Sahni.S: Fundamentals of Data structures. Computer Science Press, 1978.
2. Tanenbaum, A.M., and Augenstein, M.J.: Data Structures with Pascal, Prentice - Hall International, 1985.
3. Stubbas, D.: Data Structures with Abstract Data Types and Modula2, Brooks & Cole Pub. Co. 1987.
4. Trembley & Sorenson: An Introduction to Data Structures with Applications; Tata McGraw Hill.
5. Kruse, R. L., Leung, B. P., and Tondo, C. L.: Data Structures and Program Design in C; Prentice - Hall of India, 1999.
6. The *C Programming* Language by Brian W.Kernighan , Dennis M. Ritchie

7. Mark Allen Weiss, “Data Structures and Algorithm Analysis in C”, 2nd Edition Addison- Wesley, 1997.
8. Schaum‟s Outline Series, “Data Structure”, TMH, Special Indian Ed., Seventeenth Reprint, 2009.
9. Mary E. S. Loomes, “Data Management and File Structure”, PHI, 2nd Ed., 1989.
10. Michael J. Folk and Bill Zoellick, “File Structures” (Second Edition).
11. Mark Allen Weiss, “Data Structures and Algorithm Analysis in C”, 2nd Edition Addison- Wesley, 1997.
12. Schaum‟s Outline Series, “Data Structure”, TMH, Special Indian Ed., Seventeenth Reprint, 2009.
13. Mary E. S. Loomes, “Data Management and File Structure”, PHI, 2nd Ed., 1989.

School of Computer and Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA-II)

Course Code: CA464/CA466 Title of the Course: Object Oriented Programming (OOP) & Lab

L-T-P: 3-0-3 Credits : 3+1.5

Prerequisite Course / Knowledge (If any): Must have done a programming course

Course Outcomes (COs)

After completion of this course successfully, the students will be able to……

- CO-1: Describe the object-oriented design concepts. (Understand)
- CO-2: Analyse a given computational problem. (Analyze)
- CO-3: Design classes for a given Computational problem (Create)
- CO-4: Apply the UML concepts to model a problem. (Apply)
- CO-4: Create Java programs for the object-oriented design of the given problem. (Create)
- CO-6: Create Java programs which require to use advanced features of Java such as Exception handling, Interfaces, GUI package etc. (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	3									
CO2		1		2	3							
CO3			1	2								3
CO4			3	2	1							
CO5				1	2							3
CO6			3	1	2							

Detailed Syllabus:

UNIT -I: Object Oriented Fundamentals and Modeling: Need for OOP paradigm, What is object orientation and OO Development, Modelling, Abstraction, Three models of OOD, Object and class concepts, Links and Association Concepts, Generalization and Inheritance, N-ary associations, Aggregation, Abstract classes, multiple inheritance, metadata, Reification, Constraints, Derived data, packages.

UNIT -II: Java Basics: History of Java, java data types, variables, scope and life time of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java program, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, parameter passing, recursion, nested and inner classes, exploring string class. Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces. Exploring java.io.

UNIT -III: State Modelling and Interaction Modelling: Events, states, Transitions and Conditions, State Diagram, Nested state diagram, Concurrency Use-Case model, Sequence model, Activity model, procedural sequence model, Relation between class, state model and interaction model.

UNIT -IV: Hierarchical abstractions, Generalization and Aggregation, Base class object, subclass, subtype, substitutability, forms of inheritance-specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance. Member access rules, super uses, using final with inheritance, polymorphism-method overriding, abstract classes, the Object class Exception handling in Java: Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception sub classes. String handling, Exploring java.util.

UNIT -V: System Design, Class Design and Implementation Modelling: Overview of system design, performance estimation, reuse plan, Subsystems, Management of data storage and global resources, software control strategy and boundary conditions. Overview of class design, Realizing use-cases, designing algorithms, refactoring and design optimization, Overview of implementation, fine tuning of classes, generalization, and realizing associations.

Java Lab Exercises:

1. Basic features of Java such as data types, control structures, loops and arrays (2 Lab Sessions)
2. Working with classes, constructors, methods, objects. (2 Lab Sessions)
3. Using Inheritance, Polymorphism, Interfaces and abstract classes (3 Lab Sessions). Exercises can include a case study depicting OO application design using polymorphism and inheritance. (For example) Developing a Solitaire Application (Chapter 8 of "Introduction to Object Oriented Programming by Timothy Budd")
4. Exception Handling (1 or 2 Lab Session)
5. Exploring Java IO Package (2 or 3 Lab Sessions)
6. Java GUI Programming such as Applets (2 Lab Sessions)
7. Miscellaneous Topics such as generic classes, collection framework and java.util packages (1 or 2 Lab Sessions)

Reference Books:

1. Herbert Schildt, Java: The complete reference, McGraw hill.
2. Paul J. Deitel and Harvey M. Deitel , Java: How to Program, Prentice Hall.
3. T. Budd, Understanding OOP with Java, Pearson Education.
4. Michael Blaha and James Rambaugh, Object Oriented Modelling and Design with UML, 2nd edition, Eastern Economy Edition.
5. Herbert Schildt, Java: A Beginner's Guide, McGraw Hill Education (India) Private Limited.
6. Bruce Eckle, Thinking in Java, Prentice Hall.
7. Joshua Bloch, Effective Java, Createspace Independent Pub.
8. Kathy Siera, Head First Java, O'Reilly Media

School of Computer and Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA-II)

Course Code: CA514 Title of the Course: Data Base Management systems

L-T-P: 3-0-0 Credits : 3

Prerequisite Course / Knowledge (If any): Programming Methodology, Data and File Structures, Operating Systems

Course Outcomes (COs)

After completion of this course successfully, the students will be able to……

CO-1: Explain the fundamentals of relational database management systems (Understand)

CO-2: Explain the relational data model, ER model and relational algebra (Understand)

CO-3: Design the ER models for database applications (Create)

CO-4: Prepare SQL queries from the ER models (Apply)

CO-5: Evaluate the database design aspects by considering normalization principles (Evaluate)

CO-6: Explain concurrency, recovery, security, integrity, Indexing, Hashing, deadlock handling (Understand)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	3									
CO2	2			1	3							
CO3	3		1		2							
CO4			1	2	3							
CO5		3	2	1								
CO6	1		2		3							

Detailed Syllabus:

UNIT 1: Introduction: Aims and Objectives, Technology involved and current uses of the technology.
UNIT 2: Data Models: Entity-Relationship model, Network model, Hierarchical model.
UNIT 3: Database design: Normalization principles and their uses. Secondary data storage and retrieval techniques.
UNIT 4: Query Processing: Studies on query processing strategies and cost estimation.
UNIT 5: Transaction Processing: Defining Properties and studies on recovery and concurrency. Security and Integrity.
UNIT 6: Distributed Databases: Introduction, Issues on design, concurrency, recovery, deadlock handling and coordinator selection.

Reference Books:

1. A. Silberschatz, H. F. Korth and S. Sudarshan, *Database Systems & Concepts*, 6th Edition McGrawHill Publications, 1376 pages.
2. R. Elmasri, S. B. Navathe: *Fundamentals of Database Systems*, 7th Edition, Pearson Publication, US, 1168 pages.
3. Stefano Ceri, G. Pellagatti: *Distributed Databases Principles & Systems*, McGrawHill, India, 408 pages.

School of Computer and Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA-II)

Course Code: CA516 Title of the Course: Data Base Management systems lab

L-T-P: 0-0-3 Credits: 1.5

Prerequisite Course / Knowledge (If any): Programming Methodology, Data and File Structures, Operating Systems

Course Outcomes (COs) (5 to 8)

After completion of this course successfully, the students will be able to……

CO-1: Explain the fundamentals of SQL - Structured Query Language (Understand)

CO-2: Construct SQL queries to create, delete any given table structures and views including a database (Create)

CO-3: Apply set of commands to pose queries, insert new tuples, and update/delete existing tuples (Apply)

CO-4: Create nested SQL queries to retrieve /update data from/to multiple tables (Create)

CO-5: Demonstrate how to write SQL code using Triggers, Assertions, etc. (Apply)

CO-6: Demonstrate Create, Modify and delete virtual tables called views and use them wherever required (Apply)

CO-7: Experiment PL/SQL codes using cursors, anonymous PL/SQL blocks, stored procedures, and functions (Analyze)

CO-8: Develop a real-time web application using any favorite programming language of his / her choice. The application will use the relational database management system like Oracle, MySQL or any other. The student will use the knowledge gained in the course outcomes (CO-2 to CO-7) for this purpose (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	3		2								
CO2			1	2	3							
CO3		1	2	3								
CO4			2	3	1							
CO5		1		3	2							
CO6		1	3		2							
CO7	3		2	1								
CO8										1	2	3

Detailed Syllabus:

Unit 1: Introduction to SQL. Features of SQL, DDL Statements and DML commands.
Unit 2: Writing simple SQL queries using DDL statements and DML commands
Unit 3: Introduction to inner, outer and natural joins. Writing nested queries and correlated nested queries to retrieve and update the data.
Unit 4: Writing SQL queries using EXISTS, NOT EXISTS, explicit join operation, aggregate functions, group by and having classes.
Unit 5: Creating virtual tables (views). Using views in SQL queries.
Unit 6: PL/SQL programming
Unit 7: Mini project: Implement a real-time web application which makes use of database concepts

Reference Books:

1. R. Elmasri, S. B. Navathe: *Fundamentals of Database Systems*, 7th Edition, Pearson Publication, US, 1168 pages.

2. Raghu Ramakrishnan, Johannes Gehrke: *Database management systems,* McGrawHill, Singapore, 1098 pages.

School of Computer and Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA-III)

Course Code: CA 511 Title of the Course: Algorithms

L-T-P: 3-1-0 Credits : 4

Prerequisite Course / Knowledge (If any): Data Structures in under graduate level, discrete mathematical structures, knowledge of sorting algorithms and basic search strategies

Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

CO-1: Assess the inherent structure/hardness of a problem (Evaluate)

CO-2: Select an appropriate strategy to solve a problem (Understand)

CO-3 Design an algorithm that suits the time complexity requirements of the problem. (Create)

CO-4: Estimate the time and space complexities of an algorithm along with the necessary mathematical proofs when necessary. (Evaluate)

CO-5: Devise algorithms by choosing appropriate data structures (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1		2	3	1								
CO2			2	3	1							
CO3	3		1	2								
CO4	1	2	3									
CO5		3	2		1							

Detailed Syllabus

UNIT-I: Analysis of Algorithms: Asymptotic Notation; Best, worst and average case analysis of algorithms; Solving recurrence relations using substitution method, generating functions, Master's theorem etc. Warm-up to complexity analysis: Heap data structure, priority queue application, Best, worst and average case analysis of a few sorting algorithms like heap sort, insertion, bubble, selection, counting and radix sort algorithms. Strategies for problem solving

UNIT-II: Divide and Conquer strategy: Time complexity analysis for Merge Sort and Quick Sort Algorithms

UNIT-III: Greedy strategy: Theoretical foundation of greedy strategy: Matroids Algorithms for solving problems like Knapsack Problem (Fractional), Minimum Spanning Tree problem; Shortest Paths, Job Scheduling, Huffman's code etc along with proofs of corrections and complexity analysis

UNIT-IV: Dynamic Programming strategy: Identify situations in which greedy and divide and conquer strategies may not work. Understanding of optimality principle. Technique of memorization. Applications to problems like Coin change, 0/1 and 0/n- Knapsack, Shortest Paths, Optimal Binary Search Tree (OBST), Chained Matrix Multiplication, Traveling Salesperson Problem (TSP) etc.

UNIT-V: Backtracking and Branch & Bound strategies: State space tree construction, traversal techniques and solving problems like 0/1 and 0/n knapsack, TSP, Applications of Depth First Search: Topological sorting, Finding strongly connected components and game problems.

UNIT-VI: Theory of NP-Completeness: Complexity classes of P, NP, NP-Hard, NP-Complete, Polynomial reductions, Cook's theorem. Discussion of problems: Satisfiability (SAT), CNF-SAT, Min-Vertex Cover, Max-Clique, Graph Coloring, NP-Completeness proofs.

Reference Books:

1. Introduction to Algorithms-T.Cormen, C.E.Leiserson, R.L.Rivest, PHI, 3rdEdition 2009.
2. Algorithms- R.Johnsonbaugh and M.Schaefer, Pearson, 2004.
3. Fundamentals of Algorithmics - G.Brassard and P.Bratley, PH, 1996
4. The Algorithm Design Manual- Steven S. Skiena, Springer, 2009

School of Computer and Information Sciences

Name of the Academic Program: **Master of Computer Applications (MCA-III)**

Course Code : CA 512 Title of the Course : **Software Engineering**

L-T-P : 3-0-0 Credits : 3

Prerequisite Course / Knowledge (If any): It is expected that the students must have done at least one programming course at undergraduate/postgraduate level

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO-1: Explain the models of software development process (Understand)
- CO-2: Evaluate the appropriateness of different models of software development for their application in various domains (Evaluate).
- CO-3: Apply the requirements engineering to software systems. (Apply)
- CO-4: Describe Software Architectures (understand).
- CO-5: Assess the applicability of software architectures for various combinations of non-functional requirements (Evaluate level).
- CO-6: Apply object oriented and structured and structured paradigms to design software systems (Apply).
- CO-7: Apply testing strategy to test software applications (Apply).

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1		2			3						
CO2			1	2	3							
CO3		1								2		3
CO4	1	2	3									
CO5			1		2			3				
CO6		3	2		1							
CO7		3	2		1							

Detailed Syllabus:

UNIT-I: Introduction to Software Engineering

Need of software engineering, systems engineering, challenges in software engineering, Software process models, quality characteristics of software systems, Ethics in Software Engineering.

UNIT-II: Requirements Engineering

Requirements engineering process, requirements specification, structured and object-oriented analysis

UNIT-III: Software Design

Architectural design, detailed design, Structured and object-oriented design, user interface design

UNIT-IV: Software Testing

Verification, Validation, testing techniques, Testing Process

UNIT-V: Tools and Evolution

CASE Tools, Reverse engineering, Reengineering and Configuration management.

Reference Books:

1. Ian Sommerville (2016), "*Software Engineering*",10th Edition, Pearson Education Limited, Global Edition.

2. Roger S Pressman, Bruce R Maxim(2015), "*Software Engineering, A Practitioner's Approach*", 8th Edition, TataMcGraw Hill, Indian Edition

3. Grady Booch, James Rumbaugh, Ivor Jacobson(2005), "*The Unified Modeling Language User Guide*", 2nd Edition, Addison Wesley Professional.US

School of Computer and Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA-III)

Course Code : CA515 Title of the Course: **Software Engineering Lab**

L-T-P : 0-0-3 Credits : 1.5

Prerequisite Course / Knowledge (If any): It is expected that the students must have done at least one programming course at undergraduate level.

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO-1: Create user stories (Create).
- CO-2: Develop test plans for test first development (Create).
- CO-3: Design & develop the stories (Create).
- CO-4: Create the documentation (Create).
- CO-5: Develop Software requirements specification document (Create).
- CO-6: Apply object oriented and structured paradigm (Apply).
- CO-7: Generate test reports (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1			1	2								3
CO2	3			2	1							
CO3			1	2								3
CO4				1						2		3
CO5			2	3	1							
CO6	3	1	2									
CO7			3	1	2							

Detailed Syllabus:

For a given case study/problem statement, the following deliverables are to be realized

- Define stories
- Identify tasks and develop test plan for stories/task (with the help of specifications)
- Design and develop increments
- Test the increments and release the increment
- Apply object oriented and structured modelling
- Implement the case study for plan driven approach by writing use case specification, designing the system and implementing the same.

Reference Books:

1. Ian Sommerville (2016), "*Software Engineering*",10th Edition, Pearson Education Limited, Global Edition

2. Roger S Pressman, Bruce R Maxim(2015), "*Software Engineering, A Practitioner's Approach*", 8th Edition, TataMcGraw Hill, Indian Edition

Detailed Syllabus:

UNIT-I: ERP Overview: Integrated management information systems, Supply chain management, Integrated data model, Benefits of ERP, Evolution of ERP and Modern enterprise, BPR (Business Process Reengineering) & ERP, Business modelling for ERP

UNIT-II: Customer Service

UNIT-III: Production Planning and execution

UNIT-IV: Purchasing and goods receipt

UNIT-V: Financial and other metrics

UNIT-VI: ERP Packages

UNIT VII: Case studies, Insurance industry, Banking industry, Pharmaceutical industry, Health care, Consumer products, Retail industry, University, Transport Industry, Telecom industry, Public Sector Industry

UNIT VIII:: Current Trends in Implementations; Hardware / Network selections ; Data Management requirements; Integration requirements and techniques; Other Non-functional requirements necessary for implementations

Reference Books:

Learning Resources/Text Books:

1. 3. Infor ERP - LN Workbook

2. 4. Enterprise Resource Planning – Alexis Leon, Tata McGraw Hill

Suggested Reading

3. INTRODUCTION TO MATERIALS MANAGEMENT 6/E (English) 6th Edition Authors: Tony Arnold / Stephen Chapman Publishers: Pearson India
4. Manufacturing Planning and Control for Supply Chain Management (APICS / CPIM Certification Edition) Authors: Thomas E. Vollmann, CFPIM, William L. Berry, D. Clay Whybark, and F. Robert Jacobs Publishers: McGraw Hill Education
5. Accounting Handbook 6th Edition Authors: Shim Siegal Publishers: Barrons Educational Series
6. Operations Strategy 4th Edition Authors: Nigel Slack and Mike Lewis Publishers: Pearson
7. CONCEPTS IN ENTERPRISE RESOURCE PLANNING Authors: Ellen F. Monk, Bret J. Wagner
8. Enterprise Resource Planning – Ashim Raj Singla, Cengage Learning

School of Computer and Information Sciences

Name of the Academic Program: **Master of Computer Applications (MCA-III)**

Course Code : CA 513 Title of the Course: **Software Project Management**

L-T-P : 3-0-0 Credits : 3

Prerequisite Course / Knowledge (If any): It is expected that the students must have done at least one programming course at undergraduate/postgraduate level

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO-1: Explain the Project Management Organization (Understand)
- CO-2: Apply software planning and Project Management techniques. (Apply)
- CO-3: Assess the Quality of software (Evaluate).
- CO-4: Explain the Risk Management Strategies (Understand)
- CO-5: Apply Agile Management Techniques (Apply).
- CO-6: Apply Project Management concepts and techniques to an IT Project (Apply).

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1							3	1			2	
CO2					1			2				3
CO3		2		1				3				
CO4		2	3					1				
CO5	1				2			3				
CO6	2				1			3				

Detailed Syllabus:

UNIT-I: Introduction to Project Management: Objectives, Managing Projects, Programs & Portfolios, Relationship between Portfolios programs and Projects, Project Management Office, Managing Triple

Constraints, Organization Structure, Functional Organization, Projectized Organization, Matrix organization, Project Life-Cycle Vs Product-Life Cycle

UNIT-II: Project Management Techniques: Overview, Objectives, Project Selection Techniques, Work Breakdown Structure, Gantt Chart, Gantt Chart relationships, Network Diagram, Critical Path Method, Program Evaluation and Review Techniques, PERT

UNIT-III: Quality & Risk Management: Earned value management, Quality Planning, Quality Assurance & Quality Control, Quality management concepts, Responsibility Assignment Matrix and RACI, Risk Categorization, Qualitative risk analysis, Planning risk responses, Types of contract, Stakeholder Engagement Assessment Matrix.

UNIT-IV: Agile and Scrum: Objectives, Agile Manifesto, Agile Project Management Life Cycle, Agile Project Management Framework, SCRUM, SCRUM roles, Best practices of SCRUM, Overview of other agile methodologies- Extreme Programming, Feature Driven Development, Dynamic Systems Development Method etc.

UNIT-V: Project Management Tools: Introduction to Agile Project Management Tools and Generic Project Management Tools (eg. MS Project, Project.net, Kanban etc). Creating a project from Blank Project Template, Existing template and Existing project. Create project from Microsoft Excel Workbook, Microsoft SharePoint Task List, Create Base calendar, Set Baseline, create task dependencies with links, Create Project Milestones, manage resources and schedule, add constraints, Identify Critical Path using MS Project, Reports and Dashboards

Note: A case study has to be selected by Students and the activities of Project Management cycle are to be implemented using a Project Management Tool.

Suggested Readings:

Books:

1. Kathy Schwalbe, "Information Technology Project Management", 7th edition, Thomson Course Technology
2. Joel Henry, "Software Project Management, a Real-World Guide to Success", Pearson Education,

e-resources:

1. https://www.opentextbooks.org.hk/system/files/export/15/15694/pdf/Project_Management_15694.pdf
2. https://www.kornev-online.net/ITIL/Mcgraw.Hill.Software_Project_Management_2nd_Edition.pdf
3. https://www.edutechlearners.com/download/Software%20Project%20Management.pdf
4. http://seu1.org/files/level6/IT270/PM%20E-%20Book%20(6%20Edition).pdf