

## Master of Computer Application

Name of Program	<b>Master of Computer Application</b>
Abbreviation	<b>MCA</b>
Duration	<b>2 Years</b>
Eligibility Criteria	<p>Passed BCA/ Bachelor Degree in Computer Science Engineering or equivalent Degree.</p> <p style="text-align: center;">OR</p> <p>Passed B.Sc./ B.Com./ B.A. with Mathematics at 10+2 Level or at Graduation Level (with additional bridge Courses as per the norms of the concerned University).</p>
Objective of Program	The core objective of the MCA programme is to prepare the students for productive career in software industry and academia by providing an outstanding environment of teaching and research in the core and emerging areas of the discipline.
Program Outcome	<p><b>PO1 : Fundamental Knowledge Enrichment</b>  Program trains students with the core computer science and Information Technology (IT) knowledge domains. It also makes students capable of using core concepts in the conceptualization of domain specific application development.</p> <p><b>PO2 : Critical Thinking Development</b>  The program develops the skills of critical thinking, problem solving, evaluative learning of various techniques, and understanding the essence of the problem.</p> <p><b>PO3 : Advanced Emerging Technology Awareness</b>  The program trains students with the latest technologies that is being used in the industry. The continuous syllabi review adds value to the program for the outgoing students and make them ready to face challenging demands of the industry.</p> <p><b>PO4 : Advanced Tools Usage</b>  The program teaches the students to apply the advanced tools to solve real world problems.</p> <p><b>PO5 : Nurturing Project Planning and Management Capabilities</b>  The program trains students for designing and conceptualizing the software architecture, planning and managing the product development process of complex and live software projects. It also makes students understand the decision making for selection of an appropriate project management capabilities.</p> <p><b>PO6 : Real World Problem / Project Development</b>  Real world project provides the candidates exposure to work in the challenging and demanding environment of the industry. The project development training makes students employable and industry ready.</p> <p><b>PO7 : Team Work and Leadership Development</b>  Trains students to work in a team and also to take leadership of the of the project management team.</p>
Program Specific Outcomes	<p>PSO1 : Develop and strengthen the fundamental core concepts that are required to solve complex problems</p> <p>PSO2 : Develop the professional and entrepreneurship skills that needs independent logical and analytical thinking, teamwork and leadership</p> <p>PSO3 : Nurture the students to investigate for the design and development of a workable solution for a real world problem</p> <p>PSO4 : Develop students for self-learning and practicing challenging</p>

problem solution  
 PSO5 : Train students to apply managerial skills to develop business applications.  
 PSO6 : Train students to use recent computer science and application domain specific knowledge  
 PSO7 : Train students to take-up the real world challenges to develop workable solution to a domain specific problem  
 PSO8 : Inculcate the passion for continuous learning and doing research for making a successful professional career

Mapping between POs and PSOs		PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
	PO1								
	PO2								
	PO3								
	PO4								
	PO5								
	PO6								
	PO7								

Medium of Instruction English

Program Structure Semester 1

Course Code	Title	Teaching per week		Course Credits	University Examination		Internal Marks	Total Marks
		Theory	Practical		Duration	Marks		
101	Relational Database Management System	4	0	4	3 Hrs	70	30	100
102	Object Oriented Programming Methodology	4	0	4	3 Hrs	70	30	100
103	Cloud Computing	4	0	4	3 Hrs	70	30	100
104	Computer Network	4	0	4	3 Hrs	70	30	100
105	Data Structures and Design and Analysis of Algorithms	4	0	4	3 Hrs	70	30	100
106	Programming Skills - I	0	3	3	2 Hrs	70	30	100
107	Programming Skills – II	0	3	3	2 Hrs	70	30	100
108	Programming Skills – III	0	4	4	2 Hrs	70	30	100
Total		20	10	30	21 Hrs	560	240	800

Program Structure Semester 2

Course Code	Title	Teaching per week		Course Credits	University Examination		Internal Marks	Total Marks
		Theory	Practical		Duration	Marks		
201	Artificial Intelligence	4	0	4	3 Hrs	70	30	100
202	Front End Technologies	4	0	4	3 Hrs	70	30	100
203	Programming in .NET	4	0	4	3 Hrs	70	30	100
204	Python Programming Language	4	0	4	3 Hrs	70	30	100
205	iOS/Android	4	0	4	3 Hrs	70	30	100
206	Programming Skills – IV	0	3	3	2 Hrs	70	30	100
207	Programming Skills – V	0	2	2	2 Hrs	70	30	100
208	Programming Skills – VI	0	2	2	2 Hrs	70	30	100
209	Programming Skills – VII	0	3	3	2 Hrs	70	30	100
Total		20	10	30	23 Hrs	630	270	900

# MCA 1<sup>st</sup> Semester

## Course: 101: Relational Database Management System

Course Code	<b>101</b>								
Course Title	<b>Relational Database Management System</b>								
Credit	4								
Teaching per Week	4 Hrs								
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)								
Review / Revision	June 2020								
Purpose of Course	Give fundamental knowledge of Database Fundamentals like Keys & Normalisation, Oracle Database Server Architecture and Working knowledge of SQL & PL/SQL in Oracle.								
Course Objective	To acquaint the students with Database Fundamentals like Keys & Normalisation in general and Oracle Architecture in particular. Also, to get working knowledge of SQL and PL/SQL programming								
Course Outcome	<p>CO1: Students will be able to understand and evaluate the role of database management systems in IT applications within an organization.</p> <p>CO2: Students will be able to normalise any real-life database and hence they will be able to design and implement properly structured databases of real-world applications.</p> <p>CO3: Students will be able to write queries using Structured Query Language (SQL) and programs using PL/SQL.</p> <p>CO4: Students will understand how Oracle Database works and the importance of various components of Oracle Database.</p> <p>CO5: Students will understand administration and security enforcement of Oracle Database. This will help them in pursuing higher studies and career in Database Administration.</p>								
Mapping between COs with PSOs		PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
	CO1								
	CO2								
	CO3								
	CO4								
	CO5								
Pre-requisite	Nil								
Course Content	<p><b>Unit 1: Database Computing Models</b></p> <p>1.1. Notion of Keys (Super Key, Candidate Key, Primary Key, Foreign Key)</p> <p>1.2. Referential Integrity Constraint</p> <p>1.3. Functional Dependencies</p> <p>1.4. Normalization using Functional Dependencies</p> <p>1.5. Normalization using Multivalued Dependencies</p> <p>1.6. Normalization using Join Dependencies</p> <p><b>Unit 2: Overview of Oracle Database Server Architecture</b></p> <p>2.1. Architecture of Oracle Database and Oracle Instance</p> <p>2.2. Overview of Physical and Logical Structures</p> <p>2.3. Oracle Server Start-up and Shutdown</p> <p>2.4. Creating Database</p> <p><b>Unit 3: Oracle Tools and Utilities</b></p> <p>3.1. SQL</p> <p>3.2. PL/SQL Procedural Extension.</p> <p>3.2.1. Overview, PL/SQL data types &amp; Control Structures</p> <p>3.2.2. Cursors</p> <p>3.2.3. Stored Procedures &amp; Functions</p> <p>3.2.4. Database Triggers</p> <p>3.2.5. Package Creation</p>								

	<p><b>Unit 4: Database Administration</b></p> <p>4.1. Managing Users</p> <p>    4.1.1. User Authentication Methods</p> <p>        4.1.1.1. Password Authentication</p> <p>        4.1.1.2. O.S Authentication</p> <p>    4.1.2. User Configuration Setup</p> <p>        4.1.2.1. Profiles</p> <p>        4.1.2.2. Default Table space</p> <p>        4.1.2.3. Temporary Table space</p> <p>    4.1.3. Resource Management</p> <p>        4.1.3.1. Quotas</p> <p>    4.1.4. Working with user database account</p> <p>        4.1.4.1. Creating, Modifying and deleting user account</p> <p>        4.1.4.2. Changing password</p> <p>4.2. Backup &amp; Recovery</p> <p><b>Unit 5: Database Security</b></p> <p>5.1. Authentication</p> <p>5.2. Privileged Accounts &amp; Privileges</p> <p>5.3. Object Security</p> <p>5.4. System security</p> <p>5.5. Database Roles</p> <p>5.6. Database Auditing</p> <p>[Self-Study]</p> <p>Export &amp; Import Tools, Overview of Grid Based Database</p> <p>**Computing, Calling External Routines from PL/SQL</p>
Reference Books	<ol style="list-style-type: none"> <li>1. Oracle 9i PL/SQL Programming -Scott Urman- Oracle Press</li> <li>2. Oracle DBA Fundamentals-I - Oracle Press</li> <li>3. Effective PL/SQL: - Apress</li> <li>4. Expert Oracle Database Architecture 9i and 10g-Tom Kyte- Apress</li> <li>5. Effective Oracle by Design - Peter Norton - Tom Kyte-Oracle Press</li> <li>6. Oracle 24X 7 Tips and Techniques - Venkat Devraj–Oracle Press</li> <li>7. Expert Oracle Database 11gAdministration – Alpati- Wiley Student Edition</li> <li>8. Fundamentals of Database Management System- Gille neon-Wiley Student Edition</li> <li>9. SQL &amp; PL/SQL for Oracle 11g Black Book - Deshpande-McGraw Hill</li> <li>10. Beginning Oracle Database 11g Administration from novice to professional-Iggy Fernandez - Apress/Springer</li> <li>11. Oracle PL/SQL-Benjamin Rosenweig &amp; Elena Silvestrova-4/e, Pearson</li> <li>12. Database Systems Using Oracle: A simplified guide to SQL &amp; PL/SQL- Shah Nilesh- PHI</li> <li>13. Learning Oracle SQL &amp; PL/SQL: A Simplified Guide- Chatterjee, Rajeeb C- PHI</li> </ol>
Teaching Methodology	Class Work, Discussion, Self Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment 70% External Assessment

## Course: 102: Object Oriented Programming Methodology

Course Code	102								
Course Title	Object Oriented Programming Methodology								
Credit	4								
Teaching per Week	4 Hrs								
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidaysetc.)								
Review / Revision	June 2020								
Purpose of Course	This course introduces the concepts of object-oriented programming and skills necessary for developing programs in C++.								
Course Objective	<ol style="list-style-type: none"> <li>1. To make students understand concepts of object-oriented paradigm</li> <li>2. To make students develop C++ programs</li> <li>3. To make students learn capabilities of an object-oriented programming language</li> </ol>								
Course Outcome	<p>CO1- Articulate the principles of Object Oriented Problem solving and programming.</p> <p>CO-2-To demonstrate the differences between traditional imperative design and object Oriented Design</p> <p>CO-3-Outline the essential features and elements of C++ programming language.</p> <p>CO-4- To grasp and apply the concepts of class, method, constructor, abstraction, inheritance and Static Polymorphism.</p> <p>CO-5- To understand and apply Dynamic Polymorphism in real world applications.</p> <p>CO-6- To implement Genericity through the usage of Templates.</p> <p>Mapping between Cos and PSOs</p>								
Mapping between COs with PSOs		PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
	CO1								
	CO2								
	CO3								
	CO4								
	CO5								
	CO6								
Pre-requisite	Nil								
Course Content	<p><b>Unit 1: C++ Basics</b></p> <ol style="list-style-type: none"> <li>1.1 Data Types</li> <li>1.2 Pointers             <ol style="list-style-type: none"> <li>1.2.1 Pointer Arithmetic</li> <li>1.2.2 Array of Pointers</li> <li>1.2.3 Dynamic Array</li> </ol> </li> <li>1.3 ios Class</li> <li>1.4 Input and Output</li> <li>1.5 Manipulators</li> </ol> <p><b>Unit 2: Introduction to Object Oriented Programming</b></p> <ol style="list-style-type: none"> <li>2.1 Structure, classes and Objects</li> <li>2.2 Encapsulation &amp; Data Hiding</li> <li>2.3 Constructors</li> <li>2.4 Friend Functions</li> <li>2.5 Inline Functions</li> <li>2.6 Dynamic Object Creation &amp; Destruction</li> <li>2.7 Static Members</li> <li>2.8 this Pointer</li> <li>2.9 Destructors</li> </ol> <p><b>Unit 3: Object Oriented Properties</b></p> <ol style="list-style-type: none"> <li>3.1 Introduction to Object Oriented Properties</li> <li>3.2 Abstraction</li> </ol>								

	<p>3.3 Polymorphism  3.3.1 Operator Overloading  3.3.2 Function Overloading &amp; Type Conversions</p> <p>3.4 Inheritance  3.4.1 Types of Inheritance  3.4.2 Constructor &amp; Destructor calls during Inheritance</p> <p>3.5 Dynamic Polymorphism  3.5.1 Overriding  3.5.2 Virtual Functions  3.5.3 Abstract Class</p> <p><b>Unit 4: Data Files and Exception Handling</b>  4.1 Streams  4.2 File Types and Modes  4.3 File Pointers &amp; their manipulations  4.4 Sequential Input &amp; Output operations  4.5 Random access  4.6 Error handling during File operations  4.7 Exception Handling</p> <p><b>Unit 5: Generic Programming and C++ Standard Template Library (STL)</b>  5.1 Template Classes  5.2 Template Functions  5.3 Implementation of Object-Oriented Properties on Template Classes  5.4 STL  5.4.1 Algorithms  5.4.2 Containers  5.4.3 Functions  5.4.4 Iterators</p>
Reference Books	<ol style="list-style-type: none"> <li>1. The C++ Programming Language, Stroustrup, Addison Wesley</li> <li>2. The Complete Reference C++, Schildt, Tata McGraw Hill</li> <li>3. OOP in Turbo C++, Robert Lafore, Galgotia Publication</li> <li>4. C++ Primer, Lippman, Addison Wesley</li> <li>5. Object Oriented Programming with ANSI and Turbo C++, Kamthane, Pearson Education</li> <li>6. Thinking in C++, Bruce Eckel, Pearson</li> <li>7. Object Oriented Modelling &amp; Design, Rumbaugh, PHI</li> <li>8. Object Oriented Analysis &amp; Design with Application, Grady Booch, LPE</li> <li>9. Standard C++ with Object Oriented Programming, Paul S. Wang, Thomson</li> <li>10. C++ Primer Plus, Stephan Prata, Addison Wesley</li> <li>12. Programming with ANSI C++, Bhushan Trivedi, Oxford University Press</li> </ol>
Teaching Methodology	Class Work, Discussion, Self Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment 70% External Assessment

### Course: 103: Cloud Computing

Course Code	<b>103</b>								
Course Title	<b>Cloud Computing</b>								
Credit	4								
Teaching per Week	4 Hrs.								
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)								
Review / Revision	June 2020								
Purpose of Course	The purpose of the course is to make student capable of implementing the concepts, methods and tools of Cloud Computing								
Course Objective	The objective of the course is to provide comprehensive and in-depth knowledge of Cloud Computing Concepts, technologies, architecture, applications and implementation.								
Course Outcome	<p>CO1 : Explain students about the cloud and cloud computing, History &amp; Evolution ,Properties &amp; Characteristics, Advantages &amp; Disadvantages of cloud computing.</p> <p>CO2: Explain students about various service models and deployment models</p> <p>CO3: To provide students a foundation of different cloud service models IAAS,PAAS and SAAS so that they are able to use Cloud Computing Services in real world problem</p> <p>CO4: Understanding the concepts of cloud infrastructure security, data security and storage, Access control and authentication in cloud.</p> <p>CO5: Train students to use AWS and Microsoft Azure</p> <p>CO6: Explain students in brief about BigTable and Firebase</p>								
Mapping between COs with PSOs		PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
	CO1								
	CO2								
	CO3								
	CO4								
	CO5								
	CO6								
Pre-requisite	Basics of DBMS, Web Development & HTML, Networking								
Course Outcome	To give basic knowledge of cloud computing, its architecture and its benefits and how to deploy applications on well-known cloud platforms								
Course Content	<p><b>Unit 1: Introduction to Cloud &amp; its architecture</b></p> <p>1.1 Introduction &amp; Definitions</p> <p>1.2 Cloud Computing (NIST)</p> <p style="padding-left: 20px;">1.2.1 History &amp; Evolution</p> <p style="padding-left: 20px;">1.2.2 Properties &amp; Characteristics</p> <p style="padding-left: 20px;">1.2.3 Advantages &amp; Disadvantages</p> <p>1.3 Cloud Architecture overview</p> <p><b>Unit 2: Cloud Computing Models</b></p> <p>2.1 Cloud computing Stack</p> <p style="padding-left: 20px;">2.1.1 Comparison with traditional architecture</p> <p>2.2 Service Models</p> <p style="padding-left: 20px;">2.2.1 Infrastructure as a Service (IaaS)</p> <p style="padding-left: 20px;">2.2.2 Platform as a Service (PaaS)</p> <p style="padding-left: 20px;">2.2.3 Software as a Service (SaaS)</p> <p>2.3 Deployment Models</p> <p style="padding-left: 20px;">2.3.1 Public Cloud</p>								

	<p>2.3.2 Private Cloud  2.3.3 Hybrid Cloud  2.3.4 Community Cloud</p> <p><b>Unit 3: Cloud Service Models</b></p> <p>3.1 Infrastructure as a Service (IAAS)  3.1.1 Introduction to Virtualization  3.1.1.1 Hypervisors, Virtual Machine, Machine Image  3.1.2 Resource Virtualization  3.1.2.1 Server, Storage, Network  3.1.3 Amazon EC2, Eucalyptus</p> <p>3.2 Platform as a Service (PAAS)  3.2.1 Introduction to SOA  3.2.2 Cloud Platform  3.2.2.1 Computing  3.2.2.2 Storage  3.2.3 Introduction to Microsoft Azure  3.2.4 Introduction to Salesforce's Force.com</p> <p>3.3 Software as a Service (SAAS)  3.3.1 Introduction  3.3.2 Web Service &amp; WebOS</p> <p><b>Unit 4: Cloud Security</b></p> <p>4.1 Infrastructure Security  4.2 Data Security and Storage  4.3 Identity and Access Management (IAM)  4.4 AccessControl  4.5 Authentication in Cloud</p> <p><b>Unit 5: Cloud Databases (DBaaS)</b></p> <p>5.1 AWS SimpleDB &amp; RDS  5.2 Azure Table Service &amp; SQL Azure  5.3 Introduction to BigTable  5.4 Introduction to Firebase</p>
Reference Books	<ol style="list-style-type: none"> <li>1. Cloud Computing Principles and Paradigms (Wiley) Rajkumar Buyya, James Broberg, Andrzej M. Goscinski</li> <li>2. Cloud Computing: Principles, Systems and Applications Nikos Antonopoulos, Lee Gillam (Springer)</li> <li>3. Enterprise Cloud Computing: Technology, Architecture, Applications Gautam Shroff - Cambridge University Press</li> <li>4. Cloud and Virtual Data Storage Networking Greg Schulz - Auerbach</li> <li>5. Cloud Security: A Comprehensive Guide to Secure Cloud Computing Ronald L Krutz, Russel Dean Vines (John Wiley &amp; Sons)</li> <li>6. Cloud Computing (David Crookes - TMH Education)</li> <li>7. Cloud Computing Bible Barrie Sosinsky (Wiley India)</li> <li>8. Cloud Computing: Implementation, Management and Security (James F Ransome, John W Rittinghouse - CRC Press)</li> </ol>



	<p>9. Amazon Cloud Computing with Java (Aditya Yadav - Lulu.com)</p> <p>10. Grid and Cloud Database Management Fiore, Sandro, Aloisio, Giovanni - Springer</p> <p>11. Building a Database Cloud for Dummies Michael Wessler John Wiley &amp; Sons</p>
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30 % internal assessment and 70% external assessment

### Course: 104: Computer Network

Course Code	<b>104</b>								
Course Title	<b>Computer Network</b>								
Credit	4								
Teaching per Week	4 Hrs								
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)								
Review / Revision	June 2020								
Purpose of Course	This course aims towards learning fundamentals of computer network. The course teaches students about the various network technologies and popular network protocols								
Course Objective	1.To make students learn about computer network fundamentals 2.To make students familiar with services offered at each layer of the network protocol stack 3.To make students learn various protocols at data link layer , network layer, and transport layer of network.								
Course Outcome	CO1 : Understand students the fundamental aspects of the computer networks. CO2 : Explain and help students to learn fundamentals network protocols at data link layer , network layer and transport layer. CO3 : Explore students the services offered at each layer of the network protocol stack. CO4 : Train students to implement various error control, flow control, routing algorithms and security algorithms fall under data link layer , network layer and transport layer. CO5 : Explore students the concepts of Security, digital certificate, Public key Infrastructure, and similar security schemes.								
Mapping between COs with PSOs		PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
	CO1								
	CO2								
	CO3								
	CO4								
	CO5								
Pre-requisite	Nil								
Course Content	<p><b>Unit 1: Introduction to Data Communication</b></p> 1.1 Introduction to networks, Internet and its application 1.2 Network Structure 1.3 Network Architecture 1.4 The OSI Reference model & services 1.5 The TCP/IP Reference model and Comparison with OSI Model 1.6 Concepts of data transmission 1.6.1 Guided and unguided Transmission media. PSTN 1.7 Multiplexing & switching techniques 1.8 ISDN (Integrated Service Digital Network) <p><b>Unit 2: Data Link Layer</b></p> 2.1 MAC Sub layer 2.1.1 Multiple Access Protocols 2.1.2 Ethernet 2.1.3 LAN protocols & IEEE standards for LAN 2.1.4 Fibre Optic & Satellite networks 2.2 Data Link Layer protocols 2.3 Error detection & correction <p><b>Unit 3: Upper Layers</b></p>								

	<ul style="list-style-type: none"> <li>3.1 Network <ul style="list-style-type: none"> <li>3.1.1 Routing Algorithms</li> <li>3.1.2 Congestion Control Algorithm</li> <li>3.1.3 Internetworking</li> </ul> </li> <li>3.2 Transport Layer <ul style="list-style-type: none"> <li>3.2.1 Connection Management</li> </ul> </li> <li>3.3 Concepts of Session Layer</li> </ul> <p><b>Unit 4: The Presentation Layer</b></p> <ul style="list-style-type: none"> <li>4.1 Data Compression Technique</li> <li>4.2 Cryptography</li> <li>4.3 Symmetric Key Algorithms</li> <li>4.4 Public – Key Algorithms &amp; management of Public Keys</li> <li>4.5 Digital Signatures and Communications security</li> </ul> <p><b>Unit 5: The Application Layer</b></p> <ul style="list-style-type: none"> <li>5.1 Electronic Mail</li> <li>5.2 Virtual Terminals <ul style="list-style-type: none"> <li>General Purpose Applications</li> </ul> </li> </ul> <p>[Self Study] Virtual LAN</p>
Reference Books	<ol style="list-style-type: none"> <li>1. Networking Complete- 1st Edition 2002, BPB Publication (Text Book)</li> <li>2. Data Communication and Networking: Forouzan, TMH</li> <li>3. Computer Networks - A. S. Tanenbaum - Prentice-Hall</li> <li>4. Computer Networks and Distributed Processing - Martin J. - Pretice-Hall</li> <li>5. Local Area Networks: An Introduction - Stalling, William - Mc-Millan Publishing Co.</li> <li>6. Computer Networks: Protocols, Standards and Interfaces - Black – Prentice-Hall</li> <li>7. Data Networks: Concepts Theory and Practices - Black - PHI</li> <li>8. N/W Architecture - Comer - Prentice-Hall</li> </ol>
Teaching Methodology	Class Work, Discussion, Self Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment 70% External Assessment

## Course: 105: Data Structures and Design and Analysis of Algorithms

Course Code	105								
Course Title	Data Structures and Design and Analysis of Algorithms								
Credit	4								
Teaching per Week	4 Hrs								
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidaysetc.)								
Review / Revision	June 2020								
Purpose of Course	This course introduces the various data structures and algorithms involving these data structures and their logical implementation. Students also will be able to understand complex data structures like trees and their use in various Applications								
Course Objective	<ol style="list-style-type: none"> <li>1. To learn fundamental data structures like arrays, stacks, lists.</li> <li>2. To learn complex data structures like trees.</li> <li>3. To learn and compare various sorting techniques.</li> <li>4. To learn analysis of algorithms</li> </ol>								
Course Outcome	<p>CO1. Explain the futures of various non-primitive data structure and their applications. Train students for algorithms to create, insert, delete and traversing various data structure.</p> <p>CO2. Explain the basics of computational complexity analysis and big O &amp; theta notation that help for fundamental research in algorithm analysis. Students can select the efficient algorithm after analyzing a problem and identifying it's the computing requirements.</p> <p>CO3. Explain insight of basic technique for algorithm such as Divide &amp; Conquer, Greedy method, Backtracking &amp; Branch &amp; Bound</p> <p>CO4. Students will be able to work with various searching and sorting techniques and compare its computational complexity.</p> <p>CO5. Train students with to work with Hash structure, proper hash method and collision detection techniques.</p>								
Mapping between COs with PSOs		PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
	CO1								
	CO2								
	CO3								
	CO4								
	CO5								
Pre-requisite	C programming Language								
Course Content	<p><b>Unit 1: Non-Primitive Data structures.</b></p> <ol style="list-style-type: none"> <li>1.1 Arrays - its storage structures &amp; operations</li> <li>1.2 Stacks - operations and its applications in Recursion, Polish expressions etc.</li> <li>1.3 Queues - Types of queues, operations and its applications.</li> <li>1.4 Linked lists - Types of linked list, operations and its applications.</li> <li>1.5 Trees - Concept and Definitions, Operations, linked &amp; threaded storage representation of Binary Trees.</li> <li>1.6 Applications of Trees - The manipulation of Arithmetic expressions, Symbol-table construction, Syntax Analysis etc.</li> </ol> <p><b>Unit 2: Analysis of Algorithms</b></p> <ol style="list-style-type: none"> <li>2.1 Asymptotic: Big-O and Theta</li> </ol> <p><b>Unit 3: Basic techniques &amp; example algorithms for</b></p> <ol style="list-style-type: none"> <li>3.1 Divide &amp; Conquer method</li> <li>3.2 Greedy method</li> </ol>								

	<p>3.3 Backtracking 3.4 Branch &amp; Bound</p> <p><b>Unit 4: Searching and Sorting</b> 4.1 Sequential, Binary 4.2 Search Trees:- Height, Balanced tree, 2-3, tree, red-black trees weight-balanced trees 4.3 Sorting 4.3.1 Internal sorting - Insertion, Selection, Quick, 2-way merge and Heap 4.3.2 External sorting - k-way merging, Balanced merge and poly phase Merge</p> <p><b>Unit 5: Hashing</b> 5.1 Hash Tables 5.2 Hash functions 5.2.1 Division method 5.2.2 Multiplication method</p> <p>[Self Study] Graphs – Creation and Traversal</p>
Reference Books	<ol style="list-style-type: none"> <li>1. An Introduction to Data Structures with applications - Trembley – McGraw Hill</li> <li>2. Theory and Problems of Data Structure – Lipschutz Semour – McGraw Hill</li> <li>3. Algorithms + Data Structure Programs - Wirth, Niclaus - PHI.</li> <li>4. Fundamentals of Data Structures, Horwitz, E. and Sahni S. - Computer Science Press.</li> <li>5. The Art of Computer Programming, Vols. 1-2, Knuth D. - Addison Wesley.</li> <li>6. Data Structures and Algorithms - Aho A.V., Hopcroft and Ullman - Addison Wesley</li> <li>7. Data Structure &amp; "C" Programming - Vanwyte CJ - Addison Wesley.</li> <li>8. Data Structures, Algorithms And Object Oriented Programming – Tata McGraw Hill edition Geogory L. Heileman.</li> <li>9. Data Structures and the Standard Template Library - William J. Collins, Tata McGraw Hill edition.</li> <li>10. Programming with C++ and Data Structures - Maria Litvin &amp; Gary Litvin, Vikas Publishing House Pvt. Ltd.</li> <li>11. Data Structures using C &amp; C++ - Y. Langsam Moshe J. Angensterin &amp; A.M. Tenenbaum</li> <li>12. Data Structures and Algorithms in C++ - Adam Drozdek, Thomson Learning</li> <li>13. Data Structures &amp; Program Design in C - Robert Kruse, C.L. Tondo, Brnceleing PHI Pvt Ltd.</li> <li>14. Data Structures and Algorithms in Java, Lafore, Pearson</li> <li>15. Data Structures and Algorithm Analysis in Java, Mark Allen Weiss, Pearson</li> <li>16. Data Structures and Algorithms in Java, Micheal T Goodrich, Roberto Tamassia, Wiley</li> </ol>
Teaching Methodology	Class Work, Discussion, Self Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment 70% External Assessment

**Course: 106: Programming Skills - I**

Course Code	<b>106</b>
Course Title	<b>Programming Skills – I</b>
Credit	5
Teaching per Week	5 Hrs
Minimum weeks per Semester	15 (Including lab work, examination, preparation, holidays etc.)
Review / Revision	June 2020
Purpose of Course	Give fundamental knowledge of Database Models, Oracle Database Server Architecture and Working knowledge of SQL & PL/SQL in Oracle.
Course Objective	To acquaint the students with Client Server Architecture in general and Oracle Architecture in particular. Also, to get working knowledge of SQL and PL/SQL programming
Pre-requisite	None
Course Outcome	After studying the course, students will be able to understand how Oracle Database works and the importance of various components of Oracle. This course will also help students to appreciate the role of a database administrator. After successful completion, students will be able to manage Oracle database and will be able to write codes in SQL & PL/SQL necessary for an application.
Course Content	Practical based on paper no: 101 (RDBMS)
Reference Books	None
Teaching Methodology	Lab. Work
Evaluation Method	30% Internal assessment and 70% External Assessment

**Course: 107: Programming Skills - II**

Course Code	<b>107</b>
Course Title	<b>Programming Skill – II</b>
Credit	3
Teaching per Week	3 Hrs
Minimum weeks per Semester	15 (Including lab work, examination, preparation, holidays etc.)
Review / Revision	June 2020
Purpose of Course	This course helps students practically implement the concepts of object oriented programming using C++.
Course Objective	1. To make students practically learn concepts of object-oriented paradigm 2. To make students develop and code C++ programs.
Pre-requisite	Nil
Course Outcome	After studying the course, students will be able to practically solve common and complex programming problems using object-oriented paradigm. This course also helps students learn practical implementation of data files and operations upon them using object-oriented approach.
Course Content	Practical based on paper no: 102 (OOPM)
Reference Books	None
Teaching Methodology	Lab. Work
Evaluation Method	30% Internal assessment and 70% External Assessment

**Course: 108: Programming Skills - III**

Course Code	108
Course Title	Programming Skill – III
Credit	3
Teaching per Week	3 Hrs
Minimum weeks per Semester	15 (Including lab work, examination, preparation, holidays etc.)
Review / Revision	June 2020
Purpose of Course	This course introduces the various data structures and algorithms involving these data structures and their practical implementation using JAVA programming language. Students also will be able to understand and write JAVA programs using complex data structures like trees.
Course Objective	<ol style="list-style-type: none"><li>1. To practically learn implementation of fundamental data structures like arrays, stacks, lists using JAVA</li><li>2. To learn implementing complex data structures like trees using JAVA</li><li>3. To learn and compare various sorting techniques practically</li><li>4. To learn analysis of algorithms practically.</li></ol>
Pre-requisite	C programming Language
Course Outcome	After studying the course, students will be able to use data structures and their application in sorting, searching and comparison of algorithms. Students will also learn analysis of the algorithms.
Course Content	Practical based on paper no: 105 (Data Structures and Design and Analysis of Algorithms)
Reference Books	None
Teaching Methodology	Lab. Work
Evaluation Method	30% Internal assessment and 70% External Assessment

## MCA 2<sup>nd</sup> Semester

### Course: 201: Artificial Intelligence

Course Code	<b>201</b>								
Course Title	<b>Artificial Intelligence</b>								
Credit	4								
Teaching per Week	4 Hrs								
Medium of Instruction	English								
Minimum weeks per Semester	15 (Including Classwork, examination, preparation, holidays etc.)								
Effective From	June 2020								
Purpose of Course	The purpose of the course is to make the student capable of implementing the concepts, methods, and tools of Artificial Intelligence and learn their implementation in Knowledge-Based Systems Course Objective To acquaint students with concepts of Artificial Intelligence and its applications.								
Course Objective	To make students acquainted with concepts of Artificial Intelligence and its applications.								
Course Outcome	<p>CO1 : Explain students the insight of the historical and fundamental aspects the artificial intelligence.</p> <p>CO2 : Train students to represent declarative knowledge in the form of symbolic knowledge through various Knowledge Representation (KR) techniques like First Order Predicate Logic (FOPL), Semantic Network, Conceptual Graphs, Scripts, and Frames.</p> <p>CO3 : Train students to apply various searching algorithms fall under informed and uninformed search methods to solve complex problem of AI domain.</p> <p>CO4 : Explain and train students to deal with the uncertainty that inherently lies within many AI problem.</p> <p>CO5 : Expose the students with the analysis and development process of the knowledge based system development.</p> <p>CO6 : Explain students to utilize the AI problemsolving techniques in the advanced AI problem domain like Natural Language Processing (NLP) and Computer Vision (CV)</p>								
Mapping between COs with PSOs		PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
	CO1								
	CO2								
	CO3								
	CO4								
	CO5								
	CO6								
Pre-requisite	Basics of Mathematics, Data Structures								
Course Content	<p><b>Unit 1:</b></p> <p>1.1 Introduction to Artificial Intelligence</p> <p style="padding-left: 20px;">1.1.1 Definition of Artificial Intelligence</p> <p style="padding-left: 20px;">1.1.2 History of Artificial Intelligence</p> <p style="padding-left: 20px;">1.1.3 Application of Artificial Intelligence</p> <p style="padding-left: 20px;">1.1.4 Introduction to Knowledge-Based System</p> <p>1.2 Turing Problem</p> <p>1.3 Knowledge Representation</p> <p style="padding-left: 20px;">1.3.1 knowledge and Knowledge Base</p> <p style="padding-left: 20px;">1.3.2 First Order Predicate Logic (FOPL)</p> <p style="padding-left: 20px;">1.3.3 Inference Rules</p> <p><b>Unit 2:</b></p> <p>2.1 Structured Knowledge Representation</p> <p style="padding-left: 20px;">2.1.1 Associative network and Conceptual graphs</p> <p style="padding-left: 20px;">2.1.2 Frames and Scripts</p> <p style="padding-left: 20px;">2.1.3 Conceptual Dependencies</p>								



	<p>2.2 Searching</p> <ul style="list-style-type: none"> <li>2.2.1 Search Problem</li> <li>2.2.2 Initial State, action, transition model, goal test, the cost function</li> <li>2.2.3 Uninformed Search <ul style="list-style-type: none"> <li>2.2.3.1 Depth First Search</li> <li>2.2.3.2 Breadth-First Search</li> <li>2.2.3.3 Iterative Deepening Search</li> </ul> </li> <li>2.2.4 Informed Search <ul style="list-style-type: none"> <li>2.2.4.1 Heuristics</li> <li>2.2.4.2 A* Search</li> <li>2.2.4.3 Minimax</li> <li>2.2.4.5 Hill-Climbing Method</li> <li>2.2.4.6 Constraint Satisfaction Search</li> </ul> </li> </ul> <p><b>Unit 3:</b></p> <ul style="list-style-type: none"> <li>3.1 Uncertainty <ul style="list-style-type: none"> <li>3.1.1 Probability</li> <li>3.1.2 Conditional Probability</li> <li>3.1.3 Baye’s Rule</li> <li>3.1.4 Joint Probability</li> <li>3.1.5 Probability Rules</li> </ul> </li> <li>3.2 Introduction to Hidden Markov Model</li> </ul> <p><b>Unit 4:</b></p> <ul style="list-style-type: none"> <li>4.1 Knowledge Acquisition <ul style="list-style-type: none"> <li>4.1.1 Knowledge gathering</li> <li>4.1.2 Learning Models <ul style="list-style-type: none"> <li>4.1.2.1 Introduction to Supervised Learning</li> <li>4.1.2.2 Introduction to Unsupervised Learning</li> <li>4.1.2.3 Reinforcement Learning</li> </ul> </li> <li>4.1.3 Performance of Learning Model</li> </ul> </li> </ul> <p><b>Unit 5:</b></p> <ul style="list-style-type: none"> <li>5.1 Expert System</li> <li>5.2 Characteristics of Expert System</li> <li>5.3 Architecture of Expert System</li> <li>5.4 Application of AI in Natural Language Processing</li> <li>5.5 Application of AI in Computer Vision</li> </ul>
Reference Books	<ol style="list-style-type: none"> <li>1. Artificial intelligence, 3<sup>rd</sup> Edition, Kevin Knight, Elaine Rich, B. Shivashankar Nair, McGraw Hill</li> <li>2. Russell Stuart Jonathan and Norvig Peter, Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice-Hall, 2010</li> <li>3. A First Course in Artificial Intelligence, Deepak Khemani, McGraw Hill</li> <li>4. Introduction to artificial intelligence, <u>Akerkar, Rajendra</u>, PHI Learning</li> <li>5. Foundation of Artificial Intelligence and Expert Systems by V.S. Janakiraman, K. Sarukesi, P. Gopalakrishnan, Mc Millan</li> <li>6. Expert Systems Principles and Programming (3rd Edition) by Giarratano &amp; Riley, Thomson (Vikas Publishing House)</li> </ol>
Teaching Methodology	Classwork, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	<p>30% Internal assessment based on class attendance, participation, class test, quiz, assignment, seminar, internal examination, etc.</p> <p>70% External based on semester end University examination</p>

## Course: 202: Front-End Technologies

Course Code	202								
Course Title	Front-end Technologies								
Credit	4								
Teaching per Week	4 Hrs								
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)								
Review / Revision	June 2020								
Purpose of Course	To understand the concepts of HTML, CSS, Front-end Scripting technologies								
Course Objective	To teach the concepts of Front-end Scripting and its practical applications.								
Course Outcome	<p>CO1 : Explain students the insight of the fundamental aspects of the static web technology like HTML , CSS and Javascript</p> <p>CO2 : Train students to represent declarative knowledge in the Bootstrap and it's components , with help of this students can learn static web designing .</p> <p>CO3 : Train students to understand various Javascript based technologies like JQuery , JSON and AJAX.</p> <p>CO4 : Explain and train students to understand basic concepts of Angular JS.</p> <p>CO5 : Expose the students to the various UI technology and real world use of it..</p> <p>CO6 : After completion of this course, the student will be able to design and develop web pages and Interactive UI for Web Applications</p>								
Mapping between COs with PSOs		PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
	CO1								
	CO2								
	CO3								
	CO4								
	CO5								
	CO6								
Pre-requisite	Basic concepts of HTML, Web & Programming skills								
Course Content	<p><b>Unit 1: Fundamentals of Web Technology</b></p> <p>1.1 HTML</p> <p>    1.1.1 Basic HTML tags</p> <p>    1.1.2 HTML Forms</p> <p>1.2 HTML5</p> <p>    1.2.1 HTML5 new elements</p> <p>    1.2.2 HTML5 Form elements</p> <p>    1.2.3 HTML5 Attributes</p> <p>    1.2.4 Canvas</p> <p>    1.2.5 Video and Audio</p> <p>    1.2.6 Web storage</p> <p>    1.2.7 Geolocation</p> <p>    1.2.8 HTML 5 APIs</p> <p>1.3 CSS3</p> <p>    1.3.1 Introduction to CSS3</p> <p>    1.3.2 Selectors and Classes</p> <p>    1.3.3 Font and Text effect</p> <p>    1.3.4 Colors, Gradients, Background Images, and Masks</p> <p>    1.3.5 Border and Box effects etc.</p> <p>    1.3.6 Embedding Media</p> <p>1.4 JavaScript</p> <p>    1.4.1 Fundamentals of JavaScript</p> <p>    1.4.2 Syntax of JavaScript</p> <p>    1.4.3 Use of JavaScript in HTML</p> <p>    1.4.4 Validation using JavaScript</p> <p>    1.4.5 DOM</p> <p><b>Unit 2: Introduction to Bootstrap</b></p> <p>2.1 What is Bootstrap</p>								

	<p>2.1.1 What is Responsive web page  2.1.2 Advantages and features of Bootstrap  2.1.3 Setup Environment  2.1.4 Apply bootstrap to Application  2.2 Bootstrap with CSS  2.2.1 Grid system  2.2.2 Typography  2.2.3 Code, table, forms, buttons, image, responsive utilities etc.  2.3 Bootstrap components  2.3.1 What is Bootstrap components  2.3.2 Advantages of components  2.3.3 Types of Bootstrap components  2.3.3.1 Glyphicons, Drop downs, button group, input groups navigation, pagination etc.</p> <p><b>Unit 3: JQuery</b>  3.1 Introduction to JQuery  3.1.1 Syntax, Attributes, Selectors, Events  3.2 JQuery Effects  3.2.1 Hide/Show, Fade, Slide, Animation etc.  3.2.2 JQuery with HTML  3.3 Traversing  3.4 JQuery and AJAX</p> <p><b>Unit 4: AJAX and JSON</b>  4.1 Ajax Basics  4.1.1 HTTP Request and Response Fundamentals  4.1.2 The XMLHttpRequest Object XMLHttpRequest Methods  4.1.3 XMLHttpRequest Properties  4.1.4 Cross-Browser Usage Sending a Request to the Server  4.1.5 Server-Side Processing Expanding and Contracting Content  4.1.6 Form Validation  4.1.7 Ajax-Based Database Querying using any one server site scripting language  4.2 JSON  4.2.1 JSON Basics  4.2.2 Syntax  4.2.3 Datatype, Parse, Stringify, Objects, Array  4.2.4 Use of JSON using any one server site scripting</p> <p><b>Unit 5: Introduction to Angular JS</b>  5.1 Introduction to AngularJS  5.2 Directives, Expressions, Controllers, Filters, Tables, Html DOM  5.3 Modules, Forms, Includes, Views  5.4 Angular SQL  5.5 AngularJS Applications</p>
Reference Books	<ol style="list-style-type: none"> <li>1. Responsive Web Design with HTML5 and CSS3 By Ben Frain - Packt Publishing Ltd.</li> <li>2. HTML, CSS &amp; JavaScript Web Publishing in One Hour a Day, Sams Teach Yourself by Laura Lemay, Rafe Colburn, Jennifer Kyrnin – Sams Publication</li> <li>3. Training Guide Programming in HTML5 with JavaScript and CSS3 (MCS5): 70-480 by Glenn Johnson - Pearson Education</li> <li>4. Learning Bootstrap by Aravind Shenoy, Ulrich Sossou - Packt Publishing Ltd.</li> <li>5. Professional AngularJS by Valeri Karpov, Diego Netto - John Wiley &amp; Sons</li> <li>6. Ajax: Creating Web Pages with Asynchronous JavaScript and XML - Edmond Woychowsky - Prentice Hall</li> </ol>

Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30 % internal assessment and 70% external assessment

### Course: 203: Programming in .NET

Course Code	203								
Course Title	Programming in .NET								
Credit	4								
Teaching per Week	4 Hrs.								
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)								
Review / Revision	June 2020								
Purpose of Course	This course is an introduction to students to understand fundamentals of .NET technology. The course also gives students an idea about VB.NET Programming. The course also explains the concept of ASP.NET								
Course Objective	<ol style="list-style-type: none"> <li>1. To make students understand .NET Technology</li> <li>2. To make students understand VB.NET Programming</li> <li>3. To make students understand the importance of ASP.NET</li> </ol>								
Course Outcome	<p>CO-1 Study .Net Architecture.</p> <p>CO-2- Design and develop console and window based .NET application.</p> <p>CO-3-To learn basic syntax and implement small applications in C# programming language.</p> <p>CO-4-Create and manipulate GUI components in C#.</p> <p>CO-5-Create applications in C# using Object Oriented Properties.</p> <p>CO-6-Design and implement Database connectivity using ADO.Net.</p> <p>CO-7-Identify and resolver problems in C# applications.</p>								
Mapping between COs with PSOs		PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
	CO1								
	CO2								
	CO3								
	CO4								
	CO5								
	CO6								
	CO7								
Pre-requisite	Nil								
Course Content	<p><b>Unit1: Overview of Microsoft .NET Platform</b></p> <ol style="list-style-type: none"> <li>1.1 Introduction to Building Blocks of .Net Platform</li> <li>1.2 Overview of .Net Assemblies</li> <li>1.3 Common Type System</li> <li>1.4 Common Language Specification</li> <li>1.5 Common Language Runtime</li> <li>1.6 Exploring an Assembly (ildasm)</li> <li>1.7 Platform Independent Nature of .Net</li> <li>1.8 Base Class Libraries</li> </ol> <p><b>Unit2: Overview of C#</b></p> <ol style="list-style-type: none"> <li>2.1 Literals, Variables, Data Types</li> <li>2.2 Operators</li> <li>2.3 Expressions and Looping</li> <li>2.4 Constants, Arrays, Array Class, List</li> <li>2.5 String, String Builder</li> <li>2.6 Boxing and UnBoxing</li> <li>2.7 Events, Errors and Exceptions</li> </ol> <p><b>Unit3: Object Oriented Aspects of C#</b></p> <ol style="list-style-type: none"> <li>3.1 Creating Classes, Encapsulation, Object Construction &amp; Destruction</li> <li>3.2 Inheritance</li> <li>3.3 Polymorphism</li> <li>3.4 Abstraction</li> <li>3.4 Interfaces and Abstract Classes</li> <li>3.5 Delegates</li> </ol>								

	<p><b>Unit4: Application Development</b>  4.1 Creating Windows Forms with Events and Controls  4.2 Menu Creation  4.3 Inheriting Windows Forms  4.4 SDI and MDI Application  4.5 Dialog Boxes (Modal and Modeless)  4.6 Validating Controls</p> <p><b>Unit5: Accessing Data</b>  5.1 ADO.Net  5.1.1 Data Adapter  5.1.2 Data Set  5.1.3 Typed Data Set  5.2 Using Stored Procedures  5.3 Handling Exceptions  5.4 LINQ</p> <p>[Self Study]  Report Generation, Deployment</p>
Reference Books	<ol style="list-style-type: none"> <li>1. .NET Framework Essentials, Hoand Lam, Thuan L. Thai, O'REILLY</li> <li>2. Microsoft .NET Framework 4.5 Quickstart Cookbook, Jose Luis Latorre Millas, PACKT Publishing</li> <li>3. Pro C# 5.0 and the .NET 4.5 Framework, Andrew Troelsen, Apress</li> <li>4. C# IN DEPTH, Jon Skeet, Manning Publications</li> <li>5. Beginning C# 7 Programming with Visual Studio 2017, Benjamin Perkins, wrox</li> <li>6. Illustrated C#, Daniel Solis, Cal Schrottenboer, Apress</li> <li>7. The C# Programmer's Study Guide, Ali Asad, Hamza Ali, Apress</li> </ol>
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30 % internal assessment and 70% external assessment

**Course: 204: Python Programming Language**

Course Code	<b>204</b>								
Course Title	<b>Python Programming Language</b>								
Credit	4								
Teaching per Week	4 Hrs								
Medium of Instruction	English								
Minimum weeks per Semester	15 (Including Classwork, examination, preparation, holidays, etc.)								
Effective From	June 2020								
Purpose of Course	The Python language is used popularly among the people working in the area of Machine Learning (ML), Data Analytics, Artificial Intelligence, Web Application, and even the people working on Desktop Applications. This course imparts to the students understanding of Python programming language.								
Course Objective	<ol style="list-style-type: none"> <li>1. To make students understand Python Language</li> <li>2. To make students understand various components of language and its Working</li> <li>3. To prepare students to understand the use of language in the area of AI, ML, Data Analytics etc.</li> <li>4. To make students understand the important</li> </ol>								
Course Outcome	<p>CO1: Students will be able to Write, Test and Debug Python Programs.</p> <p>CO2: Students will be able to Implement Conditionals and Loops, use functions and represent Compound data using Lists, Tuples and Dictionaries in Python programs.</p> <p>CO3: Students will be able to Read and write data from &amp; to files in Python and develop Real World Application.</p> <p>CO4: Students will be able to Design and implement programs to solve real-world problems using Python Programming Language.</p> <p>CO5: Students will learn essential packages like NumPy and Matplotlib, which are necessary for Machine Learning, Data Analytics, and AI.</p>								
Mapping between COs with PSOs		PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
	CO1								
	CO2								
	CO3								
	CO4								
	CO5								
Pre-requisite	Basic programming skills								
Course Content	<p><b>Unit 1: Fundamentals of Python</b></p> <ol style="list-style-type: none"> <li>1.1 Features of Python</li> <li>1.2 Python's Integrated Development and Learning Environment (IDLE)</li> <li>1.3 Python identifiers</li> <li>1.4 Python Operators</li> <li>1.5 Python Datatypes             <ol style="list-style-type: none"> <li>1.5.1 Numeric: integer, float, complex</li> <li>1.5.2 Sequence: list, tuple, range</li> <li>1.5.3 Sets</li> <li>1.5.4 Texts</li> <li>1.5.5 Binary: bytes, bytearray</li> <li>1.5.6 Iterator</li> <li>1.5.7 Mapping: dictionary</li> </ol> </li> <li>1.6 Input/Output in Python</li> </ol> <p><b>Unit 2: Program Logic</b></p> <ol style="list-style-type: none"> <li>2.1 Control Structures             <ol style="list-style-type: none"> <li>2.1.1 If, if..else, nested if, shorthand if, shorthand if..else</li> <li>2.1.2 while loop</li> <li>2.1.3 for loop</li> <li>2.1.4 break, continue, pass</li> </ol> </li> <li>2.2 Functions in Python</li> </ol>								

	<p>2.2.1 Function declaration  2.2.2 Passing arguments to function  2.2.3 Return values  2.2.4 Variable scope and name space  2.2.5 Lambda function  2.2.6 Recursive function  2.2.7 In-built function</p> <p><b>Unit 3: Files and Modules</b>  3.1 File handling  3.1.1 Reading and writing to a file  3.1.2 Creation of new file  3.1.3 Deletion of a file  3.2 Python Modules  3.2.1 Creation of module  3.2.2 Importing a module  3.2.3 Date &amp; time module</p> <p><b>Unit 4: Exceptions, Class and Objects</b>  4.1 Exception Handling  4.1.1 try, catch, finally  4.2.2 Multiple error handling: except  4.2.3 Throwing a particular error: raise  4.2 Classes and Objects  4.2.1 Creation of class and object  4.2.2 The __init__() function  4.2.3 Self parameter  4.2.4 Modifying the property of a class  4.2.5 Inheritance &amp; Encapsulation</p> <p><b>Unit 5: Python Packages</b>  5.1 NumPy  5.1.1 Installing numpy  5.1.2 Numpy Array: dtype, shape, reshape, ndim, itemsize, empty, zeros, ones, fromiter, arrange, linspace  5.1.3 Indexing and slicing, broadcasting  5.1.4 Array manipulation: changing shapes, transpose, changing dimension, joining and splitting arrays, adding and removing elements  5.1.5 Mathematical functions and matrix library  5.2 Introduction to Matplotlib  5.2.1 Installing Matplotlib  5.2.2 Components of a plot  5.2.3 Drawing a plot  5.2.4 Drawing scatter diagram</p>
Reference Books	<ol style="list-style-type: none"> <li>1. Python Programming, Anurag Gupta, G Biswas,, Mc Graw Hill</li> <li>2. Exploring Python, Timothy A. Budd, McGraw Hill Publication</li> <li>3. Core Python Programming, R. Nageswara Rao, Dreamtech Press</li> <li>4. Learning Python, 5th Edition, Mark Lutz, O'Reilly Media</li> <li>5. Python Projects, Laura Cassell, Alan Gauld, Wrox Publication</li> <li>6. NumPy: Beginner's Guide, 3rd Edition, Ivan Idris, Packt Publishing</li> <li>7. NumPy Essentials, Leo Chin, Tanmay Dutta, Packt Publishing</li> <li>8. Matplotlib 2.x By Example, Allen Yu, Claire Chung, Aldrin Yim, Packt Publishing</li> </ol>
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment based on class attendance, participation, class test, quiz, assignment, seminar, internal examination, etc. & 70% External based on semester end University examination



### Course: 205: iOS Development using Swift

Course Code	205								
Course Title	iOS Development using Swift								
Credit	4								
Teaching per Week	4 Hrs.								
Minimum weeks/ Semester	15 (Including Class work, examination, preparation, holidays etc.)								
Review / Revision	June 2020								
Purpose of Course	This course will help the students to understand the fundamental as well as advanced concepts of iOS Programming. The course also provides them the skills necessary to develop an iOS Application from scratch to deploying it on the App Store.								
Course Objective	The objective of the course is - 1. To understand the iOS ecosystem and tools for creating iOS applications 2. To explain advanced level concepts in iOS application design and development 3. To impart knowledge of Swift programming language								
Course Outcome	CO1: Understand the iOS ecosystem and Xcode IDE. Understand the life cycle of iOS application and how to implement it in MVC. Understand Foundation framework in iOS. CO2: Understand the syntax, and semantics of the Swift programming language. Expose the students to CLI applications with Swift. CO3: Understand the UIKit framework in iOS. Understand the usage and working of UI elements in iOS application. Understanding various types of design and their implementation. CO4: Understand data persistence in mobile application. Understand working with files in iOS. Expose students with implementation and usage of database in an iOS application. CO5: Understand the usage and data extraction of sensors in iPhone. Expose the students with Location and MapKit Framework in iOS to build map-based applications. Expose the students with ad-hoc and App Store application deployment.								
Mapping between COs with PSOs		PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
	CO1								
	CO2								
	CO3								
	CO4								
	CO5								
Pre-requisite	Knowledge of C, C++ and SQL								
Course Content	<b>Unit 1: Introduction to iOS ecosystem</b> 1.1. Introduction to Xcode IDE 1.1.1. Environment setup 1.1.2. Editors, Storyboard and Simulator 1.2. Application Life-Cycle 1.3. View Controller Life-Cycle 1.4. Info.plist and App Permissions 1.5. MVC in iOS 1.6. Introduction to iOS App Frameworks 1.6.1. Foundation Framework 1.6.2. UIKit Framework 1.6.3. Swift and SwiftUI  <b>Unit 2: Introduction to Swift Programming Language</b> 2.1. Simple Values – Constant and Variable 2.2. Control Flow 2.3. Functions and Closures 2.4. Objects and Classes 2.5. Enumerations and Structures								

	<p>2.6. Protocols and Extensions  2.7. Error Handling  2.8. Generics</p> <p><b>Unit 3: UIKit: View Controllers, Views and Controls</b>  3.1. Text Views: UILabel, UITextField, UITextView  3.2. Controls: UIButton, UIDatePicker, UIPageControl, UISegmentedControl, UISlider, UIStepper, UISwitch  3.3. Content Views: UIActivityIndicatorView, UIImageView, UIPickerView, UIProgressView  3.4. Bars: UINavigationController, UISearchBar, UIToolbar, UITabBar  3.5. Images and Video: UIImagePickerController  3.6. Container View Controllers: UINavigationController, UITabBarController  3.7. Container Views: Table Views, Collection Views  3.8. Alerts: UIAlertController  3.9. Gestures: UITapGestureRecognizer, UIPinchGestureRecognizer, UIRotationGestureRecognizer, UISwipeGestureRecognizer, UIPanGestureRecognizer</p> <p><b>Unit 4: Data Persistence and Networking</b>  4.1. UserDefaults  4.2. FileManager  4.3. SQLite Framework  4.4. Core Data Framework  4.5. JSON Parsing  4.6. Working with URL and URL classes</p> <p><b>Unit 5: App Services and App Deployment</b>  5.1. Core Motion – Accelerometer, Gyroscope, Pedometer, Magnetometer, Altitude  5.2. Core Location – CLLocationManager, CLLocation, Authorization  5.3. MapKit – Map Fundamentals, Map Coordinates, Annotations and Overlays  5.4. How to deploy an Ad-Hoc app – (diawi)  5.5. Publishing an app to the AppStore</p>
Reference Books	<ol style="list-style-type: none"> <li>1. Apple Documentation [ <a href="http://developer.apple.com/documentation">developer.apple.com/documentation</a> ]</li> <li>2. The Swift Programming Language by Apple Inc. [ <a href="http://swift.org/documentation">swift.org/documentation</a> ]</li> <li>3. Hacking with Swift by Paul Hudson [ <a href="http://hackingwithswift.com">hackingwithswift.com</a> ]</li> <li>4. iOS 13 Programming Fundamental with Swift by Matt Neuberg, O’Reilly</li> <li>5. Programming iOS 13 by Matt Neuberg, O’Reilly</li> <li>6. Mastering Swift 5: Deep dive into the latest edition of the Swift programming language, 5th Edition, Packt Publishing Limited</li> <li>7. SwiftUI Essentials - IOS Edition: Learn to Develop IOS Apps Using SwiftUI, Swift 5 and Xcode 11 by Neil Smyth, Payload Media, Inc.</li> <li>8. Beginning iOS 13 &amp; Swift App Development: Develop iOS Apps with Xcode 11, Swift 5, Core ML, ARKit and more by Greg Lim</li> <li>9. Pro iPhone Development with Swift 5: Design and Manage Top Quality Apps by Wallace Wang, Apress</li> </ol>
Teaching Methodology	Class work, Discussion, Self-study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

## Course: 205: Android Application Programming

Course Code	205								
Course Title	Android Application Programming								
Credit	4								
Teaching per Week	4 Hrs.								
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)								
Review / Revision	June 2020								
Purpose of Course	The purpose of the course is to make student capable of implementing the concepts, methods and tools of mobile applications development using Android.								
Course Objective	To provide a thorough introduction to the Android environment and tools for creating Android applications.								
Pre-requisite	Basic concepts of Operating Systems, Programming skills in core Java and Knowledge of object-oriented programming is desirable. Knowledge of XML format is helpful.								
Course Outcome	<p>CO1. Train students for installing and using the Android Developer's Toolkit such as SDK Manager, Android Virtual Device, Dalvik Debug Monitor Service (DDMS), Android Debug Bridge (ADB) and make them capable to develop, manage and maintain application(Apps) using Android and publish Apps on Google Play.</p> <p>CO2. Understand the Android software stack &amp; program building blocks like activities, services and notifications to use them effectively to develop Android applications.</p> <p>CO3. Explain working with AndroidManifest, xml resources like layout and values and incorporate xml resources with Java code.</p> <p>CO4. Explain the use of java library for views, controls, menus, dialogs, graphics, media, storage, SQLiteDatabase etc... to make interactive applications.</p> <p>CO5. Train students to build Android app that access the database &amp; other resources on web server using web services.</p> <p>CO6. Train students for managing Telephony and Message services. Make them capable of using Location Manager and Google MAP related APIs for App and create the professional applications.</p>								
Mapping between COs with PSOs		PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
	CO1								
	CO2								
	CO3								
	CO4								
	CO5								
	CO6								
Course Content	<p><b>Unit 1: Introduction to Android</b></p> <p>1.1 Evolution of Android and OHA</p> <p>1.2 Architecture of Android OS</p> <p>1.3 Introduction to Android SDK</p> <p>1.4 Android Development tools: SDK Manager, Android Emulator, Android Virtual Device, Dalvik Debug Monitor Service (DDMS), Android Debug Bridge (ADB)</p> <p>1.5 Anatomy of Android App: AndroidManifest.xml, Resources &amp; R.java, Assets, Layouts &amp; Drawable Resources</p> <p><b>Unit 2: Working with User Interface in with Android Activity</b></p> <p>2.1 Widgets: Button, ImageButton, EditText, CheckBox, ToggleButton, RadioButton, RadioGroup, ProgressBar, Text Fields, ListView, Spinner</p> <p>2.2 Designing UI Layouts: LinearLayout, RelativeLayout, TableLayout</p> <p>2.3 Toast and Dialogs: AlertDialogs, TimePicker, DatePicker</p> <p>2.4 Menus: Option menu, Context menu</p> <p>2.5 Adapters for data binding: Array adapter, Cursor adapter</p> <p>2.6 Event callback methods: onClick(), onLongClick() and onTouch()</p>								

	<p><b>Unit 3: Android Application Components</b>  3.1 Activity and Activity lifecycle  3.2 Intents and Intent Filters  3.3 Implicit Intent and Explicit Intent  3.4 Linking of Activity using Intent: startActivity() &amp; startActivityForResult()  3.5 Fragments  3.6 Introduction to Service: life cycle, creation and destroy  3.7 Broadcast receiver &amp; notification</p> <p><b>Unit 4: Data Persistency in Android</b>  4.1 Shared preferences  4.2 File I/O Access: internal and external files  4.3 Working with SQLite Database - Performing insert, update, delete and query operations  4.4 Data access through web services (external databases)  4.5 Working with inbuilt Content Provider: CallLogs, Contacts</p> <p><b>Unit 5: Telephony APIs, Sensors and Leveraging Google APIs</b>  5.1 Telephony APIs      5.1.1 Working with Telephony utilities      5.1.2 Sending and receiving SMS  5.2 Location and Map      5.2.1 Incorporating Location APIs      5.2.2 Incorporating Google map      5.2.3 Geocoding and reverse Geocoding</p>
Reference Books	<ol style="list-style-type: none"> <li>1. Beginning Android 4 Application Development, WEI-MENG LEE, WROX Publication-Wiley-India</li> <li>2. Professional Android 4 Application Development by Reto Meier WROX Publication-Wiley-India</li> <li>3. Android Programming Unleashed, B.M. Harwani, Sams Publishing</li> <li>4. Beginning Android 4 Onur Cinar Apress Publication</li> <li>5. Advanced Android Application Development, Fourth Edition, By Shane Conder, Lauren Darcey, Joseph Annuzzi Jr., Pearson</li> </ol>
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30 % internal assessment and 70% external assessment

**Course: 206: Programming Skills IV**

Course Code	<b>206</b>
Course Title	<b>Programming Skills IV</b>
Credit	2
Teaching per Week	2 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2020
Purpose of Course	To understand the concepts of HTML, CSS, Front-end Scripting technologies
Course Objective	To teach the concepts of Front-end Scripting and its practical applications
Pre-requisite	Basic concepts of HTML, Web & Programming skills
Course Outcome	After completion of this course, the student will be able to design and develop web pages and Interactive UI for Web Applications
Course Content	Practical based on paper no 202 (Front End Technologies)
Reference Books	None
Teaching Methodology	Lab work, Practical Programming Exercises (to be documented in a separate journal), Self-study, and/or Assignment
Evaluation Method	30% Internal assessment & 70% External Assessment

**Course: 207: Programming Skills V**

Course Code	<b>207</b>
Course Title	<b>Programming Skills V</b>
Credit	2
Teaching per Week	2 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2020
Purpose of Course	This course is an introduction to students to understand fundamentals of .NET technology. The course also gives students an idea about C#.NET Programming. The course also explains the concept of C#.NET
Course Objective	1. To make students understand .NET Technology 2. To make students understand C#.NET Programming 3. To make students understand the importance of C#.NET
Pre-requisite	Object Oriented Programming methodology
Course Outcome	After studying the course, students will be able to understand how .NET Technology works and the importance of object-oriented programming. This course will also help students to appreciate the C#.NET programming.
Course Content	Practical based on paper no 203 (Programming in .NET)
Reference Books	None
Teaching Methodology	Lab work, Practical Programming Exercises (to be documented in a separate journal), Self-study, and/or Assignment
Evaluation Method	30% Internal assessment & 70% External Assessment

**Course: 208: Programming Skills VI**

Course Code	<b>208</b>
Course Title	<b>Programming Skills VI</b>
Credit	2
Teaching per Week	2 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2020
Purpose of Course	The purpose of the course is to make student capable of implementing the concepts, methods and tools of mobile applications development using Android/iOS.
Course Objective	To provide a thorough introduction to the Android/iOS environment and tools for creating Android/iOS applications.
Pre-requisite	Basic concepts of Operating Systems, Programmingskills in core Java and Knowledge of object-oriented programming is desirable. Knowledge of XML format is helpful.
Course Outcome	After completion of this course, the student will be capable to develop, manage and maintain mobile device-based application using Android/iOS.
Course Content	Practical based on paper no 204 (Python Programming Language)
Reference Books	None
Teaching Methodology	Lab work, Practical Programming Exercises (to be documented in a separate journal), Self-study, and/or Assignment
Evaluation Method	30% Internal assessment & 70% External Assessment

**Course: 209: Programming Skills VII**

Course Code	<b>209</b>
Course Title	<b>Programming Skills VII</b>
Credit	2
Teaching per Week	2 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2020
Purpose of Course	The purpose of the course is to make student capable of implementing the concepts, methods and tools of mobile applications development using Android/iOS.
Course Objective	To provide a thorough introduction to the Android/iOS environment and tools for creating Android/iOS applications.
Pre-requisite	Basic concepts of Operating Systems, Programmingskills and knowledge of object-oriented programming is desirable
Course Outcome	After completion of this course, the student will be capable to develop, manage and maintain mobile device-based application using Android/iOS.
Course Content	Practical based on paper no. 205 (iOS Programming using Swift / Android Application Programming)
Reference Books	None
Teaching Methodology	Lab work, Practical Programming Exercises (to be documented in a separate journal), Self-study, and/or Assignment
Evaluation Method	30% Internal assessment & 70% External Assessment