SAURASHTRA UNIVERSITY
RAJKOT

(ACCREDITED GRADE "A" BY NAAC)

FACULTY OF SCIENCE

Syllabus for

MASTER OF SCIENCE (IT & CA)

Choice Based Credit System

With Effect From: 2016-17

Program Outcome:

The program is a professional program. It is a technical course. The program is aimed to design in such a way that the students can be absorbed directly in the industries. During their studies, students has to go through a variety of subjects which are directly or indirectly useful in the industry. They have to go through a lot of practical work as well as project work. In the last semester they have to go to the industry for their project work and they have to work as per the requirement of the industry. Hence the program outcome is ob oriented.

Some of the students starts their own development and turns into a small software development industries. So the course is more focus on “Atma nirbhar”.

- **Program Specific Outcome**

During the course, students have to go through a variety of subjects, i.e. computer fundamentals, programming, mathematics and many more. The students are developed in such a way that they are able to work in area where computer works as a tool. They can do analysis, design development and implementation of the work, which help the society.

Master of Science (Information Technology & Computer Application) M.Sc.(IT & CA)

(2 years Full Time: 4 Semester Programme)

Ordinance:

O. M.Sc.(IT & CA) – 1: Candidate seeking admission to the Master of Science (Information Technology & Computer Application) must have a Bachelor's degree of minimum three years duration with 48% or more in the discipline

1. ***B. C. A. with 48% or more***
   OR
2. **B. Sc. with 48% or more**
   OR
3. **B. E. with 48% or more**
   OR
4. **B. Com. (With optional Computer Science) with 48% or more**
   OR
5. **B. Pharm. with 48% or more**
   OR
6. **B. Arch. with 48% or more**
   OR
7. **Any graduate with 48% or more and P.G.D.A.C.A. with 48% or more**
   OR
8. **Any graduate with 48% or more and P.G.D.C.A. with 48% or more**

O. M.Sc.(IT & CA) – 2: The duration of the course is full time two academic years. The examination for the Master of Science (Information Technology & Computer Application) course will be conducted under the semester system. For this purpose the academic year is divided into two semesters. No candidate will be allowed to join any other course simultaneously.

O. M.Sc.(IT & CA) – 3: Candidate who have passed an equivalent examination from any other university or examining body and is seeking admission to the M.Sc. (IT & CA) programme shall not be admitted without producing the eligibility certificate from the Saurashtra University.

O. M.Sc.(IT & CA) – 4: No candidate will be admitted to any semester examination for the Master of Science (Information Technology & Computer Application) unless it is certified by the Head of the Department/ Director of institute.

“That candidate has attended the course of study to the satisfaction of the Head of Department/Director of institute)

O. M.Sc.(IT & CA) – 5: Candidate desirous of appearing at any semester examination of the M.Sc.(IT & CA) programme must forward their application in the prescribed form to the Controller of Examination through Head of Department/Director of Institute on or before the date prescribed.

O. M.Sc.(IT & CA) – 6: No candidate will be permitted to reappear at any semester examination, which he/she has already passed.

O. M.Sc.(IT & CA) – 7: To pass the whole M.Sc.(IT & CA) examination, candidate must clear all the four semester examinations within a period of five years from the date of his/her registration, otherwise candidate has to register him/her self again as a fresh candidate and keep attendance and appear and pass all the four semester examinations .

O. M.Sc.(IT & CA) – 8: There shall be an examination at the end of each four semesters to be known as First semester examination, Second semester examination respectively, at which a student shall appear in the portion of papers practical and Project viva-voce if any, for which he has kept the semester in accordance with the regulations in this behalf.

A candidate whose term is not granted for whatsoever reason shall be required to keep attendance for that semester of terms when the relevant papers are actually taught at the institute.

O. M.Sc.(IT & CA) – 9: A candidate will be permitted to go to the next semester, irrespective he/she is failing in any number of subjects.

O. M.Sc.(IT & CA) – 10: No candidate will be allowed to reappear in examination of any subject which he/she has already passed.

Regulations:

R. M.Sc.(IT & CA) – 1:

The standard of passing the M.Sc. (IT & CA) degree examination will be as under:

(1) To pass any semester examination of the M.Sc. (IT & CA) degree, a candidate must obtain at least 40% marks in the university examination separately in each course of theory and practical.

(2) Class will be awarded based on Earned Grade Point, SGPA and CGPA as per rules of University.

R. M.Sc.(IT & CA) – 2. Marks and credit hours of each course

Marks of Internal examination, university examination and credit hours will be as under:

(1) Total marks of each theory course are 100 (university examination of 70 marks + internal examination of 30 marks).

(2) Marks of each unit in the course are equal (i.e. 14 Marks). Total marks of each course are 14x5=70 for university examination.

(3) Credit hours (lectures) for each unit in the course are equal (i.e. 12 hours). Total credit hours (lectures) of each course are 12x5=60.

(4) Total marks of each practical and project-viva course are 100. No internal

examination of marks in practical and project-viva courses.

R. M.Sc.(IT & CA) – 3. Structure of Question Paper

Question Paper contains 5 questions (each of 14 marks). Every question will be asked from corresponding unit as specified in the syllabus of each course. (i.e. Question-1 from Unit No.1 and remaining questions from their corresponding units)

Every question is divided in four parts like (a), (b), (c) and (d). Part (a) contains four objective type questions (not MCQ) like definition, reason, answer in one line, answer in one word etc., each of one marks and no internal option. Part (b) contains two questions each of two marks and student will attempt any one out of two. Part (c) contains two questions each of three marks and student will attempt any one out of two. Part (d) contains two questions each of five marks and student will attempt any one out of two.

R. M.Sc. (IT & CA) – 4: Following is the syllabus

M.Sc. (IT & CA) (Semester – 1)

SR. NO.	COURSE	No. of LECT./Lab. PER WEEK	CREDIT
1.	**CS – 01** APPLICATION DEVELOPMENT USING ADVANCE JAVA	5	5
2.	**CS – 02** ADVANCE WEB DEVELOPMENT IN Laravel	5	5
3.	**CS – 03** NoSQL DATABASE: MongoDB	5	5
4.	**CS – 04** PRACTICAL - 1 (BASED ON CS-01)	5	5
5.	**CS – 05** PRACTICAL - 2 (BASED ON CS-02 and CS-03)	5	5
6.	**CS – 06** PROJECT DEVELOPMENT (In House)	5	5
Total Credits of Semester – 1			**30**

CS – 01: APPLICATION DEVELOPMENT USING ADVANCE JAVA

course outcomes:

- Learn how to download, setup and configure the Spring Framework
- Explore the Spring Container and Modules
- Understand dependency injection
- Learn aspect oriented programming and how it is used to provide cross cutting concerns
- Understand how Spring deals with transaction management and ORM
- Hibernate: Inheritance mapping collection mapping.
- Understand the HQL.

Pre-Requisites: Students must have strong background of Java programming knowledge and exposure to J2EE technology.

Unit No.	Topics	Details
1	**Basics of Spring, Spring with IDE And IOC container**	• What is Spring • Spring Modules • Spring Application • Spring in Myeclipse • Spring in Eclipse
	Dependency Injection	• Constructor Injection • CI Dependent Object • CI with collection • CI with Map • CI Inheriting Bean • Setter Injection • SI Dependent Object • SI with Collection • SI with Map • CI vs SI • Autowiring • Factory Method
	Spring AOP	• AOP Terminology • AOP Implementations • Pointcut • Advices

2	**Spring JDBC**	- JdbcTemplate Example - PreparedStatement - ResultSetExtractor - RowMapper

		• NamedParameter • SimpleJdbcTemplate
	Spring with ORM And SpEL	• Spring with Hibernate • Spring with JPA • SpEL Examples • Operators in SpEL • variable in SpEL
	Spring 3 MVC and Remoting with Spring	• Spring with RMI • Http Invoker • Hessian • Burlap • Spring with JMS
3	**OXM Frameworks, Spring Java Mail And Web Integration**	• Spring with JAXB • Spring with Xstream • Spring with Castor • Spring with Struts2 • Login and Logout Application
	Basics of Hibernate And Hibernate with IDE	• Hibernate Introduction • Hibernate Architecture • Understanding First Hibernate application • Hibernate in Eclipse • Hibernate in MyEclipse
	Hibernate Application And Hibernate Logging	• Hibernate with annotation • Hibernate Web application • Hibernate Generator classes • Hibernate Dialects • Hibernate with Log4j 1 • Hibernate with Log4j 2
4	**Inheritance Mapping**	• Table Per Hierarchy • Table Per Hierarchy using Annotation • Table Per Concrete • Table Per Concreteusing Annotation • Table Per Subclass • Table Per Subclass using Annotation
	Collection Mapping	• Mapping List • One-to-many by List using XML • Many to Many by List using XML • One To Many by List using Annotation • Mapping Bag

		• One-to-many by Bag • Mapping Set • One-to-many by Set • Mapping Map • Many-to-many by Map • Bidirectional • Lazy Collection
5	**Component Mapping, Association Mapping, Transaction Management, HQL and HCQL**	• One-to-one using Primary Key • One-to-one using Foreign Key
	Named Query, Hibernate Caching and Integration	• First Level Cache • Second Level Cache • Hibernate and Struts • Hibernate and Spring

References Books

1. Spring and Hibernate - Santosh Kumar K. - Tata McGraw-Hill Publishing
2. Spring persistence with Hibernate - Paul Tepper Fisher and Brian D. Murphy - Apress
3. Spring 4 and Hibernate 4: Agile Java Design and Development McGraw-Hill Education, 2015

CS-02: Advance Web Development in Laravel

course outcomes:

- Student should know OOP in PHP
- Student should be able to implement Laravel framework
- Student should be able to design and code responsive website
- Student should be able to meet current modern market requirement and create fruitful products

Pre-Requisites: Strong background and Knowledge of HTML, CSS, JavaScript and PHP is mandatory.

Unit No.	Topic	Details
1	**Object Oriented Programming in PHP**	The Basics, Properties, Class Constants, Autoloading Classes, Constructors and Destructors, Visibility, Object Inheritance, Scope Resolution Operator (::), Static Keyword, Class Abstraction, Object Interfaces, Anonymous classes, Overloading, Object Iteration, Magic Methods, Final Keyword, Object Cloning, Comparing Objects, Type Hinting, Late Static Bindings, Objects and references
	Bootstrap Basics	**Introduction:** File Structure, Basic HTML Template, Global Styles, Default Grid System, Basic Grid HTML, Offsetting Columns, Nesting Columns, Fluid Grid System, Container Layouts, Responsive Design, What Is Responsive Design? **Implementation:** Typography, Code, Tables, Forms, Buttons, Images, Icons, Glyphicons, Dropdown Menus, Button Groups, Button with Dropdowns, Navigations, Navbar, Breadcrumb, Pagination, label, badges, Typographic elements, thumbnails, alerts, progress bar, wells
2	**Introduction to Laravel**	What is Laravel, features, MVC architecture, structure of laravel application (laravel directory structure),
	Installation	Basic requirements for Laravel, Using Laravel Installer, Using Composer, how does Composer work? Installation, Linux & Windows, Finding and installing new packages
	Configuration	Introduction, Environment configuration, Protecting sensitive configuration, Maintenance mode, database configuration (setting database connection parameter for laravel and artisan)
3	**Artisan**	Artisan Command Line Tool, database creation, artisan migration, migration structure, creation migration, Database seeding
	Routing in Laravel	Basic Routing, Route Parameters, Route Filters, Named Routes, Route Groups, Sub-Domain Routing, Route Prefixing, Route Model Binding, Throwing 404 Errors, Routing to Controllers

4	**Blade Template**	Template inheritance, Master layout, Extending the master layout, display variables, Blade conditional statements, Blade Loops, Executing PHP functions in blade

	SQL Interaction	Introduction, Running Raw SQL Queries, Database Transactions
5	**Eloquent ORM**	**Eloquent ORM Models:** Naming conventions, table name & primary keys, timestamps **Basic Operations:** Create, Retrieve, Update, Delete Using Models, displaying data from models in views.
	Validation	Defining The Routes, Creating The Controller, Writing The Validation Logic, Displaying The Validation Errors, Array validations, creating new validators, Error messages & custom errors **Available Validators:** Accepted, After (Date), Alpha, Alpha Dash, Alpha Numeric, Array, Before (Date), Between, Boolean, Date, Date Format, Different, Digits, Digits Between, E-Mail, Exists (Database), Image (File), In, Integer, Max, Min, Not In, Numeric, Regular Expression, Required, String Custom validation rules.

References Books

1. Online Laravel 5.2 Documentation (https://laravel.com/docs/5.2)
2. Laravel 5 Essentials, Martin Bean, Packet Publishing, ISBN 978-1-78528-301-7
3. Bootstrap, Jake Spurlock, O’reilly, ISBN: 978-1-449-34391-0

CS – 03: NoSQL DATABASE: MongoDB

course outcomes:

- To develop proficiency in the specification, representation and various other types in MongoDB using PHP.
- To be able to perform various Analytical as well as to increase the programming skills in PHP using MongoDB.
- To get a good understanding regarding various styles in Programming.
- To develop a good base for No-SQL queries.

Pre-Requisites: Knowledge of PHP is mandatory.

Unit No.	Topics	Details
1	**Introduction to NoSQL Database**	• Define NoSQL, its characteristics and history, and the primary benefits for using NoSQL databases. • Define the major types of NoSQL databases including a primary use case and advantages/disadvantages of each type. • Describe the factors affecting return on investment for using locally hosted database vs. database-as-a-service.
	Introduction to MongoDB	• MongoDB concepts – Databases, collections, and documents • Downloading Installing and running MongoDB, Installing PHP Driver for MongoDB on various OS Platforms • The Data Model and Working with Data
2	**Learning MongoDB by implementing web Application**	• Inserting documents in MongoDB, Querying documents in collection. • Doing advance queries in MongoDB, Updating documents MongoDB, • Deleting documents in MongoDB, Managing relationships between documents
	Using MongoDB with relational Databases	• MongoDB and RDBMS together • Defining the relational model
3	**Session Management**	• Understanding HTTP sessions. • Understanding PHP native session handling, • Implementing session handling with MongoDB. • Putting Session Manager. • Building user authentication module, creating login, logout and user profile.
4	**Aggregation Queries**	• Generating Sample Data. • Understanding MapReduce, • Performing MapReduce in MongoDB and PHP, Aggregation using

		group() • Listing distinct values for field • counting documents with count()
	Web Analytics using MongoDB	• Logging with MongoDB, • Extracting analytics data with MapReduce • Real-time analytics using MongoDB
5	**Handling Files with GridFS**	• What is Grid? • Storing files in GridFS • Serving files from GridFS • Reading files in chunks
	Database Management	• Database Administration • Optimization • Replication • Sharding

References Books

1. MongoDB the definitive guide - O'Reilly Kristina Chodorow & Michal Dirolf
2. MongoDB in Action - Kyle Banker Manning Sheltar Island.
3. The definitive guide to MongoDB - NoSQL Database for cloud and desktop computing. - Apress - Eelco Plugge, Peter membrey and Tim Hawkins
4. PHP and MongoDB Web Development Beginers guide - Rubayeet Islam - Open Source

CS – 04: PRACTICAL - 1 (BASED ON CS-01)

Topics	Marks
APPLICATION DEVELOPMENT USING ADVANCE JAVA	**100**

CS – 05: PRACTICAL - 2 (BASED ON CS-02 and CS-03)

Topics	Marks
• ADVANCE WEB DEVELOPMENT IN Laravel • NoSQL DATABASE: MongoDB	**100**

Note:

- Practical examination may be arranged before or after theory exam.

CS – 06: PROJECT DEVELOPMENT (In **Marks:**
Project must be developed in the computer laboratory of concern institute under the supervision of faculties of concern institute on any subject of current semester. **(At the time of Project-Viva examination student must show Project Report (In Hard Copy) along with all the Workouts in workbook, implementation of project in SDLC, Documentation, Program codes and project in running mode)**

Note :

- Project must be submitted before two week of commencement of theory exam.
- Project viva examination may be arranged before or after theory exam.
- During the project viva examination project must be run.

M.Sc. (IT & CA) (Semester – 2)

SR. NO.	COURSE	No. of LECT./Lab. PER WEEK	CREDIT
1.	**CS – 07** APPLICATOIN DEVELOPMENT USING ADVANCED ANDROID	**5**	**5**
2.	**CS – 08** INTRODUCTION TO BIG DATA AND HADOOP	**5**	**5**
3.	**CS – 09** CLOUD COMPUTING	**5**	**5**
4.	**CS – 10** PRACTICAL - 1 (BASED ON CS-07)	**5**	**5**
5.	**CS – 11** PRACTICAL - 2 (BASED ON CS-08 and CS-09)	**5**	**5**
6.	**CS – 12** PROJECT DEVELOPMENT (In House)	**5**	**5**
Total Credits of Semester – 2			**30**

CS – 07: APPLICATOIN DEVELOPMENT USING ADVANCED ANDROID

course outcomes:

- To be able to develop mobile applications using advanced android api based on
- Data storage in external and internal memory and database
- To develop app that supports animation, multimedia, camera, sensor
- To develop app that supports Network, Bluetooth-Wi-Fi
- Developing web service and retrieving data using JSON & xml
- Packaging and distributing android app

Pre-Requisites: OOPS concepts, Programming in core java, Basic Android Programming.

Unit No.	Topics	Details
1	**Basics of Android & UI Design**	• Core building blocks, Android manifest.xml file, R.java file, Basic UI widgets, Activity, Layout, Intent
	Working with view and adaptor	• Adaptors: Array adaptor, Arraylist adaptor, Base adaptor, • Views: GridView, ScrollView, WebView, SearchView, TabHost, DynamicListView, ExpandedListView
	Multimedia API	• Wallpapaer, Live Wallpaper, • Audio – Recording audio, Playing audio • Video– Recording video, Playing video • Alarm Manager • Camera - Capturing pictures, configuring camera mode settings, camera parameters, zooming camera.
2	**Data Storage & SQLite**	• Shared Preferences • Android File System • Internal storage, External storage • SQLite : Storing data using SQLite, Querying SQLite database, insert-update-delete operations, Persistent database using SQLiteOpenHelper and creating a database
	Content Provider, Intent & Notifications	• Accessing built in content providers • Searching for content • Adding, changing, and removing content • Creating content provider • Sending & Receiving Broadcast • Notifying user, Notifying with status bar
3	**Device Connectivity**	• Bluetooth Tutorial –existence of Bluetooth, enable Bluetooth, discover devices, List Paired Devices, establishing connection between devices. • Working with WiFi
	Working with Sensor	• Sensor API, • Working with different sensors :Motion Sensor, Position Sensor, Environmental Sensor, • Sensor Values, SensorManager class, Sensor Class, SensorEvent class, SensorEventListener interface, Compass Acceslerometer

		and Orientation Sensors • Reading sensor data, calibrating sensors, determining device orientation
	Android Web Service	• Introduction to web service, • Soap Vs Restful web service • Android Restful web service example with java servlet • Storing data into external database • Verifying data in android with external database
4	**JSON & XML Parsing**	• XML Parsing SAX • XML Parsing DOM • XML Pull Parser • JSON Parsing • Integrating Social Networking using HTTP
	WiFi& Bluetooth	• Monitoring and managing Internet connectivity • Managing active connections • Managing WiFi networks • Controlling local Bluetooth device • Discovering and bonding with Bluetooth devices • Managing Bluetooth connections • Communicating with Bluetooth
5	**Location Based Services and Google Maps**	• Location Based Services - Finding current location and listening for changes in location, Proximity alerts, Working with Google Maps • Showing google map in an Activity • Map Overlays • Itemized overlays • Geocoder • Displaying route on map
	Drawing, Animation and Graphics programing	• Drawing on screen – using canvas and paint • Working with bitmap, shapes • 2D Animation - Drawable, View, Property animation
	Packaging, Deploying and distributing/ selling app	• Signing certificate • Distributing android app via Google Play • Obfuscating and optimizing with ProGuard

References Books:

1. Advanced Android Application Development – Joseph Annuzzi, Lauren darcey, Shane Conder – 4th Edition, Addision – Wesley.
2. Android cookbook - Ian F. Darwin Oreilly
3. The Android Developer’s CookBook – Building Application with Android SDK – 2nd Edition, Addision – Wesley.

CS – 08: INTRODUCTION TO BIG DATA AND HADOOP

course outcomes:

- Master the concepts of HDFS and MapReduce framework
- Understand Hadoop Architecture
- Setup Hadoop Cluster and write Complex MapReduce programs
- Learn data loading techniques using Sqoop and Flume
- Perform data analytics using Pig and Hive
- Implement HBase and MapReduce integration
- Implement Advanced Usage and Indexing
- Implement best practices for Hadoop development
- Work on a real life Project on Big Data Analytics

Pre-Requisites: Knowledge of Java, SQL and Linux commands is mandatory

Unit No.	Topics	Details
1	**Introduction to Big Data and Hadoop**	Introduction/Installation of Virtual Box and the Big Data VM Introduction to Linux - Why Linux? - Windows and the Linux equivalents - Different flavors of Linux - Unity Shell (Ubuntu UI) - Basic Linux Commands (enough to get started with Hadoop)
	Understanding Big Data	Understanding Big Data - 3V (Volume-Variety-Velocity) characteristics - Structured and Unstructured Data - Application and use cases of Big Data Limitations of traditional large Scale systems How a distributed way of computing is superior (cost and scale) Opportunities and challenges with Big Data
	HDFS (The Hadoop Distributed File System)	HDFS Overview and Architecture - Deployment Architecture - Name Node, Data Node and Checkpoint Node (aka Secondary Name Node) - Safe mode - Configuration files - HDFS Data Flows (Read vs Write) How HDFS addresses fault tolerance? - CRC Check Sum - Data replication - Rack awareness and Block placement policy

		- Small files problem HDFS Interfaces - Command Line Interface - File System - Administrative - Web Interface Advanced HDFS features - Load Balancer - DistCp - HDFS Federation - HDFS High Availability - Hadoop Archives
	NoSQL Databases - 1 (Theoretical Concepts)	NoSQL Concepts - Review of RDBMS - Need for NoSQL - Brewers CAP Theorem - ACID vs BASE - Schema on Read vs. Schema on Write - Different levels of consistency - Bloom filters Different types of NoSQL databases - Key Value - Columnar - Document - Graph Columnar Databases concepts
2	**MapReduce – 1 (Theoretical Concepts)**	MapReduce overview - Functional Programming paradigms - How to think in a MapReduce way? MapReduce Architecture - Legacy MR vs Next Generation MapReduce (aka YARN/MRv2) - Slots vs Containers - Schedulers - Shuffling, Sorting - Hadoop Data Types - Input and Output Formats - Input Splits - Partitioning (Hash Partitioner vs Customer Partitioner) - Configuration files - Distributed Cache MR Algorithm and Data Flow - Word Count Alternatives to MR - BSP (Bulk Synchronous Parallel) - Adhoc querying - Graph Computing Engines

	Higher Level Abstractions for MR (Pig)	Introduction and Architecture Different Modes of executing Pig constructs Data Types Dynamic invokers Pig streaming Macros Pig Latin language Constructs (LOAD, STORE, DUMP, SPLIT etc) User Defined Functions Use Cases
3	**MapReduce – 2 (Practical)**	Developing, debugging and deploying MR programs - Stand alone mode (in Eclipse) - Pseudo distributed mode (as in the Big Data VM) - Fully distributed mode (as in Production) MR API - Old and the new MR API - Java Client API - Hadoop data types and custom Writables/WritableComparables - Different input and output formats - Saving Binary Data using SequenceFiles and Avro Files Hadoop Streaming (developing and debugging non Java MR programs - Ruby and Python) Optimization techniques - Speculative execution - Combiners - JVM Reuse - Compression MR algorithms (Non-graph) - Sorting - Term Frequency - Inverse Document Frequency - Student Data Base - Max Temperature - Different ways of joining data - Word Co-Occurrence MR algorithms (Graph) - PageRank - Inverted Index
	Higher Level Abstractions for MR (Hive)	Introduction and Architecture Different Modes of executing Hive queries Metastore Implementations HiveQL(DDL & DML Operations) External vs Managed Tables Views

		Partitions & Buckets User Defined Functions Transformations using Non Java Use Cases Comparison of Pig and Hive
4	**NoSQL Databases - 2 (Practical)**	HBase Architecture - Master and the Region Server - Catalog tables (ROOT and META) - Major and Minor compaction - Configuration files - HBase vs Cassandra Interfaces to HBase (for DDL and DML operations) - Java API - Client API - Filters - Scan Caching and Batching - Command Line Interface - REST API Advance HBase Features - HBase Data Modeling - Bulk loading data in HBase - HBase Coprocessors - EndPoints (similar to Stored Procedures in RDBMS) - HBase Coprocessors - Observers (similar to Triggers in RDBMS)
5	**Spark**	- Introduction to RDD - Installation and Configuration of Spark - Spark Architecture - Different interfaces to Spark - Sample Python programs in Spark
	Setting up a Hadoop Cluster using Apache Hadoop	Cloudera Hadoop cluster on the Amazon Cloud (Practice) - Using EMR (Elastic Map Reduce) - Using EC2 (Elastic Compute Cloud) SSH Configuration Stand alone mode (Theory) Distributed mode (Theory) - Pseudo distributed - Fully distributed
	Hadoop Ecosystem and Use Cases	Hadoop industry solutions Importing/exporting data across RDBMS and HDFS using Sqoop Getting real-time events into HDFS using Flume Creating workflows in Oozie Introduction to Graph processing Graph processing with Neo4J

		Processing data in real time using Storm Interactive Adhoc querying with Impala

References Books

1. MapReduce Design Patterns Building Effective Algorithms and Analytics for Hadoop and Other Systems By Donald Miner, Adam Shook Publisher: O'Reilly Media
2. Professional Hadoop Solutions By Boris Lublinsky, Kevin T. Smith, Alexey Yakubovich
3. Hadoop The Definitive Guide by Tom White
4. Hadoop Operations, Eric Sammer
5. Hadoop for Dummies by Dirk Deroos

CS – 09: CLOUD COMPUTING

course outcomes:

- To describe cloud computing architecture and services
- To identify cloud platforms and services
- To identify design issues of cloud computing
- To analyze the security factors of implementing cloud environment
- To understand the server virtualization and its implementation
- To review real time applications of cloud computing

Pre-Requisites: Knowledge of Advance Computer Networks is mandatory

Unit No.	Topics	Details
1	**Overview of Computing Paradigm**	• Recent trends in Computing: Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing • Evolution of cloud computing: Business driver for adopting cloud computing
	Introduction to Cloud Computing	• Cloud Computing (NIST Model): Introduction to Cloud Computing, History of Cloud Computing, Cloud service providers • Properties, Characteristics & Disadvantages: Pros and Cons of Cloud Computing, Benefits of Cloud Computing, Cloud computing vs. Cluster computing vs. Grid computing • Role of Open Standards
	Cloud Computing Architecture	• Cloud computing stack: Comparison with traditional computing architecture (client/server), Services provided at various levels, How Cloud Computing Works, Role of Networks in Cloud computing, protocols used, Role of Web service: • Service Models (XaaS): Infrastructure as a Service(IaaS),Platform as a Service(PaaS), Software as a Service(SaaS) • Deployment Models: Public cloud , Private cloud, Hybrid cloud, Community cloud
2	**Infrastructure as a Service(IaaS)**	• Introduction to IaaS: IaaS definition, Introduction to virtualization, Different approaches to virtualization, Hypervisors, Machine Image, Virtual Machine(VM): Resource Virtualization: Server ,Storage, Network, Virtual Machine(resource) provisioning and manageability, storage as a service, Data storage in cloud computing(storage as a service) • Examples: Amazon EC2, Renting, EC2 Compute Unit, Platform and Storage, pricing, customers, Eucalyptus
	Cloud Security	• Infrastructure Security: Network level security, Host level security,

		Application level security • Data security and Storage: Data privacy and security Issues, Jurisdictional issues raised by Data location,Identity & Access Management, Access Control • Trust, Reputation, Risk • Authentication in cloud computing, Client access in cloud, Cloud contracting Model, Commercial and business considerations
3	**Platform as a Service(PaaS)**	• Introduction to PaaS: What is PaaS, Service Oriented Architecture (SOA) • Cloud Platform and Management: Computation, Storage • Examples: Google App Engine, Microsoft Azure,SalesForce.com, Force.com platform
	Software as a Service(PaaS)	• Introduction to SaaS • Web services • Web 2.0 • Web OS • Case Study on SaaS
	Service Management in Cloud Computing	• Service Level Agreements(SLAs) • Billing & Accounting • Comparing Scaling Hardware: Traditional vs. Cloud • Economics of scaling: Benefitting enormously • Managing Data: Looking at Data, Scalability & Cloud Services, Database & Data Stores in Cloud,Large Scale Data Processing
4	**Virtualization**	• Virtualization objectives • Virtualization implementation • Virtual servers introduction • Xen server-Hyper V – I, Hyper V – II, VMWare – I, VMWare – II
5	**Case Study on Open Source & Commercial Clouds**	• Eucalyptus • Microsoft Azure • Amazon EC2

Reference Books

1. Kenneth Hess, Amy NewMan – Practical Virtualization Solutions – Prentice Hall, 2010
2. Shahed Latif, Tim Mather, Subra Kumaraswamy – Cloud Security and Privacy : An Enterprise perspective on risks and compliance – O’Reilly Media Inc., 2009
3. Gautam Shroff – Enterprise Cloud Computing: Technology, Architecture, Applications – Cambridge University Press, 2010
4. Cloud Computing Bible, Barrie Sosinsky, Wiley-India, 2010

Andrzej M. Goscinski, Wile, 2011

6. Cloud Computing: Principles, Systems and Applications, Editors: Nikos Antonopoulos, Lee Gillam, Springer, 2012
7. Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Ronald L. Krutz, Russell Dean Vines, Wiley-India, 2010
8. George Reese – Cloud Application Architectures: Building Applications and Infrastructures in the cloud – O'Reilly Media Inc., 2009
9. Anthony T. Velte, Toby J. Velte, Robert Elsenpeter – Cloud Computing A practical Approach

CS – 10: PRACTICAL - 1 (BASED ON CS-07)

Topics	Marks
APPLICATOIN DEVELOPMENT USING ADVANCED ANDROID	100

CS – 11: PRACTICAL - 2 (BASED ON CS-08 and CS-09)

Topics	Marks
• INTRODUCTION TO BIG DATA AND HADOOP • CLOUD COMPUTING	100

Note:

- Practical examination may be arranged before or after theory exam.

CS – 12: PROJECT DEVELOPMENT (In Marks:

Project must be developed in the computer laboratory of concern institute under the supervision of faculties of concern institute on any subject of previous semester or current semester. **<u>(At the time of Project-Viva examination student must show Project Report (In Hard Copy) along with all the Workouts in workbook, implementation of project in SDLC, Documentation, Program codes and project</u>**

Note:

- Project must be submitted before two week of commencement of theory exam.
- Project viva examination may be arranged before or after theory exam.

- During the project viva examination project must be run.

M.Sc. (IT & CA) (Semester – 3)

SR. NO.	COURSE	No. of LECT./Lab. PER WEEK	CREDIT
1.	**CS – 13** HYBRID MOBILE APPLICATIONS DEVELOPMENT USING WEB TECHNOLOGIES	5	5
2.	**CS – 14** WEB APPLICATION DEVELOPMENT USING DJANGO	5	5
3.	**CS – 15** PROGRAMMING WITH R FOR DATA SCIENCE	5	5
4.	**CS – 16** PRACTICAL - 1 (BASED ON CS-13)	5	5
5.	**CS – 17** PRACTICAL - 2 (BASED ON CS-14 and CS-15)	5	5
6.	**CS – 18** PROJECT DEVELOPMENT (In House)	5	5
Total Credits of Semester – 3			**30**

CS-13 : Hybrid Mobile Applications Development Using Web Technologies

course outcomes:

- Focuses on developing multiplatform mobile applications using the Web skills (HTML5, CSS and Javascript).
- Understand AngularJS basic and advanced in depth concepts.
- Using the Cordova hybrid application framework to develop and target multiple mobile platforms with a single codebase.
- Using Ionic framework, one of fastest growing mobile application frameworks, that is built with mobile-optimized HTML5 and CSS based components and AngularJS.
- Understand NodeJS concepts.
- Publish mobile app on play store and app store.
- Understand UI development with Ionic and then using Cordova's modules to access the native mobile platform's capabilities from Javascript.

Pre-Requisites:

- Basic Programming Knowledge
- Basic Knowledge of HTML, CSS and Java Script
- Good Knowledge of Bootstrap
- Familiarity with AngularJS.

Sr. No	Topic	Details
1	**Introduction to Hybrid application, development platforms**	What is hybrid application? Need of hybrid application development Tool and platforms in used for development of hybrid mobile application development o Phonegap-cordova o Ionic o Mobile angular UI • Step by step installation of coredova using git and npm • Introduction to HTML 5 and HTML 5 APIs o Forms validation o Audio video tags o Data storage APIs ▪ Local storage ▪ Web sql ▪ IndexedDB\ • Introduction to CSS, Sscss, less • Using bootstrap.css with mobile application development
2	**Java Script for Mobile Application Development**	• Introduction to Java Script • Variables, Scopes and functions in Java Script • What is jquery? • Forms, data validation and storage using jquery o Storage on client side(HTML 5 storage APIs) o Sending data over server side (may serverside be PHP or NodeJs) • What is angularjs? • Role of angularjs in platforms like mobile angular UI and

		Iconic • $scope and $rootScope • Config() and Run() • Directives in angularjs ○ Ng-model ,Ng-bind, Ng-app, Ng-click, Ng-show/ng-hide, Ng-init, Ng-submit, Ng-repear, User-define dierctives • Filters in angularjs, Angular forms, Angular validation, Angular module, angular controller, angular factory, service • ui-router (restful application development) • $state, $statParms, $statProvider.stat() • MVC architecture of angularjs • Implementation of model (FACTORY/SERVICE), controller and view for data handling • Development a TO-DO, task application using angularjs
3	**Iconic – 1**	• Introduction to iconic platform for hybrid mobile application development • Step by step installation of iconic • Command line interface handling of iconic-1 • Creating project in iconic (CLI APPROACH) • Component of iconic-1 ○ Colors, header, button, list, card, forms, checkgbox, radiobuttons, range, select (drop down), tabs, grid • Iconic java script components ○ Action sheet, backdrop, content, forms, model, popover, popup, SideMenu, SlideBox • Platform management in iconic-1 • Plugins for iconic-1 • ngCordova plugins • stateful approach of developing iconic applications ($stateProvider.state()) ○ passing data into the state by URL ○ States hierarchy • Injection of controllers and factories in modules • Icon and Splash Screen for iconic applications • Implementation of MVC architecture (exactly as MVC of angularJS) • http request and promises in factory() • populating data into view from controller • Themes in iconic
4	**Interaction with server side PHP**	• Database connection to MySQL • Associative arrays and array handling in PHP ○ Array_push(), array_pop(), array_search(),

		in_array() • Reading JSON as input ○ file_get_contents(“php://input”); ○ json_encode() and json_decode() • data communication and interaction with client side using JSON • CRUD operation with PHP and MySQL ○ C=create (INSERT QUERY), R=read (SELECT QUERY), u = update (UPDATE QUERY), D=delete(DELETE QUERY) • Introduction to Nodejs • Architecture of Nodejs • Step by step installation of Nodejs • Introduction to express-Nodejs • Create a server and listen to port in Nodejs
5	**Accessing Native Services using Iconic and Application Signing and Development**	• Iconic-cordova integration, iconic-camera, iconic-native audio, iconic-media, iconic-InApp browser • Introduction to git • Basic commands in git ○ Push, pull, commit, rollback, status, init, branch • Git branches • Git push & pull operations • Significance and importance of git in development of applications • Maintaining version of applications using git • Android ○ Signing application (keystore) ○ Publishing application on play store • IOS • Build and publication application in App store

References Books:

- Mobile App Development with Ionic 2 Cross-Platform Apps with Ionic, Angular, and Cordova By Chris Griffith Publisher: O'Reilly Media Final Release Date: April 2017
- Learning Ionic – Arvind Ravulavaru – PACKT Publishing, July – 2015
- Ionic in Action: Hybrid Mobile Apps with Ionic and AngularJS - Jeremy Wilken, Manning Publications, 2015
- Learning PHP, MySQL & JavaScript 4e (Learning Php, Mysql, Javascript, Css & Html5)
- AngularJS - O'Reilly Media By Brad Green, Shyam Seshadri

Useful Links:

https://www.javatpoint.com/nodejs-tutorial
https://www.tutorialspoint.com/html5/index.ht
m https://www.tutorialspoint.com/ionic

https://www.w3schools.com/angular/

CS – 14 : Web Application Development using Django

course outcomes:

- Understand how to learn a web development framework.
- Understand how to use Python and Django to develop modern web applications.
- Gain functional knowledge of Python, Databases and the Django framework.
- Understand current web development best practices.
- Build and deploy a Python Django web application that incorporates a database.

Pre-Requisites:

- Basic programming knowledge.
- Object Oriented Programming knowledge.
- Knowledge Python would be desired, not mandatory.

Sr. No	Topic	Details
1	**Introduction to Python and Python Syntax, Language Components / Collections & Functions,**	A Brief History of Python, Strengths and Weaknesses, Python Versions. Installing Python, Environment Variables, and Executing Python from the Command Line, IDLE, Editing Python Files, Getting Help, Dynamic Types, Python Reserved Words, Naming Conventions, Basic Syntax, Comments, String Values, String Operations, The format Method, String Slices, String Operators, Numeric Data Types, Conversions, Simple Input and Output, The print Function. Control Flow and Syntax, Indenting, if Statement, Relational Operators, Logical Operators, True or False, Bit Wise Operators. The while Loop, break and continue, The for Loop, Lists, Tuples, Sets, Dictionaries, Sorting Dictionaries, Copying Collections, Summary, Defining Your Own Functions, Parameters, Function Documentation, Keyword and Optional Parameters, Passing Collections to a Function.
2	**Introduction to Web framework and DJango DJango Template System**	HTTP Client-Server Request – Response, concept of web framework and web application. Introduction to Django, MVC Design Pattern, Django installation, setting up database, starting project. Django project architecture, Understanding manage.py, Understanding settings.py, Understanding init .py and wsgi.py, Understanding urls.py and Python regular expression, Understanding admin.py, Understanding models.py, Understanding views.py , Running Django development server

		Template system basics, Using template system, basic

		template tags and filters, using templates in views, template loading.
3	**Interaction with Database**	Configuring database, defining model, basic data access, inserting and updating data, selecting objects, deleting objects.
4	**Django Admin Site & Forms, Views and URLConfs**	Activating the Admin interface, Creating super user for Admin site, Using the Admin site, Using Admin site, django.contrib package. Form basics, GET and POST methods , Form validation, Rendering forms , ModelForm, Understanding the view layer, Requesting a web page via URL, Rendering web page via view function, Render HTTPResponse to templates, Understanding context data and Python dictionary type.
5	**Session and Cookies & Testing and Deploying web application**	Cookies: Getting and Setting Cookies. Session: Django's session framework: enabling sessions, using session in views, session outside views. Testing Django, Python's unittest2 library, Deploying Django application on GitHub / Amazon Web Service.

References Books:

- John V Guttag. "Introduction to Computation and Programming Using Python", Prentice Hall of India
- Learning Website Development using DJano – Ayman Hourieh – PACKT Publishing
- Pro DJango – Marty Alchin - APress
- The Definitive Guide to Djano: Web Development done Right – Adrian Holovaty, Jacob K. Moss

CS-15 : Programming with R for Data Science

course outcomes:

- The main objective of this syllabus is to ensure the working aspects of R-Programming.
- Here, Students will be able to learn R programming with various level of strategic inputs such as Vectors, Arrays, Matrices, Strings and Factors etc.
- The course also covers the understanding the aspects of Packages and at last Visualize the data in the form of graph in various ways.

Pre-Requisites:

- A basic understanding of any of the computer programming language will help in understand the R programming concepts.
- Relevant knowledge of Linux OS needed if working in Open source OS for various IDE’s

Sr. No	Topic	Details
1	**Introduction to Data Analysis and Fundamentals of R**	• Overview of Data Analytics, Need of Data Analytics • Classification of Data: Structured, Semi-Structured, Unstructured, • Characteristics of Data, Applications of Data Analytics. • Setup with R Studio • R Commands, Variables, Data Types. • Vectors ○ Sequences, Lengths, Names, Indexing vectors, Vector Recycling and Repetition • Matrices and Arrays ○ Creating Arrays and Matrices, ○ Row, Columns and Dimensions ○ Row, Column and Dimension names, Array Arithmetic • Lists ○ Creating Lists, Atomic and Recursive Variables, List Dimensions and Arithmetic ○ Indexing Lists, Converting Between Vectors and Lists ○ Combining Lists, NULL. Pair lists • Data Input ○ Data Input from Keyboard, Input from files(CSV), input from files using scan, Reading data from a file using readLines, Masking Input and output formats, Checking Files from cmd. • Data Frames ○ Creating Data Frames, Indexing Data Frames, Basic Data Frames Manipulation

2	**Environment, Functions, String, Factors, Flow Control and Loops**	• Environments • Functions – Creating and calling Functions, Passing functions to and from other functions, Variable scope, Commands to Functions, Functions and Functional Programming, Function Objects and Function Calls, Debugging, Interactive Tracing and Editing, Conditions: Errors and Warnings, Testing R Software. • Strings

		○ Constructing and Printing Strings, Formatting Numbers, Special Characters, Changing Case, Extracting Substrings Splitting Strings, File Paths • Factors ○ Creating Factors, Changing Factor Levels, Dropping Factor Levels, Ordered Factors, Converting Continuous Variables to Categorical, Converting Categorical Variables to continuous, Generating Factor Levels, Combining Factor Levels • Flow Control and Loops ○ Flow Control – if and else, Vectorized if, Multiple selections ○ Loops – repeat, while, for, lapply, sapply, • Advance Loops – Replication, Looping over Lists, Looping Over Arrays, Multiple Inputs, Split-Apply-Combine, the plyr package.
3	**Creating Packages and working with date & time**	• Packages ○ Loading Packages – The search path, Libraries and Installed packages ○ Installing Packages ○ Maintaining Packages • Dates and Time
4	**Data Visualization and Graphics**	• Reading and getting data into R (External Data): Using CSV files, XML files, Web Data, JSON files, Databases, Excel files. • Working with R Charts and Graphs: Histograms, Boxplots, Bar Charts, Line Graphs, Scatterplots, Pie Charts
5	**Analytics Using R**	• Big Data analytics using R. • Business Foundation Analytics Using R • Data Flow and Management for Business Operations and Problem Solving • Typical Analytical Process Flow • Data Collections Method • Data Summarization and Presentation • Managing Data using Analytics Tools (R) • Data Manipulation and Report Generation Using R

References Books:

- Data Manipulation with R by Phil Spector ISBN 978-0-387-74731-6
- Learning R by Richard cotton
  - *Reference Link: https://books.google.co.in/books?id=7dyzAAAAQBAJ&printsec=frontcover#v=onepage&q&f=false*
- The R Book by Michael J. Crawley
  - *Reference Link: https://books.google.co.in/books?id=XYDl0mlH-moC&printsec=frontcover&dq=r+programming&hl=en&sa=X&redir_esc=y#v=onepage&q=r%20pro gramming&f=false*
- Software for Data Analysis Programming with R. by John M. Chambers
  - *Reference Link: http://www.e-reading.club/bookreader.php/137298/Software_for_Data_Analysis*

CS – 16: PRACTICAL - 1 (BASED ON CS-13)	
Topics	**Marks**
Hybrid Mobile Applications Development Using Web Technologies	**100**

CS – 17: PRACTICAL - 2 (BASED ON CS-14 and CS-15)	
Topics	**Marks**
• WEB APPLICATION DEVELOPMENT USING DJANGO • PROGRAMMING WITH R FOR DATA SCIENCE	**100**

Note:

- Practical examination may be arranged before or after theory exam.

CS – 18: PROJECT DEVELOPMENT (In Marks:
Project must be developed in the computer laboratory of concern institute under the supervision of faculties of concern institute on any subject of current semester. **(At the time of Project-Viva examination student must show Project Report (In Hard Copy) along with all the Workouts in workbook, implementation of project in SDLC, Documentation, Program codes and project in running mode)**

Note :

- Project must be submitted before two week of commencement of theory exam.
- Project viva examination may be arranged before or after theory exam.
- During the project viva examination project must be run.

M.Sc. (IT & CA) (Semester – 4)

CS – 19: INDUSTRIAL PROJECT DEVELOPMENT Marks:

Project must be developed at industrial organization. **(At the time of Project-Viva examination student must show Project Report (In Hard Copy) along with all the Workouts in workbook, implementation of project in SDLC, Documentation, Program codes (Optional) and project in running mode).**

Guidelines:

(1) Institute/College/Department has to make arrangement for the students for project development in various software development organizations in industry.

(2) Project work must be developed at the industrial organization, not at the paid or free project training institute.

(3) Internal guide from institute and external guide from Industry must be allocated for supervision

(4) Coding standards should be followed meticulously. At the minimum, the code should be self-documented, modular, and should use the meaningful naming convention.

(5) The documentation should include a chapter on “Learning during Project Work”, i.e. “Experience of Journey during Project Duration”.

SrNo	Evaluation Criteria	Marks
1	EXPLANATION OF CODE	75
2	EXPLANTION OF ANALYSIS AND DESIGN	75
3	DOCUMENTATION	75
4	PRESENTATION	75
	Total Marks	300