SHIVAJI UNIVERSITY, KOLHAPUR.

Accredited By NAAC with 'A' grade

2019-2020

Revised Syllabus For

M.Sc. Computer Science Part-I

(Subject to modifications to be made time to time)

Syllabus to be implemented from June 2019

The name of the programme shall be Master of Science (**M.Sc**. **Computer Science)**

Duration of the Program

- The M.Sc. programme will be a full-time two years i.e. 4 semesters. Pattern of examination will be Semester System.

Medium of Instruction

- The medium of Instruction will be English only**.**

Admission Procedure

- Eligibility: B.Sc. Computer Science (Entire/ optional) / B.Sc. IT/ BCS/BCA(under Science faculty), B.Sc. Mathematics, B.Sc. Statistics,B.Sc. Electronics
- Admission through University Entrance exam only.
- Only entrance marks should be considered for admission process.
- Reservation of Seats as per rules of Government of Maharashtra**.**

Project work

At the end of I, II & III semester student has to carry out a project work.

1. Projectwork at end of semester should be done in groups, each student must be given a responsibility for a distinct module and care should be taken to see the progress of individual modules is independent of others.
2. Students should take guidance from an internal guide and prepare a Project Report on "Project Work" to be submitted to the Department after evaluation.
3. The Project Report should contain an Introduction to Project, which should clearly explain the project scope in detail. Database/Webpage/ UI designs and a list of output reports should be included along with references.
4. The project Work should be of such a nature that it could prove useful or should be relevant from the societal/commercial/research angle.
5. The project report will be duly accessed by the internal guide of the project and internal marks will be communicated by the concerned guide.
6. Project viva-voce by the University panel will be conducted as part of Evaluation.

At the end of the Fourth semester of study, a student will be examined in the course
"Industrial / Research Project ".

1. Fourth semester Project work can be carried out as industrial training of four months in the Industry or in the Institute as Research project with prior permission of the Institute.
2. Project viva-voce by the University panel will be conducted at the end of semester.
3. The project report should be prepared in a format prescribed by the University, which also specifies the contents and methods of presentation.
4. Project work may be done individually or in groups in case of bigger projects.

5. The major project work carry 50 marks for internal assessment and 150 marks for External viva. The external viva shall be conducted by a panel of external examiners.

OR

1. The student will be allowed to formulate a proposal for start-up and the same will be rated equivalent to an industrial project. A detailed problem statement showinginnovation along with markability,business plan and cash flow will be part of the Evaluation criteria.

Research Seminar

At the end of fourth semester student shalldeliver seminar on one of the advanced topic chosen in consultationwith the guide after compiling the information from the latest literature and also internet. The concepts must be clearly understood and presented by student. Prior to presentation, he/she shall carry out the detailed literature survey from standard references such as International & National journals and periodicals recently published reference books etc. A hard copy of the report (A4 size, 12 fonts, Times New Roman, Single spacing both side printed) should be submitted to the Department before delivering the seminar.This seminar will be evaluated internally for 100 marks by the respective guides.

Assessment

 The final total assessment of the candidate is made in terms of an internal assessment and an external assessment for each course.

1. For each theory paper, 20% marks will be based on internal assessment and 80% marks for semester examination (external assessment), unless otherwise stated.

2. Internal assessment of theory papers should be in the form of two internal tests of 10 marks each. Total 20 marks.

3. The projects will be evaluated by the university appointed panel.

4. The final practical examination will be conducted by the university appointed panel at the end of semester for each lab course and marks will be submitted to the university by the panel. The pattern of final Practical Examination will be as follows-

1	Coding and Execution of Program	60 Marks
2	Viva-voce	20 Marks
3	Journal	20 Marks
	Total	100 marks

6. The internal marks will be communicated to the University at the end of each semester, but before the semester end examinations. These marks will be considered for the declaration of the results.

Nature of question paper

Nature of question paper is as follows for University end semester examination

a. **Theory Examination**: There will be seven (7) questions of 16 Marks and out of which four (4) to be attempted from question no 2 to 6. Question No.1 is compulsory and is of multiple choice questions.

b. **Practical Examination**:

      i. Duration of Practical Examination: 3 Hrs

      ii. Nature of Question paper: There will be three questions out of which any two questions to be attempted and each question carries 30 Marks.

Standard of Passing

Internal as well as external examination will be held at the end of semester. The candidate must score 40% marks in each head of internal as well as external Examination.

Board of Paper Setters /Examiners

For each Semester and examination there will be one board of Paper setters and examiners for every course.

Award of Class

There will be numerical marking on each question. At the time of declaration of the result the marks obtained by the candidate is converted into classes as per University norms.

Credit system implementation

As per the University norms

<u>Clarification of Syllabus</u>

The syllabus Committee should meet at least once in a year to study and clarify any difficulties from the Institutes. The Workshop on syllabi should be organised at the beginning of every semester.

<u>Revision of Syllabus</u>

As the computer technology is changing very fast, revision of the syllabus should be considered   every 3 years.

PEO's for M.Sc. Program

Program Educational Objectives is to prepare graduates to:

1. Apply and continuously acquire knowledge, both theoretical and applied, related to core areas of computer science
2. Demonstrate the ability to work effectively as a team member and/or leader in an ever-changing professional environment
3. Work productively as computer professionals (in traditional careers, graduate school, or academia) by demonstrating effective use of oral and written communication, working competently as a member of a team unit, adhering to ethical standards in the profession.

Program Outcomes (POs)

At the end of the Master of Science (Computer Science) Programme, graduating students/graduates will be able to:

1. Communicate computer science concepts, designs, and solutions effectively and professionally
2. Apply knowledge of computing to produce effective designs and solutions for specific problems
3. Identify, analyse, and synthesize scholarly literature relating to the field of computer science Use software development tools, software systems, and modern computing platforms.
4. Prepare for academic roles through NET/SET/PhD
5. Apply design and development principles in the construction of software systems of varying complexity.

Program Specific Outcomes (PSOs)

1. Demonstrate understanding of the principles and working of the hardware and software
   aspects of computer systems.
2. Ability to understand the structure and development methodologies of software systems.
3. Possess professional skills and knowledge of software design process. Familiarity and practical competence with a broad range of programming language and open source platforms.
4. Be acquainted with the contemporary issues, latest trends in technological development and thereby innovate new ideas and solutions to existing problems.

Student Contact hours per week : **32** Hours (Min)	Total Marks for M.Sc.-I : **1400**

Course structure

M.Sc. Part I - Semester I

Course Code	Title of the Course	Credits	Teaching Scheme (h/w)		Evaluation Scheme (marks)		
			L	P	CIE	SE	Total
CC-101	Design and Analysis of Algorithms	4	4	-	20	80	100
CC-102	Python Programming	4	4	-	20	80	100
CC-103	Database Management System	4	4	-	20	80	100
OE-104	Cyber Security	4	4	-	20	80	100
CCPR-105	Python Lab	4	-	6	20	80	100
CCPR-106	Database Lab	4	-	6	20	80	100
CCPR-107	Project	4	-	6	20	80	100
Total		28	16	18	140	560	700

M.Sc. Part I - Semester II

Course Code	Title of the Course	Credits	Teaching Scheme (h/w)		Evaluation Scheme (marks)		
			L	P	CIE	SE	Total
CC-201	Web Technology	4	4	-	20	80	100
CC-202	Advanced Java	4	4	-	20	80	100
CC-203	Android Fundamentals Development	4	4	-	20	80	100
CE-204	Elective-I: CE-204.1: Software Project Management CE-204.2: Data Science Foundations CE-204.3: Application Security Analyst CE-204.4:Cloud Computing	4	4	-	20	80	100
CCPR-205	Web Technology Lab	4	-	6	20	80	100
CCPR-206	Advanced Java Lab	4	-	6	20	80	100
CCPR-207	Project	4	-	6	20	80	100
Total		28	16	18	140	560	700

Theory Lectures : 60 Minutes Each	Total Credits for M.Sc. -I (Semester I & II) : 56
CC-Core Course	
OE – Open Elective	
CCPR-Core Course Practical	
SWM- SWAYAM UGC online Courses	
CE – Core Elective (Within department) : Core elective papers shall be minimum 2 or more	
Practical Examination is Semester wise.	
Separate passing is mandatory for Theory, Internal, Practical andProject	

M.Sc. Part II - Semester III

Course Code	Title of the Course	Credits	Teaching Scheme (h/w)		Evaluation Scheme (Marks)		
			L	P	CIE	SE	Total
SWM -301	Artificial Intelligence	4	4	-	20	80	100
CC-302	Advanced Web Technology	4	4	-	20	80	100
CC-303	PHP	4	4	-	20	80	100
CE-304	Elective-II: CE-304.1: Software Quality Assurance CE-304.2: Advanced Data Science CE-304.3: Network Security Analyst CE-304.4: Internet of Things	4	4	-	20	80	100
CCPR-305	Advanced Web Technology Lab	4	-	6	20	80	100
CCPR-306	PHP Lab	4	-	6	20	80	100
CCPR-307	Project	4	-	6	20	80	100
Total		28	16	18	140	560	700

M.Sc. Part II - Semester IV

Course Code	Title of the Course	Credits	Teaching Scheme (h/w)		Evaluation Scheme (Marks)		
			L	P	CIE	SE	Total
CCPR - 401	Research Seminar	4	-	8	100	-	100
CCPR-402	Industrial / Research Project	8	-	4	50	150	200
Total		12	-	12	150	150	300

Student Contact hours per week : **16 Hours** (Min)	Total Marks for M.Sc-II : **1000**
Theory Lectures : 60 Minutes Each	Total Credits for M.Sc-II (Semester III & IV) : **40**
CC-Core Course	
CE – Core Elective (Within department) : Core elective papers shall be minimum 2 or more	
OE – Open Elective	
SWM –SWAYAM UGC Online courses	
CCPR – Core Course Practical.	**Total Credits for M.Sc. Course : 96**
Separate passing is mandatory for Theory, Internal, practical and	**Total Marks for M.Sc. Course : 2400**

Project	

MSc-I Semester-I (Computer Science)
CC-101: Design and Analysis of Algorithms
(Choice Based Credit System)
To be implemented from the academic year 2019-2020
External Marks-80 Internal Marks -20 Theory-04 hrs. /week

_____

Course outcomes:
1. Analyse the asymptotic performance of algorithms.
2. Demonstrate a familiarity with data structures and algorithms.
3. Compare algorithms based on time & space complexity.
4. Employ graphs to model real life problems, when appropriate. Develop algorithms that employ graph computations as key components, and analyse them.
5. Mapping of data structures like Stack, Queue and Linked List to real life problems.
6. Master the implementation of linked data structures such as linked lists and binary trees.
7. Be familiar with advanced data structures such as balanced search trees, hash tables, Red-Black trees, Btrees.
8. Understand Divide & Conquer approach, Greedy algorithm, Backtracking approach for algorithm design.
9. Be familiar with Branch and Bound&Dynamic programming

_____

UNIT-I   **[15]**
Algorithm Analysis: Introduction to algorithms, analysing and designing algorithms, Growth functions, asymptotic notations, **Divide and conquer**: The maximum sub array problem, matrix multiplication, solving recurrences: Substitution method, recursion tree method, master method. Binary Search, Max-Min problem, Sorting (Merge Sort, Quick Sort).
Hashing: Hashing, Direct address tables, Hash tables, Hash functions, collision resolution techniques.

Unit-II   **[15]**
Data Structures: Stacks, Queues, Linked list, Trees, General tree, Binary tree, binary search tree, operations on binary search tree, AVL tree, Red-Black Trees B-trees. **Graphs**: Representations of graph, Traversing Graphs, Breadth-first search, Depth-First Search, topological sort.

Unit-III   **[15]**
Greedy Algorithm: General Characteristics of greedy algorithms, Problem solving using Greedy Algorithm - Activity selection problem, Elements of Greedy Strategy, Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm), Shortest paths, The Knapsack Problem, Job Scheduling Problem, Huffman code.

Unit-IV   **[15]**
Backtracking: Introduction,N Queen Problem, Subset Sum, Hamiltonian Cycle
Branch and Bound – Introduction**,** 0/1 Knapsack, Travelling Salesman problem,
Dynamic programming: Introduction, Tabulation, memoization, Optimal Substructure
Property in Dynamic Programming

References:

1. Introduction to algorithms, Third Edition. by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, PHI
2. Fundamentals of Computer Algorithms, Second edition. By Ellis Horowitz, SartajSahani, SanguthevarRajasekaran, University Press.
3. Data structures and algorithm analysis in C, Second edition. By Mark Allen weiss

4. Fundamental algorithms by Donald E. Knuth, Pearson Education.

MSc-I Semester-I (Computer Science)
CC-102: Python Programming
MSc (Choice Based Credit System)
To be implemented from the academic year 2019-2020
External Marks-80 Internal Marks -20 Theory-04 hrs. /week

_____

Course Outcomes:
1. Understand principles of Python
2. Understand object oriented programming
3. Demonstrate file handling techniques
4. Understand how Python can be used for application development
5. Design Real life problems and think creatively about solution of them
6. Apply a solution clearly and accurately in a program using python

_____

Unit-I **(15 hrs)**
Installing Python, Simple program using Python, Expressions and Values, Variables and
Computer Memory, error detection, Multiple line statements, Designing and using functions,
functions provided by Python, Tracing function calls in memory model, omitting return
statement. Working with Text: Creating Strings of Characters, Using Special Characters in
Strings, Creating a Multiline String , Printing Information, Getting Information from the
Keyboard
Unit-II **(15 hrs)**
A Boolean Type, Choosing Statements to Execute, Nested If Statements, Remembering the
Results of a Boolean Expression Evaluation, A Modular Approach to Program Organization,
Importing Modules , Defining Modules, Testing Code Semi automatically Grouping
Functions Using Methods: Modules, Classes, and Methods , Calling Methods the Object-
Oriented Way, Exploring String Methods, Underscores.
Unit-III **(15hrs)**
Storing Collections of Data Using Lists: Storing and Accessing Data in Lists, Modifying
Lists, Operations on Lists, Slicing Lists, Aliasing, List Methods, Working with a List of
Lists. Repeating Code Using Loops: Processing Items in a List, Processing Characters in
Strings, Looping Over a Range of Numbers, Processing Lists Using Indices, Nesting Loops
in Loops, Looping Until a Condition Is Reached, Repetition Based on User Input, Controlling
Loops Using Break and Continue Reading and Writing Files: Kinds of files, Opening a File,
Techniques for Reading Files, Files over the Internet, Writing Files, Writing Algorithms that
use the File- Reading Techniques, Multiline Records
Unit-IV **(15hrs)**
Storing Data Using Other Collection Types: Storing Data Using Sets, Storing Data Using
Tuples, Storing Data Using Dictionaries, Inverting a Dictionary
Creating Graphical User interface: Building a Basic GUI, Models, Views, and Controllers,
Customizing the Visual Style Widgets, Object-Oriented GUIs, Regular expressions
Databases: Overview, Creating and Populating, Retrieving Data, Updating and Deleting,
Using NULL for Missing Data, Using Joins to Combine Tables, Keys and Constraints.

References:

1. Practical Programming: An introduction to Computer Science Using Python, second edition, Paul Gries, Jennifer Campbell, Jason Montojo, The Pragmatic Bookshelf.
2. Python for Informatics: Exploring Information, Charles Severance
3. Learning Python, Fourth Edition, Mark Lutz, O'Reilly publication
4. Introduction to Python for Computational Science and Engineering (A beginner's guide), Hans Fangohr
5. John V Guttag. "Introduction to Computation and Programming Using Python", Prentice Hall of India
6. R. Nageswara Rao, "Core Python Programming", Dreamtech

MSc-I Semester-I (Computer Science)
CC-103: Database Management System
MSc (Choice Based Credit System)
To be implemented from the academic year 2019-2020
External Marks-80 Internal Marks -20 Theory-04 hrs. /week

_____

Course Outcomes:
   After successful completion of the course, the student will be able to
   1) Define the terminology, features, classifications, and characteristics embodied in database systems.
   2) Demonstrate an understanding of the relational data model.
   3) Transform an information model into a relational database schema and to use a data definition language and/or utilities to implement the schema using a DBMS.
   4) Formulate, using SQL, solutions to a broad range of query and data update problems.
   5) Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.
   6) Use an SQL interface of a multi-user relational DBMS package to create, secure, populate, maintain, and query a database.
   7) Use PL/SQL for handing data in a database as per the user's requirement using programming features.
   8) Define various cursors and its implementation along with procedure and functions.

_____

UNIT-I [15]
Introduction to DBMS: Database Concept, Characteristics and architecture of DBMS, Database users, 3-tier architecture of DBMS-its advantages over 2-tier, Viewsof data – Schemas and instances, Data independence.Data models: Conventional data models, NDM & HDM, Physical Data Organization-Hashed files, Indexed files, B-trees
UNIT-II [15]
Database Design and E-R Model: Entities, Attributes, Relationships, Representation of entities, relationship set, Generalization, aggregation Structure of relationalDatabase and different types of keys, E-R diagrams, E-R design Issues in Relationaldatabase design, Functional dependencies.Normalization and Database Recovery systems: Codd's rules, Normalization,Database Recovery System- Failure classification, storage structure, recovery andatomicity, log-based recovery. Role of DBA.
UNIT-III [15]
Introduction to RDBMS: History, Generations and characteristics, difference between DBMS & RDBMS. Data constraint- primary key, foreign key, unique key, null, not null,default key etc.SQL: Introduction to SQL, Features of SQL, Basic data types, SQLstatements/commands, Set operations in SQL,order by and group by clause,likebetween, in, like, create index , view and join command Nested queries, GRANTand REVOKE, Commit, Rollback, Savepoint.Join concept: Simple, Equi, non-equi, Self, Outer join.View-Introduction, Create, Update, Drop, Index.SQL functions: MAX, MIN SORT, COUNT, AVERAGE, Numeric, String, Date Functions,Type conversion functions.
UNIT-IV [15]

Introduction to PL /SQL: Introduction, Difference between SQL AND PL/SQL, Blockdefinition structure and Data types, Block Functions - %Type, %RowType, Controlstatements, Looping statements and sequential statement, Exceptionhandling.Simple PL/SQL blocks.

Cursor management: meaning, types and importance, implicit and explicit cursormanagement using simple example.Trigger: meaning importance and types of trigger, examples using triggerProcedures-Definition, creating procedures, passing parameters.Function-Definition, syntax and calling methods, passing parameters.

Reference Books -
1. Introduction to database systems C. J. Date PearsonsEducation8th
2. Database system concept Korth, SilberschatzandSudarshanMGH 5th
3. Fundamentals of DatabaseSystemsElmasriNavathe PearsonEducation5th
4. SQL /PL SQL For Oracle 11G BlackBookDr.Deshpande WileyDreamtech2012
5. ORACLE PL/SQL Programming Scott Ulman TMH 9th
6. SQL, PL/SQL the programminglanguage of OracleIvanBayross BPB 4th
7. Advance Database ManagementSystemChakrabharati/DasguptaWileyDreamtech2011
8. Understanding SQL Martin Gruber BPB 2nd
9. SQL Scott Ulman TMH 4th

MSc-I Semester-I (Computer Science)
OE-104: Cyber Security
CBCS (Comp.Sci / Other Dept.)
MSc (Choice Based Credit System)
To be implemented from the academic year 2019-2020
External Marks-80 Internal Marks -20 Theory-04 hrs. /week

_____

Course Outcomes:

1) Realize the need for Cyber Security
2) Understand the need for Security in day to day communications
3) Understand the vulnerabilities in the Network and Computer System
4) Understand the cyber law and Cyber Forensics

_____

Unit-I   **(15 hr)**
Introduction to Cyber Security:Overview of Cyber Security, Internet Governance –
Challenges and Constraints, Cyber Threats:- Cyber Warfare-Cyber Crime-Cyber terrorism-
Cyber Espionage, Need for a Comprehensive Cyber Security Policy, Need for a Nodal
Authority, Need for an International convention on Cyberspace. Cyber Security
Vulnerabilities and Cyber Security Safeguards :Cyber Security Vulnerabilities-Overview,
vulnerabilities in software, System administration, Complex Network Architectures, Open
Access to Organizational Data, Weak Authentication, Unprotected Broadband
communications, Poor Cyber Security Awareness. Cyber Security Safeguards- Overview,
Access control, Audit, Authentication, Biometrics, Cryptography, Deception, Denial of
Service Filters, Ethical Hacking, Firewalls, Intrusion Detection Systems, Response,
Scanning, Security policy, Threat Management.

Unit-II   **(15 hr)**
Securing Web Application, Services and Servers: Introduction, Basic security for HTTP
Applications and Services, Basic Security for SOAP Services, Identity Management and Web
Services, Authorization Patterns, Security Considerations, Challenges. Intrusion Detection
and Prevention: Intrusion, Physical Theft, Abuse of Privileges, Unauthorized Access by
Outsider, Malware infection, Intrusion detection and Prevention Techniques, Anti-Malware
software, Network based Intrusion detection Systems, Network based Intrusion Prevention
Systems, Host based Intrusion prevention Systems, Security Information Management,
Network Session Analysis, System Integrity Validation.

Unit –III   **(15 hr)**
Cryptography and Network Security: Introduction to Cryptography, Symmetric key
Cryptography, Asymmetric key Cryptography, Message Authentication, Digital Signatures,
Applications of Cryptography. Overview of Firewalls- Types of Firewalls, User
Management, VPN Security Protocols: - security at the Application Layer- PGP and
S/MIME, Security at Transport Layer- SSL and TLS, Security at Network Layer-IPSec.

Unit –IV   **(15 hr)**
Cyberspace and the Law: Introduction, Cyber Security Regulations, Roles of International
Law, the state and Private Sector in Cyberspace, Cyber Security Standards. The INDIAN
Cyberspace, National Cyber Security Policy 2013.Cyber Forensics: Introduction to Cyber
Forensics, Handling Preliminary Investigations, Controlling an Investigation, Conducting

disk-based analysis, Investigating Information-hiding, Scrutinizing E-mail, Validating E-mail header information, Tracing Internet access, Tracing memory in real-time.

References:
1. Preston Gralla, How Personal and Internet Security Work, Que Publications
2. Alfred Basta and Wolf Halton, Computer Security Concepts, Issues and Implementation,
   Cengage Learning
3. Digital Defense: A Cybersecurity Primer by Joseph Pelton , Indu B. Singh
4. Cryptography and Network Security: Principles and Practice by William Stallings
5. Computer and Information Security Handbook by John R. Vacca .
6. Cyberlaw: The Law of the Internet and Information Technology by Brian Craig .
7. Cyber Warfare: Techniques, Tactics and Tools for Security Practitioners 2nd Edition by Jason Andress (Author), Steve Winterfeld (Author)

MSc-I Semester-I (Computer Science)
CCPR-105: Python Lab
MSc (Choice Based Credit System)
To be implemented from the academic year 2019-2020
External Marks-80 Internal Marks -20 Practical -06 hrs. /week

_____

Lab assignments based on Python Programming course.

MSc-I Semester-I (Computer Science)
CCPR-106: Database Lab
MSc (Choice Based Credit System)
To be implemented from the academic year 2019-2020
External Marks-80 Internal Marks -20 Practical -06 hrs. /week

_____

Lab assignments based on Database Management System course.

MSc-I Semester-I (Computer Science)
CCPR-107: Project
MSc (Choice Based Credit System)
To be implemented from the academic year 2019-2020
External Marks-80 Internal Marks -20 Practical -06 hrs. /week

_____

Course outcomes:
1) Gain skills as they apply knowledge effectively in diverse contexts.
2) Analyse and model requirements and constraints for the purpose of designing and implementing software artefacts and IT systems
3) Design and implement software solutions that accommodate specified requirements and constraints, based on analysis or modelling or requirements specification
4) Present a clear, coherent and independent exposition of software applications, alternative IT solutions, and decision recommendations to both IT and non-IT personnel via technical reports of professional standard and technical presentations.
5) Team work: Work effectively in different roles, to form, manage, and successfully produce outcomes from teams, whose members may have diverse cultural backgrounds and life circumstances, and differing levels of technical expertise.

A software module development should be carried out as part of Project.

MSc-I Semester-II (Computer Science)
CC-201: Web Technology
MSc (Choice Based Credit System)
To be implemented from the academic year 2019-2020
External Marks-80 Internal Marks -20 Theory -04 hrs. /week

_____

Course Outcomes:
After Completion of this course the student would be able to;
1.To familiarize a student with windows and web-based application
2.To provide a student with the solid foundation of the syntax and semantics of C# as well as architecture of the .NET framework
3.Debug and deploy ASP.NET web applications
4.Discuss the insights of internet programming and implement complete application over the web
5.To inculcate skills pertaining to data access technology geared to facilitate the development of disconnected systems using .NET platform.
6.To familiarize the student with the development of windows-based application using
C#
7.To familiarize the student with the development of web-based application using ASP.NET
8.Handle various toolkit like AJAX
9.Utilize the concepts of JavaScript

_____

UNIT -I (15 Hrs)
Introduction: .NET framework and its architecture, CLR, JIT, CTS, Metadata, .NET Revolution, Characteristics of C#, Programming structure of C#, scope of variables, boxing & unboxing, Nullable Data types, Conditional statements, Arrays, Loops, class, inheritance, polymorphism, Exception handling, delegates

UNIT- II (15 Hrs)
Introduction to C#: Windows form controls, containers, Data Controls, Dialog Controls, sample application development, Deployment of C# application
ASP.NET: Page Life Cycle, ASP.NET server-side Controls, Data Controls, Server-side Validation Controls, Language Integrated Queries (LINQ), Web services,

UNIT- III (15 Hrs)
State management: Server side & Client-side state management, Caching in ASP.NET: Page caching, data caching, fragment caching AJAX: Introduction to AJAX, Various extenders of AJAX
ADO.NET: ADO.NET Architecture, ADO.NET namespaces, Working with ADO.NET, Data controls, FormView Control, Working with Repeater, connected architecture, Disconnected Architecture, Three-layer Architecture, Working with Master pages

UNIT- IV **(15 Hrs)**

Microsoft SQL Server: Database designing, Tables, Views, Stored Procedures, taking backup of database, creating script of database

Java Script: JavaScript Overview, Data types, variables, scope of variables, casting, data type conversion rules, Expressions and operators. Arrays. Built-in functions, and Built-in objects- String, Date, Math. Three types of dialog boxes-alert, prompt, confirm. Custom Functions.

References:

1.C# 4.0 The Complete Reference by Herbert Schildt
2.Essential C# 4.0" by Mark Michaelis and Eric Lippert
3."C# in Depth" by Jon Skeet
4."Head First C#: A Learner's Guide to Real-World Programming with C#, XAML, and .NET" by Jennifer Greene and Andrew Stellman
5."Microsoft Visual C# Step by Step" by Sharp John
6."Let Us C#" by Yashavant P Kanetkar
7.Javascript: The Complete Reference by Thomas Powell & Fritz Schneider
8.JavaScript Pocket Reference, 3rd Edition By David Flanagan
9.Professional JavaScript for Web Developers, 4th Edition by Matt Frisbie

MSc-I Semester-II (Computer Science)
CC-202: Advanced Java
MSc (Choice Based Credit System)
To be implemented from the academic year 2019-2020
External Marks-80 Internal Marks -20 Theory-04 hrs. /week

_____

Course Objectives:

1) The student will be able to develop distributed business applications, develop web pages using advanced server-side programming through servlets and Java server pages.
2) Demonstrate approaches for performance and effective coding
3) Develop Java client/server applications.
4) Develop distributed applications using RMI
5) Develop component-based Java software using JavaBeans
6) Develop server side programs in the form of servlet
7) Understand the multi-tier architecture of web-based enterprise applications using Enterprise JavaBeans (EJB) ,use Struts frameworks, which gives the opportunity to reuse the codes for quick development and map Java classes and object associations to relational database tables with Hibernate mapping files

_____

Unit -1   **(15 Hrs)**
Overview of Java programming, Java Servlets: Servlet basics, servlet life cycle , Generic and HTTP servlets, The Servlet API, javax.servlet and javax.servlet.http package, session tracking using session and cookies, web deployment descriptor, web.xml.
Remote Method Invocation–Introduction, architecture, defining remote objects, creating stubs and skeleton, object serialization, dynamically loaded classes, RMI activation, registering remote objects, marshaled objects.

Unit-II   **(15 Hrs)**
 Java Database Connectivity.JDBCoverview,Architecture, Types of JDBC Drivers, DriverManager class, database connection statements,Resultset, transaction, Metadata and Aggregate functions , callable statements, Connection to various back ends.
Java Server Pages (JSP): Introduction to JSP tags and directive, Request String, User Sessions,   Cookies,Session   objects.

Unit-III   **(15 Hrs)**
Java Beans: Basics of designing JavaBeans, Java Bean design patterns, creating and using properties, using events to communicate with other components.
Enterprise Java Bean: Preparing a Class to be a JavaBean, Creating a JavaBean, JavaBean Properties,Types of beans, Stateful Session bean, Stateless Session bean, Entity bean

Unit –IV   **(15 Hrs)**
Struts Framework: An introduction to Struts, building a simple struts application. Action Servlet, Model, view and Controller layers, validator, declarative exception handling, Introduction to struts tag libraries and struts configuration files.
Spring and Hibernate: Spring API libraries, Designing spring applications. Spring persistence using JPA. Spring web flow, Using spring MVC to build web pages. Integrating and configuring hibernate. Building a simple application.

References:-
1. Java 2 Complete Reference - (Tata McGraw Hill)
2. Java server pages
3. Java 2EE – Ivan Bayross (PHI)
4. Java 2 Black Book –(DreamTech)
5. Orfali, "The essential Distributed Object Survival Guide".
6. Valesky, "Enterprise Java Beans", Addison Wesley.
7. A Complete Reference Struts (Second Edition) - JamesHomes. Tata McGraw-Hill Edition.
8. Struts 2 - Black Book. (Second Edition). Kogent Solutions Inc. dreamtech press.

MSc-I Semester-II (Computer Science)
CC-203: Android Fundamentals Development
MSc (Choice Based Credit System)
To be implemented from the academic year 2019-2020
External Marks-80 Internal Marks -20 Theory-04 hrs. /week

Course outcomes:
1) Understand Android Studio Environment and application structure.
2) Demonstrate different layouts, views, activities and intents
3) Testing and debugging of application.
4) Design good user interface for the application.
5) Able to store, retrieve and load data
6) Demonstrate background tasks & events
7) Understand publishing of app.

_____

Unit-I **(15hrs)**
The Basics: Introduction to Android, Creating First Android App,Layouts, Views and Resources, Scrolling Views and Lists, Activities and Intents: The Activity Lifecycle and Managing State, Starting Activities with Implicit Intents, Background Tasks, Connecting to the Internet, Concept: AsyncTaskLoader, Testing and Debugging & Backwards Compatibility: Debugging app, Testing app, Support libraries.

Unit- II **(15hrs)**
User Interface: User Interaction and intuitive navigation, User input, Menus, Navigation styles, Delightful User Experience, Themes and Styles, Material Design, Adapt layouts for multiple devices and orientations, Accessibility, Localization, Testing UI

Unit-III **(15hrs)**
Data -- Saving, Retrieving, Loading: Storing Data in app Concepts: Overview to storing data, Storing Data using SQLite, Store data using SQLite database, Sharing Data: Content Resolvers and Content Providers, Using Content Resolvers to access data, Content Providers, Loading Data using Loaders, Using Loaders to Load and Display Data

Unit-IV **(15hrs)**
Background Tasks and Events:Using Services for long running tasks, Services, Summary of the different ways to do background work, Triggering, Scheduling, and Optimizing Background Tasks, Alarms and Broadcast Receivers, Transferring Data Efficiently
Permissions and Libraries: Permissions, Libraries, Security best practices, Notifications and Widgets: Notifications, Widgets, Publishing App: Monetizing app, Making and publishing APKs

References:
1. Professional Android 4 Application Development Reto Meier Wrox
2. Android Application Development: Programming with the Google SDK 2009 by Rick Rogers, John Lombardo, ZigurdMednieks, G. Blake Meike
3. Android Programming for Beginnersby John Horton, Published by Packt Publishing Ltd.

4. Programming Android, Book by G. Blake Meike, Laird Dornin, Masumi Nakamura, and Zigurd R. Mednieks, O'Reilly Publishers.

MSc-I Semester-II (Computer Science)
Elective-I:CE-204
CE-204.1: Software Project Management
To be implemented from the academic year 2019-2020
External Marks-80 Internal Marks -20 Theory-04 hrs. /week

Course outcomes:
1) To understand Software Project Models and Software Management Concepts.
2) To understand the various methods of Cost Estimation.
3) To Study about Software Quality Management.
4) To Study about Emerging Trends in Software Management.
5) To understand Project Evaluation.
_____

Unit-I   (15)
Project Management: Concept of project Management, Project Organization, Planning a software project, Project management life cycle, Risk management, Identification of Risks, Risk Analysis, Risk Planning and Monitoring.
Unit-II   (15)
Software Project Estimation: Concept of Project Estimation, Different methods of software project estimation (COCOMO model, Delphi cost estimation etc.), Function point analysis, Software Project Management Tools and Techniques- PERT & Gantt Charts
Unit-III   (15)
Software Quality Management and Testing: Quality Assurance and Standards, Quality Planning, Quality control, Role of testing in Software, development, Testing Procedure, Defect Management. Team Structure, Team Communication, Managing customer expectations, Group Behaviour.
Unit -IV(15)
Project Evaluation and Emerging Trends : Strategic Assessment–Technical Assessment–Cost Benefit Analysis–Cash Flow Forecasting– Cost Benefit Evaluation Technique–Risk Evaluation–Software Effort Estimation. Emerging Trends: people Focused Process Models.

Reference:
1. Software Project management By Edwin Bennatan
2. Software Engineering By Roger S. Pressman
3. Software Engineering concepts by Richard Fairley
4. Software Project Management by S.A. Kelkar
5. Software Engineering by IAN Sommerville
6. System Analysis and Design Methods By J.L Whitten ,L.D.Bentley and K.C. Dittman
7. Ramesh Gopalaswamy , "Managing and global Software Projects", Tata McGraw Hill Tenth
Reprint, 2011.

MSc-I Semester-II (Computer Science)
Elective-I:CE-204
CE-204.2: Data Science Foundations
To be implemented from the academic year 2019-2020
External Marks-80 Internal Marks -20 Theory-04 hrs. /week

Course outcomes:
After successful completion of this course students are able to:
1) Understand the concept of Probability and Information Theory
2) Compute probabilities of events
3) Explain Bayes rule and compute probabilities using Bayes rule
4) To formulate and solve classification problem
5) To investigate clustering techniques for particular data
6) To mine frequent pattern in a dataset
7) To get acquainted with natural language processing techniques

_____

Unit – I (15hrs)
Probability and Information Theory: Introduction to Probability, Random Variables, probability Distributions, Marginal Probability, Conditional Probability, The Chain Rule of Conditional Probabilities, Independence and Conditional Independence, Expectation, Variance and Covariance, Common Probability Distributions, Useful Properties of Common Functions, Bayes' Rule, Information theory, Structured Probabilistic Models

Unit – II (15hrs)
Data understanding and data cleaning, concept of supervised and unsupervised learning. Classification techniques: Classification: Preliminaries, general approach to solve classification problem, Decision tree induction, Rule-based classifier, Nearest-Neighbor classifier, Bayesian Classifiers, Support Vector Machine.

Unit – III (15hrs)
Cluster analysis: Introduction, Types of Clustering, Types of Clusters. K-means algorithm, Agglomerative Hierarchical Clustering, DBSCAN, Prototype based clustering and Density based clustering, introduction to Web Mining. Association analysis: Problem definition, Frequent Itemset Generation, Apriori Principle, apriori algorithm, Maximal Frequent itemset, closed frequent itemset. FP-growth algorithm

Unit – IV (15hrs)
Introduction to Natural Language Processing (NLP), Data Preparation: punctuation removal, stop-words removal, numeric value removal, frequent words removal, rare words removal, spelling correction, tokenization, stemming, lemmatization. Feature Engineering: count vectors as features, Term Frequency-Inverse Document Frequency (TF-IDF), TF-IDF vectors as features, word level TF-IDF, N-Gram level, TF-IDF, Character level TF-IDF, Inverse

Document Frequency, word embedding as features, Text/NLP based features, Topic Models as features, word2vec.

References:
1) Bird, S., Klein, E., &Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".
2) Data Mining concepts and techniques --- Jiawei Han and MichelineKamber , Elsevier
3) Data Mining: Introductory and Advanced Topics - Margaret H. Dunham, Pearson education
4) Data Mining: Practical Machine Learning Tools and Techniques, Ian H. Witten, Eibe Frank
5) Goodfellow, I., Bengio, Y., Courville, A., &Bengio, Y. (2016). *Deep learning* (Vol. 1). Cambridge: MIT press.
6) Introduction to Data Mining – Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson education.
7) Kumar, E. (2013). *Natural language processing*. IK International Pvt Ltd.
8) Ross, S. M. (2014). *Introduction to probability models*. Academic press.

MSc-I Semester-II (Computer Science)
Elective-I:CE-204
CE-204.3: Application Security Analyst
To be implemented from the academic year 2019-2020
External Marks-80 Internal Marks -20 Theory-04 hrs. /week

Course outcomes:
1) Understand application development life cycle
2) Identify vulnerabilities, entry points, error code etc.
3) Get acquainted with password cracking techniques and prevention techniques.
4) Understand authentication and authorization vulnerabilities.
5) Demonstrate use of SQL injection.
6) Investigate Session & browser manipulations

_____

Unit-I   **(15hrs)**
Introduction: standard Systems Development Lifecycle (SDLC) practices and process, enterprise information technology (IT) architecture, key vulnerabilities to applications , Risk management frameworks and processes, Site Mapping & Web Crawling, Server & Application Fingerprinting, Identifying the entry points, File extensions handling, Page enumeration and brute forcing, **Comments in code** – view source, Looking for leftovers and backup files, Admin interfaces, Robots.txt, Analysis of error code

Unit-II   **(15hrs)**
Basic security concepts: hardware and software vulnerabilities
Malware: Types of Malware, Virus, Trojan, Key logger, File Binder, Testing FUD and Countermeasure.
Password Cracking and Prevention: Introduction, Password Cracking Techniques, Dictionary Based Attack, Brute force Attack, Cracking common files password, Cracking web based password, Password reset flaws, Password change flaws, Cracking Wifi password, Countermeasures for users, Countermeasures for System Administrators.

Unit-III   **(15hrs)**
Authentication & Authorization vulnerabilities: Authentication concepts, scenarios, User enumeration, Direct page requests, Parameter modification, Lack of SSL at login pages , Bypassing weak CAPTCHA mechanisms, Login without SSL, Authorization: RBAC, Authorization bypassing, Parameter tampering, Forceful browsing, Rendering based Authorization, Client side validation attacks, Insecure direct object reference.
File handling: Insecure file handling, Path traversal, Canonicalization, Uploaded files backdoors, Insecure file extension handling, Directory listing, File size, File type, Malware upload

Unit-IV   **(15hrs)**
Input vulnerabilities: SQL injection, Common implementation mistakes - authentication bypassing using SQL Injection, Cross Site Scripting, Reflected VS. Stored XSS

Command injection, Special chars - '&<>, empty

Session & browser manipulation: Session & browser manipulation attacks, Session management techniques, Cookie based session management, , cookie properties, Cookies - secrets in cookies, tampering, Exposed session variables, Missing Attributes – httpOnly, secure Session validity after logoff, Long session timeout, Session keep alive – enable/disable, Session id rotation, Cross Site Request Forgery (CSRF), Open redirect , **IT Auditing**–Definition, Objectives and strategies

References:

1. Bryan Sullivan and Vincent Liu, Web Application Security, A Beginner's Guide, McGraw Hill
2. Michael Cross, Developer's Guide to Web Application Security, Syngress Publishers
3. Justin Clarke , SQL Injection Attacks and Defense, Elsevier
4. Nigel Chapman, Authentication and Authorization on the Web, Publisher   Macavon Media, 2012
5. https://owasp.org/

Elective-I:CE-204
CE-204.4: Cloud Computing
To be implemented from the academic year 2019-2020
External Marks-80 Internal Marks -20 Theory-04 hrs. /week

Course outcomes:
1) Introduce the broad perceptive of cloud architecture and model
2) Apply different cloud programming model as per need.
3) Explore some important cloud computing driven commercial systems such as Google Apps, Microsoft Azure and Amazon Web Services and other businesses cloud applications
4) To learn how to use Cloud Services.
5) To implement Task Scheduling algorithms.
6) To build Private Cloud.

_____

Unit – I: (15hrs)
Fundamentals of Cloud Computing: Overview of Cloud Computing, Evolution of Cloud Computing, Types of Clouds, KeyCharacteristics of Cloud Computing, Intranets and Clouds. Benefits and challenges of cloudcomputing, Usage scenarios and Applications, Regulatory issues, major players in cloudcomputing.

Unit – II (15hrs)
Cloud Models & Services:
Cloud Models – Benefits of Cloud Models, Public, Private, Hybrid, and Community Clouds Types of Clouds Services: SaaS, PaaS, IaaS, DaaS, MaaS, CaaS. Service Providers: Google App Engine, Microsoft Azure, Amazon EC2, IBM, Sales Force; Introduction to MapReduce, GFS, HDFS, Hadoop Framework.

Unit – III (15hrs)
Essentials & Collaborating with Cloud:
Hardware and Infrastructure – Clients, Security, Network, Services; Accessing Cloud –
Platforms, Web Applications, Web APIs, Web Browsers; Cloud Storage – Overview, Cloud Storage Providers; Standards – Application, Client, Infrastructure, Service; Centralizing Email
Communications, Collaborating on Calendars, Schedules & Task Management, Event Management, Project Management and Contact Management.

Unit-IV (15hrs)
Virtualization & Security for Clouds
Need for Virtualization – Pros and Cons of Virtualization, Types of Virtualization, System VM,
Process VM, Virtual Machine Monitor – Virtual Machine Properties, HLL VM, Hypervisor –

VMWare, Virtual Box, Hyper-V; Case Studies on Cloud Data Centres.
Security in Clouds – Cloud security challenges, SaaS as Service Security; Common
Standards
– Open Cloud Consortium, Distributed Management Task Force, Standards for Application
Developers; Standards for Messaging – Standards for Security, End User access to cloud
computing, mobile internet devices and the cloud.

Reference Books:
1. Bloor R., Kanfman M., Halper F. Judith Hurwitz "Cloud Computing for Dummies" (Wiley
India Edition) 2010, ISBN 978-0-470-48470-8.
2. Barrie Sosinsky, "Cloud Computing Bible, " Wiley India Pvt. Ltd. 2012
3. George Reese, "Cloud Application Architectures", Shroff/O' Reilly, 2009.
4. John W. Rittinghouse& James F. Ransome, "Cloud Computing: Implementation,
Management and Security", CRC Press, 1st Edition, 2009, ISBN 978-1439806807.
5. Antohy T Velte, Toby J. Velte, Robert Elsenpeter, Cloud Computing: "A Practical
Approach", McGraw Hill, 2009. ISBN 978-0-07-068351-8
6. Michael Miller, Cloud Computing: "Web-Based Applications That Change the Way You
Work and Collaborate Online", Que Publishing, August 2008. ISBN 978-0-7897-3803-5
7. George Reese, "Cloud Application Architecture", O'Reilly and Associates.

MSc-I Semester-II (Computer Science)
CCPR-205: Web Technology Lab
To be implemented from the academic year 2019-2020
External Marks-80 Internal Marks -20 Practical -06 hrs. /week

Course outcomes:
1) Debug and deploy ASP.NET web applications
2) Discuss the insights of internet programming and implement complete application over the web
3) Use the features of Dot Net Framework along with the features of C#
4) Build and host web applications using ASP.NET
5) Develop and deploy Windows applications
6) Handle data by using ADO.NET architecture
7) Create database-driven ASP.NET web applications and web services
8) Handle various toolkit like AJAX
9) Utilize the concepts of JavaScript
10) Develop and deploy a website using HTML

MSc-I Semester-II (Computer Science)
CCPR-206: Advanced Java Lab
To be implemented from the academic year 2019-2020
External Marks-80 Internal Marks -20 Practical -06 hrs. /week

Course outcomes:
1. Define & explain applet Life cycle
2. Differentiate local and remote applet
3. Write the code for a simple Java applet
4. Explain applet tag and its parameter
5. Use the methods of the Applet and Component classes required for a basic applet
6. Describe the classes in the AWT package that relate to the Applet class
7. Describe the AWT graphics explain controls and how to apply them in the container
8. Develop simple programs using Event class and Event Listener Interface
9. Develop a program for steps to connect a database

10. Describe the Basics of JDBC
11. Explain the different Types of JDBC drivers & their advantages and Disadvantages
12. Develop program to use JDBC to query a database and modify
13. Describe life cycle of servlet
14. Develop program using javax.servlet package
15. Explain JSP Architecture and its Life cycle
16. Develop simple program

MSc-I Semester-II (Computer Science)
CCPR-207: Project
To be implemented from the academic year 2019-2020
External Marks-80 Internal Marks -20 Practical -06 hrs. /week

Course outcomes:
6) Gain skills as they apply knowledge effectively in diverse contexts.
7) Analyse and model requirements and constraints for the purpose of designing and implementing software artefacts and IT systems
8) Design and implement software solutions that accommodate specified requirements and constraints, based on analysis or modelling or requirements specification
9) Present a clear, coherent and independent exposition of software applications, alternative IT solutions, and decision recommendations to both IT and non-IT personnel via technical reports of professional standard and technical presentations.
10) Team work: Work effectively in different roles, to form, manage, and successfully produce outcomes from teams, whose members may have diverse cultural backgrounds and life circumstances, and differing levels of technical expertise.

A software module development should be carried out as part of Project work.