GONDWANA UNIVERSITY
GADCHIROLI

NATIONAL EDUCATION POLICY(NEP) 2020

Scheme and Syllabus
of
M.Sc. (Computer Science)

Prepared by BOS Member (Computer Science):

- **Dr. K. D. Kalaskar**
- **Dr. S. B. Kishor**
- **Dr. R. N. Jugele**
- **Dr. P. B. Dhumane**
- **Mr. V. K. Tembhurne**
- **Dr. M. L. Jiwtode**
- **Dr. M. N. Quadri**
- **Mr. Abhishek Acharya**

General Guidelines

1. The duration of M.SC. (COMPUTER SCIENCE) the course shall be of two academic years consisting of Four semesters with University examinations at the end of each semester namely:
    a. M.SC. (COMPUTER SCIENCE) Part I: Semester I & Semester II Examination
    b. M.SC. (COMPUTER SCIENCE) Part II: Semester III & Semester IV Examination
2. The examination specified in the preceding paragraph shall be held semester vise at such places and such dates as may be appointed by the university.
3. Subject to their compliance with the provisions of this Direction and of other directions/ordinances in force from time to time, the candidate who has prosecuted a regular course of study for not less than one academic year before the examination shall be eligible for admission to the examination.
4. **Eligibility: - For M.SC. (COMPUTER SCIENCE) - I**

    Candidates who have passed Bachelor of Computer Science (B.Sc. (Computer Science) /Bachelor of Computer Application (BCA)/ Bachelor of Computer Science (BCS)/ Bachelor of Science in Information Technology/ BE in (Computer Science/Computer Engineering/Computer Technology / Information Technology) degree courses
5. The fees for the examination shall be as prescribed by the university from time to time.
6. The scope of the subject of M.Sc. (Computer Science) shall be as indicated in the respective syllabi in force from time to time.
7. The medium of instruction shall be English.
8. The theory papers and practicals in which an examinee is to be examined, the maximum marks prescribed for each paper and practical, and the minimum passing mark which the examinee must obtain to pass in the subject and the examination is prescribed in the Later part**.**
9. In the Fourth semester, there will be one Major project based on computer topics Students need to develop one application on the latest technology as prescribed by the department.
10. **Absorption Scheme from other Universities to Gondwana University**
11. A No objection certificate from Previous College and/or from a previous university is required along with the following satisfaction of criteria,

Admission to 3rd Semester if student cleared 1st year of M.Sc. (COMPUTER SCIENCE).

Syllabus
of

M.Sc. – I
(Computer Science)
(Sem – I)

Scheme

M.Sc. – I Computer Science – Semester-I

Subject	Course/ Paper Code	Paper Name	Teaching Scheme			Credits Assigned				% of Assessment			
			Theory	Prac.	Tut.	Theory	Prac.	Tut.	Total	IA	UE	Total	Min. Passing Marks
Core	01MSCCS01	Scripting Languages and Information Retrieval	4	-	-	3	-	-	3	20	80	100	40
	01MSCCS02	Python Programming	4	-	-	3	-	-	3	20	80	100	40
	01MSCCS03	Advanced Java	4	-	-	3	-	-	3	20	80	100	40
Elective PGMSCC04 (Student Shall select any one from the elective group)	01MSCCS04.1	Cloud Computing	4	-	-	3	-	-	3	20	80	100	40
	01MSCCS04.2	Discrete Mathematics											
	01MSCCS04.3	Theory of Computation & System Programming											
	01MSCCS04.4	Network Security											
	01MSCCS04.5	Computer Graphics											
RM	01MSCCS05	Research Methodology & Publication Ethics	4	-	-	3		-	3	20	80	100	40
LAB - I	01MSCCS06	Based on 01MSCCS01 & 01MSCCS02	-	4	-	-	2	-	2	25	75	100	50
LAB - II	01MSCCS07	Based on 01MSCCS03 & 01MSCCS04	-	4	-	-	2	-	2	25	75	100	50
Ability Enhancement	01MSCCS08	Seminar	-	1	-	-	1	-	1	50	-	50	20
Total			20	9	-	15	5	-	20	200	550	750	320

- **Core:** Major theory papers in the concerned subject.
    - From Elective Courses, students need to select only one paper.
- **IA** (Internal Assessment): It will be evaluated by Internal Examiner appointed by the college in consultation with the University. (Refer to Appendix A)
- **UE** (University Examination): It will be evaluated by External Examiner appointed by the university. (Refer to Appendix B)
- **For Paper Patterns refer to Appendix C**

- **In Course/Paper Code**
    - 01: Represent it as a Semester One.
    - MSC: Represent is MASTER OF SCIENCE
    - CS: Represent the subject COMPUTER SCIENCE
    - Last two numbers: Represent Paper No. for Ex. 01 Represent Paper no. 1

- **Lab*:**

  1) Not more than two students should be allowed to do practicals on a single PC.

  2) Wherever possible Practicals should be performed using Open-Source Software.

M.Sc. (Computer Science) - I (SEMESTER – I)
Paper Code: 01MSCCS01
PAPER – I: Scripting Languages and Information Retrieval

Credit: 3 **Max. Marks: 80**

Program Learning Outcome:

1. Structure text and image content for the web using HTML5 and Learn semantic markup new to the HTML standard
2. Style a web page using Cascading Style Sheets (CSS) and validate a web page.
3. Describe the Document Object Model (DOM) using JavaScript and Summarize managing web page styles using JavaScript and CSS also will understand how to script forms.

Unit I: Advanced HTML and CSS

HTML: Introduction HTML Documents, Basic structure of an HTML document creating an HTML document Markup Tags, Heading-Paragraphs, Line Breaks, and HTML Tags. Elements of HTML Introduction to elements of HTML- Working with Text, Lists in HTML, Tables, and Frames, Working with Hyperlinks, Images and Multimedia, Forms in HTML, advanced input elements, input restrictions, use of <Meta> tag, Insert audio and video in HTML, Image Map in HTML.

CSS:-Creating Style Sheet, CSS Properties, CSS Styling (Background, Text Format, Controlling Fonts) Working with block elements and objects, Working with Lists and Table, CSS Id and Class Box Models (Introduction, Border properties, Padding Properties, Margin properties), CSS Advanced (Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class, Navigation Bar, Image Sprites, Attribute sector), CSS Color Creating page Layout and Site Designs.

UNIT II: Java Script

Java Script: The Nature of JavaScript, Script, Script Writing Basic, Auditioning of Interactivity to a Web Page. Creating Dynamic Web Pages, Java Scripting Your forms. Creating Scrolling Messages Animating Graphics, creating a Floating Toolbar, Setting Up Tool Bar, Window, Designing Image Map Navigation

UNIT III: VB Script

Introduction VB Script: Evolution of Scripting Language, Introduction to VB Script, Features of VB Script, Data Types in VB Script, Elements of VB Script: Identifiers, Operators, Control Statements, Control Structure

Functions: Variant Function, Math Function, formatting Function, String Manipulation Function, Type Conversion Methods Supported by VB Script, Arrays in VB Script, Regular Expression

UNIT IV: Information Retrieval

Information Retrieval- Boolean Retrieval, The Term Vocabulary and Postings Lists, Dictionaries and Tolerant Retrieval, Index Construction, Index Compression, Scoring, Term Weighting, and The Vector Space Model, Computing Scores in a Complete Search System, Evaluation in Information Retrieval, Relevance Feedback and Query Expansion, XML Retrieval, Probabilistic Information Retrieval

Books:

1) Beginning CSS: Cascading Style Sheets for Web Design, Wiley India, author Ian Pouncey, Richard York.
2) HTML 5 in simple steps publication Kogent Learning, Dreamtech Press
3) Mastering HTML, CSS & Javascript Web Publishing by Laura Lemay (Author), Rafe Colburn (Author), Jennifer Kyrnin (Author), BPB publication

References:

1) C. Xavier, "Web Technology and Design", ISBN-812214508/9788122414509.
2) Dr. S. B. Kishor, Rajani Singh, "Web Designing", Das Ganu Prakashan

M.Sc. (Computer Science) - I (SEMESTER – I)
Paper Code: 01MSCCS02
PAPER-II: PYTHON PROGRAMMING

Credit: 3 **Max. Marks: 80**

Program Learning Outcome:

1. Python is a general-purpose programming language that is becoming ever more popular for data science.
2. To understand why Python is a useful scripting language for developers.
3. To learn how to design object-oriented programs with Python classes.

Unit I: Basic Elements, Control Statements, String Manipulation, and Collection

Introduction to Python, Features of Python, Different Methods to Run Python, Basic Syntax, Writing and Executing simple programs, Comments, Basic Elements (Data Types, Constants, Variable, Operator, Precedence of Operator, Expression, Type Conversion, Objects), Indentation in Python, Input and Output in Python: Printing on screen, reading data from keyboard, import function

Control Statements: Branching (if, else, elif), Iteration (while, for, Nested Loop), Terminating Loops, Skipping Specific conditions,

String Manipulation: Declaring strings, strings function. Range and enumerate functions, Manipulating Collections: Tuples, Lists, Sets, Dictionaries, Built-in methods of lists, sets, and dictionaries, Mutable and Immutable Objects.

Unit II: Functions, Modules, and Exception Handling

Functions Definition, Advantage of function, types of function, Function Calling, Function parameters formal parameters, actual parameter, and Anonymous function, Function Arguments (Required, Keyword, Default), Global and local variable, Recursion,

Exceptions, Built-in Exceptions, Argument of an Exception, Python Standard Exceptions (IndexError, OverflowError, ZeroDivisionError, RuntimeError), Exception Handling. User Defined Exception and Raising Exception

Unit III: Module, File Handling, and Object-Oriented Programming

Modules, Built-in Modules (math, statistics, time, random), Creating Modules, packages

File Handling (Opening, Closing, Writing, Reading),

Class Definition, Object Creation, Built-in Attribute Methods, Encapsulation, Data Hiding, Inheritance, Multi-Level Inheritance, Polymorphism (Method Overriding, Operator Overloading)

Unit IV: Arrays and Data Visualization

Arrays in Python, Numpy Module, Creating Arrays (array, zeros, ones, empty, linspace, arrange, random), Two-Dimensional Array, Indexing, Slicing, Iterating, Copying, Splitting, Shape Manipulation (reshape, transpose, resize), Arithmetic Operations on Arrays.

Data Visualization in Python (matplotlib Module, pyplot, plot(), hist, scatter, bar charts, Formatting, figure(), subplot(), text(), xlabel(), ylabel(), title(), Plotting Simple Mathematical Functions (sin x, x2)

Books:

1) Bill Lubanovic, "Introducing Python", Shroff Publication
2) Joseph Joyner, "Python Programming for Beginners", ISBN 13-9781633830394

References:

1) Paul Gries, Jennifer Campbell, Jason Montojo, Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf
2) Adesh Pandey, Programming Languages – Principles and Paradigms, Narosa
3) Taming Python By Programming, Dr. Jeeva Jose, Khanna Publishing
4) Introduction to Computation and Programming Using Python with Application to Understanding Data - John V. Guttag, PHI (2016)
5) https://www.numpy.org/devdocs/user/quickstart.html
6) https://matplotlib.org/users/pyplot_tutorial.html

M.Sc. (Computer Science) - I (SEMESTER – I)

Paper Code: 01MSCCS03

PAPER – III: ADVANCED JAVA

Credit: 3 **Max. Marks: 80**

Program Learning Outcome:

1. Learn the Internet Programming, using Java Applets.
2. Create a full set of UI widgets and other components, including windows, menus, and buttons, checkboxes, and text fields, scrollbars and scrolling lists, using Abstract Windowing Toolkit (AWT) & Swings.
3. Apply event handling on AWT and Swing components.
4. Create dynamic web pages, using Servlets and JSP.

Unit I: Introduction to Java, Objects, and Classes

History of Java, Features of Java, Java Development Cycle: Source Code, Byte Code, Java Development Kit (JDK), The Java Virtual Machine, JRE, Linker, Loader. Programming Concepts of Basic Java: Identifiers and Keywords, Operators, Variables, Types of Variables, Data Types, Control structures, decision-making statements, Arrays, Array Methods, Strings, and their methods. Oops Concepts of Java: Declaring Objects, Methods, Scope Rules, Modifiers of Class, Constructor, Overloading Constructor, Uses of Static and Final Keywords, Dynamic Method Dispatch, Abstract Classes.

Unit II: Inheritance, Packages, Exception Handling, Multithreading, and Collection Classes

Inheritance and Interfaces, Implementing and extending Interface, Packages: Importing Packages and Classes, User define packages. Types of Exceptions Try-Catch Block, Multiple Catch, Nested Try, Throw, Throws, and Finally, Built-in and User-Defined Exceptions. Multithreading: Multithreading Concept, Thread Life Cycle, Thread Class and Runnable Interface, Type Priorities, Synchronization. Collections: Introduction to the Collection Framework, Interfaces: Implementation.

Unit III: Applet, AWT, and Swings

Applet: Applet Class, Architecture, Life Cycle, Display Methods, HTML Applet Tag, Passing Parameter to Applet. AWT: Working with Windows (Frames and Panel), Controls (Label, TextField, Button, Checkbox, Scrollbar, List, Choice) Layout Managers: Border Layout, Flow Layout, Grid Layout. Menus. Swings: Introduction and Event Handling.

Unit IV: Servlet and JSP

Servlet: Introduction to Servlet (Http Servlet), Life Cycle of Servlet, Handing Get and Post Request (Http), Data Handling Using Servlet, Creating and Cookies, Session Tracking Using Http Servlet. JSP: Getting Familiar with JSP Server, First JSP, Adding Dynamic Contents via Expressions Scriptlet, Mixing Scriptlet, and HTML, Directives, Declaration, Tags, and Session.

<u>Books:</u>

1) Dr. S. B. Kishor, Dr. Rajani Singh “Programming in JAVA”, Das Ganu Publication, ISBN-978-93-84336-49-3
2) E. Balaguruswamy, “Programming with Java - A Primer”, Tata McGraw-Hill, New Delhi, Third Edition, ISBN0-07-061713-9
3) Marty Hall, Larry Brown “Core Servlets and JavaServer Pages: Volume 1: Core Technologies”, 2nd Edition, August 2003, Pearson Publication, ISBN: 0130092290

<u>References:</u>

1) Steven Holzner, “Java 2 Programming Black Books”, ISBN-13: 978-1588800978

M.Sc. (Computer Science) - I (SEMESTER – I)
Paper Code: 01MSCCS04.1
PAPER – IV- Elective – I
CLOUD COMPUTING

Credit: 3 **Max. Marks: 80**

Program Learning Outcome:

1. To understand the concept of cloud computing.
2. To appreciate the evolution of cloud from the existing technologies.
3. To have knowledge on the various issues in cloud computing.
4. To be familiar with the lead players in cloud.
5. To appreciate the emergence of cloud as the next generation computing paradigm.

Unit I: Fundamentals of Cloud Computing

Overview of Cloud Computing, Evolution of Cloud Computing, Types of Clouds, Key Characteristics of Cloud Computing, Intranets, and Clouds. Benefits and challenges of cloud computing, Usage scenarios, and publications, Regulatory issues, major players in cloud computing.

Unit II: 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: 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: 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.

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. John W. Rittinghouse & James F. Ransome, "Cloud Computing: Implementation, Management and Security", CRC Press, 1st Edition, 2009, ISBN 978-1439806807.
3. Antony T Velte, Toby J. Velte, Robert Elsenpeter, Cloud Computing: "A Practical Approach", McGraw Hill, 2009. ISBN 978-0-07-068351-8
4. 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
5. James E Smith, Ravi Nair, "Virtual Machines", Morgan Kaufmann Publishers, 2006. ISBN 9788131203293

References:

1. George Reese, "Cloud Application Architecture", O'Reilly and Associates.
2. Haley Beard, "Cloud Computing Best Practices for Managing and Measuring Processes for On-demand Computing", Applications and Data Centers in the Cloud with SLAs, Emereo Pty Limited, July 2008.

M.Sc. (Computer Science) - I (SEMESTER – I)

Paper Code: 01MSCCS04.2

PAPER – IV - Elective – II

DISCRETE MATHEMATICS

Credit: 3 **Max. Marks: 80**

Program Learning Outcome:

1. Introduce concepts of mathematical logic for analyzing propositions and proving theorems.
2. Use sets for solving applied problems, and use the properties of set operations algebraically and Work with relations and investigate their properties.
3. Investigate functions as relations and their properties and will understand the basic concepts of graphs, digraphs, and trees.

Unit I: Fundamental of Sets and Mathematical Logic Fundamental

Sets and Subsets, Operations on Sets, Sequence, Matrices.

Logic-Proposition and Logical Operation Conditional Statements, Methods of Proof, Mathematical Induction

Mathematical Logic- Statements and Notation, Equivalence of Formulas, Duality, Connectives, Normal Forms, Principle Disjunctive Normal Form, Principle Conjunctive Normal Form, Theory of Inference for the Statement Calculus, Inference Theory of the Predicate Calculus.

Unit II: Counting, Relation, and Diagraph

Counting: Permutation, Combination, Pigeonhole Principle, and Recurrence Relations. **Relational and Digraphs**- Product Sets and Partitions, Relations and Digraphs, The Matrix of a Relation, Paths in Relations and Digraphs, Properties of Relations, Equivalence Relations, Computer Representation of Relations and Digraph, Manipulation of Relations, Transitive Closure and Warshall"s Algorithm.

Unit III: Graph Theory, Lattices, and Boolean-Algebra

Graph Theory: Basic Concept of Graph Theory, Euler Paths, Circuits, Hamiltonian Paths, and Circuits. **Additional Relations and Structure**-Partially Ordered Sets, Lattices, Hasse Diagram, Principle of Duality, Distributive Lattice, Sub Lattice, Complemented Lattice, **Boolean-Algebra:** Introduction, Functions of Boolean algebra‟s, Boolean Function as Boolean Polynomials

Unit IV: Groups, Languages, and Finite State Machines

Groups: Binary Operations, Products, and Quotients of Groups, Subgroup, Abelian Group, Normal Subgroup, Semi Groups, Products and Quotients of Semi Groups.

Languages: Definition, Languages of Machine, Grammar, Derivation Trees

Finite-State Machines: Introduction to Finite State Machine, Moore Machines

Books:

1) Dr. S. B. Kishor, "Discrete Mathematics", Das Ganu Prakashan, ISBN-978-93-81660-21-8
2) Bernard Kolman, Robert C. Busby, Sharon C. Ross, "Discrete Mathematical Structures", Prentice-Hall Publication, ISBN No.–0132297515.
3) Discrete Mathematical Structures with Application to computer science, Publication Tata McGraw –Hill, ISBN-0-07-065142-6,

References:

1) Goodacre, "Discrete Mathematics with Graph Theory", PHI Publication, ISBN -0136020798.
2) J.K. Sharma, "Discrete Mathematics", McMillan Publication, ISBN -9780230322301.
3) Rajendra Akerkar, "Discrete Mathematics", Publication Pearson

M.Sc. (Computer Science) - I (SEMESTER – I)
Paper Code: 01MSCCS04.3
PAPER – IV - Elective - III
THEORY OF COMPUTATION & SYSTEM PROGRAMMING

Credit: 3 **Max. Marks: 80**

Program Learning Outcome:

1. Defines machine models formally.
2. Synthesizes finite automata with specific properties.
3. To introduce students to the fundamental model of the processing of high-level language programs for execution on computer systems.
4. To understand and implement Assembler, Loader, Linkers, Macros & Compilers.

Unit I: Finite Automation and Regular Expression

Finite Automation and Regular Expression: Finite State Systems, Basic Definitions, Non - Deterministic Finite Automata, Finite Automata with Moves, Regular Expressions, Two Way Finite Automata, Finite Automata with Output, Application on Finite Automata.

Properties of Regular Sets: The Pumping Lemma for Regular Sets, Close Properties of Regular Sets, Decision Algorithms for Regular Sets.

Context-Free Grammars: Motivation and Introduction, Context-Free Grammar, Derivation Tree, Simplification of Context-Free Grammars, Chomsky Normal form, Greibach Normal form, The Existence of Inherently Ambiguous Context-Free Languages.

Unit II: Push Down Automata and Turing Machine

Push Down Automata: Informal Description, Definitions, Push Down Automata, and Context-Free Languages.

Turing Machine: Introduction, The Turing Machine Model, Computable Languages and Functions, Techniques Turing Machine Construction, Modification of Turing Machines, Church"s Hypothesis, Turing Machine As Enumerators, Restricted Turing Machine Equivalent to The Basic Model. **Chomsky:** Regular Grammar, Unrestricted Grammar, Context-Sensitive Languages, Relation between Classes of Languages.

Unit III: Introduction to Device Drivers

Introduction to Device Drivers: Role of Device Drivers, Splitting The Kernel, Classes of Devices and Modules, Security Issues, Version Numbering, Building and Running Modules Kernel Modules Vs. Applications, Compiling, and Loading, Kernel Symbol Table, Preliminaries, Interaction and Shutdown, Module Parameters, Doing It in User Space.

Unit IV: Assembly and Machine Languages

Assembly and Machine Languages: CPU Architecture of 8086 Family, Function, Procedure and General Purpose Registers, Memory Segmentation and Address Computation, Addressing Modes, Instructions Set and formats. Different Types of Instruction, Processing of Binary, ASCII, and BCD Data, Stacks, Calls, Returns, Near and Far Procedures. Interrupts and Their Routines, Definitions and Recursive Macros, Assemblers, Overview of Compilation Process. **Loaders and Linkers:** Loading Schemes, Linking, Relocation, and Program Relocation.

Books:

1. Donovan J.J, "Systems Programming", New York, TMH, ISBN-0-07-460482-1
2. Dhamdhere D.M., "System Programming", TMH, ISBN-0-7-133311-8
3. John E. Hoperott and Jeffery D.Ullman, "Introduction to Automata Theory, Languages,and Computation".

References:

1. Adam Hoover, "System Programming with C and UNIX", Pearson, ISBN-0136076602.

M.Sc. (Computer Science) - I (SEMESTER – I)

Paper Code: 01MSCCS04.4

PAPER – IV - Elective - IV

NETWORK SECURITY

Credit: 3 **Max. Marks: 80**

Program Learning Outcome:

1. understanding of Network Security Concepts
2. Identification of Security Threats and Attacks
3. Implementation of Network Security Measures
4. Knowledge of Cryptographic Techniques
5. Network Vulnerability Assessment and Penetration Testing

Unit 1: Introduction to Network Security

Understanding Network Security: Definitions, goals, and importance of network security. Security Threats and Attacks: Types of threats (e.g., malware, hacking, social engineering) and common attack vectors. Security Principles: CIA Triad (Confidentiality, Integrity, Availability) and other security principles.

Unit 2: Network Fundamentals and Protocols

Networking Basics: OSI and TCP/IP models, understanding network layers. Network Devices and Technologies: Routers, switches, hubs, and their roles in network security. Common Network Protocols: TCP, UDP, IP, HTTP, DNS, FTP, and their security implications.

Unit 3: Cryptography and Encryption

Introduction to Cryptography: Symmetric vs. asymmetric encryption, hashing, digital signatures. Cryptographic Algorithms: DES, AES, RSA, ECC, and their strengths and weaknesses. Key Management: Key generation, distribution, storage, and exchange.

Unit 4: Network Fundamentals and Threats

Introduction to Network Security: Overview of network security concepts, objectives, and principles. Network Protocols and Architecture: Understanding the OSI and TCP/IP models, common protocols, and their vulnerabilities. Network Attacks: Types of network attacks, such as DDoS, Man-in-the-Middle (MitM), Spoofing, etc. Network Security Devices: Firewalls, Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), and their configurations.

Books:

1. Network Security Essentials: Applications and Standards" by William Stallings, Pearson Publication, ISBN-13: 978-0136108054
2. "Network Security: Private Communication in a Public World" by Charlie Kaufman, Radia Perlman, and Mike Speciner, Prentice Hall Publication, ISBN-13: 978-0130460196
3. "Firewalls and Internet Security: Repelling the Wily Hacker" by William R. Cheswick, Steven M. Bellovin, and Aviel D. Rubin, Addison-Wesley Professional Publication, ISBN-13: 978-0201634662

References:

1. "The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" by Dafydd Stuttard and Marcus Pinto, Wiley Publication, ISBN-13: 978-1118026472
2. "CISSP (ISC)2 Certified Information Systems Security Professional Official Study Guide" by Mike Chapple, James Michael Stewart, and Darril Gibson, Sybex Publication, ISBN-13: 978-1119475934

M.Sc. (Computer Science) - I (SEMESTER – I)
Paper Code: 01MSCCS04.5
PAPER – IV - Elective - V
COMPUTER GRAPHICS

Credit: 3 **Max. Marks: 80**

Program Learning Outcome:

1. Understands the core concepts and mathematical foundations of computer graphics.
2. Students will know the fundamental computer graphics algorithms and data structures.
3. Students will be able to understand different modeling approaches and methods.
4. Understand the detailed knowledge about basic shading and texture mapping techniques.

Unit I: Digital Image Fundamentals

Digital Image Fundamentals: What is Digital Image Processing? fundamental Steps in Digital Image Processing, Components of an Image processing system, and elements of Visual Perception. Image Sensing and Acquisition, Image Sampling and Quantization, Some Basic Relationships between Pixels, Linear and Nonlinear Operations.

Unit II: Image Transforms

Image Transforms Two-dimensional orthogonal & unitary transforms, properties of unitary transforms, two-dimensional discrete Fourier transform. Discrete cosine transforms, sine transform, Hadamard, transform, Haar transform, Slant transform, KL transform

Unit III: Image Enhancement

Image Enhancement: Image Enhancement in the Spatial domain, Some Basic Gray Level Trans -formations, Histogram Processing, Enhancement Using Arithmetic/Logic Operations. Basics of Spatial Filtering Image enhancement in the Frequency Domain filters, Smoothing Frequency Domain filters, Sharpening, Frequency Domain filters, homomorphic filtering

Unit IV: Model of Image Degradation

Model of image degradation/restoration process, noise models, Restoration in the Presence of Noise, Only-Spatial Filtering Periodic Noise Reduction by Frequency Domain Filtering, Linear Position-Invariant Degradations, inverse filtering, minimum mean square error (Weiner) Filtering, Color Fundamentals. Color Models, Pseudo color Image Processing., processing basics of full-color image processing

Books:

1. Donald Hearn, M. Pauline Baker, “Computer Graphics”, Prentice Hall Publication, Year- 1994, “2nd Edition”.

2. Steven Harrington, “Computer Graphics A Programming Approach”, McGraw-Hill International Edition, Year- 1987, ISBN 0-07-026753-7

References:

1. Apurva A. Desai, “Computer Graphics”, Prentice Hall Publication, Year-2006, ISBN No. 978-81-203-3524-0

2. V. K. Pachghare, “Comprehensive Computer Graphics: Including C++”, Laxmi Publications

M.Sc. (Computer Science) - I (SEMESTER – I)
Paper Code: 01MSCCS05
PAPER-V
RESEARCH METHODOLOGY & PUBLICATION ETHICS

Credit: 3 **Max. Marks: 80**

Program Learning Outcome:

1. Demonstrate the ability to choose methods appropriate to research aims and objectives
2. Understand the limitations of particular research methods
3. Develop skills in qualitative and quantitative data analysis and presentation
4. Develop advanced critical thinking skills
5. Demonstrate enhanced writing skills

Unit I: Introduction

Definition of Research, Meaning, Types, Objectives of Research, Research Process, Significance and Scope of Research, Criteria of Good Research

RESEARCH PROBLEM HYPOTHESIS - Identification of Research Problem, Research Question, Formulation of Research Proposal, Hypothesis – Null and Alternative Hypothesis, Testing of Hypothesis

Unit II: Research Design Plan

Meaning of Research Design, Concept of Research Design, Characteristics of good research design, Use of advanced technology in Research Design

Sampling Data Collection & Analysis - Meaning and definition of Sampling, Sampling Design, and Probability; Non-probability Sampling. Collection of Primary; Secondary Data, Methods of Data collection - Observation, Questionnaire, Survey, interviews, brainstorming, scales Tests, etc., Validity and Reliability, Analysis of quantitative data based on various tools and its presentation with tables and graphs, Statistical Tools and techniques of data analysis.

Unit III: Report Writing Evaluation

Principles of Report writing and guidelines according to style manuals, Contents of Report Writing; Presentation of preliminary part, main body and reference section of the Report, writing research abstract, Writing a research paper, Evaluation of Research Report

Unit IV: Scientific Conduct & Publication Ethics

Introduction to philosophy: definition, nature and scope, concept, branches. Ethics: definition, moral philosophy

Scientific Conduct: - Ethics with respect to science and research, Intellectual honesty and research integrity, scientific misconducts: falsification, fabrication, and plagiarism

Publication Ethics: Publication ethics: definition, introduction, and importance. Best practices/standards-setting initiatives and guidelines: COPE, WAME, etc., Conflicts of interest, Predatory publishers and journals

Books:

1) Dr. S. Sachdeva, "Research Methodology", Himalaya Publication
2) The Student's Guide To Research Ethics, Paul Oliver, McGrew Hill
3) Dr. S. B. Kishor, Dr. Ajay S. Kushwaha, Dr. Gitanjali J, "Research and Publication Ethics", published by DAS Ganu, Prakashan, ISBN 978-93-84336-71-4

References:

1) C. R. Kothari, Research Methodology Methods and Techniques, New Age Publication
2) Dr. S. B. Kishor, "Research Methodology", published by DAS Ganu, Prakashan
3) Ethical Choice in Research, Harries Cooper, Aditya Books, ISBN 9781433821684

M.Sc. (Computer Science) - I (SEMESTER – I)

Paper Code: 01MSCCS06

Lab – I

(based on 01MSCCS01 & 01MSCCS02)

Credit: 2 **Max. Marks: 100**

Practical on Scripting Languages and Information Retrieval (01MSCCS01)

1. Write a program using HTML5 with the following CSS specifications-
   i. The background color of the college name should be green.
   ii. The text color of the college name should be Red.
   iii. The heading should be large with the font "comic sans ms".
   iv. The description of the college should be displayed in blue color in a paragraph.
2. Write HTML5 code with CSS as follows-
   i. Create a form to accept the name, age, and email address of the user.
   ii. Create a submit button to send the data.
   iii. The heading of the form should have a background color red and a different font style.
3. Write HTML5 code with CSS as follows-
   b. Create a user registration form having the fields – name, gender, address, and fields of interest.
   c. Give an appropriate heading with a larger size and center alignment.
   d. Set the background color of the body to light grey and the color of the text to blue.
   e. Create submit button to submit form data and a reset button for clearing entries in the form.
4. Write a JavaScript program to associate the Method with the Object.
5. Write a JavaScript program to illustrate the different properties of Document Objects.
6. Write a JavaScript Program to Create a Dynamic Web Page.
7. Write a JavaScript Program to Generate User ID at Runtime.
8. Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 and outputs HTML text that displays the resulting values in an HTML table format.
9. Write a JavaScript code that displays text "TEXT-GROWING" with increasing font size in the interval of 100ms in RED COLOR, when the font size reaches 50pt it displays "TEXT SHRINKING" in BLUE color. Then the font size decreases to 5pt.
10. Develop and demonstrate an HTML5 file that includes a JavaScript script that uses functions for the following problems: a) Parameter: A string b) Output: The position in the string of the left-most vowel c) Parameter: A number d) Output: The number with its digits in the reverse order.
11. Write a VB Script program to find the factorial of a given number.
12. Write a VB Script to demonstrate the program to find the largest among two numbers.
13. Write VB Script to create dialog boxes.
14. Demonstration of the array in VB Script.

Practical: Python Programming (01MSCCS02)

1) Write a Python Program to add two Numbers.
2) Write a Python Program to find the Square root.
3) Write a Python Program to generate Random Numbers.
4) Write a Python Program to check if a number is positive, negative, or zero.
5) Write a Python Program to check number is odd or even
6) Write a Python Program to find the sum of natural numbers
7) Programs on Python List, Dictionary, and Object-Oriented Concepts
8) Write a Python program that can compute the factorial of a given number.
9) Write a Python program to find Armstrong numbers between 100 to 999.
10) Write a Python program to find the sum of prime numbers between 1 to 100.
11) Write a Python program to find the reverse of the given number.
12) Write a Python program to check if a given positive integer is a power of three.
13) Write a Python program to check if a number is a perfect square.
14) **a)** Write a function nearly equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation.

    b) Write a function to compute the gcd and lcm of two numbers. Each function shouldn't exceed one line.

M.Sc. (Computer Science) - I (SEMESTER – I)

Paper Code: 01MSCCS07

Lab- II

(based on 01MSCCS03 & Selected Elective)

Credit: 2 **Max. Marks: 100**

Practical List for ADVANCED JAVA (01MSCCS03)

1. Write a program to accept two numbers and display the result using the command line argument.
2. Write a program for sorting a list of a number using Array.
3. Write a java program to print the following output.

```
A
A  B
A  B  C
A  B  C  D
A  B  C  D  E
```

4. Write a java program to no. of evens and no. odd numbers in an array of size 10. Also, calculate the sum of evens and the sum of odds.
5. Write a java program to find the sum of prime numbers ranging from 1 to 100.
6. Write a program to calculate multiplication and division using the static method.
7. Write a program of Constructor Overloading to calculate the area of the Room.

i. Default constructor

ii. Constructor with one argument.

iii. Constructor with three arguments.

8. Write a program to demonstrate Single Inheritance.
9. Write a program to calculate the area of a rectangle and circle using Interface.
10. Write a java program to demonstrate the try…catch mechanism.
11. Write a java program to show the use of throw, throws, and finally keywords.
12. Write a program that throws IO Exception. (Accept student Name and age from keyboard and display.
13. Write a program to demonstrate user-defined exception (use division of two no's & throw user define exception if the result is smaller than0.01)
14. Write a java program to demonstrate Threads using Thread class and also with Runnable interface.
15. Write a program to demonstrate the Linked List class.
16. Write a program to demonstrate the List class.
17. Write a program to demonstrate ListIterator Interface.
18. Write an AWT program to accept the user's details.
19. Write an Applet program to create a Login page with having Username and Password.
20. Design a user interface using an applet to accept two values and calculate the sum of these numbers.
21. Write a program to create a frame using the JFrame class
22. Write a program to demonstrate Generic Servlet Program.

Practical: CLOUD COMPUTING (01MSCCS04.1)

1. **Implementing Infrastructure as a Service (IaaS):**
   - Choose a cloud platform (e.g., AWS, Azure, Google Cloud) that supports IaaS.
   - Create virtual machines (VMs) and configure networking, storage, and security settings.
   - Deploy an application or service on the VMs and manage them using the cloud platform's management tools.
2. **Deploying Platform as a Service (PaaS) applications:**
   - Select a PaaS provider (e.g., Heroku, Google App Engine, Azure App Service).
   - Develop a web application using a supported programming language or framework.
   - Configure the PaaS environment and deploy the application.
   - Monitor and scale the application as needed using the PaaS provider's tools.
3. **Building and deploying containers with Docker:**
   - Install Docker on your local machine or a cloud VM.
   - Create a Dockerfile to define the container's configuration and dependencies.
   - Build a container image and push it to a container registry (e.g., Docker Hub, AWS ECR).
   - Deploy the container on a cloud platform or a container orchestration service (e.g., Kubernetes).
4. **Exploring Serverless Computing:**
   - Choose a serverless computing platform (e.g., AWS Lambda, Azure Functions).
   - Write a simple function in a supported programming language.
   - Configure triggers and event sources for the function.
   - Deploy the function to the serverless platform and test its execution.
5. **Implementing Software as a Service (SaaS):**
   - Choose a SaaS provider (e.g., Salesforce, Google Workspace, Microsoft 365).
   - Explore the available SaaS applications and their features.
   - Customize and configure the SaaS application to meet specific requirements.
   - Integrate the SaaS application with other cloud services or on-premises systems.
6. **Designing and implementing a hybrid cloud architecture:**
   - Set up a private cloud infrastructure using virtualization technologies (e.g., VMware, OpenStack).
   - Connect the private cloud to a public cloud provider (e.g., AWS, Azure) using VPN or direct connect services.
   - Design and implement workload migration strategies between the private and public clouds.
   - Manage and monitor the hybrid cloud environment using appropriate tools and services.
7. **Exploring cloud-native development and microservices architecture:**
   - Learn about cloud-native development principles and microservices architecture.
   - Choose a cloud platform that supports container orchestration (e.g., Kubernetes).
   - Develop and deploy microservices-based applications using containerization and orchestration tools.
   - Monitor and scale the microservices using the cloud platform's management tools.

Practical: DISCRETE MATHEMATICS (01MSCCS04.2)

1. **Implementing basic mathematical operations:**
    - Write a program that performs arithmetic operations such as addition, subtraction, multiplication, and division on integers or floating-point numbers.
    - Implement algorithms for finding the greatest common divisor (GCD) and least common multiple (LCM) of two numbers.
2. **Generating prime numbers:**
    - Write a program that generates a list of prime numbers up to a given limit using algorithms like the Sieve of Eratosthenes or trial division.
    - Implement functions to check if a given number is prime or not.
3. **Implementing combinatorial algorithms:**
    - Write programs to calculate permutations and combinations of a given set of elements.
    - Implement algorithms for generating combinations or permutations recursively or iteratively.
4. **Implementing graph algorithms:**
    - Write programs to represent and manipulate graphs using adjacency matrix or adjacency list representations.
    - Implement algorithms for traversing graphs, such as depth-first search (DFS) and breadth-first search (BFS).
    - Implement algorithms for finding shortest paths, such as Dijkstra's algorithm or Bellman-Ford algorithm.
5. **Implementing sorting and searching algorithms:**
    - Write programs to implement various sorting algorithms like bubble sort, insertion sort, selection sort, merge sort, quicksort, or heapsort.
    - Implement searching algorithms like linear search, binary search, or interpolation search.
6. **Implementing algorithms for number theory:**
    - Write programs to find the factors of a given number.
    - Implement algorithms for testing if a number is prime, checking if two numbers are coprime, or finding modular inverses using the extended Euclidean algorithm.
7. **Implementing cryptography algorithms:**
    - Write programs to implement encryption and decryption algorithms like Caesar cipher, Vigenère cipher, or RSA encryption.
    - Implement algorithms for generating and verifying digital signatures using techniques like the RSA algorithm or the ElGamal algorithm.
8. **Implementing algorithms for counting and probability:**
    - Write programs to calculate factorials, permutations, and combinations efficiently using techniques like memoization or dynamic programming.
    - Implement algorithms for simulating random events and calculating probabilities, such as rolling dice or drawing cards from a deck.

Practical: TOC & SP (01MSCCS04.3)

1. **Implementing Finite Automata:**
    - Write a program that simulates a deterministic finite automaton (DFA) or a non-deterministic finite automaton (NFA).
    - Implement algorithms to determine whether a given input string is accepted or rejected by the automaton.
    - Extend the program to convert an NFA to a DFA using algorithms like the subset construction method.
2. **Implementing Regular Expressions:**
    - Write a program that parses and evaluates regular expressions.
    - Implement algorithms to match a given input string against a regular expression using techniques like Thompson's construction or the NFA simulation.
3. **Implementing Context-Free Grammars:**
    - Write a program that parses and evaluates context-free grammars (CFGs).
    - Implement algorithms like the CYK algorithm or the Earley parser to determine whether a given input string can be derived from a CFG.
4. **Implementing Pushdown Automata:**
    - Write a program that simulates a pushdown automaton (PDA).
    - Implement algorithms to determine whether a given input string is accepted or rejected by the PDA.
    - Extend the program to convert a CFG to an equivalent PDA using algorithms like the Chomsky normal form or the Greibach normal form.
5. **Implementing Turing Machines:**
    - Write a program that simulates a Turing machine (TM).
    - Implement algorithms to determine whether a given input string is accepted or rejected by the TM.
    - Extend the program to simulate more complex TM variations like multi-tape TMs or non-deterministic TMs.
6. **Implementing Computability and Complexity:**
    - Write programs to solve classic computability problems like the Halting problem or the Post Correspondence Problem.
    - Implement algorithms to analyze the time and space complexity of programs or algorithms using techniques like Big O notation or time complexity analysis.
7. **Implementing Formal Language Operations:**
    - Write programs to perform operations on formal languages, such as union, concatenation, or Kleene closure.
    - Implement algorithms to convert regular expressions to NFAs or DFAs, or vice versa.

Practical: NETWORK SECURITY (01MSCCS04.4)

1. **Configuring Firewall Rules:**
   - Set up a firewall (e.g., iptables, Windows Firewall) on a local machine or a network device.
   - Configure firewall rules to allow or block specific network traffic based on protocols, ports, or IP addresses.
   - Test the effectiveness of the firewall rules by attempting to access blocked services or ports.
2. **Implementing Virtual Private Networks (VPNs):**
   - Set up a VPN server (e.g., OpenVPN, IPSec) on a local machine or a cloud-based virtual machine.
   - Configure VPN clients to connect securely to the VPN server.
   - Test the VPN connection to ensure data confidentiality and integrity.
3. **Conducting Network Vulnerability Assessments:**
   - Use vulnerability scanning tools (e.g., Nessus, OpenVAS) to scan a network for potential vulnerabilities.
   - Analyze the scan results and identify security weaknesses or misconfigurations.
   - Develop a plan to remediate the identified vulnerabilities and implement necessary security patches or configuration changes.
4. **Implementing Intrusion Detection and Prevention Systems (IDS/IPS):**
   - Set up an IDS/IPS system (e.g., Snort, Suricata) on a local machine or a network device.
   - Configure the IDS/IPS rules to detect and prevent network attacks or suspicious activities.
   - Test the IDS/IPS system by simulating various attack scenarios and analyzing the generated alerts.
5. **Configuring Secure Remote Access:**
   - Set up a secure remote access solution (e.g., SSH, VPN) for accessing network resources remotely.
   - Configure secure authentication methods (e.g., public-key authentication, two-factor authentication) for remote access.
   - Test the remote access solution to ensure secure and authenticated access to network resources.
6. **Implementing Network Encryption:**
   - Set up Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates on web servers.
   - Configure HTTPS for secure communication between clients and servers.
   - Test the SSL/TLS implementation to ensure secure and encrypted data transmission.
7. **Conducting Network Traffic Analysis:**
   - Use network traffic analysis tools (e.g., Wireshark, tcpdump) to capture and analyze network packets.
   - Analyze network traffic patterns and identify potential security threats or anomalies.
   - Develop strategies to mitigate network security risks based on the analysis results.
8. **Implementing Network Access Control:**
   - Set up a network access control solution (e.g., 802.1X, NAC appliances) to enforce network access policies.
   - Configure authentication mechanisms and access rules to control network access for devices and users.
   - Test the network access control solution by connecting devices and users to the network and verifying the enforcement of access policies.

Practical: COMPUTER GRAPHICS (01MSCCS04.5)

1. Explain MATLAB’s workspace, Save, Load, and Clear All functions.

2. Write a program in MATLAB to enter two arrays of any dimension and multiply them after checking the condition.

3. Write a program in MATLAB to enter a date so that program will display the day coming on that date. If the first of the month is on Wednesday.

4. Write a program in MATLAB to generate the sine wave

5. Write a program in MATLAB to demonstrate the subplots.

6. Write a program in MATLAB to demonstrate the 3-D plot

7. Write a program in MATLAB to create a tree using Huffman’s Encoding.

8. Write a program in MATLAB to demonstrate the multidimensional array.

9. Write a program in MATLAB to demonstrate Data Brushing.

10. Write a program in MATLAB to enter– the ordinate of a point and find its position with respect to the coordinate system.

11. Write a program in MATLAB to enter two arrays of any dimension and add them after checking the condition.

12. Write a program in MATLAB to enter the Cartesian coordinate of a point and convert them into equivalent polar coordinates.

13. Write a program in MATLAB to find the transpose of a matrix of 4 * 5

14. Enter the following complex number, z = 2-j3 then

(a) Get the real and the imaginary parts of z

(b) Get the magnitude and the phase angle of z

(c) If y = 3+j5, calculate the following: y+z, y-z, y×z

15. What is meant by relational and logical operators, write a program to demonstrate each one.

M.Sc. (Computer Science) - I (SEMESTER – I)
Paper Code: 01MSCCS08
SEMINAR

Credit: 1 **Max. Marks: 50**

Program Learning Outcome:

1. To analyze a current topic of professional interest and present it before the audience.
2. To familiar with basic technical writing concepts and terms, such as audience analysis, jargon, format, visuals, and presentation and acquired the basic skills to for performing literature survey and paper presentation.
3. To improve skills to read, understand, and interpret material on technology.
4. To improve communication and writing skills and prepare the report.

The seminar must be based on some current trends related to IT/Computer Science/Computer Application. A student must present the PowerPoint presentation along with Seminar Report. Students are requested to follow the following guidelines while choosing & preparing their seminars.

Guidelines to Seminar

1. The name of the seminar topic must be the latest on the current trends and should not be repeated. A student must submit the topic at the beginning of the semester.
2. The Seminar topic is to be approved by the departmental allocated guide.
3. Students are allowed to use graphics/animation/audio-video aids for their presentations.
4. Seminar work will be evaluated by an internal examiner. Refer Appendix B
5. Students are requested to submit their seminar reports on or before the deadline with the concern of their respective guides otherwise students will be responsible for any appropriate action.
6. The seminar report should be submitted to the department in the following format, printed in double line space using A4 size bond paper, with a left margin of 1.5” and a right margin of 1.0” with proper spiral binding to be done. Only one copy needs to be submitted.
7. Students are requested to obtain the necessary certificates and declarations to be duly enclosed in the report.

Syllabus
of

M.Sc. – I
(Computer Science)
(Sem II)

M.Sc. – I Computer Science – Semester-II

Subject	Course/ Paper Code	Paper Name	Teaching Scheme			Credits Assigned				% of Assessment			
			Theory	Prac.	Tut.	Theory	Prac.	Tut.	Total	IA	UA	Total	Min. Passing Marks
(Major) Mandatory	02MSCCS01	Web Designing using ASP.Net	4	-	-	3	-	-	3	20	80	100	40
	02MSCCS02	Data Warehouse and SQL	4	-	-	3	-	-	3	20	80	100	40
	02MSCCS03	Android Application Development	4	-	-	3	-	-	3	20	80	100	40
Elective 02MSCCS04 (Student Shall select any one from the elective group)	02MSCCS04.1	Cyber Security & IPR	4	-	-	3	-	-	3	20	80	100	40
	02MSCCS04.2	Soft Skills											
	02MSCCS04.3	Digital and Cyber Forensics											
	02MSCCS04.4	Operation Research											
	02MSCCS04.5	Data Visualization											
OJT	02MSCCS05	On Job Training	4	-	-	3	-	-	3	25	75	100	50
LAB - I	02MSCCS06	Based on 02MSCCS01 & 02MSCCS02	-	4	-	-	2	-	2	25	75	100	50
LAB - II	02MSCCS07	Based on 02MSCCS03 & 02MSCCS04	-	4	-	-	2	-	2	25	75	100	50
Ability Enhancement	02MSCCS08	Seminar	-	1	-	-	1	-	1	50	-	50	20
Total			**20**	**9**	**-**	**15**	**5**	**-**	**20**	**205**	**545**	**750**	**330**

M.Sc. (Computer Science) - I (SEMESTER – II)
Paper Code: 02MSCCS01
PAPER - I: WEB DESIGNING USING ASP.NET

Credit: 3 **Max Marks:80**

Program Learning Outcome:

1. Compare the features and differences between ASP and ASP.NET to understand the advantages of using ASP.NET for web development.
2. Upgrade existing HTML pages and ASP pages to ASP.NET, and implement data addition functionality using ADO.NET in an ASP.NET site.
3. Utilize web controls and Visual Studio.NET to create a user interface for web applications, implement data validation, navigation between forms and pages, and perform data binding and transformation on web forms using ADO.NET.
4. Create, discover, and invoke web services, test web applications using appropriate testing techniques, debug and deploy web applications, and implement security measures such as authentication and authorization using Windows Authentication and Forms Authentication.

Unit I: Web Development and Asp .Net

Comparison of Asp and Asp .Net, Features of Asp .Net, Benefits of Asp .Net, Web forms and Their Components, Overview of Web Services. Web Application Basics: Web forms Model, Web forms Internals, Asp.Net Core Server Controls, Working with Page.

Unit II: Creating Web forms Application

Upgrading HTML Pages to Asp.Net, Asp Pages to Asp.Net, Adding Data in an Asp.Net Site: ADO.Net, Paging Through Data Sources, Creating Web forms Application: Creating an Asp.Net Web Application Project, Responding to Events, Namespace Fundamentals Maintaining State Information.

Unit III: Creating a User Interface

Using Web Controls, Using Visual Studio.Net, Validation and Rich Control, Validating Data, Navigating Between forms, Navigation between Pages, Data Binding: Bind Data to The UI, Transform and Filter Data Storing and Retrieving Data with ADO.Net, Accessing Data with ADO.Net, Using Data Sets on Web forms, Processing Transactions, Catching and Correcting Errors: Using Exception Handling, Using Error Pages, Logging Exceptions.

Unit IV: Web Services

Creating Web Services, Discovering Web Services, Instantiating, and Invoking Web Services, Testing Web Applications: Creating Tests, Running Tests. Debugging, Building and Deploying Web Applications, Creating an Installation Program, Maintaining Security: Authenticating and Authorizing Users, Using Windows Authentication, Using forms Authentication.

Books:

1) Russel, “Mastering Asp.Net”, BPB Publication,
2) Matthew Macdonald, “Asp.Net the Complete References”, TMH.

References:

1) Mitchell and Atkinson, “Active Server Pages 3.0 (in 21 Days)”, Tecmedia
2) David Buser, John Kauffman, Juan T. Llibre, Brian Francis, Dave Sussman, Chris Ullman, Jon Duckett, “Beginning Active Server Pages 3.0”, Wrox Press

M.Sc. (Computer Science) - I (SEMESTER – II)
Paper Code: 02MSCCS02
PAPER - II: DATA WAREHOUSE AND SQL

Credit: 3 **Max Marks:80**

Program Learning Outcome:

1. To develop research-oriented applications of data mining and data warehousing.
2. To understand the necessity and importance of data pre-processing, data integration, data discretization.
3. To learn the concepts of OLAP technology, data mining methods, various classification and prediction methods.
4. To able to apply accuracy and error measures, methods of cluster analysis, graph mining and mining sequence patterns in biological data.

Unit I: Data Warehousing and OLAP

Introduction to Data Warehousing: Characteristics of a Data Warehouse, Data Warehouse Architectural Strategies, Design Considerations, Data Content, building a Data Warehouse, Metadata, Tools for Data Warehousing, Performance Considerations, Crucial Decisions in Designing a Data Warehouse, Different Case Studies. Various Technological Considerations: OLTP and OLAP Systems, Data Modeling, Managed Query Environment (MQE).

Unit II: Data Mart and Data Mining Tools

Data Mart: Data Mart, Type of Data Mart, Loading a Data Mart, Metadata for a Data Mart, Data Model for a Data Mart, Software Component for a Data Mart, Tables in Data Mart, Security in Data Mart.

Data Mining and Tools: Introduction, From Data Warehouse to Data Mining, Steps of Data Mining, Data Mining Algorithm, Database Segmentation, Predictive Modeling, Link Analysis, Tools for Data Mining.

Unit III: SQL Server, Components, and Queries

SQL Server Architecture: SQL Server Data Storage Architecture, the Data Engine, and System Databases.

SQL Components: SQL‟s Basic Object, Data Types, Transact-SQL Functions, Scalar Operators, and Null Values. Data Definition Language, Data Manipulation Language, Queries, Modification of Table Contents, Stored Procedures and User-Defined Functions, Views.

Unit IV: Data Integrity, User Security, and Concurrency Control

Managing Data Integrity: Data Integrity Controls, Working with Constraints, DML Triggers. **Managing User Security:** Security Architecture, Implementing SQL Server Principles and Authentication, Implementing Permission in SQL Server.

Backup and Concurrency Control: Transaction Architecture, Locking, Backup Types, Performing Database Restore, Replication, Using Transaction Logs, Using Triggers, and Replication Methods.

Books:

1) C.S.R. Prabhu, "Data Warehousing", PHI, 3rd Ed., ISBN-978-81-203-3421-2.
2) Dusan Petkovic, "Microsoft SQL Server 2008, Beginner S Guide", TMH Pub. , ISBN-0071540383.
3) Michel Lee, Gentry Bieker, "Mastering SQL Server 2008", Sybex Pub, ISBN-047028904x.

References:

1) Jiawei Han, Micheline Kamber, Jian Pei, "Data Mining Concepts and Techniques", Elsevier Pub., ISBN-9780123814791.
2) Alex Berson, "Data Warehousing, Data Mining & OLAP", TMH, ISBN 0-07-058741-8
3) Robert Vieira, "Beginning Microsoft SQL Server 2008 Programming", Wrox Publication, ISBN-9780470257012.

M.Sc. (Computer Science) - I (SEMESTER – II)

Paper Code: 02MSCCS03

PAPER - III: ANDROID APPLICATION DEVELOPMENT

Credit: 3 **Max Marks:80**

Program Learning Outcome:

1. Install and configure Android application development tools.
2. Design and develop user Interfaces for the Android platform.
3. Save state information across important operating system events.
4. Apply Java programming concepts to Android application development.

Unit I: Introduction to Open Source & Android

Introduction to Open Source: What is Open Source, License Issues (MPL, GPL, and LGPL) and Open Source Vs Traditional Development Methodologies **Introduction to Android:** Introducing Android, History of Mobile Software Development, Open Handset Alliance, The Android Platform, Layers of Android, Android SDK, Kinds of Android Components, Building a Sample Android Application

Unit II: Android Application Design Essentials

Anatomy of Android Applications, Android Terminologies, Application Context, Activities, Services, Intents, Receiving and Broadcasting Intents, Android Manifest File and its common settings, Using Intent Filter, Permissions, Managing Application resources in a hierarchy, Working with different types of resources.

Unit III: Android User Interface Design Essentials

User Interface Screen Elements, Designing User Interfaces with Layouts, Drawing, and Working with Animation.

Unit IV: Using Common Android APIs

Using Android Data and Storage APIs, managing data using SQLite, Sharing Data between Applications with Content Providers, Using Android Networking APIs, Using Android Web APIs, and Using Android Telephony APIs.

Books:

1) Lauren Darcey and Shane Conder, “Android Wireless Application Development”, Pearson Education.
2) W. Frank Ableson, Robi Sen, Chris King, “Android in Action”, 2nd Edition, Manning Publications Co, ISBN 978-1-935182-72-6.
3) Chris Haseman, “Android Essentials”, Apress Publications, ISBN-13: 978-1-4302- 1064-1.
4) James Steele, Nelson To, “The Android Developer’s Cookbook-Building Applications with the Android SDK”, Addison-Wesley Publications, ISBN-13: 978-0-321- 74123-3.

References:

1) Lucas Jordan, Pieter Greyling, “Practical Android Projects”, Apress Pub., ISBN- 13: 978-1-4302-3243-8.
2) Reto Meier, “Professional Android 2 Application Development”, Wiley India Pvt. Ltd.
3) Mark L Murphy, “Beginning Android”, Wiley India Pvt Ltd.
4) Zigurd Mednieks, Laird Dornin, G. Blake Meike & Masumi Nakamura, “Programming Android”, O’Reilly Publications.

M.Sc. (Computer Science) - I (SEMESTER – II)

Paper Code: 02MSCCS04.1

PAPER – IV- Elective – I

CYBER SECURITY & IPR

Credit: 3 **Max. Marks: 80**

Program Learning Outcome:

1. Conduct digital investigations that conform to accepted professional standards and are based on the investigative process: identification, preservation, examination, analysis, and reporting.
2. Cite and adhere to the highest professional and ethical standards of conduct, including impartiality and the protection of personal privacy. Apply event handling on AWT and Swing components.
3. Identify and document potential security breaches of computer data that suggest violations of legal, ethical, moral, policy, and/or societal standards.
4. Learner will understand and implement the concept of IPR.

Unit I: Introduction to Internet Security

Need for Internet Security, Adopting Security Policies, Strategies for a Secure Network, Ethics of Computer Security, Security Threats and Levels, Security Plan (RFC 2196). Classes of Attacks: Stealing Passwords, Social Engineering, Bugs and Backdoors, Authentication Failures, Protocol Failures, Information Leakage, Exponential Attacks – Viruses and Worms, Denial-of-Service-Attacks, Botnets, Active Attacks.

Unit II: Computer Security

Introduction to Viruses, Trojan Horses, Worms, Bombs, Protection against Viruses, Structure of Virus. Firewalls: Introduction to Firewalls, Kinds of Firewalls: Packet Filters, Application-Level Filtering, Circuit-Level Gateways, Dynamic Packet Filters, Distributed Firewalls, Firewall Engineering: Rulesets, Proxies, Building a Firewall from Scratch, Firewall Problems and Testing Firewalls**.**

Unit III: Principles & Acquisition of IPR

Philosophical Aspects of Intellectual Property Laws, Basic Principles of Patent Law, Patent Application Procedure, Drafting of a Patent Specification, Understanding Copyright Law, Basic Principles of Trade Mark, Basic Principles of Design Rights, International Background of Intellectual Property.

Unit IV: Patents (Ownership and Enforcement of Intellectual Property) :

Patents - Objectives, Rights, Assignments, Defences in case of Infringement. Copyright - Objectives, Rights, Transfer of Copyright, work of employment Infringement, Defences for infringement Trademarks-Objectives, Rights, Protection of good will, Infringement, Passing off, Defences. Designs - Objectives, Rights, Assignments, Infringements, Defences of Design Infringement**.**

<u>Books:</u>

1) W. R. Cheswick, S. M. Bellovin, A. D. Rubin, “Firewalls and Internet Security-Repelling the Wily Hacker”, Addison-Wesley Publication, “2nd Edition”, ISBN No- 0-201- 63466-X
2) V. V. Sopale, “Managing Intellectual Property: The Strategic Importance”, 2nd Edition, PHI Publication
3) Peter Weill, Jeanne Ross, “IT Governance: How Top Performers Manage IT Decision Rights or Superior Results” 1st Edition, Harvard Business Review Press, June 2004, ISBN-101591392535, ISBN -13: 978-1591392538.

<u>References:</u>

1) Kenneth Einar Himma, “Internet Security: Hacking, Counter Hacking, and Society”, Jones & Bartlett Publication, “1st Edition”, ISBN No– 978-0-7637-3536-4
2) Man Young Rhee, “Internet Security: Cryptographic Principles, Algorithms, and Protocols”, J. Wiley Publication, ISBN No- 978-0-4708-5285-9

M.Sc. (Computer Science) - I (SEMESTER – II)
Paper Code: 02MSCCS04.2
PAPER – IV- Elective – II
SOFT SKILLS

Credit: 3 **Max. Marks: 80**

Program Learning Outcome:

1. To effectively communicate through verbal/oral communication and improve the listening skills. Write precise briefs or reports and technical documents. Soft Skills.
2. To actively participate in group discussion / meetings / interviews and prepare & deliver presentations. Become more effective individual through goal/target setting, self-motivation and practicing creative thinking.
3. To function effectively in multi-disciplinary and heterogeneous teams through the knowledge of team work, Inter-personal relationships, conflict management and leadership quality.

Unit I: Self Analysis

SWOC Analysis: Strengths, Weaknesses, Opportunities, Challenges, Master Plans. **Enthusiasm:** Characteristics & traits of an Enthusiastic person, Conducive Condition for Enthusiasm, Ways to be positive. **Etiquette:** Introduction, Need, Types, Factor Influencing Etiquette, Classification

Unit II: Attitude & Grooming

Attitude: Definition, Factors, Components, Characteristic, Types of Attitudes, TIENS Pak 8 Positive Attitude for Success. **Grooming:** Need for Personal Grooming, Importance of Proper Dressing, The ways to Groom your Personality, Introduce with Grace, ways to Introduce.

Unit III: Goal Setting, Planning & Time Management

Goal Setting: Introduction, Importance, ways to set Personal Goals in life, reasons for Goal Setting, failures, SMART Goals, Tips for Setting Goals. **Planning:** Importance of having a plan, Personal Development Plan, Master Plan, Weekly Planner, Prioritizing Work, To-Do list. **Time Management:** Essential Elements, Ways to Diagnose Poor Time Management, Tips for Effective Time Management, Techniques.

Unit IV: Communication Skills, Public Speaking, Team Work and Leadership

Communication Skills: Definition, Features, Elements, Types, 9'C of Communication, Barriers to Communication. **Public Speaking:** Importance, Characteristic of a Good Speaker, ways to overcome the fear of Public Speaking, Listening skills. **Team Work:** Keys to successful Teamwork, Benefits of Teamwork. **Leadership:** Qualities of Good Leader, Leadership Styles.

Books:

1) Dr. S. B. Kishor & Swapnil Bhagat, "Soft Skills", Das Ganu Publication
2) Prashant A. Dhanwalkar (Manusmare), "Soft Skills Development", Sai Jyoti Publication.
3) R C Sharma and Krishna Mohan, "Business Correspondence and Report Writing",TMH
4) Pravin Bhatia, S. Chand, "Professional Communication Skills".

Reference:

1) Pravin Bhatia and Meera Banerjee, "Developing Communication Skills".
2) The Communicator by Board of Editors, Orient Black Swan Publication.

M.Sc. (Computer Science) - I (SEMESTER – II)
Paper Code: 02MSCCS04.3
PAPER – IV- Elective – III
DIGITAL AND CYBER FORENSICS

Credit: 3 **Max. Marks: 80**

Program Learning Outcome:

1. Digital evidence collection and preservation: Demonstrate proficiency in collecting and preserving digital evidence from various sources, including computers, mobile devices, networks, and cloud storage, while adhering to legal and ethical standards.
2. Cybercrime investigation and analysis: Apply investigative techniques and tools to analyze digital evidence, identify cyber threats, and investigate cybercrimes such as hacking, data breaches, fraud, and digital identity theft.

Unit I: Introduction to Networking

Introduction to Networking: Networking Hardware, Networking Software, Internet, Web Phishing. **History and Future of Cybercrime**: Old and New Crimes, The Internet Spawns Crime, Worms Verses Viruses, Broadband, Wireless. **Introduction to Computer forensics**: Computer forensics Definitions, Computers' Roles in Crimes, Computer forensics Tasks, prepare for an Investigation, Collect Evidence, Preserve Evidence, Recover Evidence, Document Evidence, Challenges Associated with Making "Cybercrime" Laws, Jurisdictional Issues, Introduction to Computer Hardware

Unit II: Computer Crimes and Criminals

Computer Crimes and Criminals: Crimes, Violent Crimes: Computers Included in Terrorism, Assault Threats, Stalking, Child Pornography, Nonviolent Crimes, Trespass, Theft, Fraud, Vandalism, Address Books, Chat Logs, E-Mail, Images, Movies, Internet Browser History, Etc. Crime Timeline, Modify Access Create (Mac) Dates Associated with Files, Criminals, and Crime Fighter, understanding "Cyber, Criminals" and Their Victims, "Cyber Investigators", Protecting Yourself on The Internet, Anti-Virus and Firewall Software.

Unit III: Collecting and Preserving Digital Evidence

Collecting and Preserving Digital Evidence: Admissibility of Evidence, Must Be Legally Obtained, (Obeying The 4th Amendment and Other Federal and State Laws), Must Be Competent, Relevant, and Material, Types of Evidence, Physical, Direct, Circumstantial, Demonstrative, Documentary, Documenting Evidence with Tags and Logs, Maintaining The Chain of Custody, Processes for Collecting Computer Evidence. **Building a Cybercrime Case**: Bodies of Law, Constitutional Law, Criminal Law, Civil Law, Administrative Regulations, Levels of Law, Local Laws, State Laws, Federal Laws, International Laws, Levels of Culpability, Intent, Knowledge, Recklessness, Negligence, Level and Burden of Proof, Criminal Versus Civil Cases, Vicarious Liability, Laws Related to Computers, CFAA, DMCA, Can-Spam.

Unit IV: Computer Hardware, Software

Computer Hardware: Computer Architectures, Components, Power Supply, Motherboard, Ethernet, Com, Parallel Port, Modem Etc. **Computer Software**: Operating: Systems, Types of Operating Systems, **Working Preserving and Recovering Digital Evidence**: Disk Imaging, Creating a Message Digest or Hash Code for a Disk, Where Data Hides; Deleted and Erased Data, File Systems, Files, Modify Access Create (Mac) Dates to Establish Time Line, File Headers - Info About File Type.

Books:

1) Debra Littlejohn Shiner, “Scene of the Cybercrime”.

2) Vakul Sharma, “Handbook of Cyber Laws”, McMillan

References:

1) Michael Cross, “Scene of the Cybercrime” Second Edition.

M.Sc. (Computer Science) - I (SEMESTER – II)
Paper Code: 02MSCCS04.4
PAPER – IV- Elective – IV
OPERATION RESEARCH

Credit: 3 **Max. Marks: 80**

Program Learning Outcome:

1. Mathematical modeling: Formulate real-world problems as mathematical models using optimization techniques such as linear programming and integer programming.
2. Problem-solving: Analyze and solve optimization problems using algorithms and solution methods, including dynamic programming and simulation.
3. Decision-making: Evaluate decision-making processes considering uncertainty, risk, and sensitivity analysis, and propose improvements for better outcomes.

Unit 1: Introduction to OR

Overview of OR, Characteristics of OR problems, Formulation of OR problems, Linear programming, and simplex method, Duality in linear programming, Sensitivity analysis, Transportation problem, Assignment problem

Unit 2: Network Analysis

Introduction to network analysis, Shortest path problem, Minimal spanning tree problem, Maximum flow problem, Critical path method (CPM), Program Evaluation and Review Technique (PERT), Resource-constrained project scheduling

Unit 3: Decision Analysis

Decision-making under uncertainty, Expected value of perfect information, Decision trees, Bayesian analysis, Value of information, Utility theory

Unit 4: Heuristics and Metaheuristics

Overview of heuristics and metaheuristics, Hill climbing, Simulated annealing, Genetic algorithms, Ant colony optimization, Tabu search, Particle swarm optimization, Applications of heuristics and metaheuristics in computer science

Books:

1. Introduction to Operations Research, Frederick Hillier and Gerald Lieberman, ISBN: 978-1259917020W.
2. Operations Research: An Introduction, Taha Hamdy, ISBN: 978-0134444017.
3. Network Flows: Theory, Algorithms, and Applications, Ahuja, Magnanti, and Orlin, ISBN: 978-0136175490.

References:

1. Decision Analysis for Management Judgment, Decision Analysis for Management Judgment, ISBN: 978-1119561534.
2. Metaheuristics: From Design to Implementation, El-Ghazali Talbi, ISBN: 978-0470482916

M.Sc. (Computer Science) - I (SEMESTER – II)
Paper Code: 02MSCCS04.5
PAPER – IV- Elective – V
DATA VISUALIZATION

Credit: 3 **Max. Marks: 80**

Program Learning Outcome:

1. Create visually appealing and effective data visualizations by understanding various design principles like perception, Gestalt principles, color theory, etc.
2. Data visualization through data collection and analysis, enabling them to effectively gather and assess data and apply statistical techniques for data analysis.

Unit 1: Introduction to Data Visualization, Data Visualization Principles

Introduction to Data Visualization: Definition and importance of data visualization, Historical Overview of Data Visualization, **Data Visualization Principles:** Perception and Cognition Principles in Data Visualization, Type of Data and Appropriate Visual Representations, Gestalt principles and visual perception, Color theory and usage in Visualization, Design principles for Effective Visualization.

Unit 2: Techniques for Exploring and Understanding data

Data Collection in data visualization: Objectives, Identify Data Sources, Data Quality and Reliability, Data Storage and Organization, Data Privacy and Security, Data Sampling, Data Documentation; **Statistical Techniques for Data Visualization:** Descriptive Statistics, Correlation Analysis, Regression Analysis, Hypothesis Testing, Time Series Analysis, Clustering and Classification, Statistical Distributions; **Static Data Visualization Techniques:** Bar charts, line charts, Pie Charts, Scatter plots, Pie charts, area charts, bubble charts, Histograms, Heatmaps, Tree maps, Choropleth Maps, Box Plots, and Word Clouds, Geospatial visualization techniques; **Exploratory Data Analysis (EDA):** Data cleaning and preprocessing for visualization.

Unit 3: Interactive Data Visualization

Interactive Data Visualization: Principles of interactivity in visualization, Selection, filtering, and brushing techniques, **Specialized Data Visualization Techniques:** Network visualization and graph, analysis, Time series visualization and forecasting, Text and sentiment visualization, Hierarchical and tree-based visualization, Case studies of successful and misleading visualizations.

Unit 4: Data Visualization Tools and Technologies, Advanced Data Visualization and Applications

Overview of popular Data Visualization Tools: Tableau, Power BI, QlikView, D3.js (web-based interactive visualization); Understanding the Strengths and Weaknesses of different Data Visualization Tools, **Practical Application of Data Visualization Tools:** Hands-on Exercises (e.g., Tableau, Power BI, QlikView); Design Thinking and Iterative Design Process; Various Application of Data Visualization in different domains (e.g., healthcare, finance, social media); Collaboration and teamwork in data visualization projects, Data Visualization Project.

Books:

1. Kieran Healy, "Data Visualization - A Practical Introduction", Princeton University Press, ISBN: 9780691181622
2. Claus O. Wilke, “Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures”, First Edition, O'Reilly Publication, ISBN: 9789352138111

References:

1. Edward R. Tufte, “The Visual Display of Quantitative Information”, Graphics Pr., ISBN: 9781930824133

M.Sc. (Computer Science) - I (SEMESTER – II)
Paper Code: 02MSCCS05
ON JOB TRAINING(OJT)

Credit: 3 **Max. Marks: 100**

---

1. **When to Carry out?**

   On-the-Job Training (OJT) programs are carried out after the completion of Semester I or during a break between academic terms i.e. The break between Semester I and Semester II provides an ideal timeframe for a dedicated OJT program. Students can focus solely on their OJT without the demands of regular coursework.

   It is essential for a college or educational institution should provide a formal letter to students in the mid of semester I so that he/she can apply by specifying the requirement to carry out On-the-Job Training (OJT) in the name of a company. This letter serves as an official communication and outlines important details related to the OJT program. A student may contact the company for OJT.

   The evaluation of OJT will be held at the end of semester II, though it is a continuous process till semester IV where students will implement what they have learned in the final project i.e. Preparing the basic background about the computer project during Semester 2 for execution in Semester 4 can greatly contribute to a successful and well-structured On-the-Job Training (OJT) experience.

2. **Where to Carry-out?**

   A list of potential places where On-the-Job Training (OJT) can be carried out by the students:

   - Software Development Companies
   - Tech Startups
   - IT Consulting Firms

   On-the-Job Training (OJT) can be carried out in a company, having a GST registration, or falls under the Micro, Small, and Medium Enterprises (MSME) category.

3. A Basic guideline to help students and educators effectively plan and lay the groundwork for OJT is given below:

- **Understanding the Project Scope:**
  - In Semester 2, students should receive an overview of the OJT program and understand the scope of the project they will undertake in Semester 4. Clearly define the problem statement and objectives of the project.
- **Selecting Project Topics**:
  - Allow students to propose project topics aligned with their interests and career aspirations. This can help increase their motivation and engagement in the OJT program. A Max. of two students can work on the same topic.
- **Preparing Literature Review**:
  - Encourage students to conduct a literature review during Semester 2. This involves researching existing solutions and technologies related to the project topic. It will help them understand the state-of-the-art and identify potential areas of innovation for their project.
- **Identifying Tools and Technologies:**
  - In Semester 2, students should explore and become familiar with the tools, programming languages, and technologies relevant to their chosen project. This will give them a head start when they begin the implementation phase in Semester 4.
- **Project Proposal:**
  - By the end of Semester 2, students should prepare a detailed project proposal. It should include the problem description, research objectives, proposed solution, methodology, and a timeline for implementation in Semester 4.
- **Mentor Assignment**:
  - During Semester 2, assign mentors or supervisors to each student based on their project topic and area of expertise. The mentors can provide guidance and support throughout the OJT journey.

- **Setting Milestones**:
    - Define key milestones and deliverables for the project to be achieved in Semester 4. This helps in tracking progress and ensuring that the project stays on schedule.
- **Ethical Considerations**:
    - Discuss ethical considerations, such as data privacy, intellectual property, and code plagiarism, with students to ensure responsible and ethical conduct throughout the project.
- **Resource Availability**:
    - In Semester 2, assess the resources required for the project's successful implementation in Semester 4. This includes hardware, software, datasets, and access to any external resources or APIs.
- **Preparing a Work Plan**:
    - A student should create a detailed work plan for Semester 4. This plan should outline the tasks, activities, and deadlines for each phase of the project.
- **Communication and Documentation**:
    - Emphasize the importance of clear communication and thorough documentation throughout the OJT program. Encourage students to maintain regular progress reports and update their mentors on their work. Finally, a final report of OJT should be submitted by the students before the commencement of the semester II examination.
- **Certificate:**
    - Students who undergo On-the-Job Training (OJT) should submit the certificate of completion to their College/Department. The certificate serves as proof that the students have completed the OJT program in the company and fulfilled the requirements set by the college or University.

4. **Evaluation**

During the evaluation, the examiner must check the following points and assign the marks,

- Clear and measurable learning objectives for the evaluation.
- Utilize a combination of quantitative and qualitative assessment methods.
- Assess project outcomes, deliverables, and the application of learned skills.
- Evaluate technical proficiency, problem-solving abilities, and creativity.
- Consider students' communication, collaboration, and teamwork skills.

An evaluator may Provide constructive feedback to students for improvement and growth. An external examiner will follow the following table while allocating the Marks out of 75. The internal examiner will allocate the 25 marks in proportion.

Sr No	Points	Marks	Remarks
1	Project Outcomes and Deliverables	25	Evaluate students' technical skills and proficiency in applying their theoretical knowledge to real-world tasks during the OJT.
2	Problem-Solving and Creativity	10	Measure their ability to solve complex problems, think critically, and demonstrate creativity in finding innovative solutions.
3	Communication and Collaboration	10	Evaluate students' communication skills, how well they interact within a group, and their ability to work collaboratively.
4	Project Documentation	15	Assess the completeness and quality of project documentation, such as design documents, technical specifications, and user guides.
5	Presentation	15	Consider the effective use of visual aids, such as slides, charts, and multimedia, to enhance the presentation.
Total		**75**	

Note: A project must be guided by regular faculty having Ph.D. If the department does not have any faculty with Ph.D. degrees, then a synopsis of such students must be approved by the college/department from the subject faculty of adjacent colleges having Ph.D. Degree faculty and same should work as an internal practical examiners organized by University.

M.Sc. (Computer Science) - I (SEMESTER – II)
Paper Code: 02MSCCS06
Lab - I
Based on 02MSCCS01 & 02MSCCS02)

Credit: 2 **Max. Marks: 100**

Practical on Web Designing Using ASP.NET (02MSCCS01)

- Design simple web applications using ASP.NET.
- Design web applications with different validations.
- Design online database application.
- Design data report application.
- Design web application for uploading files on the web.
- Design AJAX application.
- Design localized web applications.
- Design WPF browser application.
- Authentication and authorization in asp.
- Deployment and publishing of websites.

Practical in DATA WAREHOUSE AND SQL (02MSCCS02)

A. Create a table DONAR with the following fields (Dno, Dname, City, Age, Sex, BG, Quantity, date).

B. Insert the following records into the table DONAR.

Dno	Dname	City	Age	Se x	BG	Quantity	Date
101	RAJESH RAO	CHANDRAPUR	28	M	O+ve	100	25-AUG-11
102	ANAND SHARMA	NAGPUR	20	M	O+ve	200	26-AUG-11
103	VISHAL DESHPANDE	HYDERABAD	23	M	O-ve	250	26-AUG-11
104	SHRUTI RAKHUNDE	CHANDRAPUR	22	F	A+ve	100	27-AUG-11
105	ANUSHREE DHAKATE	-	22	F	A-ve	200	26-AUG-11
106	VIJETA DHAKATE	BALLARPUR	22	F	O+ve	100	25-AUG-11
107	AAMIR TAJA	CHANDRAPUR	21	M	O+ve	250	27-AUG-11
108	AMIR KHAN	DURGAPUR	25	M	O+ve	100	25-AUG-11

C. Perform the following queries on the above table.

1) Find all donors whose name starts between alphabets “A” to “S”.
2) Find all donors who belong to the city of CHANDRAPUR.
3) Find all donors who do not belong to CHANDRAPUR city.
4) Find all donors who belong to either CHANDRAPUR or NAGPUR city.
5) Find all donors whose city value contains NULL.
6) Arrange all donors in the sorted order whose age is between 18 and 22.
7) Find all male donors.
8) Find all male donors having O+Ve blood group.
9) Find all donors who donated their blood between 25-AUG-10 and 26-AUG-11.
10) Find all donors who donated more than 100 ml of blood.
11) Find all-female donors who belong to the city CHANDRAPUR having blood group “O+Ve” in the sorted order of the city.
12) Display all donors according to their age.
13) Display the donor list in a recent order of donation date.
14) Display all distinct blood group types.
15) Update the age of all donors by 1.
16) Mr. RAJESH RAO changed his name to RAMESH RAO and he is shifted to DURGAPUR. Note the above changes in the table.

17) Due to certain reasons, all the donors who donated their blood on the date “26-AUG-11” are rejected. Hence delete their information.

18) Find the donors' names whose first name starts with the letter “A” and ends with “D“ irrespective of the case letter.

19) Find the donor names whose last name starts between alphabets “D” to “S” (Ex. DESHPANDE, SHARMA)

20) Find the total number of donors having the O+Ve group.

21) Find the total quantity of blood of group A+Ve.

22) The average age of female donors of the O+Ve group by rounding the age to the next digit.

23) Display all donors whose name pronounces “AAMIR”.

24) Find the donors who donated their blood in the month of AUG.

25) Find the donors who donated the blood on 15th Aug. of the year.

Functions

Perform the following queries on table donar (Functions)

1. Find the donar names whose first name starts with the letter ‘A’ and ends with ‘D’ irrespective of the case letter.

   (Ex. ANAND) Hint: Use SUBSTR and INSTR functions to extract the first name.

2. Find the donar names whose last name starts between alphabet ‘D’ to ‘S’

   (Ex. DESHPANDE, SHARMA)

   Hint: Use SUBSTR and INSTR functions to extract the first name.

3. Find the total number of donars having an O+ve group.

4. Find the total quantity of blood of group A+ve.

5. The average age of female donar of the O+ve group by rounding the age to the next digit.

   Hint: use Ceil function to round the age to the next digit.

6. Display all donars whose name pronounces like ‘AAMIR’;

7. Find the donars who donated blood in the month of AUG.

8. Find the donars who donated the blood on 15th Aug. of year.

9. Display all donar names in lowercase.

10. Find donars whose first name is five characters long.

11. Find every 3rddonar in the list. Donar numbers are assigned as consecutive no.

Perform Below Queries on Employee Table

1. List Manager of the Employees with Salary of EMP
2. Join EMP and DEPT and list departments 20,40 with or without any employees working in those departments.
3. Display all the employees working in NEW YORK
4. Find the employees who earn the maximum salary in their department (Hint: - Use Subqueries)

ORACLE (PL/SQL)

1) Create the following Tables and Execute the respective PL/SQL blocks.

- Create a table employee with the fields (empno, ename, job, hiredate, jadate&sal).
- Create table Math with fields (numb, square, cube & square_root).
- Create a table Patient with fields (p name, age, prescription).
- Create a table Music album with fields (title, hero, singer, quantity).
- Create table Stu with fields (name & marks).
- Create table error with fields (error_no& description).
- Create a table DONAR where the following fields (Donar no., donar name, city, age, Sex, Blood group, quantity of blood given, date of donation)

2) Write a PL/SQL block to accept the employee number and display his/her job, joining date, and salary of an employee. Define the variable using % rowtype.

3) Write a PL/SQL block to accept three paper marks and display the result if a student scores more than 35 marks on each paper and also specify the class.

4) Write a PL/SQL block to find the square, cube, and square root of nos. between 1 & 25 using a loop.

5) Write a program to divide a number by a character number. If any error occurs it should be handled properly, and store the error number and its description in a table called errorh.

6) Write a PL/SQL block to accept and insert valid data into the table patient. Write appropriate user-defined exceptions.

7) Write a PL/SQL block, to display only the title and quality of all albums stored in the table music album.

8) Write a PL/SQL to delete the records from the table music album where the quantity is less than 4 using a cursor.

9) Write a PL/SQL block to display the employee all having salary>some value. The value some value can be passed during execution or through bind variables.

10) Write a PL/SQL block to accept the title and display other information; it must handle the exception properly.

M.Sc. (Computer Science) - I (SEMESTER – II)
Paper Code: 02MSCCS07
Lab - II
Based on 02MSCCS03 & Selected Elective)

Credit: 2 **Max. Marks: 100**

Practical on Android Application Development (02MSCCS03)

1. Develop an application that uses GUI components, Font, and Colors
2. Develop an application that uses Layout Managers and event listeners.
3. Develop a native calculator application.
4. Write an application that draws basic graphical primitives on the screen.
5. Develop an application that makes use of the database.
6. Develop an application that makes use of an RSS Feed.
7. Testing your Android development environment perform the following operations.
    a. Add the sample application to a project in your Android Studio workspace.
    b. Create an Android Virtual Device (AVD) for your sample project.
    c. Create a launch configuration for your sample project.
    d. Run your sample application in Android Emulator.
8. Write a program to build your first Android Application “Hello World” with common activity.
9. Write a program that will implement the Sub-menu in the Android application.
10. Write a program that will implement the Context menu (Floating List of Menu Items) in the Android application.
11. Write a program to display the use of Relative Layout Views with different attributes.
12. Write a program to display the use of Linear Layout Views with different attributes.
13. Write a program to implement a menu that uses checkable items in Menu.
14. Write a program to implement a Custom Button and handle the displayed message on the button press.
15. Write a program to implement the Table layout in View Group that displays child View elements in rows and columns.
16. Write a program to implement the List View in your Android application.
17. Write a program to implement tween animation and rotate the text in your Android application.
18. Write a sample program to create a progress bar for your Android applications.
19. Write a program to show how to use the Date picker control of ADK in your Android applications.
20. Write a program that enables you to draw an image using a bitmap class object.
21. Implement an application that implements Multi-threading
22. Implement an application that writes data to the SD card.
23. Implement an application that creates an alert upon receiving a message.
24. Write a mobile application that creates an alarm clock
25. Write a program that shows you how to handle any type of interruption in your Android application.
26. Write a program that allows you to set an image as wallpaper.
27. Write a program that allows you to get images from the web and displayed them using the Image View.
28. Write a program that shows you how to create a scroll view when text is not visible onone page.
29. Write a program that will show you how to run any video file.
30. Develop a native application that uses GPS location information.

Practical: Cyber Security & IPR (02MSCCS04.1)

1. **Introduction to Cyber Security:**
   - Familiarize students with the basic concepts of cyber security, including threats, vulnerabilities, and attacks.
   - Discuss the importance of cyber security in protecting sensitive information and preventing unauthorized access.
2. **Password Security:**
   - Teach students about the significance of strong passwords and password management.
   - Conduct a hands-on exercise on creating strong passwords and using password managers.
3. **Network Security:**
   - Demonstrate different types of network attacks, such as DoS (Denial of Service) and DDoS (Distributed Denial of Service) attacks.
   - Guide students on implementing basic network security measures, such as firewalls and intrusion detection systems.
4. **Malware Analysis:**
   - Introduce students to various types of malware, such as viruses, worms, and Trojans.
   - Provide practical examples of malware analysis techniques, including static and dynamic analysis.
5. **Social Engineering:**
   - Explain the concept of social engineering and its impact on cyber security.
   - Conduct a simulated phishing exercise to raise awareness about social engineering attacks and teach students how to identify and avoid them.
6. **Web Application Security:**
   - Discuss common web application vulnerabilities, such as SQL injection and cross-site scripting (XSS).
   - Guide students on performing security assessments and vulnerability scanning on web applications.
7. **Intellectual Property Rights (IPR) Protection:**
   - Educate students about the importance of IPR and its role in protecting creative works.
   - Conduct a case study on copyright infringement and discuss the legal implications.
8. **Digital Forensics:**
   - Introduce students to the field of digital forensics and its role in investigating cybercrimes.
   - Provide hands-on experience with digital forensics tools and techniques, such as data recovery

and analysis.

9. **Incident Response:**
   - Teach students about the incident response process and the steps involved in handling a cyber security incident.
   - Conduct a simulated incident response exercise to help students understand the importance of timely and effective responses.

10. **Ethical Hacking:**
    - Introduce students to ethical hacking and its role in identifying vulnerabilities and improving security.
    - Guide students on performing ethical hacking exercises, such as penetration testing and vulnerability assessment.

Practical: SOFT SKILLS (02MSCCS04.2)

Case Study 1: Effective Communication Scenario: Sarah is a project manager leading a team of developers. She needs to communicate project updates to the stakeholders, including the client and the senior management.

Challenge: Sarah struggles with effectively conveying complex technical information concisely and understandably.

Case Study 2: Teamwork and Collaboration Scenario: John works in a marketing agency where teamwork and collaboration are crucial for successful campaign execution. He is assigned to a cross-functional team responsible for launching a new product.

Challenge: John finds it difficult to collaborate effectively with team members from different departments due to conflicting opinions and lack of coordination.

Case Study 3: Adaptability and Problem-Solving Scenario: Lisa works as a customer service representative in a fast-paced retail store. She often encounters challenging situations, such as handling difficult customers or resolving unexpected issues.

Challenge: Lisa struggles to adapt to rapidly changing circumstances and lacks problem-solving skills to address customer complaints effectively.

Case Study 4: Leadership and Emotional Intelligence Scenario: Mark is a team leader in a software development company. He is responsible for managing a team of developers and ensuring timely project delivery.

Challenge: Mark struggles with motivating and inspiring his team members, resulting in low morale and decreased productivity.

Practical: DIGITAL AND CYBER FORENSICS (02MSCCS04.3)

1. **Case Study: Cyber Attack Investigation**
    - A company's network has been compromised, resulting in a data breach. As a digital forensics investigator, outline the steps you would take to investigate this cyber-attack. What evidence would you look for, and how would you analyze it to identify the attacker?
2. **Case Study: Mobile Device Forensics**
    - A suspect's smartphone has been seized as part of a criminal investigation. Describe the process you would follow to acquire and analyze digital evidence from the mobile device. What types of data would you extract, and how would you ensure the integrity of the evidence?
3. **Case Study: Malware Analysis**
    - A computer has been infected with a new strain of malware. As a malware analyst, explain the steps you would take to analyze the malware. What techniques and tools would you use to identify its behavior, extract indicators of compromise, and determine its impact on the compromised system?
4. **Case Study: Incident Response**
    - A company's network has experienced a security incident involving unauthorized access and data exfiltration. As an incident responder, outline the steps you would take to contain and mitigate the incident. How would you preserve and analyze digital evidence to identify the attacker's entry point and the extent of the compromise?
5. **Case Study: File System Analysis**
    - A suspect's computer has been seized in a fraud investigation. Describe the techniques you would employ to analyze the computer's file system. How would you recover deleted files, examine file metadata, and identify any evidence related to the fraudulent activities?

Practical: OPERATION RESEARCH (02MSCCS04.4)

1. **Shortest Path Problem**
   - Implement Dijkstra's algorithm to find the shortest path between two nodes in a network.
   - Develop a program to solve the shortest path problem using the Bellman-Ford algorithm.
2. **Minimum Spanning Tree**
   - Write a program to find the minimum spanning tree of a connected graph using Kruskal's algorithm.
   - Implement Prim's algorithm to find the minimum spanning tree of a weighted graph.
3. **Decision Trees**
   - Develop a program to construct a decision tree for a given decision problem using algorithms like ID3 or C4.5.
   - Implement the decision tree to make decisions based on uncertain outcomes and probabilities.
4. **Multi-Criteria Decision Making**
   - Write a program to implement the Analytic Hierarchy Process (AHP) for multi-criteria decision-making problems.
   - Develop an application to solve decision problems using the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) method.
5. Decision Analysis with Decision Trees:
   - Write a program to analyze decision problems using decision trees and calculate expected monetary values (EMVs).
   - Develop an application to perform sensitivity analysis and determine the optimal decision strategy.

Practical: DATA VISUALIZATION (02MSCCS04.5)

1. **Introduction to Data Visualization**
   - Familiarize students with the importance of data visualization in understanding and communicating complex data.
   - Introduce different types of visualizations, such as bar charts, line graphs, scatter plots, and heat maps.
2. **Exploratory Data Analysis**
   - Provide a dataset and guide students on exploring and visualizing the data using appropriate charts and graphs.
   - Teach students how to identify patterns, trends, and outliers through data visualization.
3. **Design Principles for Effective Data Visualization**
   - Discuss the key design principles for creating effective and impactful visualizations, such as simplicity, clarity, and consistency.
   - Conduct exercises where students evaluate and improve existing visualizations based on design principles.
4. **Data Visualization with Python**
   - Introduce students to popular Python libraries for data visualization, such as Matplotlib, Seaborn, and Plotly.
   - Guide students on creating various types of visualizations using these libraries, including bar plots, line plots, and scatter plots.
5. **Interactive Data Visualization**
   - Teach students how to create interactive visualizations using libraries like Bokeh or Plotly.
   - Conduct exercises where students build interactive dashboards or explore interactivity features like tooltips and filters.
6. **Geographic Data Visualization**
   - Introduce students to geographic data visualization using libraries like GeoPandas or Folium.
   - Guide students on creating maps, choropleth visualizations, and spatial analysis using geographic data.
7. **Network Data Visualization**
   - Teach students how to visualize network data using libraries like NetworkX or Gephi.
   - Conduct exercises where students create network graphs, analyze network properties, and visualize relationships.
8. **Data Visualization with Tableau**
   - Introduce students to Tableau, a popular data visualization tool.
   - Guide students on creating interactive dashboards, visualizing complex datasets, and using advanced features in Tableau.
9. **Storytelling with Data Visualization**
   - Discuss the importance of storytelling in data visualization and how to effectively communicate insights through visual narratives.
   - Conduct exercises where students create data-driven stories using visualizations to convey a clear message.

M.Sc. (Computer Science) - I (SEMESTER – II)
Paper Code: 02MSCCS08
SEMINAR

Credit: 1 **Max. Marks: 50**

Program Learning Outcome:

1. To analyze a current topic of professional interest and present it before the audience.
2. To familiar with basic technical writing concepts and terms, such as audience analysis, jargon, format, visuals, and presentation and acquired the basic skills to for performing literature survey and paper presentation.
3. To improve skills to read, understand, and interpret material on technology.
4. To improve communication and writing skills and prepare the report.

The seminar must be based on some current trends related to IT/Computer Science/Computer Application. A student must present the PowerPoint presentation along with Seminar Report. Students are requested to follow the following guidelines while choosing & preparing their seminars.

Guidelines to Seminar

1) The name of the seminar topic must be the latest on the current trends and should not be repeated. A student must submit the topic at the beginning of the semester.
2) The Seminar topic is to be approved by the departmental allocated guide.
3) Students are allowed to use graphics/animation/audio-video aids for their presentations.
4) Seminar work will be evaluated by an internal examiner. (Refer to Appendix B)
5) Students are requested to submit their seminar reports on or before the deadline with the concern of their respective guides otherwise students will be responsible for any appropriate action.
6) The seminar report should be submitted to the department in the following format, printed in double line space using A4 size bond paper, with a left margin of 1.5” and a right margin of 1.0” with proper spiral binding to be done. Only one copy needs to be submitted.
7) Students are requested to obtain the necessary certificates and declarations to be duly enclosed in the report.

Appendix A

Internal Assessment

1. The internal assessment marks shall be awarded by the concerned approved teacher by the university.
2. The internal assessment marks shall be sent to the University after the Assessment in the prescribed format and direction by University.
3. General guidelines for Internal Assessment are:
   a) The internal assessment marks assigned to each theory paper on the basics of the performance in any two assignments (each of 10 marks) as described below and selected by the concerned teacher.
      1. Class Test / Sessional Examination
      2. Certification from IIT Spoken Tutorial / Swayam / NPTEL / PARAKH etc.
      3. Application-Oriented Case Study/ Project
      4. Online Test
      5. Theory Assignments
      6. Programming Assignments
      7. Study tour
      8. Industrial visits
      9. Visit educational institutions and research organizations, do fieldwork, Conference, etc
      10. Group discussions
      11. Conference/ Webinar
      12. Seminar Presentation
      13. Publishing Research Paper
      14. Review of Research Papers
      15. Open Book Test
      16. Participation in Departmental Activities

   b) There shall be no separate/extra allotment of workload to the concerned teacher related to the above assignments. He/ She shall conduct the internal assessment activity during the regular teaching days/periods as a part of the regular teaching activity.

   c) The concerned teacher/department/college shall have to keep a record of all the above activities until six months after the declaration of the results of that semester.

   d) At the beginning of each semester, every teacher/department/college shall inform his / her students unambiguously of the method he/she proposes to adopt and the scheme of marking for internal assessment. (Prescribed in the syllabus of respective Subjects)

   e) A teacher shall announce the schedule of activity for internal assessment in advance in consultation with HOD.

Appendix B

Seminar and Practical Marks Distribution

- **Seminar**

In the Seminar, the student will have to deliver a seminar on any topic relevant to the syllabus/subject encompassing the recent trends and development in that field/subject. The topic of the seminar will be decided at the beginning of each semester in consultation with the supervising teachers. The student has to deliver the seminar which will be followed by a discussion.

The students should submit the seminar report typed and properly bound in one copy to the head of the department along with a soft copy on CD. The said shall be evaluated by the concerned supervisor/head of the department. The concerned teacher/department/college shall have to keep a record of Seminar Reports until six months after the declaration of the results of that semester.

Marks Distribution: A seminar mark will be allocated by Internal Examiner as per the following format

Sr. No.	Particulars	Total Marks (50)	Min. Marks for Passing (40%)
a.	Seminar Report & Documentation	30	12
b.	Seminar Presentation	20	08
Total		50	20

- **Practical Assessment:**

Time: Minimum 2 Hours 30 Min. for conducting the practical examination subject to the condition of the availability of computers and printers at the center.

Marks Distribution: Internal (Approved by the University) & External Examiner will allocate the practical marks in the ratio of (25:75) to students as per the following format. Both examiners together will fill (upload) the marks on the date of the practical examination.

Sr. No.	Particulars	Marks in %	Time
a.	Writing, Execution, and Printout of Program-I	40%	1 Hour
b.	Writing Program/Case Studies-II	20%	30 Min
c.	Practical Record	20%	
d.	Viva Voce	20%	

Note: In a day, examiner can conduct a maximum of two Practical Examinations of different classes/courses but not of the same class.

Appendix C

The Pattern of the Question Paper

General Rules and Regulations regarding a pattern of question papers for the semester-end examinations as given below:

1. There will be four units in each paper.
2. The maximum mark for each theory paper will be 80.
3. The question paper will consist of five questions, each with 16 marks.
4. Four questions will be based on four units with internal choice.
5. The fifth question will be compulsory with questions from each of the four units having equal weightage and there will be no internal choice.

M.Sc. (Computer Science)

Year : [I/II] **Semester : [I/II/III/IV]**

Paper Code: **Paper: Name of Paper**

Time: 3 Hours] **[Max. Marks: 80**

Note 1) All questions are compulsory and carry equal marks.
2) Draw a Neat and Labeled diagram and use supporting data wherever necessary.
3) Avoid vague answers and write specific points/answers to related questions.

Q1 EITHER (From Unit1)	
a)	8
b)	8
OR	
c)	8
d)	8
Q2 EITHER (From Unit2)	
a)	8
b)	8
OR	
c)	8
d)	8
Q3 EITHER (From Unit3)	
a)	8
b)	8
OR	
c)	8
d)	8
Q4 EITHER (From Unit4)	
a)	8
b)	8
OR	
c)	8
d)	8
Q5 Solve all questions	
a) (From Unit 1)	4
b) (From Unit 2)	4
c) (From Unit 3)	4
d) (From Unit 4)	4