SYLLABUS FOR COMPUTER SCIENCE WITH HONOURS WITH RESEARCH

SEM. VII

4th Year [Four Year Under Graduate Program with Honours with Research] (FYUGPHWRS)

Under Single Major Single Minor

(To be implemented from Session 2023-24)

Proposed Syllabus for four years Computer Science (Major) Programme							
Year	**Semester**	**Paper Code**	**Paper**	**Credits**	**Periods/Week**	**Exam. Marks**	**Total Marks**
3rd Year	**VII**	UCMSMAJ47017	Computer Vision	3	3	60	60
		UCMSMAJ47017 T	Computer Vision (Tutorial)	1	1	00	
		UCMSMAJ47018	Fundamentals of Cryptography	3	3	40	60
		UCMSMAJ47018 L	Fundamentals of Cryptography (Tutorial)	1	1	20	
		UCMSMAJ47019	R Programming	3	3	40	60
		UCMSMAJ47019 L	R Programming (Lab)	1	1	20	
	VIII	UCMSMAJ48020	**Research Methodology**	4	4		
		IARD	**MAJOR RESEACRH PRJECT / DISSERTATION**	12	12		

NOTE: Tutorials should involve problem solving session/activity related to the subject taught.

SYLLABUS FOR COMPUTER SCIENCE WITH HONOURS WITH RESEARCH

Discipline: Science [√] Arts, Humanities & Social Science ☐

Commerce ☐ BBA ☐ BCA ☐

Subject Name: Computer Science

Subject Code: **UCMSMAJ47017** (Will be provided by the University)

Semester: Semester I ☐ Semester II ☐ Semester III ☐ Semester IV ☐

Semester V ☐ Semester VI ☐ Semester VII [√] Semester VIII ☐

Course Name: **COMPUTER VISION**

Course Code: (Will be provided by the University)

Course Credit: Theoretical 3 Practical/Tutorial 1

Marks Allotted: Theoretical 60 Practical/Tutorial 0

Continuing Evaluation Attendance

Course Type (tick the correct alternatives):

Major Core	[√]	AEC	☐
Interdisciplinary/ DSE	☐	SEC	☐
Minor / Generic Elective	☐	VAC	☐
Research Project/Dissertation	☐	Vocational	☐

Is the course focused on employability / entrepreneurship? YES [√] NO ☐

Is the course focused on imparting life skill? YES [√] NO ☐

Is the course based on Activity ? YES [√] NO ☐

Remarks by Chairman, UG BOS, if any

UG BOS Meeting Reference Number: Date:

Course Code: **UCMSMAJ47017**

Course Name: **COMPUTER VISION**

Brief Course Description

Data Science is an interdisciplinary field that combines statistics, mathematics, computer science, and domain knowledge to extract meaningful insights from data. This course provides a comprehensive introduction to the principles, methods, and tools used in data science for data collection, cleaning, analysis, visualization, and predictive modeling. Students will learn how to transform raw data into valuable information that supports decision-making and problem-solving across various domains such as business, healthcare, education, finance, and research. The course emphasizes both theoretical foundations and practical applications using modern data science techniques and tools.

Prerequisite(s) and/or Note(s)

Students should have a basic understanding of mathematics, statistics, and computer programming concepts. Familiarity with data handling, spreadsheets, and logical problem-solving will be beneficial. Prior exposure to programming languages such as Python or R is desirable but not mandatory.

Course Objectives

This course aims to introduce students to the fundamental concepts, techniques, and tools of Data Science. It focuses on data collection, preprocessing, analysis, visualization, and interpretation to support data-driven decision-making. The course also provides exposure to statistical methods, machine learning techniques, and real-world applications of data science.

Knowledge Acquired

Upon completion of the course, students will gain knowledge of the Data Science lifecycle, data preprocessing methods, exploratory data analysis, statistical techniques, machine learning algorithms, and data visualization principles. They will understand how data can be transformed into meaningful insights for solving business, scientific, and societal problems.

Skills Gained

Students will develop practical skills in handling and analyzing large datasets, performing statistical analysis, creating visualizations, and implementing machine learning models using modern data science tools and programming environments. They will also learn to communicate analytical findings effectively through reports and presentations.

Competency Developed

The course develops competency in applying data science methodologies to real-world problems, extracting actionable insights from complex datasets, and building predictive models for decision support. Students will be able to work independently or as part of a team in data-driven projects while adhering to ethical and professional standards in data analysis.

Detailed Syllabus

4th Year: Semester 7

UCMSMAJ47017	COMPUTER VISION	[Credits: 3, Lectures 45]

UNIT 1 – INTRODUCTION TO COMPUTER VISION AND DIGITAL IMAGES

(6 Lectures)

Definition and brief history of computer vision, real-world applications such as medical imaging, autonomous vehicles, face unlock, and barcode scanning; the human visual system versus cameras; image formation through pinhole camera and lenses; pixels and resolution; grayscale and RGB color images; common image formats including JPEG, PNG, and BMP; and basic concepts of sampling and quantization.

UNIT 2 – IMAGE ENHANCEMENT AND SIMPLE OPERATIONS (8 Lectures)

Brightness and contrast adjustment; histograms and histogram equalization; image resizing, cropping, and rotation; smoothing and sharpening filters; noise removal; and edge detection using Sobel, Prewitt, and Canny methods.

UNIT 3 – FINDING FEATURES AND MATCHING IMAGES (8 Lectures)

Features are such as corners and edges; the Harris Corner Detector; the scale invariance problem; SIFT and HOG descriptors for creating feature fingerprints; feature matching between images; Hough Transform for detecting lines and circles through voting; and RANSAC for removing wrong matches.

UNIT 4 – RECOGNIZING OBJECTS AND SEGMENTING IMAGES (8 Lectures)

Image segmentation using thresholding, region growing, and K-means clustering; the differences between classification, detection, and segmentation; template matching and image pyramids; Bag of Visual Words representation; basic face detection concepts; and motion tracking using optical flow.

UNIT 5 – MODERN DEEP LEARNING AND APPLICATIONS (15 Lectures)

why classical methods reached limits and how machine learning replaced hard-coded rules; neural networks and deep learning basics; convolutional neural network building blocks including convolution, pooling, and fully connected layers; famous architectures LeNet, AlexNet, VGGNet, and ResNet at a conceptual level; object detection methods including two-stage R-CNN and one-stage YOLO; semantic segmentation using FCN and U-Net; real-world applications in medical imaging, autonomous vehicles, agriculture, and security; ethical challenges including bias, privacy, adversarial attacks, and energy costs; and future trends including Vision Transformers, generative AI, and multimodal AI.

Suggested Readings

1. **Szeliski, Richard.** *Computer Vision: Algorithms and Applications* (2nd Edition). Springer, 2022.
2. **Gonzalez, Rafael C., and Woods, Richard E.** *Digital Image Processing* (4th Edition). Pearson Education, 2018.
3. **Forsyth, David A., and Ponce, Jean.** *Computer Vision: A Modern Approach* (2nd Edition). Pearson Education, 2011.
4. **Howse, Joseph, and Minichino, Joe.** *Learning OpenCV 4: Computer Vision with Python* (3rd Edition). Packt Publishing, 2020.
5. **Solem, Jan Erik.** *Programming Computer Vision with Python*. O'Reilly Media, 2012.
6. **Goodfellow, Ian, Bengio, Yoshua, and Courville, Aaron.** *Deep Learning*. MIT Press, 2016.
7. **Shanmugamani, Rajalingappaa.** *Deep Learning for Computer Vision*. Packt Publishing, 2018.
8. **Singh, Himanshu, and Gandhi, Bhavya.** *Practical Machine Learning and Image Processing*. Apress, 2019.

UBCAMAJ47017T	COMPUTER VISION	[Credits: 1, Lecture Hours:15]

Computer Vision tutorial as assigned and advised by teacher(s).

SYLLABUS FOR COMPUTER SCIENCE WITH HONOURS WITH RESEARCH

Discipline: Science [√] Arts, Humanities & Social Science ☐

Commerce ☐ BBA ☐ BCA ☐

Subject Name: COMPUTER SCIENCE

Subject Code: **UCMSMAJ47018** (Will be provided by the University)

Semester: Semester I ☐ Semester II ☐ Semester III ☐ Semester IV ☐

Semester V ☐ Semester VI ☐ Semester VII [√] Semester VIII ☐

Course Name: **FUNDAMENTALS OF CRYPTOGRAPHY**

Course Code: (Will be provided by the University)

Course Credit: Theoretical 3 Practical/Tutorial 1

Marks Allotted: Theoretical 60 Practical/Tutorial 0

Continuing Evaluation Attendance

Course Type (tick the correct alternatives):

Major Core	[√]	AEC	☐
Interdisciplinary/ DSE	☐	SEC	☐
Minor / Generic Elective	☐	VAC	☐
Research Project/Dissertation	☐	Vocational	☐

Is the course focused on employability / entrepreneurship? YES [√] NO ☐

Is the course focused on imparting life skill? YES [√] NO ☐

Is the course based on Activity ? YES [√] NO ☐

Remarks by Chairman, UG BOS, if any

UG BOS Meeting Reference Number: Date:

Course Code: **UCMSMAJ47018**

Course Name: **FUNDAMENTALS OF CRYPTOGRAPHY**

Brief Course Description

This course introduces the fundamental concepts and techniques of cryptography used to secure digital communication and information systems. It covers classical and modern encryption methods, cryptographic algorithms, authentication mechanisms, digital signatures, hash functions, key management, and their applications in network and cybersecurity. The course provides both theoretical understanding and practical knowledge of how cryptographic techniques protect data confidentiality, integrity, authenticity, and privacy in modern computing environments.

Prerequisite(s) and/or Note(s)

Students should possess a basic understanding of computer systems, networking concepts, and elementary mathematics, including number systems and logical reasoning. Prior exposure to cybersecurity concepts is beneficial but not mandatory. The course is designed to provide foundational knowledge suitable for undergraduate students from computing and related disciplines.

Course Objectives

The course aims to develop an understanding of the principles and importance of cryptography in securing information and communication systems. It seeks to familiarize students with various encryption techniques, cryptographic protocols, authentication methods, and security applications while enabling them to analyze and apply appropriate cryptographic solutions to address modern cybersecurity challenges.

Knowledge Acquired

Upon completion of the course, students will gain knowledge of cryptographic fundamentals, including symmetric and asymmetric encryption, hashing algorithms, digital signatures, key exchange mechanisms, authentication protocols, and public key infrastructure. They will understand the role of cryptography in securing networks, web applications, digital transactions, and information systems against unauthorized access and cyber threats.

Competency Developed

Students will develop the ability to identify security requirements and apply suitable cryptographic techniques to protect data and communications. They will gain competencies in evaluating cryptographic solutions, implementing basic encryption and authentication mechanisms, understanding secure communication protocols, and contributing effectively to cybersecurity and information assurance practices in professional environments.

Detailed Syllabus

4th Year: Semester 7

UCMSMAJ47018 FUNDAMENTALS OF CRYPTOGRAPHY [Credits: 3, Lectures 45]

UNIT 1: INTRODUCTION TO CRYPTOGRAPHY AND SECURITY CONCEPTS

(5 Lectures)

Introduction to cryptography, its importance in information security, and its role, symmetric key cryptography, asymmetric key cryptography, hybrid cryptography, plaintext, ciphertext, encryption, decryption, keys, ciphers, cryptanalysis, and cryptology. Security goals: confidentiality, integrity, authentication, non-repudiation, and availability. Applications of cryptography in web security, online banking, e-commerce, mobile communications, cloud computing, and secure messaging. Overview of hash functions, Common attacks on cryptographic systems: brute-force, known-plaintext, password, and social engineering attacks.

UNIT 2: CLASSICAL CRYPTOGRAPHY TECHNIQUES (10 Lectures)

Introduction to classical cryptography and traditional encryption methods used for secure communication, Substitution and transposition ciphers: Monoalphabetic Cipher and Polyalphabetic Cipher, Caesar Cipher, Playfair Cipher, Rail Fence Cipher, and Columnar Transposition Cipher along with their encryption and decryption techniques, Limitations of classical cryptography.

UNIT 3: MODERN SYMMETRIC KEY CRYPTOGRAPHY (10 Lectures)

Symmetric key cryptography: use of a shared secret key for encryption and decryption, Concepts of block and stream ciphers, differences, and applications, Data Encryption Standard (DES), Triple DES (3DES), and Advanced Encryption Standard (AES), basic structures, key sizes, advantages, and limitations, Key management : key generation, distribution, storage, and session keys. Applications of symmetric encryption in file protection, disk encryption, database security, wireless networks, and VPNs.

UNIT 4: PUBLIC KEY CRYPTOGRAPHY AND DIGITAL SIGNATURES (10 Lectures)

Public Key Cryptography: Need for asymmetric encryption, Public key and private key, Key pair generation, Advantages and limitations. RSA Cryptography: RSA, Mathematical foundation basics, Prime numbers in cryptography, RSA key generation, RSA encryption and decryption, $C=M^e$ (modn), $M=C^d$ (modn). Diffie-Hellman, Key Exchange: Secure key exchange problem, Diffie-Hellman principles, Shared secret generation, Applications, Digital Signatures: definition digital signatures, Signature generation, Signature verification, Integrity verification, Non-repudiation, Applications of Public Key Cryptography: Secure email, SSL/TLS, E-commerce security, Digital certificates, Secure authentication.

UNIT 5: HASHING AND AUTHENTICATION **(10 Lectures)**

Hash Functions: Definition of hashing, One-way functions, Fixed-length output, Hashing properties, Collision resistance, Hash algorithms: MD5 and SHA family. Fundamentals of password security: password hashing, salting, and secure password storage, Message authentication and data integrity verification, Basic concepts of digital certificates and Certificate Authorities (CA), Secure authentication methods: passwords, One-Time Passwords (OTP), 2FA and multi-factor authentication (MFA) for protecting user accounts and systems.

Suggested Readings:

Textbooks

1. "Cryptography and Network Security" — William Stallings — 8th Edition
2. "Understanding Cryptography" — Christof Paar& Jan Pelzl
3. "Applied Cryptography" — Bruce Schneier
4. "Introduction to Modern Cryptography" — Jonathan Katz & Yehuda Lindell
5. "Cryptography Made Simple" — Nigel Smart

Reference Books

6. "Serious Cryptography" — Jean-Philippe Aumasson
7. "Network Security Essentials" — William Stallings
8. "Handbook of Applied Cryptography" — Alfred Menezes
9. "Computer Security Principles and Practice" — William Stallings
10. "Everyday Cryptography" — Keith Martin
11. "Practical Cryptography" — Niels Ferguson & Bruce Schneier
12. "Introduction to Cryptography with Coding Theory" — Trappe & Washington

UCMSMAJ47018T FUNDAMENTALS OF CRYPTOGRAPHY [CREDITS: 1, LECTURE:15]

Fundamentals of Cryptography tutorial as assigned and advised by teacher(s).

SYLLABUS FOR COMPUTER SCIENCE WITH HONOURS WITH RESEARCH

Discipline: Science [√] Arts, Humanities & Social Science ☐

Commerce ☐ BBA ☐ BCA ☐

Subject Name: COMPUTER SCIENCE

Subject Code: **UCMSMAJ47019** (Will be provided by the University)

Semester: Semester I ☐ Semester II ☐ Semester III ☐ Semester IV ☐

Semester V ☐ Semester VI ☐ Semester VII [√] Semester VIII ☐

Course Name: **R PROGRAMMING**

Course Code: (Will be provided by the University)

Course Credit: Theoretical 3 Practical/Tutorial 1

Marks Allotted: Theoretical 40 Practical/Tutorial 20

Continuing Evaluation Attendance

Course Type (tick the correct alternatives):

Major Core	[√]	AEC	☐
Interdisciplinary/ DSE	☐	SEC	☐
Minor / Generic Elective	☐	VAC	☐
Research Project/Dissertation	☐	Vocational	☐

Is the course focused on employability / entrepreneurship? YES [√] NO ☐

Is the course focused on imparting life skill? YES [√] NO ☐

Is the course based on Activity ? YES [√] NO ☐

Remarks by Chairman, UG BOS, if any

UG BOS Meeting Reference Number: Date:

Course Code: **UCMSMAJ47019**

Course Name: **R PROGRAMMING**

Brief Course Description

This course introduces the fundamentals of R programming for data analysis, statistical computing, and visualization. Students will learn R syntax, data structures, control statements, data manipulation, graphical representation of data, and basic statistical analysis techniques using R. The course emphasizes practical applications of R in data science and analytics.

Prerequisite(s) and/or Note(s)

Basic knowledge of computer fundamentals and elementary mathematics/statistics is desirable. No prior programming experience is required.

Course Objectives

The course aims to provide students with a strong foundation in R programming, data handling, statistical analysis, and data visualization. It enables learners to develop analytical skills and apply R tools for solving real-world data-driven problems.

Knowledge Acquired

Students will gain knowledge of R programming concepts, data structures, control structures, data manipulation techniques, visualization methods, statistical analysis, probability distributions, and hypothesis testing using R.

Skills Gained

Students will develop skills in writing R programs, managing and analyzing datasets, creating informative visualizations, performing statistical computations, and importing/exporting data from various sources.

Competency Developed

Upon completion of the course, students will be able to use R effectively for data analysis, statistical modeling, visualization, and decision-making tasks, preparing them for advanced studies and careers in data science, analytics, and research.

Detailed Syllabus
4th Year: Semester 7

UCMSMAJ47019	R PROGRAMMING	[Credits: 3. Lectures 45]

Unit 1: Introduction to R Programming **(3 Lectures)**

Overview of R programming: Evolution of R language, features of R programming language, what is R? , Why R? , Advantages of R over Other Programming Languages, Downloading and installing of R language, setting up the R environment - Interactivity and Code Execution: R basic syntax - R Studio - R command Prompt, R script file, comments

Unit 2: Data Structures in R **(8 Lectures)**

Vectors: Creating vectors using c(), Indexing and subsetting vectors in R, Operations on Vectors, Understanding vectorized operations and efficiency – *Matrices* : Creating matrices using matrix(), *Basic matrix operations* : addition, subtraction, multiplication, Transposing matrices - *Lists* : Creating lists with list(), Nested lists and their components, Indexing and extracting values from lists, Manipulating list components – *Arrays* : Creating arrays using array(), Manipulating and transforming array data - *Data Frames:* Introduction to Data Frames, Manipulating Data Frames, Indexing, filtering, and subsetting data frames, Applying functions to columns, Aggregating and summarizing data- *Factors:* Categorical data representation with factors, Modifying and Working with Ordinal Factors in R, Merging and Simplifying Factor Levels in R, Converting factors to other data types.

Unit 3: Control Structures in R **(7 Lectures)**

Conditional Statements: Basic if and else statements, else if and multiple conditions, Vectorized conditions with ifelse() - Loops in R : For Loop, Nested For Loops, break and next statements, While Loop, Repeat Loop.

Unit 4: Variables, Data Types and operators **(5 Lectures)**

Concept of Variables: Variable assignment, Finding Variable, Deleting Variables – Atomic Data Types in R (Numeric, Integer, Logical, Character, Complex and Raw) - Operators in R: Arithmetic Operators, Logical Operators, Comparison Operator, Assignment Operators.

Unit 5: Data Manipulation and Cleaning **(7 Lectures)**

Working with Strings: String manipulation - Data Cleaning: Handling missing values with is.na() and na.omit() - Reshaping Data : Reshaping data frames with reshape(), melt(), cast() - Data Import and Export: Reading data from CSV, Excel, and other formats.

Unit 6: Data Visualization in R **(7 Lectures)**

Concept of Plotting, Base R plotting functions: plot(), hist(), boxplot(), Customizing plots (titles, labels, colors), Saving plots to file (PDF, PNG) - Visualizations with ggplot2: ggplot2 package, Creating scatter plots, bar plots, histograms, and more, Faceting in ggplot2: Splitting Data into Subplots, Themes in ggplot2 - 3D Visualizations: concept of 3D visualization, Generating 3D scatter plots, surface plots.

Unit 7: R in Statistical Analysis and Resampling Techniques **(8 Lectures)**

Concept of R in Statistics: R as a tool for statistical analysis, calculation: mean, median, mode, variance, standard deviation, creating frequency tables and cross-tabulations - Probability Distributions in R: Normal Distribution in R, Plotting of Normal Distribution over Histogram in R - Hypothesis Testing: hypothesis testing concepts, T-tests, Chi-squared tests.

Suggested Readings:

1. ***The Book of R*** by Tilman M. Davies, no starch press (San Francisco)
2. ***The Art of R Programming – A Tour of Statistical Software Design*** , by : Norman Matloff, no starch press(San Francisco)
3 ***The R Book***, By – Michael J. Crawley, Edition – Second, Published by – Wiley
4. ***R Graphics Cookbook: Practical Recipes for Visualizing Data***, Winston Chang, Edition – Second, Published by – Shroff/O'Reilly

UCMSMAJ47019L R PROGRAMMING [Credits: 1, Lecture Hours:15]

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

1. Download and install R-Programming environment and install basic packages using install. Packages() command in R.
2. Learn all the basics of R-Programming (Data types, Variables, Operators etc.)
3. Implement R-Loops with different examples.
4. Learn the basics of functions in R and implement with examples.
5. Implement data frames in R. Write a program to join columns and rows in a data frame using c bind () and r bind () in R.
6. Implement different String Manipulation functions in R.
7. Implement different data structures in R(Vectors ,Lists ,Data Frames)
8. Create a data set and do statistical analysis on the data using R.
9. Write an R program to demonstrate variable declaration, data types, arithmetic operations, and output display.
10. Create two vectors and perform addition, subtraction, multiplication, division, and indexing operations.
11. Create two matrices and perform matrix addition, subtraction, multiplication, and transpose operations.
12. Create a list containing student information (Name, Roll No., Marks, Grade) and demonstrate accessing and modifying list elements.
13. Create a three-dimensional array and display specific elements, rows, and columns.
14. Create a student data frame and perform sorting, filtering, subsetting, and column-wise calculations.
15. Create nominal and ordinal factors, modify factor levels, and convert factors into character and numeric formats.
16. Write an R program using if-else and nested if-else statements to determine grades based on student marks.
17. Write R programs using for, while, and repeat loops to generate multiplication tables and factorials.
18. Perform string concatenation, substring extraction, case conversion, and string length calculation.
19. Create a dataset with missing values and use is.na() and na.omit() to identify and remove missing entries.
20. Read data from a CSV file, perform basic analysis, and export the processed data to a new CSV file.
21. Create a histogram, bar chart, boxplot, and pie chart using sample datasets.
22. Create scatter plots, bar plots, and histograms with appropriate labels and themes using ggplot2.
23. Calculate mean, median, mode, variance, and standard deviation. Perform a t-test and Chi-Square test on a given dataset and interpret the results.
24. Write an R Program for "Hello World ".
25. Write an R Program to Add Two Vectors
26. Find the Sum, Mean and Product of the Vector in R Programming
27. Create an R Program to Take Input From the User
28. Create an R Program to Find the Minimum and Maximum
29. How to create R Multiplication Table
30. Write a R Program to Check Prime Number
31. Write a R Program to check Armstrong Number
32. Write a R Program to Print the Fibonacci Sequence
33. Write a R Program to Check for Leap Year
34. Write a R Program to Check if a Number is Odd or Even in R Programming
35. Write a R Program to print the elements of a vector.