CENTRAL UNIVERSITY OF KERALA

School of Physical Sciences

DEPARTMENT OF COMPUTER SCIENCE

Bachelor of Computer Applications (BCA)

Specialization: Data Science/Artificial Intelligence and Machine Learning

Programme Structure

Multidisciplinary, Internship embedded, value-added/experiential/skill-oriented courses with multiple entry/exit options as per NEP 2020

August 2025

Central University of Kerala
Department of Computer Science

BCA Honours with research Programme

Programme Objectives

1. To gain multidisciplinary knowledge and skills
2. To identify and analyze the real world problems and develop algorithms for solving the problems.
3. To understand the huge volume and value of data for finding solutions for the development of the society.
4. To equip the students with computational and modeling techniques.
5. To impart skills in Data Science, Business Intelligence, Big data Analytics, Artificial Intelligence, Machine Learning, Network Security, Full Stack Development etc.

Programme Outcome

PO	Attribute	Programme Outcomes
PO1	Ability to apply knowledge	Gain knowledge, skills and apply theoretical concepts to solve real world problems
PO2	Modern Tool Usage	Learn, select and apply appropriate techniques, resources, modern tools to model or analyze the real life activities
PO3	Environment and Sustainability	Demonstrate ethical, social, legal and environmental responsibilities for sustainable development
PO4	Individual and Team work	Function effectively as an individual and as a member or leader in teams in multidisciplinary fields
PO5	Communication	Communicate effectively, able to write reports, documentation and make presentations

Programme Specific Outcomes

1. Ability to identify, formulate and develop solutions to computational challenges and to analyse, design and develop cost effective solutions to the societal problems.
2. Ability to design, implement and evaluate computational system to meet desired needs within realistic constraints.
3. An ability to function effectively on teams to accomplish shared computing design, evaluation or implementation goals.
4. Understanding of professional, ethical, legal, security and social issues and responsibilities.
5. An ability to communicate and engage effectively with diverse stakeholders for the efficient development of software applications.

YEAR WISE CREDIT DISTRIBUTION FOR FOUR YEAR UG
BCA Honours With Research

Sem	Major Core	Discipline Elective	MDC	AEC	SEC	VAC	Intern	Research	Total
Sem 1	12	-	-	4	4	-	-	-	20
Sem 2	12	-	3	4	2	-	-	-	21
Sem 3	14	-	2	-	2	3	-	-	21
Sem 4	14	-	-	-	2	3	-	-	19
Sem 5	4	12	2	-	-	-	2	-	20
Sem 6	4	12	3	-	-	-	-	-	19
Sem 7	8	8	-	-	-	-	2	4	22
Sem 8	12	-	-	-	-	-	-	8	20
Total	80	32	10	8	10	6	4	12	162

3 Years BCA Programme	**Total Credits = 120**
4 Years BCA (Honours) / BCA (Honours with Research)	**Total Credits = 162**

Multiple Entry - Exit options

There is provision to join the course at the starting of fourth year after securing Bachelor in Computer Application (BCA) or BSc in Computer Science or equivalent. The following multiple exit option will be made available to the students joining BCA Research Program:

a. **One year:** Under Graduate Certificate in Computer Application
b. **Two years**: Under Graduate Diploma in Computer Application
c. **Three years**: Bachelor in Computer Application (BCA)
d. **Four years**:
   Bachelor in Computer Application with Honours : BCA (Honours)
   or
   Bachelor in Computer Application Honours with Research:
   BCA (Honours with Research)

Regulations:

NEP 2020 based CBCS regulations of the University for the four year undergraduate programmes will be strictly followed for the continuous and end semester assessment.

Specialization:

The students can opt either Data Science or AI & ML specialization during the seventh semester. Four baskets of professional electives are available for specializing in specific area such as Data Science, Artificial Intelligence and Machine Learning (AI & ML), Full Stack Development, Cyber Security.

Central University of Kerala

Department of Computer Science
BCA Honours with research

Programme Structure

Semester: I

Sr. No.	Course Code	Course Title	Hours/week			Evaluation			Credit	Exam hours
			L	T	P	CA	ESA	Total		
1	25ECLUG1AEC01	Basic English Grammar and Usage	2	0	0	20	30	50	2	2
2	25HINUG1AEC02	Indian Language : Different forms of Hindi literature	2	0	0	20	30	50	2	2
	25MALUG1AEC02	Indian Language : Malayalam Language & Communication	2	0	0	20	30	50	2	2
	25KANUG1AEC02	Indian Language : General Kannada	2	0	0	20	30	50	2	2
3	25CSCUG1SEC01	Unix	0	0	4	20	30	50	2	2
4	25CSCUG1SEC02	Introduction to Generative AI	2	0	0	20	30	50	2	2
5	25CSCUG1MAJ01	Digital Principles and Computer Organization	3	0	0	40	60	100	3	3
6	25CSCUG1MAJ02	C Programming	3	0	0	40	60	100	3	3
7	25CSCUG1MAJ03	Discrete Mathematics	3	0	0	40	60	100	3	3
8	25CSCUG1MAJ04	C Programming Lab	0	0	4	20	30	50	2	2
9	25CSCUG1MAJ05	Python Programming Lab	0	0	2	50	-	50	1	-
TOTAL			**15**	**0**	**10**	**270**	**330**	**600**	**20**	**---**

Semester: II

Sr. No.	Course Code	Course Title	Hours/week			Evaluation scheme			Credit	Exam hours
			L	T	P	CA	ESA	Total		
1	25ECLUG2AEC01	English for Academic Purposes	2	0	0	20	30	50	2	2
2	25HINUG2AEC02	Indian Language : Hindi Grammar and Conversation	2	0	0	20	30	50	2	2
	25MALUG2AEC03	Indian Language: Malayalam Literature and Society	2	0	0	20	30	50	2	2

	25KANUG2AEC02	Indian Language: Creative Writing in Kannada Literature	2	0	0	20	30	50	2	2
3	25MANUG2SEC01	Personality Development and Professional Communication	2	0	0	20	30	50	2	2
4	25XXXUG2MDCXX	Multidisciplinary Course 1	3	0	0	40	60	100	3	3
5	25CSCUG2MAJ01	Data Structures	2	1	0	40	60	100	3	3
6	25CSCUG2MAJ02	Computational Mathematics	2	1	0	40	60	100	3	3
7	25CSCUG2MAJ03	Object Oriented Programming using Java	3	0	0	40	60	100	3	3
8	25CSCUG2MAJ04	Data Structures Lab	0	0	4	20	30	50	2	2
9	25CSCUG2MAJ05	Object Oriented Programming Lab	0	0	2	50	-	50	1	-
TOTAL			**16**	**2**	**6**	**290**	**360**	**650**	**21**	**---**

Multidisciplinary Courses offered by Department of Computer Science					
COURSE CODE	COURSE TITLE	Hours/Week			CREDITS
		L	T	P	
25CSCUG2MDC01	Python Programming	3	0	0	3
25CSCUG2MDC02	Introduction to Generative AI	3	0	0	3
25CSCUG2MDC03	Computational Biology	3	0	0	3

Courses offered for Indian Language in First Semester						
COURSE CODE	Semester	COURSE TITLE	Hours/Week			CREDITS
			L	T	P	
25HINUG1AEC02	1	Different forms of Hindi literature	2	0	0	2
25MALUG1AEC02	1	Malayalam Language and Communication	2	0	0	2
25KNDUG1AEC02	1	General Kannada	2	0	0	2

Courses offered for Indian Language in Second Semester						
COURSE CODE	Semester	COURSE TITLE	Hours/Week			CREDITS
			L	T	P	
25HINUG2AEC02	2	Hindi Grammar and Conversation	2	0	0	2
25MALUG2AEC03	2	Malayalam Literature and Society	2	0	0	2
25KNDUG2AEC02	2	Creative Writing in Kannada Literature	2	0	0	2

Syllabus

Semester 1

Major Core Courses

25CSCUG1MAJ01 Digital Principles and Computer Organization

L -T - P : 3 - 0 - 0 Credit : 3

Participatory and skill development course

Course Objectives: Understand the basic concepts, digital principles and organization of computers in sequential and parallel processing.

Course Outcomes:
CO1: Understand the basics of Digital Electronics and Binary Number System
CO2: Learn the implementation of Combinational Circuit and Sequential Circuits
CO3: Understand the organization of basic computers
CO4: Understand the concept of Parallel Processing
CO5: Understand the concept of memory organization

Prerequisite: Introductory course and no prerequisites required

Course Contents:

Module 1
Number Systems and Codes: Binary, Octal and Hexa decimal number system, ASCII code, Excess-3 Code, Gray code. Digital Logic: Basic Gates – NOT, OR, AND, Universal Logic Gates – NOR, NAND, Exclusive-OR Gates.
Module 2
Combinatorial Logic Circuits: Boolean Laws and Theorems, Sum of Products method, Product-of sums method. Karnaugh Map – Pairs, Quads, Octets – Don't Care Conditions, Data Processing Circuits: Multiplexers, Demultiplexers, Decoder, Encoders, Parity Generators.
Module 3
Arithmetic Circuits: Binary Addition, Binary Subtraction, 2'S Complement Representation, 2'S Complement Arithmetic, Arithmetic Building Blocks, Sequential Circuits, Flip flops: RS, JK, D, T and Master Slave flip flops.
Module 4
Basic Computer organization and Design: Instruction codes, stored program organization, Computer registers and common bus system, Computer instructions, Instruction cycle: Fetch and Decode, Micro programmed Control: Control memory organization.
Module 5
Central Processing Unit: General register organization, instruction formats, addressing modes, Data transfer and manipulation. CISC and RISC, Parallel processing, Pipelining. Input-output organization: Peripheral devices, I/O interface, Memory organization: Memory hierarchy.
Text Book:
1. Digital Principles and Applications – Donald P Leach, Albert Paul Malvino, Goutam Saha, 8th

edition, McGraw-Hill Education, 3rd reprint 2015.
2. Computer System Architecture, M. Morris Mano, Pearson Education, 3rd edition., 2007

Books for Reference:

1. Digital design, R.Anantha Natarajan, PHI Learning, 2015.
2. Principles of digital Electronics, K.Meena, PHI Learning, 2013.
3. Digital Computer Fundamentals, Thomas C. Bartee TMH 2007.
4. Digital Circuits and Design, S. Salivahanan and S. Arivazhagan, Vikas Publishers, 2005.
5. Computer Organization and Architecture, V.Rajaraman and T.Radhakrishnan, PHI learning, 2015.
6. Computer Organization, Carl Hamacher, Zvonko Vranesic, Safwat Zaky, McGraw Hill Education, 2015.
7. Computer Organization and Architecture, Smruti Ranjan Sarangi, McGraw Hill Education.

25CSCUG1MAJ02 C Programming

L -T - P : 3 - 0 - 0 Credit : 3

Participatory, problem solving and skill development course

Course Objectives: Understand the various steps in program development and to develop applications in C using programming constructs for solving problems.

Course Outcomes:

- Decompose a problem into functions and to develop modular reusable code.
- Usage of arrays, pointers, strings and structures to write C programs.
- Develop C programs for any given problem
- Perform input and output operations using programs in C
- Develop simple applications in C using basic constructs

Prerequisite: Introductory programming course and no prerequisites required

Course Contents:

Module 1 BASICS OF C PROGRAMMING

Introduction to C: Introduction, structure of C program, writing compiling and executing first C program, character set in C, keywords, identifiers, basic data types, variables, constants, formating input and output, operators in C, type conversion and typecasting.

Decision control and Looping: Introduction to Decision Control Statements, conditional Branching Statements, iterative Statements, nested Loops, the Break and Continue Statements, goto Statement.

Module 2 ARRAYS AND STRINGS

Arrays: Introduction, declaration of arrays, accessing the elements of an array, storing values in arrays, operations on arrays. Two-dimensional Arrays: Declaring two-dimensional arrays, initializing two-dimensional arrays, accessing the elements of 2D arrays, operations on two-dimensional arrays, multidimensional Arrays.

Strings: Introduction, operations on strings, miscellaneous string and character functions.

Module 3 FUNCTIONS AND POINTERS

Functions: Introduction, modular programming - Function prototype, function definition, passing parameters to functions, function call, Built-in functions (string functions, math functions) – Recursion, Binary Search using recursive functions.

Pointers: Pointer operators – Pointer arithmetic – Arrays and pointers – Array of pointers – Parameter passing: Pass by value, Pass by reference.

Module 4 STRUCTURES, UNIONS AND FILES

Structures and Unions : Introduction to structures, nested structures, array of structures, structures and functions, self referential structures, dynamic memory allocation, singly linked list, typedef, Introduction to unions, storage classes.

Files: Modes of file operations, Reading, writing and updating contents in a file, Ftell, Fseek.

TEXT BOOKS:

1. ReemaThareja, "Programming in C", Oxford University Press, Second Edition, 2016.
2. Kernighan, B.W and Ritchie,D.M, "The C Programming language", Second Edition, Pearson Education, 2015.
3. C: The Complete Reference, By Herbert Schildt.

REFERENCES:

1. E. Balagurusamy, Computer fundamentals and C, McGraw-Hill
2. Yashavant Kanetkar, Let Us C, BPB.
3. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill

25CSCUG1MAJ03 Discrete Mathematics

L -T - P : 3 - 0 - 0 Credit : 3

Participatory, problem solving and skill development course

Course Objectives: To provide theoretical aspects of mathematically representing real world problems and digitally modelling it.

Course Outcomes:

- Understand the principles of logic and proof techniques
- Acquire the preliminary concept of counting, permutation, and combination.
- Understand the fundamental mathematical structures such as sets, relations, and functions
- Familiarize the basic characteristics of graph theory and its real-life applications.

Prerequisites: Basic knowledge in logical skill and mathematics

Course Contents:

Module 1

Mathematical Logic and Proofs: Propositional logic, propositional equivalences, truth tables, deduction, resolution, normal forms, predicates and quantifiers, arguments, rules of inference, introduction to proofs, theorem proving.

Module 2

Basic Structures: Sets, set operations, functions, sequences and summations, cardinality of sets, Matrices.

Relations: Relations and their properties, n-ary relations and their applications, representing relations, closures of relations, equivalence relations, partial orderings.

Module 3

Number Theory and Cryptography: Divisibility and modular arithmetic, integer representations and algorithms, primes and greatest common divisors, solving congruences, cryptography .

Counting: The basics of counting, the pigeonhole principle, permutations and combinations.

Module 4

Graph Theory: Graphs and graph models, graph terminology and special types of graphs, representing

graphs and graph isomorphism, connectivity, Euler and Hamilton paths, shortest-path problems.

Text Books:

- Kenneth H. Rosen, Discrete Mathematics and its Applications, 8th Edition, Tata McGraw Hill, 2019.

Reference Books

1. Kolman, R.C. Busby and S.C. Ross, Discrete Mathematical Structures, Pearson, 2018
2. J.P. Trembley and R. Manohar, Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw Hill, 2017.
3. C.L. Liu, D. Mohapatra, Elements of Discrete Mathematics – A Computer Oriented Approach, Tata McGraw Hill, 2017
4. R.P. Grimaldi, Discrete and Combinatorial Mathematics, Pearson Education, 2007.

25CSCUG1MAJ04 C Programming Lab

L -T - P : 0 - 0 - 4 Credit : 2

Experimental, problem solving and skill development course

List of exercises

Implement the following programs using C language

1. a) Program to convert degrees Celsius to Fahrenheit and vice versa
   b) Program to read radius of a circle and to find area and circumference
2. a) Display three input numbers in sorted (non-decreasing) order
   b) Program to read percentage of marks and to display appropriate message
3. Program to find the roots of quadratic equation (demonstration of switch case statement)
4. Display Pascal triangle
5. Given the first term, difference/multiplier and number of terms (n > 0), display the first n terms of the arithmetic/geometric progression?
6. Display the first n (n > 0) terms of the Fibonacci sequence
7. Write a program to find the sum of digits of a number
8. Compute character grade from the marks (0 ≤ marks ≤ 100) of a subject using both else-if ladder and switch case (Grading Scheme: 80-100 : A, 60 - 79: B, 50 - 59: C, 40-49: D, 0-39: F)
9. Check if a given positive integer number is a prime number or not?
10. Compute prime factors of a positive integer number
11. Program to perform multiplication of given two matrices
12. Program to do the arithmetic operations available in a calculator by passing the values through Command line arguments.
13. Program to read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters.
14. Program to check whether a given string is palindrome or not.
15. Program to accept line of text and find the abbreviation by printing the first character of each word.
16. a) Program to reverse a string using pointer
    b) Program to swap two numbers using pointers
17. Program to demonstrate student structure to read and display records of n students. Display the rank list also.
18. Exercise Programs on functions and recursion.

a) Define a function that takes three numbers as arguments and returns the smallest of them.

b) Write a program which makes use of function to display all such numbers which are divisible by 7 but are not a multiple of 5, between given range.

c) Define functions to find mean, median, mode for the given numbers.

d) Define a function which generates Fibonacci series up to n numbers.

e) Write a program to find factorial of number by using recursion.

19. Program to find ${}^{N}C_{r}$ using function
20. Program to write a file for storing the employee details.
21. Program to reverse each word available in a text file.
22. Program to read a random line from a text file.
23. Program to check whether a given number is already available in a file and to replace it with another number.
24. Write a program to get the marks of three different courses and store it in a file and make provision to update the marks of students after publishing the revaluation results.

25CSCUG1MAJ05 Python Programming Lab

L -T - P : 0 - 0 - 2 Credit : 1

Experimental, problem solving and skill development course

List of exercises

1. Demonstration on installation of Python software and different IDEs to write, compile, debug and execute python programs.
2. Write a program to Demonstrate the following Operators in Python with suitable examples.
   i) Arithmetic Operators ii) Relational Operators
   iii) Assignment Operator iv) Logical Operators
   v) Bit wise Operators vi) Ternary Operator
   vii) Membership Operators
3. Write program to demonstrate use of conditional statements for displaying the rank list of students.
   a) 'if' statement b) 'if … else' statement c) Nested 'if' statement
4. Write programs to demonstrate the following statements in Python for printing the odd numbers except a particular number entered by the user.
   a) while loop b) for loop c) break d) continue
5. Exercise Programs on Lists.
   a) Write a Python script to display elements of the list in reverse order
   b) Write a Python script to find the minimum and maximum elements without using built-in operations in the lists.
   c) Write a Python script to remove duplicates from a list.
   d) Write a Python script to create a list with different data types.
6. Exercise Programs on Tuples
   a) Write a Python script to find the repeated items of a tuple.
   b) Write a Python script to create a tuple with different data types
   c) Write a Python script to replace last value of tuples in a list with element 100.
      Sample list: [(10, 20, 40), (40, 50, 60), (70, 80, 90)]
      Expected Output: [(10, 20, 100), (40, 50, 100), (70, 80,100)]
   d) Write a Python script to sort a tuple by its float element.

Sample data: [('item1', '12.20'), ('item2', '15.10'), ('item3','24.5')]
Expected Output: [('item3', '24.5'), ('item2', '15.10'), ('item1', '12.20')]

7. Exercise Programs on Sets and Dictionaries.
   a) Write a Python script to add members in a set.
   b) Write a Python script to perform Union, Intersection, difference, and symmetric difference of given two sets.
   c) Write a Python script to check whether a given key already exists or not in a dictionary.
   d) Write a Python script to concatenate following dictionaries to create a new one.
   e) Sample Dictionary: dic1={1:10,2:20} dic2={3:30, 4:40} dic3={5:50,6:60} Expected Result: {1: 10, 2: 20, 3: 30, 4:40, 5: 50, 6: 60}
   f) Write a Python program to map two lists into a dictionary.
8. Exercise Programs on functions and recursion.
   f) Define a function that takes three numbers as arguments and returns the largest of them.
   g) Write a program which makes use of function to display all such numbers which are divisible by 7 but are not a multiple of 5, between given range.
   h) Define functions to find mean, median, mode for the given numbers.
   i) Define a function which generates Fibonacci series up to n numbers.
   j) Implement a python script for factorial of number by using recursion.
9. Exercise programs on Date and Time Modules.
   a) Write a Python script to get the current time in Python.
   b) Write a Python script to print next 5 days starting from today.
10. Exercise programs on Exception Handling.
   a) Write a Python script to handle simple errors by using exception handling mechanism
   b) Write a Python script to handle multiple errors with one except statement.
   c) Write a program in Python to handle user defined exception for given problem
11. Exercise programs on Strings
   a) Implement Python Script to perform various operations on string using string libraries.
   b) Implement Python Script to check given string is palindrome or not.
   c) Implement python script to accept line of text and find the number of characters, number of vowels and number of blank spaces in it
   d) Implement python script that takes a list of words and returns the length of longest one.
12. Exercise programs on Files
   a) Write a Python program to read an entire text file.
   b) Write a Python program to append text to a file and display the text.
   c) Write a Python program to read a file line by line store it into an array.
   d) Write a python program to find the longest words.
   e) Write a Python program to read a random line from a file.
13. Exercise programs on Regular Expressions
   a) Write a Python script to check that a string contains only a certain set of characters (in this case a-z, A-Z and 0-9).

b) Write a Python script to check whether password is valid or not. Conditions for a valid password are: Should have at least one number. Should have at least one uppercase and one lowercase character. Should have at least one special symbol. Should be between 6 to 20 characters long.

14. Exercise programs on Object Oriented Programming.

a) Write a Python Program for student information system and displaying of rank list of passed students using class and objects.

b) Write a Python Program for calculating the bill of consultation fee, medicine charges and room rent of all the patients in a hospital using class and objects.

c) Write a Python script to implement inheritance for employee registration having different designation, salary calculation methods and display the computed salary with employee name.

Programme Specific Skill Enhancement Course

25CSCUG1SEC01 UNIX

L -T - P : 0 - 0 - 4 Credit : 2

Introductory, problem solving and skill development course

COURSE OBJECTIVES:

- Execute the basic commands, directory and file related, pipe and filter related commands on the shell.
- Implement shell scripts using this editor involving decision control, looping and control flow statements.

COURSE OUTCOMES:

- Understand the basic commands and shell scripts
- Develop programs to perform numerical calculations and text processing
- Set up the permissions to access the files and directories

Prerequisite: Introductory course and no prerequisites required

List of exercises

1. Basic UNIX Commands

- pwd, ls, cd, mkdir, rmdir, touch, rm, cp, mv
- Hidden files, absolute vs relative paths

2. File Permissions and Ownership

- chmod, chown, chgrp
- File modes (read, write, execute)
- Umask and permission bits

3. Text File Manipulation

- cat, more, less, head, tail, wc
- nano, vim, or vi editors

4. Process Handling

- ps, top, kill, nice, renice
- Background & foreground jobs: &, jobs, fg, bg

5. Filters and Redirection

- grep, cut, sort, uniq, tr, sed, awk

- Piping (|), redirection (>, >>, <, 2>)

6. Basic Shell Scripting for simple programs

- Writing simple bash scripts
- Variables, conditionals (if, case)
- Loops (for, while, until)
- Input/output in scripts

7. Advanced Shell Scripting

- Functions in bash
- Command-line arguments ($1, $2, …)
- Debugging scripts (set -x, bash -x)

8. User and Group Management

- adduser, passwd, usermod, groups, su, sudo

9. Scheduling Tasks

- cron, crontab, at, batch

10. Archiving and Compression

- tar, gzip, gunzip, zip, unzip

11. Networking Commands

- ping, netstat, ifconfig or ip, scp, ssh, wget, curl

12. Regular Expressions in grep, sed, awk

- Pattern matching, substitutions, data extraction

13. Compiling C Programs in UNIX

- Using gcc, makefiles, and debugging with gdb

14. Installation & Basic Setup

- Install a UNIX/Linux OS (e.g., Ubuntu, Fedora)
    - Creating bootable media (USB/DVD)
    - Dual boot or VM installation (using VirtualBox/VMware)
    - Basic terminal usage
    - Understanding file system hierarchy (/, /home, /etc, etc.)

Text Books/References:

1. Yashwant Kanethkar .Unix Shell programming. By BPB publications, 2005.
2. Sumitabha Das. UNIX Concepts and Applications. Tata McGraw-Hill Education, 2006.
3. Brian W. Kernighan ,Rob pike “The Unix programming environment”, Prentice Hall of India Private Ltd.2004.

General Skill Enhancement Courses

25CSCUG1SEC02 Introduction to Generative AI

L -T - P : 2 - 0 - 0 Credit : 2

Introductory and skill development course

Course Objective: The objective of the course is to provide theoretical and mathematical aspects of using Generative AI.

Course outcomes:

1. Knowledge to be gained:
   (i) fundamental concepts of Generative AI.
   (ii) Influence of generative AI in real world applications.
2. Skill to be gained:

(iii) Skills in the usage of Generative AI.
(iv) Critical analyzing and logic skills in the usage of Generative AI.

3. Competency to be gained:
   (v) Generative AI applications in real world problem.

Prerequisites: Basic knowledge in logical skill and mathematics

Course Contents:

Module 1

Introduction and scope of Generative AI. Overview of generative models and their applications. Importance of Generative AI in various domains. Basics of Artificial Intelligence (AI). Introduction to Neural Networks. Basics of Machine Learning. Supervised, Unsupervised and Reinforcement Learning.

Module 2

Basics of Deep Learning, Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM). Attention Mechanism, Transformer Model, Generative Adversarial Networks (GAN), Introduction to Prompt Engineering. Generative AI Models, Text Generation Models: GPT-3, GPT-4.

Module 3

Large Language Model Meta AI (LLaMA), Claude, Gemini, Grok. DALL-E AI image generator, AI voice generators, AI video generators, Multimodal AI models: CLIP, Flamingo. Common Challenges in Large Language Model : Hallucination, Ambiguity in Prompt Design, Bias and Fairness.

Text Books/References:

1. David Foster, Generative Deep Learning, O'Reilly Media, 2019.
2. Ian Goodfellow, Yoshua Bengio, and Aaron Courville , Deep Learning, MIT Press, 2016.
3. Yoav Goldberg, Morgan & Claypool Publishers, Neural Network Methods for Natural Language Processing, 2017.
4. Cathryn van Kessel, Christopher H. Clark, AI in Social Studies Education, Teachers College Press, 2025.
5. Arokiaraj David, Jeganathan Gomathi Sankar, Generative AI and Implications for Ethics, Security, and Data Management, IGI Global, 2024.

25CSCUG1SEC03 Digital Fluency

L - T - P : 2 – 0 - 0 Credit : 2

Introductory and skill development course

Course Objectives: To learn the terms and concepts used in the digital era and to identify the emerging technologies in the Information Technology field.

Course Outcomes:

CO1: Understand the fundamentals of computer and emerging technologies
CO2: Discover the arena of Internet and its possibilities
CO3: Effective usage of software applications and tools
CO4: Awareness of cyber security and attacks

Prerequisite: Introductory course and hence no prerequisites

Course Contents:

Module 1 : Fundamentals of Computers

Components of a computer system, System software & Application Software, Operating System, Problem Solving Steps, Computer Network, Web Server, Client Server Computing, Distributed system.

Module 2: Emerging technologies & Applications

Artificial Intelligence, Machine Learning, Deep Learning, Database management for Data Science and Big data Analytics, Internet of Things (IoT), Industrial Internet of Things (IoT), Cloud computing and its service models, Cyber security and types of Cyber Attacks.

Module 3: Skill Enhancement using technologies

Effective Communication Skills, Creative problem solving and critical thinking, Collaboration and Team work skills, Innovation & Design thinking, Use of tools in enhancing skills.

Text Books/References:

1. Rajaraman V, Neeharika Adabala, Fundamentals of Computers, PHI Learning Pvt. Ltd**.,** 2014.
2. S. P. SajjanS. B. Ramoshi, Digital Fluency, Ekalavya E-educate, 2021.
3. Volker Lang, Digital Fluency: Understanding the Basics of Artificial Intelligence, Blockchain Technology, Quantum Computing and Their Applications for Digital Transformation, Apress, 2021.
4. Christian Briggs, Kevin Makice, Digital Fluency: Building Success in the Digital Age, 2012.

Semester 2
Major Core Courses

25CSCUG2MAJ01 Data Structures

L -T - P : 3 - 0 - 0 Credit : 3

Introductory, problem solving and skill development course

Course objectives: Understand fundamental data structures and algorithms and to develop problem solving skills using data structures.

Course Outcomes:

- Understand basic concepts of data structures
- Apply data structures to solve problems effectively
- Analyze and implement various nonlinear data structures
- Compare alternative implementations of data structures with respect to performance

Prerequisite: Introductory course and hence no prerequisites required

Course Contents:

Module 1: Introduction to Data Structures:

Algorithms and flowcharts, basic analysis of algorithms, complexity of algorithms, introduction and definition of data structures, classification of data structures, arrays, static and dynamic memory allocation.

Module 2: Linked lists

Linked lists: Basic Concepts: Definition and representation of linked list, types of linked lists: singly linked list, doubly liked list, header liked list, circular linked list, representation of linked list in memory. Operations on Singly linked lists: Traversing, searching, insertion, deletion, memory allocation, garbage collection.

Module 3: Stacks and Queues

Stacks: Basic concepts – Definition and representation of stacks, operations on stacks. Applications of stacks: Infix, postfix and prefix notations, conversion from infix to postfix using stack, evaluation of postfix expression using stack. **Queues:** Basic concepts – Definition and representation of queues, types of queues: Simple queues, circular queues, double ended queues, priority queues, operations on simple queues.

Module 4: Trees and Graphs

Trees: Basic terminology, binary trees, binary tree representation, binary tree traversals, Applications of trees. **Graphs:** Terminology and Representation, graph traversals-breadth first search, depth first search, spanning trees, shortest paths.

TEXT BOOKS:

1. Ellis Horowitz and Sartaj Sahni: Fundamentals of Data Structures
2. Seymour Lipschutz, "Data Structures with C", Schaum's outLines, Tata Mc Graw Hill, 2011

REFERENCES:

1. M. A. Weiss, Data Structures and Algorithm Analysis in C, Pearson Education Asia, 2002.
2. Tanenbaum: Data structures using C, Pearson Education.
3. Kamathane: Introduction to Data structures, Pearson Education.
4. Y. Kanitkar: Data Structures Using C, BPB.

25CSCUG2MAJ02 Computational Mathematics

L -T - P : 2 - 1 - 0 Credit : 3

Introductory, problem solving and skill development course

Course Objective:

The objective of the course is to provide theoretical and practical aspects of mathematically representing real world problems and digitally modelling it.

By completing this course, students will obtain the following course outcomes:

1. Knowledge to be gained:
   (i) Fundamental concepts of computational mathematics
   (ii) Representation of real world problems into computational algorithms
   (iii) Skills in representation data and implementation of mathematical concepts on computers
   (iv) Influence of data representation on computers on numerical algorithms.
2. Skill to be gained:
   (v) Skills in representation of data and implementation of mathematical concepts using AI related toolbox/packages in Python/MATLAB
   (vi) Critical analyzing and logic skills in developing computational algorithms.
3. Competency to be gained:
   (vii) Computational modelling of any real world problem

Prerequisite: Basic knowledge in logical skill and mathematics

Course Contents:

Module 1

Mathematical Statistics – Concepts of Probability and Random Variables, Classical Relative Frequency and Axiomatic Definition of Probability, Addition Rule, Conditional Probability, Multiplication Rule, Bayes Rule, T Test, χ^2 Test

Module 2

Solution of Algebraic and Transcendental Equations - Bisection method, Regula – Falsi Method, Newton_ Raphson method, Solution of Linear System of Equations and Matrix Inversion – Gaussian Elimination Method, Jacobi's Method, Gauss – Seidel Iteration Method, Eigen Value Problems – Power Method.

Module 3

Interpolation – Lagrange's Interpolation Formulae, Newton's Forward Difference Interpolation Formula, Numerical Differentiation and Integration – Trapezoidal Rule, Simpson's Rules, Ordinary Differential Equations – Euler Method, Runge-Kutta Methods. Any one of the finite difference schemes for partial differential equations.

Module 4

Graph theory – Introduction to Graphs, Subgraphs, Trees and Traversals, Connectivity, Network Flows, Shortest Paths, Graph Applications.

Text Books/References:

1. Md. Masud Rana, Wei Xu, Youguang Guo, Fundamentals of Computational Methods for Engineers, Bentham Books, 2022
2. Steven C. Chapra, Raymond P. Canale, Numerical Methods for Engineers, Mc Graw Hill Education, 2015
3. Timmy Siauw, Alexandre M. Bayen, An Introduction to MATLAB® Programming and Numerical Methods for Engineers, Academic Press, Elsevier, 2014
4. K. Erciyes, Discrete Mathematics and Graph Theory – A Concise Study Companion and Guide, Springer, 2021.

25CSCUG2MAJ03 Object Oriented Programming Using JAVA

L-T-P : 3-0-0 Credit : 3

Introductory, problem solving and skill development course

Course Objectives:

To learn the object oriented programming concepts to find the solutions of real world problems.

Course Outcomes:

- Understand the principles of object oriented programming (OOP).
- Develop proficiency in writing JAVA programs with core OOP concepts.
- Apply object oriented programming concepts in solving real time scenarios.
- Use exception handling and multithreading mechanism to create efficient applications.

Prerequisite: Introductory course and hence no prerequisites required

Course Contents:

Module 1

Overview of Object oriented programming, Basics of JAVA, Variables, Constants, Keywords, Datatypes, Operators, Type Conversion and Casting, Enumerated Types, Control Statements, Loops, Arrays.

Module 2

Class, Object, Methods, Overloading, Constructors, Static Members, this Keyword, Access Control, Garbage Collection, String Class, String Tokenizer.

Module 3

Inheritance, Types, Method Overriding, Super, Final, Abstract Classes, Interfaces, Extending and Implementing Interfaces, Applying Interfaces, Accessing Interface Variables, Packages, Creating, Defining and Accessing Packages, Importing Packages.

Module 4

Concepts of Exception, Types, Exception Handling Techniques, Built-in Exceptions, User Defined Exception, Concepts of thread, Creating a thread, Thread Life Cycle, Multithreading, Synchronizing Threads.

Text Books:

- Herbert Schildt, Java the Complete Reference, 13th Edition, McGraw Hill, Education, 2024.

Text Books/References:

1. E. Balagurusamy, Programming with Java, McGraw Hill Education, 2019.
2. Herbert Schildt, Coward Danny, Java: A Beginner's Guide, 2024.
3. **B**ruce Eckel, Thinking in Java, Prentice Hall, 2006.
4. Kathy Sierra, Bert Bates, Head First Java, O'REILLY, 2009.

25CSCUG2MAJ04 Data Structures Lab

L -T - P : 0 - 0 - 4 Credit : 2

Experimental, problem solving and skill development course

List of Exercises

Data structures through C lab experiments

1. Write a C program to find duplicate values in a list of values
2. Write a C program to reverse a given number
3. Write a C program to implement single linked list and its operations
4. Write a C program to implement Stack and its operations using arrays
5. Write a C program to implement Stack and its operations using linked list
6. Write a C program to evaluate postfix expression through stack using arrays
7. Write a C program to implement Queue and its operations using arrays
8. Write a C program to implement Queue and its operations using linked list
9. Write a C program to implement priority Queues and its operations using arrays
10. Write a C program to implement Binary tree and its operations using arrays
11. Write a C program to implement Binary tree and its operations using arrays
12. Write a C program to implement Binary tree and its operations using linked list
13. Write a C program to implement Binary tree traversal techniques using arrays
14. Write a C program to implement Graph traversal techniques using arrays

REFERENCES:

1. Ellis Horowitz and Sartaj Sahni: Fundamentals of Data Structures
2. Seymour Lipschutz, "Data Structures with C", Schaum's outLines, Tata Mc Graw Hill, 2011
3. M. A. Weiss, "Data Structures and Algorithm Analysis in C", 2nd ed, Pearson Education Asia, 2002.
4. Tanenbaum: Data structures using C, Pearson Education.
5. Kamathane: Introduction to Data structures, Pearson Education.
6. Y. Kanitkar: Data Structures Using C, BPB.

25CSCUG2MAJ05 Object Oriented Programming Lab

L -T - P : 0 - 0 - 2 Credit : 1

Experimental, problem solving and skill development course

List of Exercises

1. Write a program to read two numbers from user and print their sum and product.
2. Write a program to send the name and surname of a student through command line arguments and print a welcome message for the student.
3. Write a java program to find the Fibonacci series & Factorial of a number using recursive and non-recursive functions.
4. Write a java program to multiply two given matrices.
5. Write a Java program that checks whether a given string is a palindrome or not.
6. Write a Java program to perform mathematical operations. Create a class called AddSub with methods to add and subtract. Create another class called MulDiv that extends from AddSub class to use the member data of the super class. MulDiv should have methods to multiply and divide. Demonstrate the divide by zero exception handling also. A main function should access the methods and perform the mathematical operations.
7. Create a JAVA class called Student with details such as USN, NAME, BRANCH, PHONE, PERCENTAGE. Write a JAVA program to create n Student objects and print the USN, Name, Branch, Phone, and percentage of these objects with suitable headings.
8. Write a Java program that displays the number of characters, lines and words in a text.
9. Write a Java program to create a class called Shape with methods called getPerimeter() and getArea(). Create a subclass called Circle that overrides the getPerimeter() and getArea() methods to calculate the area and perimeter of a circle.
10. Write a Java program to create a class Employee with a method called calculateSalary(). Create two subclasses Manager and Programmer. In each subclass, override the calculateSalary() method to calculate and return the salary based on their specific roles.
11. Write a Java program using an interface called 'Bank' having function 'rate_of_interest()'. Implement this interface to create two separate bank classes 'SBI' and 'PNB' to print different rates of interest. Include additional member variables, constructors also in classes 'SBI' and 'PNB'.
12. Write a Java program for finding the cube of a number using a package for various data types and then import it in another class and display the results.
13. Write a Java program that reads a list of integers from the user and throws an exception if any numbers are duplicates.
14. Create an exception subclass UnderAge, which prints "Under Age" along with the age value when an object of UnderAge class is printed in the catch statement. Write a class exceptionDemo in which the method test() throws UnderAge exception based on the variable age passed to it.

Multidisciplinary Courses

25CSCUG2MDC01– Python Programming

L -T - P : 3 - 0 - 0 Credit : 3

Problem solving and employability based skill development course

Course Objective:

The objective of the course is to provide theoretical and practical aspects of programming concepts using python.

By completing this course, students will obtain the following course outcomes:

1. Knowledge to be gained:
   (i) Interpret the fundamental Python syntax and the usage of Python Control flow statements.
   (ii) Proficiency in the handling of strings and functions.
   (iii) Methods to develop Python programs by utilizing the data structures like lists, tuples and sets.
   (iv) Identify the commonly used operations in file systems.
   (v) Articulate the Object-Oriented Programming concepts such as encapsulation, inheritance and polymorphism.
2. Skill to be gained:
   (vi) Problem solving and programming capability using python
3. Competency to be gained:
   (vii) Design and implement of programs using python to solve real world problems

Prerequisites: Nil

Course Content:

Module 1

Introduction to Python, Basic Syntax, Variables, Data Types, Operators, Understanding python blocks. Conditional Statements, Looping, and Control Statements.

Module 2

Strings and Number System, String Methods, Basic String Operations, String Slicing, Searching and Manipulating Strings. Introduction to Lists, List slicing, Copying Lists, Processing Lists, List Methods, Tuples, Basic operations of Tuples.

Module 3

Introduction to Files, Processing files and records, Inbuilt functions, User defined functions, Local Variables, Global Variables, Recursion. Generating Random Numbers.

Module 4

Introduction to object oriented concepts, Classes and Objects, Class attributes, Instance attributes, constructor, Inheritance, Function polymorphism and class polymorphism.

Text Books:

1. Kenneth A. Lambert, The Fundamentals of Python: First Programs, Cengage Learning, 2011.
2. Think Python Second Edition, by Allen B. Downey, Orielly publishing, 2015
3. Introduction to Computation and Programming Using Python. John V. Guttag, The MIT Press,2016.

25CSCUG2MDC02 Introduction to Generative AI

L -T - P : 3 - 0 - 0 Credit : 3

Participatory and multidisciplinary skill development course

Course Objective: The objective of the course is to provide theoretical and mathematical aspects of using Generative AI.

Course outcomes:

1. Knowledge to be gained:
   (i) fundamental concepts of Generative AI.
   (ii) Influence of generative AI in real world applications.
2. Skill to be gained:
   (iii) Skills in the usage of Generative AI.
   (iv) Critical analyzing and logic skills in the usage of Generative AI.
3. Competency to be gained:
   (v) Generative AI applications in real world problem.

Prerequisites: Basic knowledge in logical skill and mathematics

Course Content:

Module 1

Introduction to Generative AI. Scope of Generative AI. Overview of generative models and their applications. Importance of Generative AI in various domains. Brief discussion on ethical considerations and challenges. Basics of Artificial Intelligence (AI). Introduction to Neural Networks.

Module 2

Basics of Machine Learning (ML), Types of ML: Supervised, Unsupervised and Reinforcement Learning.

Deep Learning: Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), LSTMs, Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), Attention Mechanism, Transformers.

Module 3

Introduction to Prompt Engineering. Generative AI Models and Architectures. Text Generation Models: GPT-3, GPT-4, LLaMA, Claude, Gemini, Falcon, Grok, Cohere, Midjourney, FLAN-T5, BERT. Generative Codes using Co-Pilot.

Module 4

DALL-E AI image generator, AI voice generators, AI video generators, Multimodal AI models: CLIP, Flamingo. Common Challenges in LLM challenges: Hallucination, Ambiguity in Prompt Design, Bias

and Fairness.

Text Books/References:

1. Generative Deep Learning" by David Foster, O'Reilly Media, 2019.
2. Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press, 2016.
3. Neural Network Methods for Natural Language Processing" by Yoav Goldberg, Morgan & Claypool Publishers, 2017.
4. AI in Social Studies Education, Cathryn van Kessel, Christopher H. Clark, Teachers College Press, 2025.
5. Generative AI and Implications for Ethics, Security, and Data Management, Arokiaraj David, Jeganathan Gomathi Sankar, IGI Global, 2024.

25CSCUG2MDC03 Computational Biology

L -T - P : 3 - 0 - 0 Credit : 3

Participatory and multidisciplinary skill development course

Course Objective: The objective of the course is to provide knowledge about the computational concepts in Biology.

Course outcomes:

1. Knowledge to be gained:
   (i) Fundamental concepts of Biology.
   (ii) Influence of technology and computational methods in the applications of Biology.
2. Skill to be gained:
   (iii) Skills in the usage of biological data and databases.
   (iv) Logic skills in the classification and Information retrieval from biological databases.
3. Competency to be gained:
   (v) Usage of the applications of network biology in biological systems.

Prerequisites: Nil

Course Content:

Module 1:

Introduction to Computational Biology, scope and importance of computational approaches in biology, interdisciplinary nature of computational biology, applications in genomics, proteomics, and drug discovery. Basic introduction to molecular docking and its significance.

Module 2:

Types of biological data and databases. Classification and features of biological databases. Sequence and molecular file formats and basic sequence conversion tools. Primary and secondary sequence databases. Information retrieval from biological databases. Genome, protein, and disease databases. Overview of NCBI resources and the Entrez system.

Module 3:

Introduction to molecular biology – Nucleic acids – DNA – RNA – Proteins, Gene –Genome – Gene expression – Translation – Transcription – Protein synthesis, Chromosomes – Maps and sequences. Amino acids as the fundamental components of proteins, classification and

properties of amino acids, levels of protein structure.

Module 4:

Graph theory: basics, types of graphs; computational representations of graphs - data structures, adjacency matrix. Representation of biological networks using graph: gene regulatory networks; protein structure networks; metabolic networks. Software tools for network analysis. Applications of network biology in understanding protein structure and biological systems.

Text Books

1. Xiong.Jin. *Essentaial bioinformatics.* Cambridge University Press, 2006.
2. Singh, G. B., Fundamentals of Bioinformatics and Computational Biology, Springer, 2015.
3. Jones, N. C., & Pevzner, P. A., An Introduction to Bioinformatics Algorithms, MIT Press, 2004.

References

4. Pevzner, P. A., & Compeau, P., Bioinformatics Algorithms: An Active Learning Approach, Active Learning Publishers, 2015.
5. Lesk, A. M., Introduction to Bioinformatics, Oxford University Press, 2014.
6. Mount, D. W., Bioinformatics: Sequence and Genome Analysis, Cold Spring Harbor Laboratory Press, 2004.
7. Baxevanis, A. D., & Ouellette, B. F. F., Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins, Wiley-Interscience, 2005.