File Ref.No.Ac.A3/Faculty of Technology/2023

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

(Abstract)

Faculty of Technology - Programmes offered at the Department of Computer Science - Resolution of the Academic Council - Communicated - Orders issued.

ACADEMIC A SECTION

No.CUSAT/AC(A).A3/4081/2024 Dated,KOCHI-22,26.09.2024

Read:-Item No. I (f) (8) of the minutes of the meeting of the Academic Council held on 30.04.2024

ORDER

The Academic Council considered along with the recommendations of it's standing committee, the Minutes of the Faculty of Technology held on 08.04.2024 and resolved to approve the following :

i. Revised Scheme and Syllabus of M.Tech Computer Science & Engineering with Specialization in Data Science and Artificial Intelligence (Full-Time Programme) with effect from 2024 admissions (Appendix I).

ii. Revised Scheme and Syllabus of M.Tech Computer Science & Engineering (Artificial Intelligence and Software Engineering) (Full-Time Programme) with effect from 2024 admissions (Appendix II).

iii. Revised Scheme and Syllabus of M.Tech Computer Science & Engineering (Data Science and Artificial Intelligence) (Part-Time Programme) with effect from 2025 admissions (Appendix III).

iv. Revised Scheme and Syllabus of Five Year Integrated M.Sc in Computer Science (Artificial Intelligence and Data Science) (Appendix IV).

v. Amendment to clause 9.4 of M.Tech Regulation, 2020 as,

9.4(a) A student of the Full-Time M.Tech programme shall acquire a minimum of 28 credits in the first two semesters before he/she registers for the third semester.

9.4(b) The minimum credit requirement mentioned in 9.4(a) is not applicable to the Part-Time M.Tech programme.

9.4(c) A student shall complete the M.Tech programme in 8 consecutive semesters in the case of Full-Time programmes and 10 consecutive semesters in the case of Part-Time programme by acquiring a minimum total credit requirement of 72.

vi. Awarding of B.Sc and M.Sc degree to 2021 & 2022 batch students:

To award B.Sc in Computer Science Degree to those students who successfully completes 6 semesters and to award M.Sc (Five Year Integrated) in Computer Science (Artificial Intelligence and Data Science) to those whose successfully completes 10 semesters. This is applicable to 2021 & 2022 admissions only.

Orders are issued accordingly.

Dr. Arun A U *
Registrar

To:

1. The Dean, Faculty of Technology
2. Chairperson, BoS under Faculty of Technology
3. The Head, Department of Computer Science
4. All AR/DR Examination wing - with a request to forward to concerned sections
5. The Director, IQAC/ DoA
6. CIRM/Conference Sections
7. PS To VC/PVC;PA To Registrar/CE.

* This is a computer generated document. Hence no signature is required.

DEPARTMENT OF COMPUTER SCIENCE
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

PROGRAMME STRUCTURE & SYLLABUS
[2024 ADMISSIONS ONWARDS]

Five Year Integrated M.Sc. in Computer Science (Artificial Intelligence & Data Science)

Vision

- To globally excel in innovative research, teaching, and technology development inspired by social obligation.

Mission

- To contribute to knowledge development and dissemination.
- To facilitate learning and innovative research in frontier areas of computer science.
- To drive students for technology development to solve problems of interest.
- To create socially responsible professionals.

Program Outcomes: Integrated M.Sc.

PO1: Demonstrate a comprehensive understanding of fundamental principles and concepts in basic sciences.

PO2: Analyze, evaluate, and synthesize complex scientific information and data using appropriate methods and techniques.

PO3: Apply scientific reasoning and critical thinking adeptly to recognize, assess, and resolve problems encountered in various scientific contexts.

PO4: Utilize computational power, programming languages, and modern technologies proficiently to address scientific challenges, effectively integrating technological solutions into problem-solving processes.

PO5: Communicate scientific information effectively and demonstrate proficiency in the use of modern scientific tools and technologies for experimentation, data collection, analysis, and interpretation.

PO6: Adhere to ethical principles and practices in the conduct of scientific research and professional activities, and work collaboratively with others.

PO7: Engage in lifelong learning and professional development to enhance the knowledge and skills in basic sciences.

Programme Specific Outcomes

After the completion of Five Year Integrated M.Sc. in Computer Science (Artificial Intelligence & Data Science) programme, the students will have:

PSO 1: Deep understanding and proficiency in advanced topics such as Algorithms, Software Engineering, Artificial Intelligence and Data Science

PSO 2: Training in research methodologies, experimental design, and critical analysis to contribute to the advancement of computer science through original research

PSO 3: Specialization opportunities in areas such as Computer Vision, Natural Language Processing, Bioinformatics, Software Engineering, Cyber Security, and Cyber Physical Systems based on personal interests and career goals

PSO 4: Understanding the ethical implications of technology and the responsibility of computer scientists to develop and use technology in an ethical and society responsible way.

I. Academic programme pathways offered by the Department of Computer Science

A. Computer Science (Artificial Intelligence & Data Science) Major

1. **3-year UG Programme**: B.Sc. in Computer Science will be awarded to those who complete a three-year degree programme securing a minimum of 133 credits, out of which a minimum of 68 credits should be from Computer Science discipline and have satisfied the minimum course requirements as mentioned in Table No 1.
2. **4-year UG Programme (Honours with Research): B.Sc. (Honours with Research) in Computer Science (Artificial Intelligence & Data Science)** will be awarded to those who complete a four-year degree programme securing a minimum of 177 credits, out of which a minimum of 104 credits(including 12 credits Research Project) should be from Computer Science (Artificial Intelligence & Data Science) discipline and have satisfied the minimum course requirements as mentioned in Table No 1 and 2.
3. **4-year UG Programme (Honours): B.Sc. (Honours) in Computer Science (Artificial Intelligence & Data Science)** will be awarded to those who complete a four-year degree programme securing a minimum of 177 credits, out of which a minimum of 104 credits should be from Computer Science (Artificial Intelligence & Data Science) discipline and have satisfied the minimum course requirements as given in Table No 1 and 2.
4. **5-year Integrated PG Degree** : **Five Year Integrated M.Sc. in Computer Science (Artificial Intelligence & Data Science)** will be awarded to those who complete a five-year degree programme securing a minimum of 221 credits out of which a minimum of 148 credits should be from Computer Science (Artificial Intelligence & Data Science) discipline and have satisfied the minimum course requirements as given in Table No 1,2 and 3.

B. Minor Degree in Computer Science

1. **3-year UG Programme**: Minor Degree in Computer Science will be awarded to those who complete a three-year degree programme in any of the discipline mentioned in the FYUGP regulations and completed a minimum of 27 credits in Computer Science, out of

which 24 credits should be from CS Minor-DSC courses and 03 credits from a Skill Enhancement Course -CS SEC

2. **4-year UG Programme**: Minor Degree in Computer Science will be awarded to those who complete a four-year degree programme in any of the discipline mentioned in the FYUGP regulations and completed a minimum of 35 credits in Computer Science, out of which 32 credits should be from CS Minor DSC courses and 03 credits from a CS Skill Enhancement Course-CS SEC

C. Discipline mention in Computer Science:

A Discipline mention in Computer Science will be awarded to those who complete a three year or four-year degree programme in any of the discipline mentioned in the FYUGP regulations and completed a minimum of 12 credits in Computer Science DSC Course in the first three years of the progamme.

.

Five Year Integrated M.Sc. in Computer Science (Artificial Intelligence & Data Science) Scheme (2024 Admission onwards)

Table 1 : Semester 1 to 6 Scheme for the Five Year Integrated M.Sc. in Computer Science (Artificial Intelligence & Data Science) 2024 Admission onwards

Semester	Computer Science Major	Computer Science Minor	**Foundation Courses** MDC: Multi-Disciplinary Courses AEC: Ability Enhancement Courses* SEC: Skill Enhancement Courses VAC: Value Added Courses*				
	Major **4 credits**	**Minor** **4 credits**	**MDC*** **3 credits**	**AEC*** **3 credits**	**SEC** **3 credits**	**VAC*** **3 credits**	**Total credits**
I	1	2	1	2*			21
II	1	2	1	2*			21
III	1	2	1			2*	21
IV	4				1	1*	22
V	5				1		23
VI	5 (Out of the five courses one course is a 4 credit Mini Project)				1		23
		Internship**					2
Total credits(courses)	**68 (17)**	**24 (6)**	**9 (3)**	**12 (4)**	**9 (3)**	**9 (3)**	**133**

***Courses offered commonly to all Integrated M.Sc. programme at University Level

**Not counted as a course

Minor Degree in Computer Science: Total Credits required= 27 [24 credits should be from CS Minor-DSC courses and 03 credits from a CS Skill Enhancement Course(CS SEC)]

Exit option 1(Major Degree): B.Sc. in Computer Science (Total credit requirements = 133)

Table 2 : Semester 7 and 8 Scheme for the Five Year Integrated M.Sc. in Computer Science (Artificial Intelligence & Data Science) 2024 Admission onwards

<table>
<tr><th colspan="2">Semester</th><th>Computer Science Major 4 credits</th><th>Seminar 2 credits</th><th>Laboratory 2 credits</th><th>Mini Project</th><th>Elective 4 Credits</th><th>Computer Science Minor 4 credits</th><th>Foundation Courses</th><th>Total Credits</th></tr>
<tr><td colspan="2">VII</td><td>4</td><td>1</td><td></td><td></td><td></td><td>1*</td><td></td><td>22</td></tr>
<tr><td rowspan="2">VIII</td><td>B.Sc. Honors with Research**</td><td></td><td></td><td>1</td><td>12 Credits</td><td>1</td><td>1*</td><td></td><td>22</td></tr>
<tr><td>B.Sc. Honors</td><td></td><td></td><td>1</td><td>4 Credits</td><td>3</td><td>1*</td><td></td><td>22</td></tr>
<tr><td rowspan="4">Total credits(courses)</td><td rowspan="2">B.Sc. Hons. (Research)[2]</td><td>80 (20)</td><td>2(1)</td><td>2(1)</td><td>16(2)</td><td>4(1)</td><td rowspan="2">32(8)</td><td rowspan="2">39(13)</td><td rowspan="2">177</td></tr>
<tr><td colspan="5">Total Credit Requirements from Computer Science discipline =104</td></tr>
<tr><td rowspan="2">B.Sc. Hons[1]</td><td>80 (20)</td><td>2(1)</td><td>2(1)</td><td>8(2)</td><td>12(3)</td><td rowspan="2">32(8)</td><td rowspan="2">39(13)</td><td rowspan="2">177</td></tr>
<tr><td colspan="5">Total Credit Requirements from Computer Science discipline =104 Credits</td></tr>
</table>

*4-year UG Programme students seeking minor Degree in Computer Science must earn 8 more credits in the Fourth year.

**Eligibility for a student to be considered for the 4-year UG Degree (Honors with Research) programme is CGPA 8.0 up to sixth semester. The number of seats and the selection criteria shall be fixed by the DCS Department Council.

[1]Exit option 2 : B.Sc. (Honours with Research) in Computer Science (Artificial Intelligence & Data Science)

[2]Exit option 3: B.Sc. (Honours) in Computer Science (Artificial Intelligence & Data Science)

Table 3 : Semester 9 and 10 Scheme for the Five Year Integrated M.Sc. in Computer Science (Artificial Intelligence & Data Science) 2024 Admission onwards

<table>
<tr><th>Semester</th><th>Computer Science (Artificial Intelligence & Data Science) Major</th><th>Seminar (2 credits)</th><th>Lab (2 credits)</th><th>Project</th><th>Elective 4 Credits</th><th>Computer Science Minor 4 Credits</th><th>Foundation Courses</th><th>Total credits</th></tr>
<tr><td>IX</td><td></td><td></td><td></td><td>Project Phase I (14 Credits)</td><td>2</td><td></td><td></td><td>22</td></tr>
<tr><td>X</td><td></td><td></td><td></td><td>Project Phase II (22 Credits)</td><td></td><td></td><td></td><td>22</td></tr>
<tr><td rowspan="2">Total credits(Courses)</td><td>80(20)</td><td>2(1)</td><td>2 (1)</td><td>Honours with Research 52(4) / Honours 44(4)</td><td>Honours with Research 12(3) / Honours 20(5)</td><td rowspan="2">32(8)</td><td rowspan="2">39</td><td rowspan="2">221</td></tr>
<tr><td colspan="5">Total Credit Requirements from Computer Science discipline =148 Credits</td></tr>
<tr><td colspan="9">Five Year Integrated M.Sc. in Computer Science (Artificial Intelligence & Data Science)</td></tr>
</table>

Five Year Integrated M.Sc. in Computer Science (Artificial Intelligence & Data Science) Course structure

(2024 Admission onwards)

Semester I

Course Code	Course Level	Course Name	The course can be taken towards obtaining credits for:	Credits	Hours/ week L-T-P	Marks distribution		
						CA	ESE	Total
24-813-0101	100-199	Computational Thinking with Python	CS Major -DSC, CS Minor-DSC CS Disci	4	4-1-2	50	50	100
24-813-0102	100-199	Practical Applications of AI	CS Minor-DSC CS Disci	4	4-1-0	50	50	100
24-813-0103	100-199	Computational Thinking for Problem Solving	CS MDC	3	3-1-0	50	50	100
Credit Requirements		21 Credits (AEC: 6 Credits, MDC: 3 Credits, Major pathway: 4, Minor pathway: 8 Credits) Cumulative credits: 21						

L: Lecture, T: Tutorial, P: Practicum

CS Major-DSC: Core course for students Majoring in Computer Science

CS Minor-DSC: Core course for students Minoring in Computer Science

CS Disci-DSC: Core course for students who choose discipline mention in Computer Science

CS MDC: Multidisciplinary course offered to students whose Major or Minor pathways are different from Computer Science

AEC: Ability Enhancement Course(Offered at University Level

Semester II

Course Code	Course Level	Course Name	The course can be taken towards obtaining credits for:	Credits	Hours/ week L-T-P	Marks distribution		
						CA	ESE	Total
24-813-0201	100-199	Fundamentals of Programming	CS Major -DSC, CS Minor-DSC CS Disci	4	4-1-2	50	50	100
24-813-0202	100-199	Computer Fundamentals 1	CS Minor-DSC CS Disci	4	4-1-0	50	50	100
24-813-0203	100-199	Foundations of Programming	CS MDC	3	3-1-0	50	50	100
Credit Requirements		21 Credits(AEC: 6, MDC: 3, Major pathway: 4, Minor pathway: 8) Cumulative credits: 42						

Semester III

Course Code	Course Level	Course Name	The course can be taken towards obtaining credits for:	Credits	Hours/ week L-T-P	Marks distribution		
						CA	ESE	Total
24-813-0301	200-299	Data Structures	CS Major -DSC, CS Minor-DSC CS Disci	4	4-1-2	50	50	100
24-813-0302	200-299	Computer Fundamentals II	CS Minor-DSC CS Disci	4	4-1-0	50	50	100
24-813-0303	200-299	Fundamentals of Data Structures	CS MDC	3	3-1-0	50	50	100
Credit Requirements		21 (VAC: 6, MDC: 3, Major pathway: 4, Minor pathway: 8) Cumulative credits: 63						

Semester IV

Course Code	Course Level	Course Name	The course can be taken towards obtaining credits for:	Credits	Hours/ week L-T-P	Marks distribution		
						CA	ESE	Total
24-813-0401	200-299	Advanced Programming with Java	CS Major -DSC	4	4-1-2	50	50	100
24-813-0402	200-299	Digital Logic and Computer Organization	CS Major -DSC	4	4-1-0	50	50	100
24-813-0403	200-299	Introduction to Artificial Intelligence	CS Major -DSC	4	4-1-0	50	50	100
24-813-0404	200-299	Database Management Systems	CS Major -DSC	4	4-1-2	50	50	100
24-813-0405	200-299	Python for Data Science and Machine Learning	CS SEC	3	3-1-2	50	50	100
Credit Requirements		22 (VAC: 3, SEC: 3, Major pathway: 16) Cumulative credits:85						

Semester V

Course Code	Course Level	Course Name	The course can be taken towards obtaining credits for:	Credits	Hours/ week L-T-P	Marks distribution		
						CA	ESE	Total
24-813-0501	300-399	Mathematics for Computing	CS Major -DSC	4	4-1-0	50	50	100
24-813-0502	300-399	Fundamentals of Data Science	CS Major -DSC, CS Minor-DSC CS Disci	4	4-1-0	50	50	100
24-813-0503	300-399	Operating System	CS Major -DSC	4	4-1-0	50	50	100
24-813-0504	300-399	Theory of Computation	CS Major -DSC	4	4-1-0	50	50	100
24-813-0505	300-399	Design and Analysis of Algorithms	CS Major -DSC	4	4-1-2	50	50	100
24-813-0506	200-299	R for Data Science	CS-SEC	3	3-1-2	50	50	100
Credit Requirements		23 (SEC: 3, Major pathway: 20) Cumulative credits: 108						

Semester VI

Course Code	Course Level	Course Name	The course can be taken towards obtaining credits for:	Credits	Hours/ week L-T-P	Marks distribution		
						CA	ESE	Total
24-813-0601	300-399	Machine Learning	CS Major -DSC CS Minor-DSC CS Disci	4	4-1-2	50	50	100
24-813-0602	300-399	Agile Software Engineering	CS Major -DSC	4	4-1-0	50	50	100
24-813-0603	300-399	Computer Networks	CS Major -DSC CS Minor-DSC CS Disci	4	4-1-0	50	50	100
24-813-0604	300-399	Natural Language Processing	CS Major -DSC	4	4-1-0	50	50	100
24-813-0605	300-399	Mini Project -1	CS Major -DSC	4	0-0-4	50	50	100
24-813-0606	200-299	Web Technologies	CS -SEC	3	3-1-2	50	50	100
Credit Requirements		23 (SEC: 3, Major pathway: 20) Cumulative credits: 131						
Internship(2 Credits) Students have to complete an internship of 2 credits (60 Hours of work) before the beginning of Semester VII. Cumulative credits: 133								

Semester VII

Course Code	Course Level	Course Name	The course can be taken towards obtaining credits for:	Credits	Hours/ week L-T-P	Marks distribution		
						CA	ESE	Total
24-813-0701	400-499	Deep Learning	CS Major -DSC	4	4-1-0	50	50	100
24-813-0702	300-399	Bigdata Analytics	CS Major -DSC, CS Minor-DSC CS Disci	4	4-1-2	50	50	100
24-813-0703	400-499	Cloud Computing and virtualization	CS Major -DSC	4	4-1-0	50	50	100
24-813-0704	400-499	Cyber Security	CS Major -DSC	4	4-1-0	50	50	100
24-813-0705	300-399	Image Processing and Computer Vision	CS Major -DSC, CS Minor-DSC CS Disci	4	4-1-2	50	50	100
24-813-0706	400-499	Seminar	CS Major -DSC	2	0-0-2	100		100
Credit Requirements		22 (Major pathway: 22) Cumulative credits: 155						

Semester VIII(Honours with Research)

Course Code	Course Level	Course Name	The course can be taken towards obtaining credits for:	Credits	Hours/ week L-T-P	Marks distribution		
						CA	ESE	Total
24-813-0801	300-399	Bioinformatics	CS Major -DSC, CS Minor-DSC CS Disci	4	4-1-0	50	50	100
24-813-080X	400-499	Elective -1	CS Major -DSE,	4	4-1-0	50	50	100
24-813-0802	400-499	Research Project	CS Major -DSC	12	0-0-12	50	50	100
24-813-0803	400-499	Full Stack AI Lab	CS Major -DSC	2	0-0-2	50	50	100
Credit Requirements		22 (Major pathway: 22) Cumulative credits: 177						

Semester VIII(Honours)

Course Code	Course Level	Course Name	The course can be taken towards obtaining credits for:	Credits	Hours/ week L-T-P	Marks distribution		
						CA	ESE	Total
24-813-0801	400-499	Bioinformatics	CS Major -DSC, CS Minor-DSC CS Disci	4	4-1-0	50	50	100
24-813-080X	400-499	Elective -1	CS Major -DSE,	4	4-1-0	50	50	100
24-813-0809	400-499	Elective -2 (MOOC*)	CS Major -DSE	4	0-0-0			100
24-813-0810	400-499	Elective -3 (MOOC*)	CS Major -DSE	4	0-0-0			100
24-813-0811	400-499	Mini Project -2	CS Major -DSC	4	0-0-0	50	50	100
24-813-0803	400-499	Full Stack AI Lab	CS Major -DSC	2	0-0-2	50	50	100
Credit Requirements		22 (Major pathway: 22) Cumulative credits: 177						

*A credit-based MOOC course of minimum 12 weeks duration from SWAYAM/NPTEL/CUSAT any other platforms approved by the Department. Students may be permitted to enrol MOOC courses approved by Department Council at any time during the programme and acquire the required credits before completing the programme.

List of Electives

24-813-0804 Advanced Optimization Techniques

24-813-0805 Blockchain Technology

24-813-0806 Information Retrieval and Web search

24-813-0807 Number Theory and Cryptography

24-813-0808 Large Language Models

Semester IX

Course Code	Course Level	Course Name	The course can be taken towards obtaining credits for:	Credits	Hours/ week L-T-P	Marks distribution		
						CA	ESE	Total
24-813-0901	500-599	Major Project Phase- I	CS Major -DSC	14	0-0-14	50	50	100
24-813-0902	500-599	Elective -4 (MOOC*)	CS Major -DSE	4	0-0-0			100
24-813-0903	500-599	Elective -5(MOOC*)	CS Major -DSE	4	0-0-0			100
Credit Requirements		22 (Major pathway: 22) Cumulative credits: 199						

*A credit-based MOOC course of minimum 12 weeks duration from SWAYAM/NPTEL/CUSAT any other platforms approved by the Department. Students may be permitted to enrol MOOC courses approved by Department Council at any time during the programme and acquire the required credits before completing the programme.

Semester X

Course Code	Course Level	Course Name	The course can be taken towards obtaining credits for:	Credits	Hours /week L-T-P	Marks distribution		
						CA	ESE	Total
24-813-1001	600-699	Major Project Phase- II	CS Major -DSC	22	0-0-22	50	50	100
Credit Requirements		22 (Major pathway: 22) Cumulative credits: 221						

Five Year Integrated M.Sc. in Computer Science (Artificial Intelligence & Data Science) Syllabus (2024 Admission onwards)

Semester 1						
24-813-0101	**Computational Thinking with Python (Course Level 100-199)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**2**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Recognizing and Defining Computational Problems	**Understand**
CO2	Designing algorithms for simple problems using computational thinking principles	**Apply**
CO3	Applying inductive and deductive reasoning, and Boolean logic to solve problems	**Apply**
CO4	Designing solutions and solution processes based on problem definitions.	**Apply**
CO5	Programming CT artifacts using Python	**Analyze**

CO – PSO Mapping

CO	PSO1	PSO2	**PSO3**	**PSO4**
CO1	3	-	**-**	**-**
CO2	3	2	**3**	**-**
CO3	3	3	**1**	**-**
CO4	1	3	**3**	**-**
CO5	-	3	**2**	**-**

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1

Elements of Computational Thinking - Understanding computational thinking - Decomposing problems, Recognizing patterns, Generalizing patterns, Designing algorithms for simple problems

Module 2

Understanding Algorithms and Algorithmic Thinking - Defining algorithms in depth, Designing algorithms, Analyzing algorithms - Big-Oh notation.

Module 3

Understanding Logical Reasoning - Inductive reasoning, Deductive reasoning. Using Boolean logic and operators. Identifying Logical Errors and Debugging. Understanding the problem definitions - Learning how to decompose problems - Converting the flowchart into an algorithm. Designing Solutions and Solution Processes

Module 4

Identifying Challenges within Solutions - Identifying errors in algorithm design - Debugging algorithms -Comparing solutions - Refining and redefining solutions

Module 5

Introduction to Python, Using Computational Thinking in Simple Challenges, Using Python in Experimental and Data Analysis Problems - Classification and Clusters, Using Computational Thinking and Python in Statistical Analysis

References

1. Applied Computational Thinking with Python - Second Edition. By Sofía De Jesús, Dayrene Martinez
2. Karl Beecher, Computational Thinking – A beginners guide to problem solving and programming, BCS, 1e, THE CHARTERED INSTITUTE FOR IT, 2017
3. Peter J. Denning, Matti Tedre, Computational Thinking, MIT Press, 2019
4. Peter William Mcowan, Paul Curzon, Power Of Computational Thinking, World Scientific, 2017

Semester 2						
24-813-0201	**Fundamentals of Programming (Course Level 100-199)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**2**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO 1	Understand the fundamentals of programming and learn to write programs.	**Understand**
CO 2	Analyze the different the programming structures such as decision making statements, loops, arrays and functions.	**Analyze**
CO 3	Understand the basic concepts of OOP and learn how to create and initialize objects using constructors.	**Apply**
CO 4	Understand and analyze the different types of inheritance..	**Analyze**
CO 5	Understand the usage of polymorphism, template classes, namespaces and exception handling	**Understand**

CO – PSO Mapping

CO	PSO1	PSO2	**PSO3**	**PSO4**

CO1	2	1	-	-
CO2	2	1	-	-
CO3	2	1	-	-
CO4	1	1	-	-
CO5	2	1	-	-

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1(8 Lectures)

Introduction to Programming: Basic Difference between Procedure Oriented Language and Object Oriented Language, Concepts of Machine level, Assembly level and High level programming, Flow charts and Algorithms. Fundamentals of C: Features of C language, structure of C program, comments, header files, data types, constants and variables, operators, expressions, evaluation of expressions, type conversion, precedence and associativity, I/O functions.

Module 2(10 Lectures)

Simple statements, Decision making statements, Looping statements, Nesting of control structures, break and continue statement. Array & String: Concept of array, One and Two dimensional arrays, declaration and initialization of arrays, String, String storage, Built-in string functions. Functions: Concept of user defined functions, prototype, definition of function, parameters, parameter passing, calling a function.

Module 3 (8 Lectures)

Introduction to OOP: Basic properties, Classes and Objects, Member functions and variables, Abstract data types (ADT), Encapsulation. Constructors: Parameterized Constructors, Copy Constructors, Dynamic Constructors, Destructors.

Module 4(10 Lectures)

Inheritance and Access Control: Member access control in classes, Friend functions and classes, Public Private and Protected Inheritance, Single – Multiple – Multilevel – Hierarchical – Hybrid inheritance.

Module 5 (8 Lectures)

Polymorphism: Runtime and compile time polymorphism, overloading functions and operators, Defining Operator Overloading, Overloading Operators, Rules for Overloading Operators, selecting friend member function for operator overloading, Virtual methods, pure virtual methods – Abstract classes. Template classes: Creating and using templates, Namespaces, Exception Handling, Inline functions

References

1. Yashavant Kanetkar: Let Us C, 15e,BPB Publications, 2016.
2. Herbert Schildt: C: The Complete Reference, 4e, McGraw Hill Education, 2017.
3. Pradip Dey, Manas Ghosh: Computer Fundamentals and Programming in C, 2e, Oxford University Press, 2013.
4. Bjarne Stroustrup: C++ Programming Language, 4e, Addison-Wesley, 2013.
5. Bjarne Stroustrup: Programming: Principles and Practice Using C++, 2e, Addison- Wesley, 2014.
6. Stanley Lippman, Josée Lajoie, Barbara Moo: C++ Primer, 5e, Addison-Wesley, 2012.

Semester 3						
24-813-0301	**Data Structures (Course Level 200-299)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**2**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Understand different asymptotic notations to analyze performance of algorithms.	**Understand**
CO2	Use elementary and advanced data structures such as Array, Linked list, Tree and Graph to solve real world problems efficiently.	**Apply**
CO3	Implement searching and sorting methods.	**Apply**
CO4	Understand different memory management techniques and their significance.	**Analyze**

CO – PSO Mapping

CO	PSO1	PSO2	**PSO3**	**PSO4**
CO1	3	3	1	-
CO2	3	2	1	-
CO3	3	2	1	-
CO4	3	2	-	-

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1

Introduction to programming methodologies – structured approach, stepwise refinement techniques, programming style, documentation. Elementary data organization - Data structure - Data structure operation - Analysis of algorithms: frequency count, definition of Big O notation, asymptotic analysis of simple algorithms - Recursive and iterative algorithms.

Module 2

Array, Records and Pointers: Introduction, Linear array, Representation of linear array in memory, Traversing linear array, Inserting and Deleting, Sorting methods, Searching methods. String - representation of strings, concatenation, substring searching and deletion.

Module 3

Linked List: Introduction, Linked list, Representation of Linked list in memory, Searching a linked list, Memory allocation, Garbage collection, Insertion and deletion in linked list, doubly linked list, Circular linked list, applications of linked list: polynomials, Memory management, memory allocation and deallocation, First-fit, best-fit and worst-fit allocation schemes.

Module 4

Stacks, Queues, Recursion - Introduction, Stacks, Queues, Operations on stacks and Queues, Implementation of Stacks and Queues using arrays and linked list, Arithmetic expression evaluation, Recursion, DEQUEUE (double ended queue), Multiple Stacks and Queues, Applications.

Module 5

Tree - Introduction, Terminology of Binary tree, Types of Binary tree, Traversing of binary tree, Header Nodes, Threads. Binary search tree – creation, insertion and deletion and search operations, applications. B-Trees, B+-Trees. Hash Tables – Hashing functions – Mid square, division, folding, digit analysis, collusion resolution and Overflow handling techniques.

References

1. Samanta D.: Classic Data Structures, 2e, Prentice Hall India, 2009.
2. Richard F. Gilberg, Behrouz A. Forouzan: Data Structures: A Pseudocode Approach with C, 2e, Cengage Learning, 2005.
3. Aho A. V., J. E. Hopcroft, J. D. Ullman: Data Structures and Algorithms, Pearson Publication, 1983.
4. Tremblay J. P., P. G. Sorenson: Introduction to Data Structures with Applications, 2e, Tata McGraw Hill, 1995.
5. Peter Brass: Advanced Data Structures, Cambridge University Press, 2008.
6. Lipschutz S.: Theory and Problems of Data Structures, Schaum's Series, 1986.
7. Wirth N.: Algorithms + Data Structures = Programs, Prentice Hall, 2004.
8. Horwitz E., S. Sahni, S. Anderson: Fundamentals of Data Structures in C, University Press (India), 2008.

Semester 4						
24-813-0401	**Advanced Programming with Java (Course Level 200-299)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**2**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Develop object-oriented programming in Java, including defining classes, invoking methods, using libraries.	**Apply**
CO2	Demonstrate the design, implementation, testing and debugging graphical user interfaces in Java.	**Apply**
CO3	Illustrate Java Swings for designing GUI applications.	**Apply**
CO4	Apply Database Connectivity and Network Programming Skills.	**Apply**
CO5	Analyze and Evaluate Java Concurrency Mechanisms and Design Patterns.	**Analyze**
CO6	Create and Design Robust Web Applications.	**Apply**

CO – PSO Mapping

CO	PSO1	PSO2	**PSO3**	**PSO4**
CO1	3	2	-	-
CO2	3	2	-	-
CO3	2	2	-	-
CO4	3	2	-	-
CO5	2	2	-	-

CO6	3	2	3	

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Module 1

Java Overview – Java Virtual Machine – Introduction to Java Programming – Operators and Expressions Control Flow statements – Defining classes and creating objects in Java – Constructors – Access Modifiers Programs using Java objects - Inheritance – Abstract classes – Access Modifiers - final class – final method

Module 2

Method overriding – Polymorphism - Packages in Java – String Handling - Exception Handling - Parameter Passing - Java.io.package classes – Input/Output Streams – Reading console input – Collection framework Accessing Collection via Iterator interface – Utility Classes in Java

Module 3

Threads in Java – Thread class and Runnable interface – Thread Synchronization - Introduction To Swing, MVC Architecture, Applications and Pluggable Look and Feel, Basic swing components : Text Fields, Buttons, Toggle Buttons, Checkboxes, and Radio Buttons. Reflection in Java - Reading Type Information - Methods

Module 4

Java database Connectivity – JDBC overview JDBC Driver types – Loading Driver class – Obtaining Connection to database – Statement – Prepared Statement –Executing queries. Network Programming With java.net Package, Client and Server Programs, Content And Protocol Handlers.

Module 5

Java Concurrency - Semaphores - Monitor patterns - Executors: Managing Thread Pools - Concurrency Utilities: Concurrent HashMap, Countdown Latch ; Design Patterns - Introduction to Design Patterns - Creational Design Patterns - Structural Design Patterns - Behavioral Design Patterns Web Programming Options in Java - Java Servlets - JavaServer Pages (JSP) - Spring Framework - Java Server Faces (JSF)

References

1. Java 6 Programming, Black Book, Dreamtech
2. The Complete Reference, 9e, McGraw-Hill, 2017, Herbert Schildt
3. Java Server Programming, Java EE6 (J2EE 1.6), Black Book, Dreamtech
4. Advanced Java Technology, By M.T. Savaliya, Dreamtech

Semester 4						
24-813-0402	**Digital Logic and Computer Organization (Course Level 200-299)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**0**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Demonstrate understanding significance of number systems, conversions, binary codes, and digital logic gates.	**Apply**
CO2	Illustrate knowledge on design of various combinational logic circuits and data processing circuits.	**Apply**
CO3	Demonstrate understanding and analysis of arithmetic algorithms.	**Analyze**
CO4	Demonstrate understanding of the basic structure, organization of computers.	**Understand**
CO5	Demonstrate understand of central processing, I/O and memory organization.	**Understand**

CO – PSO Mapping

CO	PSO1	PSO2	**PSO3**	**PSO4**
CO1	1	1	1	-
CO2	-	1	1	-
CO3	3	1	-	-
CO4	1	1	-	-
CO5	-	1	-	-

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1 (8 Lectures)

Number Systems and Codes: Binary Number system – Binary to decimal – decimal to binary – hexadecimal – ASCII code – Excess-3 Code – Gray code. Digital Logic: The Basic Gates – NOT, OR, AND - Universal Logic Gates – NOR, NAND.

Module 2 (10 Lectures)

Combinational Logic Circuits: Boolean Laws and Theorems. - Sum of Products method -Truth table to Karnaugh Map – Pairs, Quads, Octets – Don't Care Conditions- Product-of sums method -Product-of sums Simplifications. Data Processing Circuits: Multiplexers – Demultiplexers-1-of-16 Decoder – BDC- to decimal Decoders – Seven-segment Decoders – Encoders – Exclusive-OR Gates- Parity Generators and Checkers.

Module 3 (8 Lectures)

Arithmetic Circuits: Binary Addition- Binary Subtraction – 2'S Complement Representation - 2'S Complement Arithmetic – Arithmetic Building Blocks.

Module 4 (10 Lectures)

Basic Computer organization and Design: Instruction codes - stored program organization - Computer registers and common bus system - Computer instructions - Timing and control - Instruction cycle: Fetch and Decode - Register reference instructions. Micro programmed Control: Control memory organization - Address sequencing, micro instruction format and symbolic microinstructions - symbolic micro-program - binary micro program.

Module 5 (8 Lectures)

Central Processing Unit: General register organization - stack organization – instruction formats - addressing modes - Data transfer and manipulation - Program control. CISC and RISC - Parallel processing - Pipeline- general consideration. Input-output organization: Peripheral devices - I/O interface - Memory organization: Memory hierarchy - Main memory - Auxiliary memory.

References

1. Digital Principles and Applications – Donald P Leach, Albert Paul Malvino, GoutamSaha, 8th edition , McGraw-Hill Education, 3rd reprint 2015.
2. R. P. Jain, “Modern Digital Electronic”, McGraw-Hill Publication, 4thEdition.
3. William Stalling, “Computer Organization and Architecture: Designing and Performance”, Pearson Publication 10TH Edition.
4. Computer System Architecture, M. Morris Mano, Pearson Education, 3rd edition.,2007
5. Digital design, R.Anantha Natarajan, PHI Learning, 2015.

Semester 4						
24-813-0403	**Introduction to Artificial Intelligence (Course Level 200-299)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**0**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Understanding of AI Concepts	**Understand**
CO2	Demonstrate knowledge of various AI algorithms, techniques, and models	**Apply**
CO3	Apply AI techniques to solve real-world problems and demonstrate critical thinking skills	**Apply**
CO4	Understand knowledge-based systems.	**Understand**
CO5	Know ethical concerns	**Understand**

CO – PSO Mapping

CO	PSO1	PSO2	**PSO3**	**PSO4**
CO1	3	3	3	-
CO2	3	3	3	-
CO3	3	3	3	-
CO4	3	3	3	-
CO5	3	2	3	**3**

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module -1(8 Lectures)

Introduction to AI, Evolution of AI, Turing test, Categories of AI, Applications of AI, Problem Definition as a State Space Search, Production System, Control Strategies

Module-2(10 Lectures)

Problem Solving - Solving problems by searching, Uninformed and Informed search strategies (Breadth First Search, Depth First Search, uniform cost search, iterative deepening, Hill climbing, Heuristics Search Techniques: Best First Search, A* algorithm, AO* algorithm, Min-max, Alpha – Beta pruning), Constraint satisfaction problems

Module-3 (8 Lectures)

Knowledge based agents, First order logic, Propositional logic, Agents based on propositional logic, Knowledge Representation - Ontological Engineering, Planning - Classical Planning, Heuristics for Planning and Hierarchical Planning.

Module-4(8 Lectures)

Philosophy, Ethics, and Safety of AI - Limits of AI, The Ethics of AI, AI Safety, Future of AI- AI Components, AI Architectures

Module-5 (6 Lectures)

AI Components, AI applications – Language Models – Information Retrieval- Information Extraction – Natural Language Processing.

References

1. Ethem Alpaydin, Machine Learning: The New AI, MIT Press, 2016
2. Stuart Russell and Peter Norvig, Artificial Intelligence - A Modern Approach, 3e,
3. Pearson Education India, 2015
4. Andriy Burkov, The Hundred-Page Machine Learning Book, Andriy Burkov, 2019
5. Introduction to AI, Coursera
6. AI for everyone, Coursera
7. Jeff Heaton, Artificial Intelligence for Humans, CreateSpace, 2013
8. Mark Coeckelbergh, AI Ethics, MIT Press, 2020

Semester 4						
24-813-0404	**Database Management Systems (Course Level 200-299)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**2**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Apply Knowledge of Database Systems and Architectures.	**Apply**
CO2	Design and Implement Relational Databases.	**Apply**
CO3	Analyze and Normalize Database Designs.	**Analyze**
CO4	Implement Transaction Management and Concurrency Control.	**Apply**
CO5	Explore Advanced Database Concepts	**Understand**

CO – PSO Mapping

CO	PSO1	PSO2	**PSO3**	**PSO4**
CO1	3	2	1	-
CO2	3	1	1	-
CO3	3	1	1	-
CO4	3	1	1	-
CO5	3	2	1	-

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1

Introduction to Database Systems: Importance – Database architectures – Data model. Introduction to relational databases – Relational Model – Keys – Relational Algebra and Calculus. SQL fundamentals – Advanced SQL features – Embedded SQL– Dynamic SQL

Module 2

Entity-Relationship model – E-R Diagrams – Enhanced-ER Model – ER-to-Relational Mapping – Functional Dependencies – Non-loss Decomposition – First, Second, Third Normal Forms, Dependency Preservation – Boyce/Codd Normal Form – Multi-valued Dependencies and Fourth Normal Form – Join Dependencies and Fifth Normal Form- SQL Queries.

Module 3

Transaction management: Operations, Transaction Schedules, ACID properties. Concurrency control: Concurrency Control Problems,–Two-Phase Locking- Timestamp – Multiversion – Validation and Snapshot isolation– Multiple Granularity locking – Deadlock Handling, Recovery Concepts: Recovery based on deferred and immediate update – Shadow paging – ARIES Algorithm.

Module 4

Indexing – Cluster Indexes, Primary and Secondary Indexes – Index data Structures – Hash-Based Indexing – Tree base Indexing – Comparison of File Organizations – The Memory Hierarchy, RAID, Disk Space Management, Buffer Manager, Files of Records, Page Formats, Record Formats.

Module 5

Distributed Databases: Architecture, Data Storage, Transaction Processing, Query processing, and optimization – NoSQL Databases: Introduction – CAP Theorem – Document-Based systems – Key value Stores – Column-Based Systems – Graph Databases-Cloud Databases.

References

1. Abraham Silberschatz, Henry F. Korth, S. Sudharshan, “Database System Concepts”, Seventh Edition, McGraw Hill, 2020.
2. Ramez Elmasri, Shamkant B. Navathe, “Fundamentals of Database Systems”, Seventh Edition, Pearson Education, 2017
3. Raghu Ramakrishna, Johannes Gehrke,” Data base Management Systems”, TATA McGraw Hill 3rd Edition
4. M. Tamer OZSU and Patuck Valduriez,” Principles of Distributed Database Systems”, Pearson Edn. Asia, 2001.

Semester 4						
24-813-0405	**Python for Data Science and Machine Learning (Course Level 200-299)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		CS SEC	**3**	**1**	**2**	**3**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Apply Python syntax and semantics to write well-structured and efficient programs	**Apply**
CO2	Utilize functions with arguments to modularize code and improve reusability	**Understand**
CO3	Apply core data structures to organize and manipulate data in Python programs**.**	**Apply**
CO4	Apply techniques to read data from and write data to files using Python.	**Apply**
CO5	Interact with the operating system using Python libraries to automate file management and system commands	**Understand**

CO – PSO Mapping

CO	PSO1	PSO2	**PSO3**	**PSO4**
CO1	3	2	1	-
CO2	3	1	1	-
CO3	3	1	1	-
CO4	3	1	1	-
CO5	3	2	1	-

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1

Programming Environment and Python Basics: Getting Started with Python Programming - Running code in the interactive shell, Editing, Saving, and Running a script. Using editors - Jupyter. Working with data types, Numeric data types and Character sets, Keywords, Variables and Assignment statement, Operators, Expressions, Working with numeric data, Type conversions, Building Python Programs: Control statements - Selection structure (if-else, switch-case), Iteration structure(for, while), Testing the control statements

Module 2

Functions - Hiding redundancy and complexity, Arguments and return values, Variable scopes and parameter passing, Named arguments, Main function, Strings and number systems - String function, Data Representation: Lists - Basic list Operations and functions, List of lists, Slicing, Searching and sorting list, List comprehension.

Module 3

Work with tuples, Sets. Work with dates and times. Dictionaries - Dictionary functions, dictionary literals, adding and removing keys, accessing and replacing values, traversing dictionaries,

Module 4

Data Processing: The os and sys modules. Introduction to file I/O - Reading and writing text files, Manipulating binary files. NumPy - Basics, Creating arrays, Arithmetic, Slicing, Matrix Operations, Random numbers. Plotting and visualization.

Module 5

Matplotlib - Basic plot, Ticks, Labels, and Legends. Working with CSV files. – Pandas - Reading, Manipulating, and Processing Data

Textbook/ References

1. Kenneth A Lambert., Fundamentals of Python : First Programs, 2/e, Cengage Publishing, 2016
2. Wes McKinney, Python for Data Analysis, 2/e, Shroff / O'Reilly Publishers, 2017
3. Flask: Building Python web services, Jack Stouffer, Shalabh Aggarwal, Gareth Dwyer, PACKT Publishing Limited, 2018
4. Zed A Shaw, Learn Python 3 The Hard Way, Addison-Wesley, 2017
5. Allen B. Downey, Think Python: How to Think Like a Computer Scientist, 2/e, Schroff, 2016
6. Michael Urban and Joel Murach, Python Programming, Shroff/Murach, 2016

Semester 5						
24-813-0501	**Mathematics for Computing (Course Level 300-399)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**0**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Analyze the different methods for proving the correctness of the theorems and problems.	**Analyze**
CO2	Apply the basic concepts of Linear Algebra.	**Apply**
CO3	Apply the basic aspects of graph theory.	**Apply**
CO4	Apply the fundamentals of probability theory.	**Apply**

CO – PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	1	-	-
CO2	2	1	-	-
CO3	2	1	-	-
CO4	2	1	-	-

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1

Introduction – proofs – propositions – predicates and quantifiers – truth tables – first order logic – satisfiability – pattern of proof – proofs by cases – proof of an implication – proof by

contradiction – proving iff – sets – proving set equations – Russell's paradox – well-ordering principle – induction – invariants – strong induction – structural induction

Module 2

Vectors-Coordinate system-vector addition-vector multiplication-Linear combinations, span, and basis vectors-Matrix multiplication as composition-Three-dimensional linear transformations-The determinant-Inverse matrices, column space and null space- Nonsquare matrices as transformations between dimensions-Dot products and duality-Cross products-Cross products in the light of linear transformations-Cramer's rule-Change of basis-Eigenvectors and eigenvalues-vector spaces

Module 3

Graph theory – simple graphs – isomorphism – subgraphs – weighted graphs – matching problems – stable marriage problem – graph coloring – paths and walks – shortest paths – connectivity – Eulerian and Hamiltonian tours – travelling salesman problem – trees – spanning trees – planar graphs – Euler's formula – directed graphs – strong connectivity – relations – binary relations – surjective and injective relations symmetry, transitivity, reflexivity, equivalence of relations – posets and dags – topological sort.

Module 4

Probability – events and probability spaces – conditional probability – tree diagrams for computing probability – sum and product rules of probability – A posteriori probabilities – identities of conditional probability – independence – mutual independence – birthday paradox – random variables – indicator random variables.

Module 5

Probability distribution functions – Bernoulli, Uniform, Binomial, Poisson, Normal distributions – Expectation – linearity of expectations – sums of indicator random variables – expectation of products – variance and standard deviation of random variables – Markov and Chebyshev's theorems – Bounds for the sums of random variables.

References

1. Bronson, R., Costa, G.B., Saccoman, J.T. and Gross, D.,Linear algebra: algorithms, applications, and techniques.4e, 2023.
2. Eric Lehman, F Thomson Leighton, Albert R Meyer, Mathematics for Computer Science, 1e, MIT, 2010.
3. Susanna S. Epp, Discrete Mathematics with Applications, 4e, Brooks Cole, 2010.
4. Gary Chartrand, Ping Zhang, A First Course in Graph Theory, 1e, Dover Publications, 2012.
5. Michael Sipser, Introduction to Theory of Computation, 3e, Cengage, 2014.
6. Sheldon Ross, A First Course in Probability, 9e, Pearson, 2013.
7. Tom Leighton, and Marten Dijk. 6.042J Mathematics for Computer Science.Fall 2010. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu.

8. John Tsitsiklis. 6.041SC Probabilistic Systems Analysis and Applied Probability. Fall 2013.
9. Massachusetts Institute of Technology: MIT OpenCourseWare. https://ocw.mit.edu
10. Igor Pak. 18.315 Combinatorial Theory: Introduction to Graph Theory, Extremal and Enumerative Combinatorics. Spring 2005. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu
11. Albert Meyer. 6.844 Computability Theory of and with Scheme. Spring 2003. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu.
12. Shai Simonson , Theory of Computation, http://www.aduni.org/courses/theory/

Semester 5						
24-813-0502	**Fundamentals of Data Science (Course Level 300-399)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**0**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Understand the fundamental concepts and processes of data science ,encompassing data engineering , data analysis and model building.	**Understand**
CO2	Gain practical experience in data wrangling ,exploratory data analysis and visualization	**Analyze**
CO3	Develop essential skills for data mining, predictive analytics, and recommendation systems.	**Apply**
CO4	Explore the integration of data science with software development and security practices (DevSecOps).	**Apply**

CO – PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	3	-
CO2	3	2	3	-
CO3	3	2	3	-
CO4	3	2	3	-

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1

Understanding the data science lifecycle: data acquisition, preprocessing, analysis, visualization, and communication. Ethical considerations in data collection, analysis, and

responsible data science practices. Data pre-processing techniques, handling missing values, outliers, and inconsistencies. Data transformation, including feature engineering and scaling techniques.

Module 2

Data mining concepts and techniques for extracting hidden patterns and insights. Association Rule Learning:, Decision Trees: mining graph data, Cluster Analysis, Finding similar items, mining data streams, frequent item sets, link analysis, predictive models, descriptive models, and decision models.

Module 3:

Introduction to popular Python libraries for data science Pandas, NumPy. Applied statistics in Python, Statistical modelling with scipy.

Module 4:

Data visualization and exploration: creating plots and charts to explore relationships between variables, identify patterns or outliers, and communicate insights. Descriptive statistics: computing measures such as mean, median, standard deviation, or correlation coefficients to understand the distribution of data. Clustering and dimensionality reduction: Data visualization principles: choosing appropriate chart types, effective communication through visuals. Feature engineering.

Module 5

Introduction to recommendation systems and their applications. Collaborative filtering and content-based filtering techniques for recommendation. Integrating data science models into web applications and APIs. Understanding DevSecOps principles for secure development, deployment, and monitoring of data-driven applications. Case study

References/Text Books

1. Python Data Science Handbook by Jake VanderPlas (O'Reilly Media, 2016)
2. Davy Cielen, Arno DB Meysman and Mohamed Ali. Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools, Dreamtech Press 2016
3. **Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow** by Aurélien Géron (O'Reilly Media, 2019)
4. **Data Science in Production: Building Scalable Model Pipelines** by Jake VanderPlas (O'Reilly Media, 2020)

Semester 5						
24-813-0503	**Operating Systems (Course Level 300-399)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**0**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Explain the objectives and functions of operating systems	**Understand**
CO2	Analyze the tradeoffs inherent in operating system design	**Analyze**
CO3	Apply the CPU Scheduling Algorithms	**Apply**
CO4	Analyze process synchronization and deadlock mechanism in OS	**Analyze**
CO5	Understand memory management mechanism and file system in OS	**Understand**

CO – PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	1	1	-	-
CO2	1	1	-	-
CO3	1	1	-	-
CO4	1	1	-	-
CO5	1	1	-	-

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1(8 Lectures)

Overview of Operating Systems- Characteristics of OS, Types of OS, OS Operations, Resource Management, Kernel Data Structure- Operating System Structure, OS Services, System Call, Linkers and Loaders .

Module 2(8 Lectures)

Processes-Process concept, forks and pipes, Interrupt processing, Process Scheduling, Inter Process Communication-Threads And Concurrency- CPU Scheduling Algorithms

Module 3 (8 Lectures)

Process Synchronization- Critical Section Problem, Peterson's Solution, Mutex Locks, Semaphores, Deadlocks-Methods of Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock

Module 4(10 Lectures)

Memory Management and Virtual Memory - Logical versus Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation, Segmentation with Paging, Demand Paging, Page Replacement, Page Replacement Algorithms, Thrashing

Module 5 (10 Lectures)

File System- File concept, Access methods, Directory Structure, Memory Mapped Files, Blocks and Fragments, Directory tree, Inodes, File descriptors, UNIX file structure, Secondary Storage Management - Disk components, Disk scheduling, Swap-space management, Protection and Security, Routing, Connection strategies, Remote File Systems.

References

1. Operating System Principles, Abraham Silberchatz, Peter B.Galvin,Greg Gagne,10th Edition, Wiley Student Edition. 2018
2. Operating System-Internals and Design Principles, W.Stallings, 6th Edition, Pearson. Strang,
3. Gilbert. Modern Operating System, Andrew s Tanenbaum, 3rd Edition, PHI
4. Operating System A concept-based Approach, 2nd Edition, D.M.Dhamdhere, TMH/
5. Principle Of Operating Systems, B.LStuart, Cengage Learning, India Edition
6. An Introduction to Operating System, P.C.P.bhatt, PHI.

Semester 5						
24-813-0504	**Theory of Computation (Course Level 300-399)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**0**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO 1	Interpret the mathematical foundations of computation including automata theory.	**Analyze**
CO 2	Interpret the theory of formal languages and grammars.	**Analyze**
CO 3	Construct the abstract machines including finite automata, pushdown automata, and Turing machines from their associated languages and grammar.	**Apply**
CO 4	Make use of pumping lemma to show that a language is not regular / not context-free.	**Apply**
CO 5	Construct the grammar for any given finite automata, pushdown automata or Turing machines.	**Apply**

CO – PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	2	-
CO2	1	2	2	-
CO3	1	3	2	-
CO4	1	3	1	-
CO5	1	3	1	-

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1(8 Lectures)

Introduction to Automata Theory. Languages, Grammars, Automata and their applications, Type 3 Formalism, Finite state automata – Properties, Designing finite automata.

Module 2(10 Lectures)

Myhill-Nerode Theorem, Minimal FA Computation. Finite State Machines with Output-Mealy and Moore machine (Design Only), Minimization of FA, NFA, Equivalence of NFA and DFA, Finite Automata with Epsilon Transitions.

Module 3 (8 Lectures)

Regular Languages-properties, Regular Expressions-Properties, Equivalence of DFA and Regular Expressions. Pumping Lemma for Regular Languages, Applications of Pumping Lemma. Closure Properties of Regular sets.

Module 4(10 Lectures)

Push down automata, languages accepted by push down automata - Connection with Context free languages - Properties of context free languages, pumping lemmas, Context-sensitive Grammar, and Linear Bounded Automata

Module 5 (8 Lectures)

Variants of TMs -Universal Turing Machine, Multi-tape TMs, non-deterministic TMs, Recursively Enumerable Languages, Recursive languages, Properties of Recursively Enumerable Languages and Recursive Languages, Decidability and Halting Problem. Chomsky Hierarchy.

References

1. Peter Linz, An Introduction to Formal Languages and Automata, Jones & Bartlett Learning, 6e, 2016.
2. John E Hopcroft, Rajeev Motwani and Jeffrey D Ullman, Introduction to Automata Theory, Languages, and Computation, 3e, Pearson Education, 2007
3. John C Martin, Introduction to Languages and the Theory of Computation, TMH, 2007
4. Michael Sipser, Introduction to Theory of Computation, Cengage Publishers, 2013.

Semester 5						
24-813-0505	**Design and Analysis of Algorithm (Course Level 300-399)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**2**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Understand the basic concepts of design and analysis of fundamental algorithms.	**Understand**
CO2	Develop the ability to design algorithms to attack new problems.	**Apply**
CO3	Prove the correctness of algorithms.	**Analyze**
CO4	Develop the ability to analyze the complexity of algorithms.	**Apply**
CO5	Understand Complexity classes, concepts of P and NP problems.	**Understand**

CO – PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	1	-
CO2	3	3	2	-
CO3	3	2	2	-
CO4	3	3	2	-
CO5	3	2	1	-

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1

Introduction to design and analysis of algorithms, models of computation, correctness proofs, insertion sort, computational complexity, Master theorem , proof of Master theorem, merge sort, Quick sort, heaps, heap sort, binary search, binary search trees.

Module 2

Graph algorithms, BFS and DFS, Dijkstra's algorithm, proof of correctness of Dijkstra's algorithm, Complexity analysis of Dijkstra's algorithm , Negative weight edges and cycles , Bellman-Ford algorithm, proof of correctness and complexity of Bellman-Ford, All pairs shortest paths, Floyd-Warshall algorithm, proof of correctness and complexity, Minimum Spanning Trees , Prim's algorithm, Cut property, Kruskal's algorithm, proof of correctness and complexity analysis of Kruskal's Algorithm, Maximum-Flow networks, Ford-Fulkerson method, proof of correctness and complexity, Edmonds-Karp algorithm.

Module 3

Probability review, Experiments, outcomes, events, Random variables, Expectation, Linearity of Expectation, Indicator Random Variables, Hiring Problem, Quicksort , Best case and Worst case complexity, Randomized Quicksort , Average case complexity , Hashing, Chaining, Open Addressing, Universal Hashing, Perfect Hashing , Analysis of hashing operations.

Module 4

Dynamic Programming, Rod-cutting problem, Recursive formulation, Bottom-up reformulation of recursive algorithms, Optimal Substructure Property, Matrix chain multiplication, Complexity of dynamic programming algorithms, Sequence Alignment , Longest common subsequence, Greedy algorithms, Optimal substructure and greedy-choice properties , 0-1 and fractional Knapsack problems, Huffman coding.

Module 5

P vs NP, NP Hardness, Reductions, Travelling Salesman Problem, NP-Completeness, SAT, 2-SAT and 3-SAT, Vertex Cover.

References

1. Thomas H. Cormen et al, Introduction to Algorithms, 3e, MIT Press, 2009.
2. Jon Kleinberg, Eva Tardos, Algorithm Design, 2e, Pearson, 2015.
3. Robert Sedgewick, Kevin Wayne, Algorithms, 4e, AW Professional, 2011
4. Steven S. Skiena, The Algorithm Design Manual, 2e, Springer, 2011

Semester 5						
24-813-0506	**R for Data Science (Course Level 300-399)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		CS SEC	**3**	**1**	**2**	**3**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Understand the use of R for data analytics.	**Understand**
CO2	Learn to apply R programming for Text processing.	**Apply**
CO3	Perform appropriate statistical tests using R.	**Apply**
CO4	Create and edit visualizations with R.	**Apply**

CO – PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	2	-
CO2	3	3	3	-
CO3	3	3	3	-
CO4	3	3	3	-

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1

R Programming Basics: Overview of R programming, Environment setup with R Studio, R Commands, Variables and Data Types, Control Structures, Array, Matrix.

Module 2

Vectors, Factors, Functions, R packages, Reading and getting data into R (External Data): Using CSV files, XML files, Web Data, JSON files, Databases, Excel files.

Module 3

Data Visualization using R: Working with R Charts and Graphs: Histograms, Boxplots, Bar Charts, Line Graphs, Scatterplots, Pie Charts.

Module 4

Statistics with R: Random Forest, Decision Tree, Normal and Binomial distributions , Linear and Multiple Regression, Logistic Regression, Time Series Analysis.

Module 5

String Manipulation – Graphics –Creating Graphs – Customizing Graphs – Saving graphs to files – Creating three-dimensional plots

References

1. W. N. Venables, D.M. Smith and the R Development Core Team, An Introduction to R, Notes on R: A Programming Environment for Data Analysis and Graphics.
   URL: https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf
2. Norman Matloff, The Art of R Programming – A Tour of Statistical Software Design, 1e, No Starch Press, 2011.
3. Jared P. Lander, R for Everyone: Advanced Analytics and Graphics, 1e, Pearson Education India, 2014.
4. Mark Gardener, Beginning R - The Statistical Programming Language, John Wiley & Sons, Inc., 2013.

Semester 6						
24-813-0601	**Machine Learning (Course Level 300-399)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**2**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Understand and explain the different types of the learning process, and key ethical considerations.	**Understand**
CO2	Learn to effectively prepare data for machine learning models through data cleaning, feature selection, and dimensionality reduction.	**Apply**
CO3	Implement and interpret linear and non-linear regression models, while comparing various classification techniques including tree-based, kernel, and ensemble methods.	**Apply**
CO4	Gain practical knowledge in identifying data clusters using various algorithms and discovering hidden patterns through association rule learning.	**Apply**
CO5	Understand the basic building blocks of neural networks, implement the backpropagation algorithm, and explore the concept of MDPs and Q-learning	**Understand**

CO – PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	3	3
CO2	3	3	3	-
CO3	3	3	3	-
CO4	3	3	3	-
CO5	3	3	3	-

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1

Introduction to AI - What is AI? A Brief History of AI - Different types of AI - Applications of AI - Problem Solving Methods – Heuristics. Knowledge Representation and Reasoning - Planning and Decision-Making: Ethics and Societal Impact of AI.

Module 2

Machine Learning Fundamentals - Concept of Machine Learning: Definition, Applications, Types of learning (supervised, unsupervised, reinforcement) - Hypothesis Spaces and Inductive Bias - Learning Process- Machine Learning Ethics and Bias. Data Preprocessing and Feature Engineering: Data Representation - Data Preprocessing - Features and Types - Dimensionality Reduction – Feature Identification - Feature selection – Feature extraction - Feature Importance.

Module 3

Regression and Classification - Regression: Linear Regression – Non-Linear regression – evaluation metrics for regression– Classification: Binary, multi-class, and multi-label classification – lazy learners - tree-based techniques - kernel-based techniques - probabilistic techniques - and ensembled techniques – evaluation metrics for classification.

Module 4

Clustering and Rule Mining - Clustering: Partitioning based – hierarchical based – density based– grid-based – model based - Rule mining: Apriori algorithm, FB Growth - association rules. Outlier Detection - LOF.

Module 5

Artificial Neural Networks and Reinforcement Learning -Neural Networks: McCulloch-Pitts neurons, Hebb's networks, Hopfield networks, Boltzmann machines, Perceptrons, multilayer perceptrons, backpropagation. Reinforcement Learning: Markov Decision Processes (MDPs), Q-learning.

References

1. Ethem Alpaydin, Introduction to Machine Learning, 3e, MIT Press, 2014.
2. Tom M. Mitchell, Machine Learning, McGraw Hill Education; 1e, 2017.
3. Stephen Marsland, Machine Learning, An Algorithmic Perspective, 2e, CRC Press, 2015.
4. Giuseppe Bonaccorso, Machine Learning Algorithms, 1e, Packt Publishing Limited, 2017.
5. Ethem Alpaydin, Machine Learning- The New AI, MIT Press, 1e, 2016.
6. Andrew Ng, Machine Learning Yearning, ATG AI (Draft version), 1e, 2018.
7. Rohit Singh, Tomi Jaakkola, and Ali Mohammad.6.867 Machine Learning. Fall 2006. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu
8. Andrew Ng, https://www.coursera.org/learn/machine-learning

Semester 6						
24-813-0602	**Agile Software Engineering (Course Level 300-399)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**0**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Create a software product architecture using UML	**Apply**
CO2	Communicate with the development team using industry standard notations, designs and documentations.	**Apply**
CO3	Estimate the cost of a software project and apply various techniques, metrics and strategies for testing software projects.	**Analyze**
CO4	Work as a team leader by establishing goals and forming teams.	**Apply**
CO5	Understand the user requirements and plan the development work using agile project management principles.	**Analyze**

CO – PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	3	3	1
CO2	3	2	2	-
CO3	3	3	2	-
CO4	3	1	2	1
CO5	3	2	2	1

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1(8 Lectures)

Emergence of Software Engineering, Software design notations, Object-Oriented Analysis and Design using Unified Modelling Language (UML), Use Case Model Development, Object and Class Diagrams, Interaction Diagrams, Sequence models, Activity Diagrams, State Chart Diagrams, Package diagrams

Module 2

Software Life Cycle Models, Waterfall Model, Prototyping Model, Spiral Model, Software Requirements Specification, SRS Document, Function-oriented Design, , Scheduling, Critical Path Method, PERT Charts, Gantt Charts, Organization and Team Structures

Module 3

Metrics for Project Size Estimation, COCOMO Model, Software Quality, Software Quality Management System, Testing Concepts and Terminologies, Black-box Testing, White-Box Testing, Statement Coverage, Branch Coverage, Path Coverage, McCabe's Cyclomatic Complexity Metric, Software Maintenance.

Module 4

Agile Principles, Variability and Uncertainty, Work in Process, Progress, Performance, Scrum Framework, Scrum Roles, Responsibilities & Characteristics of Product Owner, ScrumMaster, Development Team, Sprints, Timeboxing, Sprint Planning, Sprint Execution

Module 5

Product Backlog, Good Product Backlog Characteristics, Requirements and User Stories, Characteristics of Good Stories, Estimation and Velocity, PBI Estimation Units, Planning Poker, Scrum Planning Principles, Product Planning (Envisioning), Portfolio Planning, Release Planning, Sprint Planning

References

1. Yashavant Kanetkar: Let Us C, 15e,BPB Publications, 2016.
2. Herbert Schildt: C: The Complete Reference, 4e, McGraw Hill Education, 2017.
3. Pradip Dey, Manas Ghosh: Computer Fundamentals and Programming in C, 2e, Oxford University Press, 2013.
4. Bjarne Stroustrup: C++ Programming Language, 4e, Addison-Wesley, 2013.
5. Bjarne Stroustrup: Programming: Principles and Practice Using C++, 2e, Addison-Wesley, 2014.
6. Stanley Lippman, Josée Lajoie, Barbara Moo: C++ Primer, 5e, Addison-Wesley, 2012.
7. Paul Deitel, Harvey Deitel: C++ How to Program, 10e, Pearson, 2016.
8. Timothy Budd: Introduction To Object-Oriented Programming, Pearson Education, 2008.
9. Walter J. Savitch, Kenrick Mock: Problem Solving with C++, 9e, Pearson Education, 2017.
10. Ira Pohl: Object-Oriented Programming Using C++, 2e, Addison-Wesley,1996.

Semester 6						
24-813-0603	**Computer Networks (Course Level 300-399)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**0**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Understand the fundamental concepts of computer networking	**Understand**
CO2	Apply various techniques for channel allocation, framing, error and flow control	**Apply**
CO3	Analyze different networks and choose addresses for networking requirements	**Analyze**
CO4	Acquire knowledge on various routing algorithms and design issues.	**Analyze**
CO5	Develop an understanding of different transport layer and application layer protocols.	**Apply**

CO – PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	1	-	2	-
CO2	1	2	2	-
CO3	1	2	2	-
CO4	1	1	2	-
CO5	1	1	2	-

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus
First module (8 Lectures)

Introduction – Uses of computer networks, Network hardware, Network software, Reference models – The OSI reference model, The TCP/IP reference model.
Physical Layer – Modes of communication, Physical topologies, Signal encoding, Network devices, Transmission media. Performance indicators – Bandwidth, Throughput, Latency Queuing time, Bandwidth–Delay product.

Second module (8 Lectures)
Elementary Data Link Protocols, Error detection and correction, Sliding Window Protocols. Medium Access Control Layer - Channel Allocation Problem - Multiple Access Protocols.

Third module (8 Lectures)
Network layer Services, Datagram and Virtual circuit services, IP datagram format and Types of Services, The Original Classful Addressing Scheme Dotted Decimal Notation - Subnet and Classless Extensions - IP Multicast Addresses. ARP Protocol. Datagram encapsulation and Fragmentation, Reassembly and fragmentation, Routing Algorithms-Distance vector routing, Hierarchical routing, Link state routing, Broadcast routing.

Fourth module (**8 Lectures**)
Transport service – Services provided to the upper layers, Transport service primitives. User Datagram Protocol (UDP). Transmission Control Protocol (TCP) – Overview of TCP, TCP segment header, Connection establishment &release, Connection management modelling, TCP retransmission policy, TCP congestion control

Fifth module (8 Lectures)
Application layer Protocols: - WWW and HTTP, FTP, DNS, SMTP, P2P File sharing, Domain Name System (DNS).

References

1. AS Tanenbaum, DJ Wetherall, Computer Networks, 5th Ed., Prentice-Hall, 2010.
2. LL Peterson, BS Davie, Computer Networks: A Systems Approach, 5th Ed., Morgan-Kauffman, 2011.
3. JF Kurose, KW Ross, Computer Networking: A Top-Down Approach, 5th Ed., Addison-Wesley, 2009.
4. W Stallings, Cryptography and Network Security, Principles and Practice, 5th Ed., Prentice-Hall, 2010

Semester 6						
24-813-0604	**Natural Language Processing (Course Level 300-399)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**0**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Define the phases of traditional NLP as well as various NLP tasks	**Apply**
CO2	Apply Hidden Markov Models, and Naive Bayes models for various NLP tasks.	**Apply**
CO3	Apply word embedding techniques and N-gram language models for Named Entity Recognition	**Apply**
CO4	Apply deep learning models like LSTM, GRU for sequence modeling, and CNN for coreference resolution	**Apply**
CO5	Apply Seq2Seq models with attention mechanisms for natural language generation	**Apply**

CO – PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	2	-
CO2	3	3	3	-
CO3	3	3	3	-
CO4	3	3	3	-
CO5	3	3	3	-

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1

Introduction to NLP, Phases of Traditional NLP - Lexical Analysis, Syntactic Analysis, Semantic Analysis, Discourse Analysis, Pragmatic Analysis. Introduction to NLP Tasks - Parts-of-Speech Tagging, Word Sense Disambiguation, Anaphora Resolution, Text classification, Recognizing Textual Entailment, Named Entity Recognition

Module 2

Introduction to Statistical NLP. Vector Space Models - Bag-of-Words, TF-IDF weighing, PPMI. Basics of Supervised and Semi-supervised Learning for various NLP tasks - Noisy Channel Model for spelling correction. Hidden Markov Models for POS Tagging, Naive Bayes model for Text Classification.

Module 3

Introduction to Neural NLP - Word Embedding - Contextual and non-contextual Word Embedding. Subword embeddings. Evaluation of word vectors. N-gram language models. Neural Networks for named entity recognition - Word window classification.

Module 4

Recurrent neural networks for language modeling and other tasks, GRUs and LSTMs for machine translation, Question answering and dialogue system, Recursive neural networks for parsing, Convolutional neural networks for Coreference resolution.

Module 5

Natural Language Generation - Seq2Seq models - Attention - Case studies and real-world applications of NLP in various domains. Introduction to Large Language Models.

References

1. Dan Jurafsky and James H. Martin. Speech and Language Processing (2024 pre-release)
2. Jacob Eisenstein. Natural Language Processing
3. Yoav Goldberg. A Primer on Neural Network Models for Natural Language Processing
4. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning
5. Delip Rao and Brian McMahan. Natural Language Processing with PyTorch.
6. Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Natural Language Processing with Transformers

Semester 6						
24-813-0605	**Mini Project – 1 (Course Level 300-399)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**0**	**0**	**4**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Create innovative solutions to real world problems by applying advanced programming techniques with requirement analysis and identification of design methodologies	**Apply**
CO2	Function effectively as an individual and as a leader in diverse teams to plan and execute project utilizing available resources within timelines, following ethical and professional norms	**Apply**
CO3	Organize and communicate technical and scientific findings effectively in written and oral forms	**Analyze**

CO – PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	3	3	1
CO2	3	2	3	3
CO3	2	3	3	2

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

The objective of this course is to apply the fundamental concepts of Software Engineering principles for the effective development of an application/research project. This course helps the learners to practice the different steps to be followed in the software development process such as literature review and problem identification, preparation of Software Requirement Specification &Software Design Document (SDD), testing, development and deployment.

Guides are allotted at the beginning of the semester. A team consists of only one student. Student should identify a topic of interest in consultation with the Guide, review the literature and gather information pertaining to the chosen topic.

The guide/supervisor shall monitor the progress being carried out by the student on a regular basis. The students should record in the daily/weekly activity diary the day-to-day account of the observations, impressions, information gathered and suggestions given, if any. It should contain the review, design and all the observations made by the student and guide. The diary shall be signed every week by the guide.

The progress of the mini project is evaluated based on a minimum of two reviews by the committee. The review committee may be constituted by the Head of the Department. A project report is required at the end of the semester.

Preparing a paper for Conference/Publication in Journals is desirable for the successful completion of course. Students are also encouraged to present projects in Project Expos conducted at state level as well as others conducted in India and abroad

Semester 6						
24-813-0606	**Web Technologies (Course Level 200-299)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		CS SEC	**3**	**1**	**2**	**3**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Develop interactive Web pages using HTML/XHTML.	**Apply**
CO2	Present a professional document using Cascaded Style Sheets	**Apply**
CO3	Construct websites for user interactions using JavaScript and JQuery	**Apply**
CO4	Know the different information interchange formats like XML and JSON.	**Apply**
CO5	Develop Web applications using PHP.	**Apply**

CO – PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	2	3	1	-
CO2	2	3	1	-
CO3	2	3	1	-
CO4	2	3	1	-
CO5	2	3	1	-

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1(7 Lectures)

Introduction to HTML/XHTML : Origins and Evolution of HTML and XHTML, Basic Syntax of HTML, Standard HTML Document Structure, Basic Text Markup, Images, Hypertext Links, Lists, Tables, Forms, HTML5, Syntactic Differences between HTML and XHTML.

Module 2(6 Lectures)

Introduction to Styles sheets and Frameworks: Cascading Style Sheets: Levels of Style Sheets - Style Specification Formats, Selector Forms, Property-Value Forms, Font Properties, List Properties, Alignment of Text, Color, The Box Model, Background Images, The span and div Tags.

Module 3 (7 Lectures)

The Basics of JavaScript: Overview of JavaScript, Object Orientation and JavaScript, General Syntactic Characteristics-Primitives, Operations, and Expressions, Screen Output and Keyboard Input, Control Statements, Object Creation and Modification, Arrays, Functions. Callback Functions, Java Script HTML DOM. Introduction to jQuery: Overview and Basics.

Module 4(6 Lectures)

XML: The Syntax of XML, XML Document Structure, Namespaces, XML Schemas, Displaying Raw XML Documents, Displaying XML Documents with CSS, XSLT Style Sheets, XML Applications. JSON(Basics Only): Overview, Syntax, Datatypes, Objects, Schema, Comparison with XML.

Module 5 (5 Lectures)

Introduction to PHP: Origins and Uses of PHP, Overview of PHP - General Syntactic Characteristics - Primitives, Operations, and Expressions - Control Statements, Arrays, Functions, Pattern Matching, Form Handling, Cookies, Session Tracking.

References

1. P. J. Deitel, H.M. Deitel, Internet &World Wide Web How To Program, 4/e, Pearson International Edition 2010.
2. Robert W Sebesta, Programming the World Wide Web, 7/e, Pearson Education Inc., 2014.
3. Bear Bibeault and Yehuda Katz, jQuery in Action, Second Edition, Manning Publications.[Chapter 1] Black Book, Kogent Learning Solutions Inc. 2009.
4. Bob Boiko, Content Management Bible, 2nd Edition, Wiley Publishers. [Chapter 1, 2]
5. Chris Bates, Web Programming Building Internet Applications, 3/e, Wiley India Edition 2009.
6. Dream Tech, Web Technologies: HTML, JS, PHP, Java, JSP, ASP.NET, XML, AJAX,

Semester 7						
24-813-0701	**Deep Learning (Course Level 400-499)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**0**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Apply basic supervised learning algorithms and optimization techniques for classification task.	**Apply**
CO2	Apply techniques for regularizing deep networks and also be proficient in model exploration and hyperparameter tuning.	**Apply**
CO3	Demonstrate the working of Convolution Operation, Sparse interactions, Parameter sharing, Equivariant representations, and Pooling.	**Apply**
CO4	Apply deep recurrent networks such as Long Short-Term Memory (LSTM) and other Gated RNNs for sequence modeling tasks.	**Apply**
CO5	Understand different types of Autoencoders including undercomplete, regularized, sparse, and denoising autoencoders.	**Understand**

CO – PSO Mapping

CO	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	**3**	**1**	**2**	**2**
CO2	**3**	**2**	**3**	**2**
CO3	**3**	**3**	**3**	**3**
CO4	**3**	**2**	**2**	**2**
CO5	**3**	**1**	**1**	**1**

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1

Introduction: Historical context and motivation for deep learning; basic supervised classification task, optimizing logistic classifier using gradient descent, stochastic gradient descent, momentum, and adaptive sub-gradient method.

Module 2

Neural Networks: Feedforward neural networks, deep networks, regularizing a deep network, model exploration, and hyperparameter tuning.

Module 3

Convolution Neural Networks: Introduction to convolution neural networks: stacking, striding and pooling, applications like image, and text classification.

Module 4

Sequence Modeling: Recurrent Nets: Unfolding computational graphs, recurrent neural networks (RNNs), bidirectional RNNs, encoder-decoder sequence to sequence architectures, deep recurrent networks - Long Short-Term Memory and Other GatedRNNs.

Module 5

Autoencoders: Undercomplete autoencoders, regularized autoencoders, sparse autoencoders, denoising autoencoders, representational power, layer, size, and depth of autoencoders, stochastic encoders, and decoders.

References

1. Ian Goodfellow, Deep Learning, MIT Press, 2016.
2. Jeff Heaton, Deep Learning and Neural Networks, Heaton Research Inc, 2015.
3. Mindy L Hall, Deep Learning, VDM Verlag, 2011

Semester 7						
24-813-0702	**BigData Analytics** **(Course Level 400-499)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**2**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Understand the fundamental concepts of BigData	**Understand**
CO2	Understand about Hadoop and its ecosystem	**Understand**
CO3	Apply Bigdata analysis using PIG, HIVE and Spark	**Apply**

CO – PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	**3**	**1**	**2**	**2**
CO2	**3**	**1**	**2**	**2**
CO3	**3**	**2**	**3**	**2**

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1 (8 Lectures)

Types of Digital Data, Big Data Characteristics, Types of Big Data, Infrastructure for Big Data, Big Data Challenges,Big Data Analytics, Application of Big data analytics,
History of Hadoop, Apache Hadoop, Analysing Data with Unix tools, Analysing Data with Hadoop, Hadoop Streaming,

Module 2 (8 Lectures)

Hadoop Echo System, Hadoop file system interfaces, Data flow Map Reduce algorithm, Failures, Job Scheduling, Shuffle and Sort.

Module 3 (8 Lectures)

Pig : Introduction to PIG, Execution Modes of Pig, Comparison of Pig with Databases, Grunt, Pig Latin, User Defined Functions, Data Processing operators.

Module 4 (**8 Lectures**)

Hive Architecture Comparison with Traditional Database, HiveQL Querying Data, Sorting And Aggregating, Map Reduce Scripts, Joins & Sub queries, HBase concepts, Advanced Usage, Schema Design, Advance Indexing

Module 5 (8 Lectures)

Spark programming. (Python and PySpark), Spark - Resilient Distributed Dataset (RDDs). Spark, RDDs, DataFrames, Spark SQL, PySpark + NumPy + SciPy, Code Optimization, Cluster Configurations

References

1. Big Data: A Revolution That Will Transform How We Live, Work, and Think by Viktor Mayer-Schonberger and Kenneth Cukier (Houghton Mifflin Harcourt, 2013)
2. **Pig: The Definitive Guide** by Julian Alvin Shaun Oak and Eric Sammer (O'Reilly Media, 2014)
3. **Learning Apache Hive** by Edward Capriolo, Noah Mischianti, and Joshua Wilson (O'Reilly Media, 2015):
4. **Hive Query Language: The Essential Guide** by Teja Deshpande and Ashish Thusoo (O'Reilly Media, 2011):
5. **High Performance Spark** by Holden Karau, Rachel Warren, and Matei Zaharia (O'Reilly Media, 2016)
6. **Learning Spark: Lightning-Fast Big Data Analytics** by Holden Karau, Rachel Warren, and Patrick Wendell (O'Reilly Media, 2015):
7. https://spark.apache.org/docs/latest/quick-start.html

Semester 7						
24-813-0703	**Cloud computing and Virtualization** **(Course Level 400-499)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**0**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Understand various basic concepts related to cloud computing technologies.	**Understand**
CO2	Analyse benefits of virtualization for computing	**Analyse**
CO3	Explore cloud technologies, architectures, and standards	**Analyse**
CO4	Understand security vulnerabilities of cloud and apply solutions	**Understand**

CO – PSO Mapping

CO	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	**3**	**1**	**1**	**1**
CO2	**3**	**2**	**2**	**1**
CO3	**3**	**3**	**3**	**3**
CO4	**3**	**2**	**2**	**3**

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1(8 Lectures)

Evolution of Computing: On-premise computing, client-server model, Distributed computing, multi-core computing. Virtualization: virtual machines, Desktop virtualization, hypervisor, microkernel, full and para virtualization. Benefits of cloud computing, Edge and fog computing, MQTT.

Module 2(10 Lectures)

Cloud architecture: Layers in cloud architecture, Hosting and management of applications. Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS). Scalability and reliability in cloud. Examples for each model. SLAs. Hybrid cloud. Hyperconverged infrastructure.

Module 3 (8 Lectures)

Usage of cloud: AWS/ Azure/ GCP. Use of lambda or cloud functions as API. Storage of data in cloud. SCSI, SAN, NAS, etc. Software defined storage. Disaster recovery. Distributed File Systems (HDFS, Ceph FS), Cloud Databases (HBase, MongoDB, Cassandra, DynamoDB), Cloud Object Storage (Amazon S3, OpenStack Swift, Ceph). Batch cloud computing: map-reduce and Hadoop.

Module 4(10 Lectures)

Web and Mobile applications communicating with cloud. Microservices vs Monolithic architectures. Applications of cloud computing healthcare, smart homes, smart grid, etc. Continuous Integration and Continuous deployment in Cloud: Automated build management, deployment and monitoring of applications. Clusters, Kubernetes, Use of Containers and docker.

Module 5 (8 Lectures)

Cloud security: Authentication and Authorization, Tokens, API Key, Identity and Access Management in cloud. Threat analysis for IoT: Types of Cyber Attacks on cloud and IoT and techniques to prevent such attacks. Securing IoT and Cloud: Encryption of data, symmetric and asymmetric key encryption. Digital Signatures and certificates.

References

1. Toby Velte, Anthony Velte, Robert Elsenpeter: Cloud Computing, A Practical Approach, 1e, McGraw-Hill Education, 2009.

2. Rajkumar Buyya, James Broberg, Andrzej Goscinski: Cloud Computing: Principles and Paradigms, 1e, Wiley, 2013.

3. Giacomo Veneri and Antonio Capasso, Hands-On Industrial Internet of Things: Create a powerful Industrial IoT infrastructure using Industry 4.0, 1st Edition, Packt Publishing, 2018.

4. Mayur Ramgir, Internet of Things: Architecture, Implementation and Security, 1st Edition, Pearson, 2019.

5. R. Buyya, S N. Srirama, Fog and Edge Computing: Principles and Paradigms, Wiley Series on Parallel and Distributed Computing, 1st Edition, Wiley, 2019.

6. Edward A. Lee and Sanjit A. Seshia, Introduction to Embedded Systems, A Cyber-Physical Systems Approach, 2nd Edition, MIT Press, 2017.

Semester 7						
24-813-0704	**Cyber Security** **(Course Level 400-499)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**0**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Understand foundational concepts in cybersecurity, including principles of confidentiality, integrity, and availability, and their application in securing information systems.	**Understand**
CO2	Identify common cyber threats, vulnerabilities, and attack vectors, and apply appropriate security measures to mitigate risks.	**Analyze**
CO3	Demonstrate proficiency in network security principles, including secure network design, implementation of access controls, and detection and prevention of network-based attacks.	**Apply**
CO4	Analyze and assess security risks in operating systems, applications, and network infrastructures, and develop strategies to address identified vulnerabilities.	**Analyze**
CO5	Apply cryptographic techniques to ensure the confidentiality, integrity, and authenticity of data in transit and at rest.	**Apply**
CO6	Develop incident response plans and procedures to effectively detect, respond to, and recover from cybersecurity incidents.	**Create**
CO7	Apply ethical hacking methodologies to identify and exploit security weaknesses in information systems, and recommend appropriate countermeasures.	**Apply**
CO8	Understand legal and ethical considerations in cybersecurity, including compliance with relevant laws, regulations, and ethical standards.	**Understand**

CO – PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	**3**	**-**	**1**	**1**
CO2	**3**	**2**	**1**	**2**
CO3	**3**	**2**	**2**	**2**
CO4	**3**	**2**	**2**	**1**
CO5	**3**	**2**	**2**	**1**
CO6	**3**	**2**	**2**	**2**
CO7	**3**	**3**	**3**	**3**
CO8	**3**	**2**	**2**	**1**

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1(8 Lectures)

Overview of Cybersecurity: Introduction to cybersecurity concepts, importance, and challenges. Security Principles: Understanding security principles, CIA triad (Confidentiality, Integrity, Availability), and security models. Threat Landscape: Exploring common cyber threats, including malware, phishing, DDoS attacks, and social engineering. Risk Management: Introduction to risk assessment, risk mitigation strategies, and risk management frameworks.

Module 2(10 Lectures)

Network Fundamentals: Basics of networking, OSI model, TCP/IP protocol suite, and network devices. Network Attacks and Defense: Common network attacks (e.g., Man-in-the-Middle, DoS attacks), and network defense mechanisms (e.g., firewalls, IDS/IPS). Secure Network Design: Principles of secure network design, subnetting, VLANs, and DMZ configuration. Cryptography in Network Security: Introduction to cryptographic techniques used in securing network communication (e.g., encryption, digital signatures, key exchange).

Module 3 (8 Lectures)

Operating System Fundamentals: Overview of operating systems, user authentication, access control mechanisms, and file systems. OS Hardening: Techniques for hardening operating systems to improve security, including patch management, disabling unnecessary services, and using secure configurations. Endpoint Security: Endpoint protection mechanisms, antivirus

software, intrusion detection system(IDS), and host-based firewalls. Secure Administration: Best practices for secure system administration, including privilege management, logging, and auditing.

Module 4(10 Lectures)

Secure Software Development Lifecycle (SDLC): Introduction to secure SDLC phases, including requirements analysis, design, implementation, testing, and maintenance. Web Application Security: Common web vulnerabilities (e.g., SQL injection, XSS, CSRF) and techniques for securing web applications (e.g., input validation, parameterized queries). Secure Coding Practices: Best practices for writing secure code, secure coding guidelines, and code review techniques. Application Security Testing: Overview of security testing techniques, including static analysis, dynamic analysis, and penetration testing.

Module 5 (8 Lectures)

Incident Response Planning: Developing an incident response plan, incident detection and classification, and incident response phases. Digital Forensics: Introduction to digital forensics principles, evidence collection, preservation, and analysis. Ethical Hacking: Overview of ethical hacking methodologies, penetration testing techniques, and tools. Legal and Ethical Considerations: Understanding legal and ethical issues in cybersecurity, including laws, regulations, and professional codes of conduct.

References

1. William Stallings and Lawrie Brown - "Computer Security: Principles and Practice" (Pearson, 4th Edition, 2017)

2. William Stallings - "Network Security Essentials: Applications and Standards" (Pearson, 7th Edition, 2017)

3. Mike Chapple, James Michael Stewart, and Darril Gibson - "CISSP (ISC)2 Certified Information Systems Security Professional Official Study Guide" (Sybex, 8th Edition, 2018)

4. Dafydd Stuttard and Marcus Pinto - "The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" (Wiley, 2nd Edition, 2011)

5. Jon Erickson - "Hacking: The Art of Exploitation" (No Starch Press, 2nd Edition, 2008)

6. Michael Sikorski and Andrew Honig - "Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software" (No Starch Press, 1st Edition, 2012)

7. Ross J. Anderson - "Security Engineering: A Guide to Building Dependable Distributed Systems" (Wiley, 2nd Edition, 2008)

Semester 7						
24-813-0705	**Image Processing and Computer Vision** **(Course Level 400-499)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**2**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Understand the fundamental concepts of signal and image processing systems.	**Understand**
CO2	Evaluate the different spatial and frequency domain filters for image enhancement and restoration.	**Apply**
CO3	Evaluate the performance of periodic noise reduction filters and image segmentation algorithms.	**Evaluate**
CO4	Understand the fundamental theories and techniques of computer vision and summarize different color and texture based feature extraction methods used for computer vision.	**Understand**
CO5	Analyse different methods to compute the motion of an object from 2D image sequences and understand the process of the depth information from stereo images.	**Analyse**

CO – PSO Mapping

CO	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	**3**	**1**	**1**	**1**
CO2	**3**	**2**	**2**	**2**
CO3	**3**	**3**	**3**	**3**
CO4	**3**	**3**	**3**	**2**
CO5	**3**	**3**	**3**	**2**

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1(8 Lectures)

Signals: Impulse Sequence - Exponential Sequence - Periodic Sequence. Linear Systems - Shift-Invariant systems - Linear Shift Invariant (LSI) systems – Convolution - Correlation. Image Transforms: Fourier Transform - Discrete Fourier Transform - Z- transform – KL Transform. Causal Systems - Random Signals - Stationary Process - Markov Process.

Module 2(10 Lectures)

Intensity Transformation and Spatial Filtering: Intensity Transformation Functions. Histogram Processing: Histogram Equalization - Histogram Matching. Image enhancement: Arithmetic/Logic operations - Image Subtraction - Image Averaging. Spatial Filtering:

Module 3 (8 Lectures)

Image degradation/Restoration process model - Noise probability density functions - Spatial Filtering: Mean Filters - Order-statistics filter - Adaptive Filters - Periodic Noise Reduction –Band-reject filters - Band-pass filters - Notch filters. Inverse filtering – Wiener

Module 4(10 Lectures)

Computer Vision: 3D structure from 2D images, Five frames of reference. Binary Image Analysis: Pixels and Neighborhoods, Applying masks to images, Counting the objects in an image, Connected components labeling. Binary image morphology, Region properties, Region adjacency graphs. Feature detection and matching: Points and patches, SIFT, SURF. Texture: Texture, Texels and Statistics, Texel based Texture Descriptions, Quantitative Texture Measures, Texture Segmentation.

Module 5 (8 Lectures)

Content based image retrieval: Image distance measures: Color similarity,Texture similarity, Shape similarity. Motion from 2D image sequences: Computing Motion Vectors. Matching in 2D: Registration of 2D data, Representation of points, Affine

References/Text Books

1. Rafael C. Gonzalez, Richard E. Woods, “Digital Image Processing”, 4th Ed., Pearson, March 2017.
2. Anil K.Jain, “Fundamentals of Digital Image Processing”, Pearson, 1st Ed., 1988.
3. William K. Pratt, “Digital Image Processing: PIKS Scientific Inside”, John Wiley & Sons, 4th Ed., 2007.
4. Azriel Rosenfield, Avinash C. Kak, "Digital Picture Processing", Morgan Kaufmann, 2nd Ed., 1982.
5. Bernd Jahne, “Digital Image Processing”, Springer, 6th Ed., 2005.
6. Linda G. Shapiro, George C. Stockman, “Computer Vision”, Prentice Hall, 1st Ed., 2001.
7. Richard Szeliski, “Computer Vision: Algorithms and Applications”, Springer, 1st Ed., 2010.
8. David A. Forsyth, Jean Ponce, “Computer Vision: A Modern Approach”, 2nd Ed., 2011.
9. Simon J. D. Prince, “Computer Vision: Models, Learning, and Inference”, Cambridge University Press, 1st Ed., 2012.
10. Ramesh Jain, RangacharKasturi, Brian G. Schunck, “Machine Vision”, McGraw-Hill, 1st Ed., 1995.

Semester 7						
24-813-0706	**SEMINAR (Course Level 400-499)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
			0	**0**	**2**	**2**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Identify, read, and interpret an academic research article from the literature that is related to his/her academic area of interest and present it before the committee.	**Understand**
CO2	Organize and communicate technical and scientific findings effectively in written and oral forms to technical and non-technical stakeholders.	**Analyze**
CO3	Demonstrate the academic discussion skills to emphasize, argue with clarity of purpose using evidence for the claims.	**Apply**

CO – PSO Mapping

CO	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	**3**	**1**	**1**	**1**
CO2	**2**	**1**	**1**	**1**
CO3	**1**	**1**	**1**	**1**

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

The student has to prepare and deliver a presentation on a research topic suggested by the department before the peer students and expert committee. They also have to prepare a comprehensive report of the seminar presented

Semester 8						
24-813-0801	**Bioinformatics** **(Course Level 400-499)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**4**	**1**	**0**	**4**

Course Outcomes (CO)

After the completion of the course, the students will be able to:

CO1	Understand and appreciate basic concepts of molecular Biology and Human genome project.	**Understand**
CO2	Illustrate and explain various sequence alignment algorithms.	**Apply**
CO3	Demonstrate and evaluate different algorithms for identifying optimal phylogenetic trees.	**Analyze**
CO4	Understand the concepts of structure prediction in molecular biology	**Understand**
CO5	Understand and demonstrate an algorithm in the literature for the domain.	**Analyze**

CO- PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	1	1	-
CO2	3	2	1	-
CO3	3	3	2	-

CO4	3	2	2	-
CO5	3	1	3	3

Syllabus

Module 1 (8 Lectures)

Bioinformatics introduction-Branches of bioinformatics-Basic concepts of molecular Biology Proteins-Nucleic acids– genes and genetic synthesis – translation- transcription- protein SynthesisChromosomes- Maps and sequences- Biological databases

Module 2 (8 Lectures)

Sequence alignment-Concepts of alignment-Gap Penalty-Pairwise sequence alignment algorithms Dot Matrix-Global & Local alignment-Multiple sequence alignment algorithms-Scoring matrices PAM, BLOSUM-Heuristic Methods -BLAST-FASTA

Module 3 (8 Lectures)

Fragment Assembly of DNA - Biological Background-human genome project – Models -Algorithms - Heuristics - Physical Mapping of DNA - Internal Graph Models – Hybridization Mapping - Heuristics - Genome rearrangements-Oriented Blocks- unoriented Blocks

Module 4 (**8 Lectures**)

Molecular Phylogeny-Phylogenetic Trees –Methods of phylogeny-Maximum Parsimony-Maximum Likelihood-Distance methods-Binary Character States- Perfect phylogeny

Module 5 (8 Lectures)

Molecular Structure Prediction- Secondary structure prediction-Protein Folding problems-Protein threading-Computing with DNA-Hamilton Path Problems-Computer aided Drug design- peptide drug-chemical drug

References

1. Rastogi, S. C., Parag Rastogi, and Namita Mendiratta. Bioinformatics: Methods and ApplicationsGenomics, Proteomics and Drug Discovery. PHI Learning Pvt. Ltd., 5e, 2022.
2. Neil James and Pavel A Pevzner, An introduction to Bioinformatics Algorithms, 4e, OUPress, 2014
3. ZhumurGhosh, BibekanandMallick , Bioinformatics : Principles and Applications, OUPress, 2015

4. Concord Bessant, Darren Oakley, Ian Shadforth, Building Bioinformatics Solutions, OUPress, 2014

5. Peter Clote and Rolf Backofen, Computational Molecular Biology-An introduction, 1e, Wiley Series, 2000

Semester 8						
24-813-0802	**Research Project (Course Level 400-499)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**0**	**0**	**12**	**12**

Course Outcomes (CO)

After the completion of the course, the students will be able to:

CO1	Identify technology/research gaps and propose creative solutions	**Create**
CO2	Create solutions to real world problems by performing requirement analysis and identification of design methodologies	**Create**
CO3	Function effectively as an individual and as a leader in diverse teams to plan and execute project utilizing available resources within timelines, following ethical and professional norms	**Apply**
CO4	Organise and communicate technical and scientific findings effectively in written and oral forms	**Apply**

CO - PSO Mapping

CO	PSO 1	PSO2	PSO3	PSO4
CO1	2	3	3	1
CO2	3	2	3	1
CO3	1	1	2	3
CO4	2	3	2	1

The course 'Project Work' is mainly intended to evoke the research, innovation and invention skills in a student. The course will provide an opportunity to synthesize and apply the knowledge and analytical skills learned, to be developed as a prototype or simulation.

The guide/supervisor shall monitor the progress being carried out by the student on a regular basis. The students should record in the daily/weekly activity diary the day-to-day account of the observations, impressions, information gathered and suggestions given, if any. It should contain the review, design and all the observations made by the student and guide. The diary shall be signed every week by the guide

Semester 8						
24-813-0803	**Full Stack AI Lab (Course Level 400-499)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		LAB	**0**	**0**	**2**	**2**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Gain practical experience across the Full Stack AI development lifecycle	**Analyse**
CO2	Master data engineering, preprocessing, and deploying AI models	**Apply**
CO3	Develop secure, cross-platform applications and complete a capstone project demonstrating acquired skills.	**Create**

CO – PSO Mapping

CO	PSO1	PSO2	PSO3	PS04
CO1	3	2	3	3
CO2	3	2	3	3
CO3	3	2	3	3

Indicative Experiments

1. Implement a program for data source exploration.
2. Implement a program for data cleaning and preprocessing.
3. Implement a program for data analysis and feature Engineering.
4. Implement different supervised learning models and evaluate its performance.
5. Implement different unsupervised learning models and evaluate its performance.

6. Implement the different model optimization techniques
7. Implement a simple deep learning model and saving as well as loading models.
8. Building Mobile Libraries (iOS/Android).
9. Cross-Platform Model Deployment.
10. Building Web /Mobile UIs with a backend API for data processing.
11. Creating Web Services (RESTful API)
12. Continuous Integration and Deployment
13. Implement the security best practices in Full Stack AI
14. Implement a capstone project to develop a real-world Full Stack AI application.

Semester 8						
24-813-0804	**Advanced Optimization Techniques** **(Course Level 400-499)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSE	**4**	**1**	**0**	**4**

Course Outcomes (CO)

After the completion of the course, the students will be able to :

CO1	Understand the basic concepts of optimization and its applications.	**Understand**
CO2	Understand the mathematical representation and classical methods for solving optimization	**Understand**
CO3	Explain and demonstrate working principles of various population-based optimization techniques	**Apply**
CO4	Explain and demonstrate working principle of various Hybrid Algorithms for optimization	**Apply**

CO - PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	2	-
CO2	3	3	2	-
CO3	3	2	1	-
CO4	3	3	3	2

Syllabus

Module 1(8 Lectures)

Introduction to optimization- formulation of optimization problems-Review of classical

methods-Linear programming- Nonlinear programming-Constraint optimality criteria constrained optimization-Population based optimization techniques.

Module 2(8 Lectures)

Genetic Algorithm - Introduction - Working principle - Representation - selection – fitness assignment - reproduction - crossover - mutation - constraint handling -advanced genetic algorithms - Applications - Artificial Immune Algorithm - Introduction- Clonal selection algorithm- Negative selection algorithm - Immune network algorithms - Dendritic cell algorithms.

Module 3(8 Lectures)

Differential Evolution - Introduction - Working principles - parameter selection - advanced algorithms in Differential evolution - Biogeography-based Optimization - Introduction - Working Principles - Algorithmic variations.

Module 4(8 Lectures)

Particle Swarm Optimization-Introduction- Working principles- Parameter selection Neighborhoods and Topologies-Convergence-Artificial Bee Colony Algorithm-Introduction Working principles-Applications-Cuckoo search based algorithm-Introduction- Working principles- Random walks and the step size-Modified cuckoo search.

Module 5(8 Lectures)
Hybrid Algorithms-Concepts- divide and conquer- decrease and conquer-HPABC-HBABC- HDABC-HGABC-Shuffled Frog Leaping Algorithm - Working principles - Parameters- Grenade Explosion Algorithm-Working principle-Applications

References

1. Dan Simon, Evolutionary Optimization Algorithms, 1e, Wiley, 2013
2. Xin-She Yang, Engineering Optimization: An Introduction with Meta-heuristic Applications, 1e, Wiley, 2010
3. S.S. Rao, Engineering Optimization: Theory and Practice, 4e,New Age International, 2013
4. R. VenkataRao, Teaching Learning Based Optimization Algorithm: And Its Engineering Applications, 1e, Springer, 2016

Semester 8						
24-813-0805	**Blockchain Technology (Course Level 400-499)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSE	**4**	**1**	**0**	**4**

Course Outcomes (CO)

After the completion of the course, the students will be able to:

CO1	Understand the fundamentals of blockchain technology	**Understand**
CO2	Understand the essentials of Bitcoin and beholding bitcoins as blockchains	**Understand**
CO3	Analyze and design the Ethereum Blockchain	**Analyze**
CO4	Analyze the powers of blockchains and their applications in various	**Analyze**
CO5	Execute a mini project on blockchain	**Apply**

CO - PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	2	-
CO2	3	3	2	-
CO3	3	2	1	-
CO4	3	3	3	2

Syllabus

Module 1(8 Lectures)

Introduction to blockchain: Structure of blockchains, Blockchain life cycle, working of a blockchain, picking a blockchain, exploring blockchain applications, building trust with blockchains, Blockchain in action: Use cases, introducing bitcoin blockchains.

Module 2(10 Lectures)

Bitcoin & Ethereum blockchains: Understanding bitcoins, comprehending bitcoins as blockchains, analyzing Ethereum blockchains, introducing ripple and factom blockchains and their importance

Module 3 (8 Lectures)

Powerful blockchain platforms: Getting introduced to Hyperledger, Hyperledger vision, Hyperledger sawtooth, understanding the blockchain fabric, understanding business, and smart blockchains, IBM Blockchains, Stellar: an optimized blockchain

Module 4(10 Lectures)

Industry impacts of blockchains: Blockchains in financial technology, Blockchains in various industries such as insurance, Government, Real-estate, health care, Telecommunication, Transportation, etc..

Module 5 (8 Lectures)

Case Study and mini-project: Study different blockchain projects as a case study and submit a report and present the work, design a blockchain application as a mini-project, and presenting the work.

References

1. Blockchain and Crypto Currency, Editors: Makoto YanoChris DaiKenichi MasudaYoshio Kishimoto, 1st Edition, Springer, 2020.
2. Blockchain or Dummies, Tiana Laurence, 1st Edition , John Wiley & Sons, Inc, , 2017.
3. Blockchain Blueprint for a new economy, Melanie Swan, 1st Edition,O'Reilly, 2017.
4. Blockchain Technology: Applications and Challenges, Panda, S.K., Jena, A.K., Swain, S.K., Satapathy, S.C. , 1st Edition, Springer, 2021
5. Blockchain and Distributed Ledgers, Alexander Lipton and Adrien Treccani, 1st Edition, World Scientific Press, 2021

Semester 8						
24-813-0806	**Information Retrieval and Web Search** **(Course Level 400-499)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSE	**4**	**1**	**0**	**4**

Course Outcomes (CO)

After the completion of the course, the students will be able to:

CO1	Understand advanced techniques for text-based information retrieval.	**Understand**
CO2	Understand Boolean and vector space retrieval models	**Understand**
CO3	Evaluate various text classification techniques	**Evaluate**
CO4	Understand Web search characteristics, web crawling and link analysis	**Understand**
CO5	Build working systems that assist users in finding useful information on the Web	**Apply**

CO - PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	2	-
CO2	3	3	2	-
CO3	3	2	1	-
CO4	3	3	3	2
CO5	3	3	3	2

Syllabus

Module 1(8 Lectures)

Taxonomy of IR Models – Classic models- Set theoretic model- Algebraic models- Probabilistic modelStructured text retrieval models- Models for browsing- Retrieval evaluations-Reference collections

Module 2(8 Lectures)

Query languages-query operations-text and multimedia languages-Text operations-document preprocessing-matrix decompositions and latent semantic indexing-text compression –indexing and searching-inverted files-suffix trees- Boolean queries-sequential searching-pattern matching

Module 3(8 Lectures)

Text Classification, and Naïve bayes-vector space classification-support vector machines and machine learning on documents-flat clustering –hierarchical clustering

Module 4(8 Lectures)

Web search basics-web characteristics-index size and estimation- near duplicates and shingling-web crawling-distributing indexes- connectivity servers-link analysis-web as a graph- PageRank-Hubs and authorities-question answering

Module 5(8 Lectures)

Online IR systems- online public access catalogs-digital libraries-architectural issues-document models - representations and access- protocols

References

1. 1. Ricardo Baezce Yates, BerthierRibeiro-Neto , Modern Information Retrieval: The Concepts and Technology behind Search, 3e, ACM Press, 2017

2. Christopher D. Manning, PrabhakarRaghavan and HinrichSchütze , Introduction to Information Retrieval, 1e, Cambridge University Press, 2008

3. Bruce Croft, Donald Metzler and Trevor Strohman, Search Engines: Information Retrieval in Practice, 1e, AW, 2009

Semester 8						
24-813-0807	**Number Theory and Cryptography** **(Course Level 400-499)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSE	**4**	**1**	**0**	**4**

Course Outcomes (CO)

After the completion of the course, the students will be able to :

CO1: Understand the fundamental principles of number theory and their applications in cryptography.

CO2: Apply modular arithmetic concepts to solve cryptographic problems and analyze cryptographic algorithms.

CO3: Demonstrate proficiency in basic cryptographic techniques, including symmetric and asymmetric encryption, hash functions, and cryptographic protocols.

CO4: Evaluate the security of cryptographic systems using number theory-based concepts such as primality testing and factorization algorithms.

CO5: Design and implement cryptographic solutions using advanced cryptographic techniques, including elliptic curve cryptography, digital signatures, and zero-knowledge proofs.

CO6: Analyze and critique cryptographic protocols and their applications in secure communication, digital signatures, and authentication.

CO7: Investigate emerging cryptographic technologies such as blockchain, quantum cryptography, and their impact on the future of secure communication and data protection.

CO8: Develop critical thinking and problem-solving skills through practical exercises, assignments, and a final project that integrates theoretical knowledge with real-world applications in cryptography

Course Outcomes (CO)

After the completion of the course, the students will be able to:

CO1	Understand the fundamental principles of number theory and their applications in cryptography.	**Understand**
CO2	Apply modular arithmetic concepts to solve cryptographic problems and analyze cryptographic algorithms.	**Analyze**
CO3	Demonstrate proficiency in basic cryptographic techniques, including symmetric and asymmetric encryption, hash functions, and cryptographic protocols.	**Apply**

CO4	Evaluate the security of cryptographic systems using number theory-based concepts such as primality testing and factorization algorithms.	**Evaluate**
CO5	Design and implement cryptographic solutions using advanced cryptographic techniques, including elliptic curve cryptography, digital signatures, and zero-knowledge proofs.	**Apply**
CO6	Analyze and critique cryptographic protocols and their applications in secure communication, digital signatures, and authentication.	**CO6: Analyze**
CO7	Investigate emerging cryptographic technologies such as blockchain, quantum cryptography, and their impact on the future of secure communication and data protection.	**Evaluate**
CO8	Develop critical thinking and problem-solving skills through practical exercises, assignments, and a final project that integrates theoretical knowledge with real-world applications in cryptography	**Analyze**

CO - PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	2	-
CO2	3	3	2	-
CO3	3	2	1	-
CO4	3	3	3	2
CO5	3	3	3	2
CO6	3	2	2	-
CO7	3	3	2	-
CO8	3	3	2	-

Syllabus

Module 1(8 Lectures)

Finite Fields: Groups, Rings and Fields. Overview of Number Theory: Introduction to prime numbers, composite numbers, and basic divisibility properties, greatest common divisor (GCD), and least common multiple (LCM). Modular Arithmetic: Understanding modular arithmetic, congruences, and

arithmetic operations modulo n. Prime Numbers: Properties of prime numbers, prime factorization, and fundamental theorems of arithmetic, Primality testing and factorization.

Module 2(10 Lectures)

Fermat's Little Theorem and Euler's Totient Function: Understanding their applications in cryptography, especially in RSA encryption and decryption. Diffie-Hellman Key Exchange: Principles and protocols of key exchange based on number theory concepts. Primality Testing: Introduction to primality testing algorithms, including probabilistic and deterministic methods. Cryptanalysis Techniques: Basic cryptanalysis techniques such as frequency analysis and brute force attacks. Quadratic Residues & Arithmetic Functions: Quadratic Residues- Quadratic Congruences, The group of Quadratic residues, Legendre symbol, Jacobi Symbol, Quadratic reciprocity.

Module 3 (8 Lectures)

Introduction to Cryptography: History, evolution, and importance of cryptography. Symmetric Encryption: Principles of symmetric key encryption, substitution ciphers, and transposition ciphers. Asymmetric Encryption: Concepts of asymmetric key encryption, RSA algorithm, and public-key cryptography. Cryptographic Hash Functions: Understanding hash functions, properties, and applications in digital signatures and data integrity.

Module 4(10 Lectures)

Elliptic Curve Cryptography: Introduction to elliptic curve cryptography, elliptic curve operations, and applications in modern cryptographic systems. Digital Signatures: Principles of digital signatures, digital signature schemes, and applications in authentication and non-repudiation. Zero-Knowledge Proofs: Overview of zero-knowledge proofs, protocols, and their applications in cryptographic protocols like secure authentication and identification.

Module 5 (8 Lectures)

Secure Communication Protocols: Overview of secure communication protocols such as SSL/TLS, SSH, and IPSec. Cryptographic Applications: Real-world applications of cryptography in secure messaging, online transactions, and digital certificates. Blockchain and Cryptocurrency: Introduction to blockchain technology, cryptographic principles in blockchain consensus mechanisms, and cryptocurrency fundamentals. Quantum Cryptography: Basics of quantum cryptography, quantum key distribution, and implications for future cryptographic systems.

References

1. Ivan Niven, Herbert S. Zuckerman, Hugh L. Montgomery - "Introduction to the Theory of Numbers" (Wiley, 5th Edition, 1991)

2. David M. Burton - "Elementary Number Theory" (McGraw-Hill Education, 7th Edition, 2010)

3. Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman - "An Introduction to Mathematical Cryptography" (Springer, 2nd Edition, 2014)

4. William Stallings - "Cryptography and Network Security: Principles and Practice" (Pearson, 7th Edition, 2016)

5. Christof Paar, Jan Pelzl - "Understanding Cryptography: A Textbook for Students and Practitioners" (Springer, 3rd Edition, 2010)

6. Lawrence C. Washington - "Elliptic Curves: Number Theory and Cryptography" (Chapman and Hall/CRC, 2nd Edition, 2008).

7. Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone - "Handbook of Applied Cryptography" (CRC Press, 1st Edition, 1996)

Semester 8						
24-813-0808	**Large Language Models** **(Course Level 400-499)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSE	**4**	**1**	**0**	**4**

Course Outcomes (CO)

After the completion of the course, the students will be able to:

CO1	Understand the architecture and functioning of Large Language Models (LLM).	**Understand**
CO2	Fine-tune pre-trained language models for various NLP tasks using Deep Learning tools	**Apply**
CO3	Design and generate prompts for generative LLMs to solve real-world challenges.	**Analyze**
CO4	Critically assess the ethical implications and societal impact of using LLMs. including elliptic curve cryptography, digital signatures, and zero-knowledge proofs.	**Evaluate**

CO - PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	2	-
CO2	3	3	2	-
CO3	3	2	1	-
CO4	3	3	3	2
CO5	3	3	3	2

Syllabus

Module 1(8 Lectures)

Large Language Models (LLM) - Introduction, Evolution of LLM, Foundation models & Instruction following LLM; Pre-training & Transfer learning; Solving Natural Language Processing (NLP) tasks using LLMs.

Module 2(10 Lectures)

Transformers - Encoder-Decoder models, Attention Mechanism; Architecture - Self-attention, Multihead attention, Layer Normalization, Positional encoding; Pre-training and fine-tuning of

Transformer based models - Autoregressive models (BERT), Generative model (GPT) and Sequence to sequence model (T5)

Module 3 (8 Lectures)

Tokenization techniques - Word & Sub-word modeling, Viterbi algorithm, Wordpeice tokenizer, Sentencepeice tokenizer, Byte Pair Encoding (BPE); Text Embeddings - Searching, classification, Clustering; Similarity Between Words and Sentences; Semantic Search

Module 4(8 Lectures)

Prompt Engineering - Introduction to Generative AI, Prompt design, Types of Prompting; Controlling model output via parameters; Use Case Ideation, Creating Custom Generative Models, Chain-ofThought Prompting, Prompt Attacks and Mitigation.

Module 5 (8 Lectures)

Ethical and Societal Implications of LLMs - Bais and Fairness, Privacy concerns, Ethical considerations, Misinformation, and Disinformation challenges, Mitigation strategies; Case study: Application of LLMs in various domains. Mini Project - Building applications from pre-trained LLMs for real-world scenarios.

References

1. Bommasani, Rishi, et al. "On the opportunities and risks of foundation models.", Center for Research on Foundation Models (CRFM), Stanford Institute for Human-Centered Artificial Intelligence (HAI), Stanford University.

2. Rogers, Anna, Olga Kovaleva, and Anna Rumshisky. "A primer in BERTology: What we know about how BERT works." Transactions of the Association for Computational Linguistics 8 (2021): 842-866.

3. Lin, Jimmy, et al. Pretrained Transformers for Text Ranking: BERT and Beyond. United States, Morgan & Claypool Publishers, 2021.

4. Pal, Ankit. "Promptify: Structured Output from LLMs." (2022) available at https://github.com/promptslab/Promptify

Semester 8						
24-813-0811	**Mini Project – 2 (Course Level 400-499)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**0**	**0**	**4**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Create innovative solutions to real world problems by applying advanced programming techniques with requirement analysis and identification of design methodologies	**Apply**
CO2	Function effectively as an individual and as a leader in diverse teams to plan and execute project utilizing available resources within timelines, following ethical and professional norms	**Apply**
CO3	Organize and communicate technical and scientific findings effectively in written and oral forms	**Analyze**

CO – PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	3	3	1
CO2	3	2	3	3
CO3	2	3	3	2

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

The objective of this course is to apply the fundamental concepts of Software Engineering principles for the effective development of an application/research project. This course helps the learners to practice the different steps to be followed in the software development process such as literature review and problem identification, preparation of Software Requirement Specification &Software Design Document (SDD), testing, development and deployment.

Guides are allotted at the beginning of the semester. A team consists of only one student. Student should identify a topic of interest in consultation with the Guide, review the literature and gather information pertaining to the chosen topic.

The guide/supervisor shall monitor the progress being carried out by the student on a regular basis. The students should record in the daily/weekly activity diary the day-to-day account of the observations, impressions, information gathered and suggestions given, if any. It should contain the review, design and all the observations made by the student and guide. The diary shall be signed every week by the guide.

The progress of the mini project is evaluated based on a minimum of two reviews by the committee. The review committee may be constituted by the Head of the Department. A project report is required at the end of the semester.

Preparing a paper for Conference/Publication in Journals is desirable for the successful completion of course. Students are also encouraged to present projects in Project Expos conducted at state level as well as others conducted in India and abroad

Semester 9						
24-813-0901	**Major Project Phase 1 (Course Level 500-599)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**0**	**0**	**14**	**14**

Course Outcomes (CO)

After the completion of the course, the students will be able to:

CO1	Identify technology/research gaps and propose creative solutions	**Analyze**
CO2	Create solutions to real world problems by performing requirement analysis and identification of design methodologies	**Apply**
CO3	Function effectively as an individual and as a leader in diverse teams to plan and execute project utilizing available resources within timelines, following ethical and professional norms	**Apply**
CO4	Organize and communicate technical and scientific findings effectively in written and oral forms	**Apply**

CO - PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	2	3	3	1
CO2	3	2	3	1
CO3	1	1	2	3
CO4	2	3	2	1

The course 'Project Work' is mainly intended to evoke the research, innovation and invention skills in a student. The course will provide an opportunity to synthesize and apply the knowledge and analytical skills learned, to be developed as a prototype or simulation. The project extends to 2 semesters and will be evaluated in the 9th and 10th semester separately, based on the achieved objectives. Importance should be given to address societal problems and developing indigenous technologies

The guide/supervisor shall monitor the progress being carried out by the student on a regular basis. The students should record in the daily/weekly activity diary the day-to-day account of

the observations, impressions, information gathered and suggestions given, if any. It should contain the review, design and all the observations made by the student and guide. The diary shall be signed every week by the guide

Phase 1 Target

- Literature study/survey of published literature on the assigned topic
- Formulation of objectives
- Formulation of hypothesis/ design/ methodology
- Formulation of work plan and task allocation.
- Block level design documentation
- Preliminary Design and Feasibility study

Students may be permitted to do the Final Year Project from top Universities in India and abroad upon getting the approval from Department Council

Semester 10						
24-813-1001	**Major Project Phase 2 (Course Level 600-699)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		DSC	**0**	**0**	**22**	**22**

Course Outcomes (CO)

After the completion of the course, the students will be able to:

CO1	Create innovative solutions to real world problems by applying advanced programming techniques with requirement analysis and identification of design methodologies	**Analyze**
CO2	Function effectively as an individual and as a leader in diverse teams to plan and execute project utilizing available resources within timelines, following ethical and professional norms	**Apply**
CO3	Organize and communicate technical and scientific findings effectively in written and oral forms.	**Apply**

CO - PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	3	1
CO2	1	1	2	3
CO3	2	3	2	1

The course 'Project Work' is mainly intended to evoke the research, innovation and invention skills in a student. The course will provide an opportunity to synthesize and apply the knowledge and analytical skills learned, to be developed as a prototype or simulation. The project extends to 2 semesters and will be evaluated in the 9th and 10th semester separately, based on the achieved objectives.

Phase 2 Targets

- Final development of product/ process, testing, results, conclusions and future directions.
- Preparing a paper for Conference Presentation/ Publication in Journals,

- Presenting projects in Project Expos conducted at state level as well as others conducted in India and abroad.
- Preparing a report in the standard format for being evaluated by the evaluation committee
- Final project presentation and viva-voce by the committee

MULTIDISCIPLINARY COURSES

Semester 1

24-813-0103	Computational Thinking for problem solving (Course Level 100-199)	TYPE	L	T	P	CREDIT
		CS MDC	3	1	0	3

Course Outcomes (CO)		Revised BT Level
After the completion of the course, the students will be able to:		
CO1	Recognizing and Defining Computational Problems	**Understand**
CO2	Designing algorithms for simple problems using computational thinking principles	**Apply**
CO3	Applying inductive and deductive reasoning, and Boolean logic to solve problems	**Apply**
CO4	Designing solutions and solution processes based on problem definitions.	**Apply**
CO5	Programming CT artifacts using Python	**Analyze**

CO – PSO Mapping

CO	PSO1	PSO2	PSO3
CO1	3	-	-
CO2	3	2	**3**
CO3	3	3	**1**
CO4	1	3	**3**
CO5	-	3	**2**

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1

Elements of Computational Thinking - Understanding computational thinking - Decomposing problems, Recognizing patterns, Generalizing patterns, Designing algorithms for simple problems

Module 2

Understanding Algorithms and Algorithmic Thinking - Defining algorithms in depth, Designing algorithms, Analyzing algorithms - Big-Oh notation.

Module 3

Understanding Logical Reasoning - Inductive reasoning, Deductive reasoning. Using Boolean logic and operators. Identifying Logical Errors and Debugging. Understanding the problem definitions - Learning how to decompose problems - Converting the flowchart into an algorithm. Designing Solutions and Solution Processes

Module 4

Identifying Challenges within Solutions - Identifying errors in algorithm design - Debugging algorithms -Comparing solutions - Refining and redefining solutions

Module 5

Introduction to Python, Using Computational Thinking in Simple Challenges, Using Python in Experimental and Data Analysis Problems - Classification and Clusters, Using Computational Thinking and Python in Statistical Analysis

References

1. Applied Computational Thinking with Python - Second Edition. By Sofía De Jesús, Dayrene Martinez
2. Karl Beecher, Computational Thinking – A beginners guide to problem solving and programming, BCS, 1e, THE CHARTERED INSTITUTE FOR IT, 2017
3. Peter J. Denning, Matti Tedre, Computational Thinking, MIT Press, 2019
4. Peter William Mcowan, Paul Curzon, Power Of Computational Thinking, World Scientific, 2017

Semester 2						
24-813-0203	**Foundations of Programming** **(Course Level 100-199)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		CS MDC	**3**	**1**	**0**	**3**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Understand the fundamentals of programming and learn to write programs.	**Understand**
CO2	Analyze the different the programming structures such as decision making statements, loops, arrays and functions.	**Analyze**

CO3	Understand the basic concepts of OOP and learn how to create and initialize objects using constructors.	**Understand**
CO4	Understand and analyze the different types of inheritance.	**Understand**
CO5	Understand the usage of polymorphism, template classes, namespaces and exception handling	**Understand**

CO – PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	-	-
CO2	3	2	-	-
CO3	3	2	**1**	-
CO4	3	2	**1**	-
CO5	3	2	**1**	-

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1(8 Lectures)

Introduction to Programming: Basic Difference between Procedure Oriented Language and Object Oriented Language. Fundamentals of C: Features of C language, structure of C program, comments, header files, data types, constants and variables, operators, expressions, evaluation of expressions, type conversion, precedence and associativity, I/O functions.

Module 2(10 Lectures)

Simple statements, Decision making statements, Looping statements, Nesting of control structures, break and continue statement. Array & String: Concept of array, One and Two

dimensional arrays, declaration and initialization of arrays, String, String storage, Built-in string functions.

Module 3 (8 Lectures)

Functions: Concept of user defined functions, prototype, definition of function, parameters, parameter passing, calling a function

Module 4(10 Lectures)

Introduction to OOP: Basic properties, Classes and Objects, Member functions and variables, Abstract data types (ADT), Encapsulation. Constructors: Parameterized Constructors, Copy Constructors, Dynamic Constructors, Destructors.

Module 5 (8 Lectures)

Inheritance and Access Control: Member access control in classes, Friend functions and classes, Public Private and Protected Inheritance, Polymorphism: Runtime and compile time polymorphism, overloading functions and operators, Defining Operator Overloading

References

1. Yashavant Kanetkar: Let Us C, 15e,BPB Publications, 2016.
2. Herbert Schildt: C: The Complete Reference, 4e, McGraw Hill Education, 2017.
3. Pradip Dey, Manas Ghosh: Computer Fundamentals and Programming in C, 2e, Oxford University Press, 2013.
2. Bjarne Stroustrup: C++ Programming Language, 4e, Addison-Wesley, 2013.
3. Bjarne Stroustrup: Programming: Principles and Practice Using C++, 2e, Addison- Wesley, 2014.
4. Stanley Lippman, Josée Lajoie, Barbara Moo: C++ Primer, 5e, Addison-Wesley, 2012.
5. Paul Deitel, Harvey Deitel: C++ How to Program, 10e, Pearson, 2016.
6. Timothy Budd: Introduction To Object-Oriented Programming, Pearson Education, 2008.
7. Walter J. Savitch, Kenrick Mock: Problem Solving with C++, 9e, Pearson Education, 2017.
8. Ira Pohl: Object-Oriented Programming Using C++, 2e, Addison-Wesley,1996.

Semester 3						
24-813-0303	**Fundamentals of Data Structures** **(Course Level 200-299)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		CS MDC	**3**	**1**	**0**	**3**
Pre-requisites: 24-813-0103, 24-813-0203						

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Understand the basic concepts of programming	**Understand**
CO2	Use elementary and advanced data structures such as Array, Linked list, Tree and to solve real world problems efficiently.	**Apply**
CO3	Implement searching and sorting methods.	**Apply**
CO4	Implement object oriented concepts in programming	**Apply**

CO – PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	**1**	**-**
CO2	3	2	**1**	**-**
CO3	3	2	**1**	**-**
CO4	3	2	**1**	**-**

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1

Introduction to programming methodologies – structured approach, stepwise refinement techniques, programming style, documentation. Elementary data organization - Data structure - Data structure operation, Review of basic programming questions

Module 2

Array, Records and Pointers: Introduction, Linear array, Representation of linear array in memory, Traversing linear array, Inserting and Deleting, Sorting methods, Searching methods.

Module 3

String - representation of strings, concatenation, substring searching and deletion.

Linked List: Introduction, Linked list, Representation of Linked list in memory, Searching a linked list,

Module 4

Stacks, Queues, Recursion - Introduction, Stacks, Queues, Operations on stacks and Queues, Implementation of Stacks and Queues using arrays and linked list, Applications.

Module 5

Tree - Introduction, Terminology of Binary tree, Types of Binary tree, Traversing of binary tree,

References

1. Samanta D.: Classic Data Structures, 2e, Prentice Hall India, 2009.
2. Richard F. Gilberg, Behrouz A. Forouzan: Data Structures: A Pseudocode Approach with C, 2e, Cengage Learning, 2005.
3. Aho A. V., J. E. Hopcroft, J. D. Ullman: Data Structures and Algorithms, Pearson Publication, 1983.
4. Tremblay J. P., P. G. Sorenson: Introduction to Data Structures with Applications, 2e, Tata McGraw Hill, 1995.
5. Peter Brass: Advanced Data Structures, Cambridge University Press, 2008.
6. Lipschutz S.: Theory and Problems of Data Structures, Schaum's Series, 1986.
7. Wirth N.: Algorithms + Data Structures = Programs, Prentice Hall, 2004.
8. Horwitz E., S. Sahni, S. Anderson: Fundamentals of Data Structures in C, University Press (India), 2008.

MINOR COURSES

Semester 1						
24-813-0101	**Computational Thinking for Problem Solving** **(Course Level 100-199)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		MIN	**4**	**1**	**2**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Recognizing and Defining Computational Problems	**Understand**
CO2	Designing algorithms for simple problems using computational thinking principles	**Apply**
CO3	Applying inductive and deductive reasoning, and Boolean logic to solve problems	**Apply**
CO4	Designing solutions and solution processes based on problem definitions.	**Apply**

CO5	Testing and Refining Computational Artifacts	**Analyze**

CO – PSO Mapping

CO	PSO1	PSO2	**PSO3**	**PSO4**
CO1	3	-	**-**	**-**
CO2	3	2	**3**	**-**
CO3	3	3	**1**	**-**
CO4	1	3	**3**	**-**
CO5	-	3	**2**	**-**

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module -1 (8 Lectures)

Elements of Computational Thinking - Understanding computational thinking - Decomposing problems, Recognizing patterns, Generalizing patterns, Designing algorithms for simple problems

Module 2(8 Lectures)

Understanding Algorithms and Algorithmic Thinking - Defining algorithms in depth, Designing algorithms, Analyzing algorithms.

Module 3 (8 Lectures)

Understanding Logical Reasoning - Inductive reasoning, Deductive reasoning. Using Boolean logic and operators. Identifying Logical Errors and Debugging

Module 4(8 Lectures)

Understanding the problem definitions - Learning how to decompose problems - Converting the flowchart into an algorithm. Designing Solutions and Solution Processes,

Module 5 (8 Lectures)

Identifying Challenges within Solutions - Identifying errors in algorithm design - Debugging algorithms -Comparing solutions - Refining and redefining solutions

References

1. Applied Computational Thinking with Python - Second Edition. By Sofía De Jesús, Dayrene Martinez

2. Karl Beecher, Computational Thinking – A beginners guide to problem-solving and programming, BCS, 1e, THE CHARTERED INSTITUTE FOR IT, 2017

3. Peter J. Denning, Matti Tedre, Computational Thinking, MIT Press, 2019

4. Peter William Mcowan, Paul Curzon, Power Of Computational Thinking, World Scientific, 2017

Semester 1						
24-813-0102	**Practical Applications of AI (Course Level 100-199)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		MIN	**4**	**1**	**0**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Understanding of AI Concepts	**Understand**
CO2	Demonstrate knowledge of various AI algorithms, techniques, and models	**Apply**
CO3	Apply AI techniques to solve real-world problems and demonstrate critical thinking skills	**Apply**
CO4	Understand knowledge-based systems.	**Understand**
CO5	Know ethical concerns	**Understand**

CO – PSO Mapping

CO	PSO1	PSO2	**PSO3**	**PSO4**

CO1	3	-	**-**	**-**
CO2	3	2	**2**	**-**
CO3	3	2	**2**	**-**
CO4	3	-	**2**	**-**
CO5	-	3	**1**	**3**

Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1(8 Lectures)

AI in Practice: Robotic Systems, Computer Vision, Natural Language ProcessingEducation, Government, Healthcare, Technology, Commerce, Manufacturing, Agriculture

Module-2(10 Lectures)

Artificial Intelligence, Machine Learning, Neural Networks, Perceptron, Deep Learning, Explainable AI, Generative AI, Prompt Engineering, GPT

Module-3 (8 Lectures)

Familiarisation of AI Software Python, R, Google Colab, Anaconda, UIPath, Power BI.

Practical Generative AI Examples, Creating presentations, Opening Excel files and draw graphs automatically, Make new pictures and music.

Module-4(8 Lectures)

Ethical concerns raised by AI, The role of ethics in the development of AI, Different ways of operationalizing fairness in the context of AI, Transparency and AI systems, AI and the Sustainable Development Goals, Applying AI to address the SDGs, The positive and negative impact of AI on the SDGs

Module-5 (6 Lectures)

Case Study 1: Contributions of AI towards developing vaccines

Case Study 2: AI for disaster management

References

1.Artificial Intelligence and Machine Learning by Vinod Chandra S. S and Anand Hareendran S, PHI, 2014.

2.Machine Learning: The New AI by Ethem Alpaydin, The MIT Press, 2016

3.https://microsoft.github.io/AI-For-Beginners/Introduction to AI, Evolution of AI, Turing test, Categories of AI, Applications of AI, Problem Definition as a State Space Search, Production System, Control StrategiesEthem Alpaydin, Machine Learning: The New AI, MIT Press, 2016

Semester 2						
24-813-0202	**Computer Fundamentals 1** **(Course Level 100-199)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		MIN	**4**	**1**	**0**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Understanding of the basic components of a computer system, including the CPU, memory, and storage	**Understand**
CO2	Gain proficiency in using common operating systems such as Windows or Linux	**Apply**
CO3	Acquire basic skills in computer programming and algorithmic thinking.	**Apply**
CO4	Understand fundamental concepts of computer networking, including protocols, topologies, and network devices.	**Understand**
CO5	Know ethical issues related to computer technology, including privacy, intellectual property, and social implications of automation	**Understand**

CO – PSO Mapping

CO	PSO1	PSO2	**PSO3**	**PSO4**
CO1	2	-	**-**	**-**
CO2	1	-	**-**	**-**
CO3	3	2	**1**	**-**
CO4	3	2	**3**	**-**
CO5	-	2	**-**	**3**

Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module -1(8 Lectures)

Introduction, Basic Applications of Computer, Components of Computer, Connecting Computer Components, Computer Hardware & Software. What is an Operating System, Basics of Popular Operating Systems.

Module-2(10 Lectures)

Word Processing: Introduction, Document Creation & Editing, Saving, Text Formatting. Microsoft Excel & using Spreadsheets: Introduction, Rows, Columns & Cells, Basics Excel Formulas and Functions

Module-3 (8 Lectures)

Introduction to Internet, WWW and Web Browsers: Basic of Computer networks; LAN, WAN; Concept of Internet; Applications of Internet; Connecting to Internet; What is ISP; Knowing the Internet; Basics of internet connectivity related troubleshooting, Search Engines; Understanding URL; Domain name and IP Address

Module-4(8 Lectures)

Communications and collaboration: Basics of electronic mail; Getting an email account; Sending and receiving emails; Accessing sent emails; Using Emails; Document collaboration; Instant Messaging; Netiquettes.

Module-5 (6 Lectures)

Computer Security and Privacy: Importance of Computer Security, Common Security Threats, Malware (Viruses, Worms, Trojans), Network Security Measures Firewalls, Encryption, Access Control, User Authentication, Privacy Concerns and Data Protection

References

1. Computer Basics Absolute Beginner's Guide- Michael Miller

2. Absolute Beginners Guide to Computing - Wallace Wang

3. Computer Fundamentals: Concepts, Systems & Applications- 8th Edition- Priti Sinha, Pradeep K, Sinha

4. Computers Made Easy from Dummy to Geek- James Bernstein

Semester 2						
24-813-0201	**Fundamentals of programming (Course Level 100-199)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		MIN	**4**	**1**	**2**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Understand the programming fundamentals and write programs.	**Understand**
CO2	Analyse the different programming structures such as decision-making statements, loops, arrays, and functions.	**Analyze**
CO3	Understanding the basic concepts of OOP and learning how to create and initialize objects using constructors.	**Understand**
CO4	Understand and analyze the different types of inheritance.	**Understand**
CO5	Understand the usage of polymorphism, template classes, namespaces, and exception handling	**Understand**

CO – PSO Mapping

CO	PSO1	PSO2	**PSO3**	**PSO4**
CO1	3	2	-	-
CO2	3	2	-	-

CO3	3	2	**1**	**-**
CO4	3	2	**1**	**-**
CO5	3	2	**1**	**-**

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1(8 Lectures)

Introduction to Programming: Basic Difference between Procedure Oriented Language and Object Oriented Language, Concepts of Machine level, Assembly level and High-level programming, Flow charts and Algorithms.

Module 2(8 Lectures)

Fundamentals of C: Features of C language, structure of C program, comments, header files, data types, constants and variables, operators, expressions, evaluation of expressions, type conversion, precedence and associativity, I/O functions.

Module 3 (8 Lectures)

Simple statements, Decision-making statements, Looping statements, Nesting of control structures, break and continue statements. Array & String: Concept of array, One and Two-dimensional arrays, declaration and initialization of arrays, String, String storage, Built-in string functions.

Module 4(8 Lectures)

Functions: Concept of user-defined functions, prototype, definition of function, parameters, parameter passing, calling a function. Introduction to OOP: Basic properties, Classes and Objects, Member functions and variables, Abstract data types (ADT), Encapsulation.

Module 5 (8 Lectures)

Inheritance and Access Control: Member access control in classes, Friend functions and classes, Public Private, and Protected Inheritance, Single – Multiple – Multilevel – Hierarchical – Hybrid inheritance.Polymorphism: Runtime and compile time polymorphism.

References/Text Books

1. Samanta D.: Classic Data Structures, 2e, Prentice Hall India, 2009.
2. Richard F. Gilberg, Behrouz A. Forouzan: Data Structures: A Pseudocode Approach with C, 2e, Cengage Learning, 2005.
3. Aho A. V., J. E. Hopcroft, J. D. Ullman: Data Structures and Algorithms, Pearson Publication, 1983.
4. Tremblay J. P., P. G. Sorenson: Introduction to Data Structures with Applications, 2e, Tata McGraw Hill, 1995.
5. Horwitz E., S. Sahni, S. Anderson: Fundamentals of Data Structures in C, University Press (India), 2008.

Semester 3						
24-813-0302	**Computer Fundamentals 2** **(Course Level 200-299)**	**TYPE**	**L**	**T**	**P**	**CREDIT**
		MIN	**4**	**1**	**0**	**4**

Course Outcomes (CO)		**Revised BT Level**
After the completion of the course, the students will be able to:		
CO1	Understanding basic concepts of Number systems and digital logic	**Understand**
CO2	Understand the basic concepts of Computer organization and architecture	**Understand**
CO3	Understand the basic working principles of operating system and its process	**Understand**

CO – PSO Mapping

CO	PSO1	PSO2	**PSO3**	**PSO4**
CO1	2	-	**-**	**-**
CO2	2	-	**-**	**-**
CO3	2	-	**-**	**-**

: Correlations Levels: 1 = Low, 2 = Medium, 3 = High, "-" = No correlation

Syllabus

Module 1(8 Lectures)

Number Systems and Codes: Binary Number system – Binary to decimal – decimal to binary – hexadecimal – ASCII code Digital Logic: The Basic Gates – NOT, OR, AND - Universal Logic Gates – NOR, NAND. Boolean Laws and Theorems.

Module-2(10 Lectures)

Basic Computer Organization and Design: Instruction codes - stored program organization - Computer registers and common bus system - Computer instructions - Timing and control - Instruction cycle: Fetch and Decode - Register reference instructions.

Module-3 (8 Lectures)

Central Processing Unit: General register organization - stack organization – instruction formats - addressing modes - Input-output organization: Peripheral devices - I/O interface - Memory organization: Memory hierarchy - Main memory - Auxiliary memory

Module-4(8 Lectures)

Overview of Operating Systems, Types of OS, OS Operations, Resource Management, Kernel Processes- Process concept, forks and pipes, Interrupt processing, Process Scheduling, CPU Scheduling Algorithms

Module-5 (6 Lectures)

Process Synchronization- Critical Section Problem, Mutex Locks, Semaphores, Deadlocks- Methods of Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance

References/Text Books

1. Digital Principles and Applications – Donald P Leach, Albert Paul Malvino, GoutamSaha, 8th edition, McGraw-Hill Education, 3rd reprint 2015. R. P. Jain, “Modern Digital Electronic”, McGraw-Hill Publication, 4th Edition.
2. William Stalling, “Computer Organization and Architecture: Designing and Performance”, Pearson Publication 10TH Edition.
3. Computer System Architecture, M. Morris Mano, Pearson Education, 3rd edition.,2007
4. Operating System Principles, Abraham Silberchatz, Peter B. Galvin, Greg Gagne,10th Edition, Wiley Student Edition. 2018
5. Operating System-Internals and Design Principles, W.Stallings, 6th Edition, Pearson.
6. Strang, Gilbert. Modern Operating System, Andrew s Tanenbaum, 3rd Edition, PHI