Course Curriculum for B Tech in Computer Science and Engineering

Computer Science and Engineering Department

NATIONAL INSTITUTE OF TECHNOLOGY DELHI

(An autonomous Institute under the aegis of Ministry of Education, Govt. of India.)

Department of Computer Science and Engineering
National Institute of Technology Delhi

About the Department

The Computer Science and Engineering Department was started in 2010 along with the foundation of NIT Delhi. Initially, only the Bachelor of Technology Programme was offered with the intake 30 which presently has been increased to 60. Now, apart from B. Tech., the department also offers Master of Technology (CSE & Analytics), and Ph.D. program which cover a number of important areas of Computer Science and Engineering. The department provides the students with a broad undergraduate and graduate curriculum, based on the application and theoretical foundations of computer science. The departmental faculties and students participate in interdisciplinary research. The department envisions producing quality graduates, capable of leading the world in the technical realm. The department is equipped with the latest configuration and high computing system with hi-speed Internet facilities. The Computer Science Program at this institute are dedicated to educate students and to advance research in computer and information technology. The department has all the facilities to carry out the related teaching and research work.

VISION

To communicate quality Computer Science Education for producing globally identifiable skilled technocrats and entrepreneurs upholding sound ethics, profound knowledge, and innovative ideas to meet industrial and societal expectations.

MISSION

- **M1** To impart value-based **technical knowledge** and skill relevant to Computer Science and Engineering through effective pedagogies and hands-on experience on the latest tools and technologies to **maximize employability**.
- **M2** To strengthen **multifaceted competence**, nurture **creativity, and innovation,** and create **entrepreneurial environment** for an ever-changing technological scenario requiring communally cognizant solutions.
- **M3** To create an appetite for **research, and higher education** in contemporary, and emerging areas of Computer Science.
- **M4** To inculcate the **moral, ethical, and social ideals** essential for prosperousnation-building.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

- **PEO1** Graduates are prepared to be employed in IT industries and be engaged in continuous learning, understanding, and applying innovative ideas while maintaining strong ethical standards.
- **PEO2** Graduates are prepared to pursue higher studies and continue to develop their professional knowledge.
- **PEO3** Graduates are equipped to do research in areas of specialization and the allied fields.

PEO4 Graduates are prepared to meet the changing needs of society through knowledge-based service, exhibit leadership qualities with demonstrable attributes in lifelong learning and become successful entrepreneurs.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1 Ability to analyze, develop and design new tools and approaches to createcutting-edge solutions for Industry.

PSO2 Ability to carry out research and education in trans-disciplinary fields to solvethe problems of national as well as international significance.

B. Tech. (Computer Science and Engineering)

Programme Framework

- Minimum Credits requirements for completion of BTech program is 160
- The Curriculum is based on the guidelines of National Education Policy(NEP)–2020
- The curriculum has embedded the Multi Exit/ Multi Entry in the BTech program
- There is provision of Major degree and Minor Degree for students
- The curriculum is designed to meet the prevailing and ongoing industrial requirements
- The curriculum includes Project based Education with Projects every year
- The curriculum is flexible and offers Choice Based Credit System (CBCS)
- The curriculum inherits the Value based Edu cation and offers Interdisciplinary/Multidisciplinary Courses
- The Curriculum offers Digital Pedagogy & Flipped Learning with adequate motivation for Entrepreneurship/ Startups
- The curriculum aims at the Holistic Development of the students
- In the proposed UG scheme, the department of CSE is proposing followingspecializations:
    1. *Artificial Intelligence and Machine Learning (Bucket 1)*
    2. *Data Science (Bucket 2)*
    3. *Information Security (Bucket 3)*
    4. *Computer Systems (Bucket 4)*
    5. *Networks and Distributed Systems (Bucket 5)*

- Total **7 electives** are proposed in the complete UG program, among them **at least 5 electives** are required from a bucket to get the specialization (with the respective bucket) with B.Tech in Computer Science and Engineering

- Students can attend 2 MOOC/NPTEL/any online courses (as per department list) among the proposed **7 electives** and the evaluation will be done by the Department as per Academic Calendar and prevailing norms
- Students can do any number of courses from the other IITs/NITs/or any other CFTI institutes. There will be the provision of credit transfer as per NIT Delhi norms
- If any student from the other branch will do **4 courses (16 credits) and one project (2 credits)**, then the student will be awarded the **minor degree in Computer Science and Engineering with the respective bucket specialization.** The project should be approved by the department of CSE Hence, in a minor degree a student should complete **18 credits other than the desired credits** of his/her major degree

Cardinal Mentions

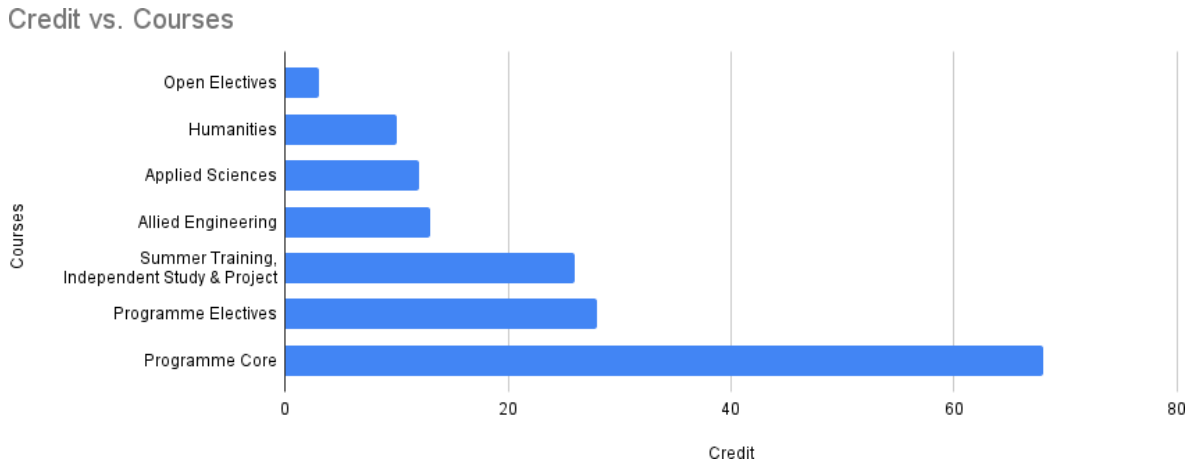
✓ The students can exit after completing 1st Year, 2nd Year and 3rd Year from the program and will be awarded Certificate, Diploma and Advanced Diploma in Computer Science and Engineering respectively. A minimum Credit requirement for Certificate is 40 Credits, Diploma is 80 Credits and Advanced Diploma is 120 Credits respectively.

✓ The other branches students can opt for Minor Degree in Computer Science and Engineering across any specialization offered by the department from 5th Semester onwards by obtaining 18 credits from Computer Science and Engineering (16 credits in course work and 02 credits in projects) from the respective specializations.

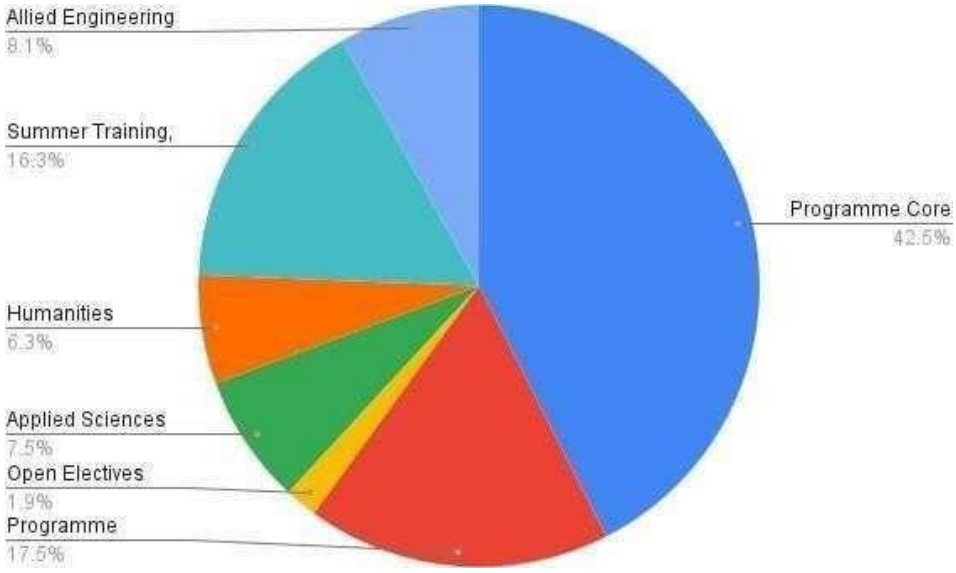
Semester wise Credit Structure

Sl. No.	Courses	Credits								Total
		1st Year		2nd Year		3rd Year		4th Year		
		1st Sem	2nd Sem	3rd Sem	4th Sem	5th Sem	6th Sem	7th Sem	8th Sem	
1	Programme Core (PC)	8	13	12	12	16	7	0	0	68
2	Programme Electives (PE)	0	0	0	0	4	8	16	0	28
3	Open Electives (OE)	0	0	0	0	0	3	0	0	3
4	Applied Sciences (AS)	4	4	4	0	0	0	0	0	12
5	Humanities (HM)	4	0	0	3	0	0	3	0	10
6	Summer Training, Independent Study & Project (ST-IS-PR)	0	2	0	1	0	2	1	20	26
7	Allied Engineering (AE)	4	1	4	4	0	0	0	0	13
Total		**20**	**20**	**20**	**20**	**20**	**20**	**20**	**20**	**160**

Credits Distribution



Credits Distribution (%)



Course Scheme

Year	First Semester						Second Semester					
	Sub Code	**Subject Name**	**L**	**T**	**P**	**C**	**Sub Code**	**Subject Name**	**L**	**T**	**P**	**C**
Ist	CSBB 103	Problem Solving & Computer Programming	2	0	2	3	CSBB 151	Data Structures	3	0	2	4
	CSBB 102	Introduction to Computer Systems	3	0	2	4	CSLB 152	Discrete Structures	3	1	0	4
	MALB 101	Advanced Calculus	3	1	0	4	CSLB 154	System Programming	3	0	0	3
	HMBB 101	Theory and Practices of Human Ethics	2	0	2	3	MALB 152	Applied Linear Algebra	3	1	0	4
	MEBB 162	Engineering Visualization	3	0	2	4	CSPB 154	Introduction to Hardware	0	0	2	1
	HMPB 102	Communication Skills	0	0	2	1	CELB 101	Environmental Sciences	2	0	0	2
	HSPB 150	Holistic Health and Sports	-	-	-	-	CSPB 100	Project I	0	0	4	2
	Total					**20**	**Total**					**20**
IInd	**Third Semester**						**Fourth Semester**					
	CSBB 202	Design and Analysis of Algorithms	3	0	2	4	CSBB 251	Computer Architecture and Organization	3	0	2	4
	CSBB 203	Operating System	3	0	2	4	CSBB 252	Artificial Intelligence	3	0	2	4
	CSBB 204	Database Management Systems	3	0	2	4	CSBB 254	Software Engineering	3	0	2	4
	MALB 202	Probability and Statistics	3	1	0	4	HMBB 251	Professional Communication	2	0	2	3
	ECBB 206	Digital Electronics and Logic Design	3	0	2	4	ECBB 254	Communication Systems	3	0	2	4
							CSPB 200	Project II	0	0	2	1
	Total					**20**	**Total**					**20**

	Fifth Semester						Sixth Semester					
	Sub Code	**Subject Name**	**L**	**T**	**P**	**C**	**Sub Code**	**Subject Name**	**L**	**T**	**P**	**C**
IIIrd	CSBB 301	Computer Networks	3	0	2	4	CSBB 351	Compiler Design	3	0	2	4
	CSLB 302	Theory of Computation	3	1	0	4	CSBB 352	Theory of App Development	2	0	2	3
	CSBB 303	Data Mining	3	0	2	4	XXXB XXX	Open Elective	3	0	0	3
	CSBB 304	Quantum Computing	3	0	2	4	CSXB XXX	Elective 2[Select in Set 2 from the respective specialization]	3	0/1	2/0	4
	CSXB XXX	Elective 1 [Select in Set 1 from the respective specialization]	3	0/1	2/0	4	CSXB XXX	Elective 3[Select in Set 2 from the respective specialization]	3	0/1	2/0	4
							CSPB 300	Project III	0	0	4	2
							CSPB 301	Internship (during summer break)	Credit will be given in next Semester			
	Total					**20**	**Total**					**20**
	Seventh Semester						**Eighth Semester**					
IVth	CSXB XXX	Elective 4 [Select in Set 3 from the respective specialization]	3	0/1	2/0	4	CSPB 400	B. Tech Project (Internship inside NIT Delhi / Outside NIT Delhi)	-	-	-	16
	CSXB XXX	Elective 5 [Select in Set 3 from the respective specialization]	3	0/1	2/0	4	CSPB 401	Mooc Course	3	0	0	3
	CSXB XXX	Elective 6 [Select in Set 3 from the respective specialization]	3	0/1	2/0	4	CSPB 402	Seminar	0	0	2	1
	CSXB XXX	Elective 7 [Select in Set 3 from the respective specialization]	3	0/1	2/0	4						
	HMLB 401	Management Principles and Practices	3	0	0	3						
	CSPB 301	Internship (completed in last summer break)	-	-	-	1						
	Total					**20**	**Total**					**20**

Bucket 1 of Elective Courses [Specialization in Artificial Intelligence and Machine Learning]												
Set-1												
	Sub Code	**Subject Name**	**L**	**T**	**P**	**C**	**Sub Code**	**Subject Name**	**L**	**T**	**P**	**C**
	CSBB 311	Machine Learning	3	0	2	4	CSBB 312	Pattern Recognition	3	0	2	4
	CSBB 313	Digital Image Processing	3	0	2	4	CSBB 314	Computer Vision	3	0	2	4
	CSLB 315	Optimization Techniques	3	1	0	4	CSBB 405	Fuzzy Logic and Applications	3	0	2	4
	CSBB 406	Cloud Computing	3	0	2	4						
Set-2												
	Sub Code	**Subject Name**	**L**	**T**	**P**	**C**	**Sub Code**	**Subject Name**	**L**	**T**	**P**	**C**
	CSBB 314	Computer Vision	3	0	2	4	CSLB 315	Optimization Techniques	3	1	0	4
	CSBB 316	Information Storage & Retrieval	3	0	2	4	CSBB 317	Soft Computing	3	0	2	4
	CSBB 405	Fuzzy Logic and Applications	3	0	2	4	CSBB 407	Natural Language Processing	3	0	2	4
	CSBB 408	Reinforcement Learning and Applications	3	0	2	4	CSBB 409	Social Network Analysis	3	0	2	4
	CSBB 412	Motion Analytics	3	0	2	4	CSBB 415	Motion Planning for Robotics	3	0	2	4
	CSBB 424	Deep Learning and Applications	3	0	2	4						
Set-3												
	Sub Code	**Subject Name**	**L**	**T**	**P**	**C**	**Sub Code**	**Subject Name**	**L**	**T**	**P**	**C**
	CSBB 405	Fuzzy Logic and Applications	3	0	2	4	CSBB 406	Cloud Computing	3	0	2	4
	CSBB 407	Natural Language Processing	3	0	2	4	CSBB 408	Reinforcement Learning and Applications	3	0	2	4
	CSBB 409	Social Network Analysis	3	0	2	4	CSBB 412	Motion Analytics	3	0	2	4
	CSBB 413	Introduction to Cognitive Computing	3	0	2	4	CSLB 414	Game Theory	3	1	0	4
	CSBB 415	Motion Planning for Robotics	3	0	2	4	CSBB 424	Deep Learning and Applications	3	0	2	4

Bucket-2 [Specialization in Data Science]												
Set-1												
	Sub Code	**Subject Name**	**L**	**T**	**P**	**C**	**Sub Code**	**Subject Name**	**L**	**T**	**P**	**C**
	CSBB 311	Machine Learning	3	0	2	4	CSLB 315	Optimization Techniques	3	1	0	4
	CSLB 321	Mathematical Foundation of Data Science	3	1	0	4						
Set-2												
	Sub Code	**Subject Name**	**L**	**T**	**P**	**C**	**Sub Code**	**Subject Name**	**L**	**T**	**P**	**C**
	CSBB 314	Computer Vision	3	0	2	4	CSLB 315	Optimization Techniques	3	1	0	4
	CSBB 323	Data Handling & Visualization	3	0	2	4	CSBB 325	Time Series Analysis	3	0	2	4
	CSBB 326	Distributed System	3	0	2	4	CSBB 406	Cloud Computing	3	0	2	4
	CSBB 409	Social Network Analysis	3	0	2	4	CSBB 421	Internet of Things	3	0	2	4
	CSBB 422	Big Data Analytics	3	0	2	4	CSBB 424	Deep Learning and Applications	3	0	2	4
Set-3												
	Sub Code	**Subject Name**	**L**	**T**	**P**	**C**	**Sub Code**	**Subject Name**	**L**	**T**	**P**	**C**
	CSBB 314	Computer Vision	3	0	2	4	CSBB 406	Cloud Computing	3	0	2	4
	CSBB 409	Social Network Analysis	3	0	2	4	CSLB 414	Game Theory	3	1	0	4
	CSBB 421	Internet of Things	3	0	2	4	CSBB 422	Big Data Analytics	3	0	2	4
	CSBB 424	Deep Learning and Applications	3	0	2	4	CSBB 425	Information Security and Privacy	3	0	2	4
	CSBB 426	Business Intelligence and Analytics	3	0	2	4	CSBB 427	Advanced Databases	3	0	2	4

Bucket 3 [Specialization in Information Security]												
Set-1												
	Sub Code	**Subject Name**	**L**	**T**	**P**	**C**	**Sub Code**	**Subject Name**	**L**	**T**	**P**	**C**
	CSBB 331	Network and Data Security	3	0	2	4	CSBB 333	Cryptography and Computer Security	3	0	2	4
	CSBB 335	Information Security	3	0	2	4	CSBB 436	Block chain Technology	3	0	2	4
Set-2												
	Sub Code	**Subject Name**	**L**	**T**	**P**	**C**	**Sub Code**	**Subject Name**	**L**	**T**	**P**	**C**
	CSBB 333	Cryptography and Computer Security	3	0	2	4	CSBB 335	Information Security	3	0	2	4
	CSBB 439	Database and Online Social Media Security	3	0	2	4						
Set-3												
	Sub Code	**Subject Name**	**L**	**T**	**P**	**C**	**Sub Code**	**Subject Name**	**L**	**T**	**P**	**C**
	CSBB 436	Block chain Technology	3	0	2	4	CSBB 439	Database and Online Social Media Security	3	0	2	4
S	CSBB 440	Introduction to Cyber Security	3	0	2	4	CSBB 444	Software Security	3	0	2	4

Bucket 4 of Elective Courses [Specialization in Computer Systems]												
Set-1												
	Sub Code	**Subject Name**	**L**	**T**	**P**	**C**	**Sub Code**	**Subject Name**	**L**	**T**	**P**	**C**
	CSBB 341	Advanced Computer Networks	3	0	2	4	CSBB 343	Computer Graphics	3	0	2	4
	CSBB 344	Object Oriented Programming	3	0	2	4	CSBB 427	Advanced Databases	3	0	2	4
Set-2												
	Sub Code	**Subject Name**	**L**	**T**	**P**	**C**	**Sub Code**	**Subject Name**	**L**	**T**	**P**	**C**
	CSLB 315	Optimization Techniques	3	1	0	4	CSBB 326	Distributed Systems	3	0	2	4
	CSBB 346	Parallel Algorithms	3	0	2	4	CSLB 348	Randomized Algorithms	3	1	0	4
	CSBB 406	Cloud Computing	3	0	2	4						
Set-3												
	Sub Code	**Subject Name**	**L**	**T**	**P**	**C**	**Sub Code**	**Subject Name**	**L**	**T**	**P**	**C**
	CSLB 315	Optimization Techniques	3	1	0	4	CSBB 406	Cloud Computing	3	0	2	4
	CSBB 452	Advanced Operating Systems	3	0	2	4	CSLB 454	Computational Complexity	3	1	0	4
	CSBB 456	Real Time Systems	3	0	2	4						

Bucket 5 of Elective Courses [Specialization in Networks and Distributed Systems]												
Set-1												
	Sub Code	**Subject Name**	**L**	**T**	**P**	**C**	**Sub Code**	**Subject Name**	**L**	**T**	**P**	**C**
	CSBB 406	Cloud Computing	3	0	2	4	CSBB 410	Wireless Sensor Networks	3	0	2	4
Set-2												
	Sub Code	**Subject Name**	**L**	**T**	**P**	**C**	**Sub Code**	**Subject Name**	**L**	**T**	**P**	**C**
	CSLB 315	Optimization Techniques	3	1	0	4	CSLB 353	Queuing Theory	3	1	0	4
	CSBB 354	Mobile Computing	3	0	2	4	CSBB 326	Distributed Systems	3	0	2	4
	CSBB 356	Network and Wireless Security	3	0	2	4	CSBB 357	High Performance Computing	3	0	2	4
	CSBB 358	Information Theory and Coding	3	0	2	4						
Set-3												
	Sub Code	**Subject Name**	**L**	**T**	**P**	**C**	**Sub Code**	**Subject Name**	**L**	**T**	**P**	**C**
	CSBB 341	Advanced Computer Networks	3	0	2	4	CSBB 352	Optical Networks	3	0	2	4
	CSBB 411	Distributed Databases	3	0	2	4	CSLB 414	Game Theory	3	1	0	4
	CSBB 472	Next Generation Networks	3	0	2	4						

List of new courses introduced in the scheme

Sub Code	**Subject Name**	**L**	**T**	**P**	**C**
CSBB 415	Motion Planning for Robotics	3	0	2	4
CSBB 411	Distributed Databases	3	0	2	4

Course Code: CSBB 103	PC (YES/ NO)	PE (YES/ NO)	OE (YES/ NO)	AS (YES/ NO)	HM (YES/ NO)	ST-IS-PR (YES/NO)	AE (YES/ NO)
	YES	NO	NO	NO	NO	NO	NO
Type of course	Program Core						
Course Title	**PROBLEM SOLVING AND COMPUTER PROGRAMMING**						
Course Objectives:	1. To understand the computational model of Computer. 2. To understand the concepts of C programming. 3. To apply functions of C programming for solving problems. 4. To understand the concept of file management in C.						
Course Outcomes	**CO1:** Understand the basics of computer and various problemsolving approaches.						**L1, L2**
	CO2: Understand the fundamentals of C programming.						**L1, L2**
	CO3: Apply functions, arrays, and structures for solving problem.						**L2, L3, L4**
	CO4: Understand the use of pointers and file management in C.						**L2, L3**

Semester	**Autumn: Yes**		**Spring:**		
I	**Lecture**	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	2	0	2	3	
Prerequisite course code as perproposed course numbers					
Prerequisite credits					
Equivalent course codes as per proposed course and old course					
Overlap course codes as per proposed course numbers					

Text Books:		
1	Title	Programming in ANSI C
	Author	E. Balagurusamy
	Publisher	TATA McGraw Hil1
	Edition	6 edition, 2012
Reference Book:		
1	Title	Let Us C
	Author	Yashwant Kanetkar
	Publisher	Infinity Science Press
	Edition	13th edition, 2012
2	Title	Schaum's Outline of Programming with C
	Author	Byron S Gottfried
	Publisher	TATA McGraw Hill
	Edition	2 d edition, 1996

3	Title	The C Programming Language
	Author	Brian Kernighan & Dennis Ritchie
	Publisher	Prentice Hal
	Edition	2nd edition, 1988

Content	**Unit 1** Introduction to Computers: Hardware and Software. Basic Model of Computation Notion of Algorithms, Flowcharts, Top down design, Bottom-up approaches of problem solving, Number system. **Unit 2** Introduction to programming language, Basics of C, Basic Data types - int, float double, char, Bool, Void. Arithmetic and logical operators: precedence and association. Flow of Control Conditional statements- If-else, Switch-case constructs, Loops- While, do-while, for. **Unit 3** Function - User defined functions, library functions, Parameter passing call by value, call by reference, recursion. **Unit 4** Arrays- Advantages and drawbacks, One dimensional, Multi-Dimensional Arrays and strings: Declaration, initialization, Accessing, Passing arrays and strings as parameters to functions. Pointers, Dynamic memory allocation, Dynamic arrays- One dimensional, Multidimensional dynamic array. **Unit 5** Structure: Declaration, Initialization, passing structure to function, Use of pointers in structures. Preprocessors, Macros, File management in C 1/0 - Opening closing and editing files. Correctness & Efficiency Issues in Programming, Time & Space measures.
Course Assessment	Continuous Evaluation 25%
	Mid Semester 25%
	End Semester 50%

Course Matrix (CO-PO-PSO Mapping)

COs	POs & PSOs													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	**3**												**1**	
CO2	**2**	**1**	**1**										**2**	**2**
CO3	**3**	**2**	**2**	**2**	**2**	**1**							**2**	**2**
CO4	**3**	**2**	**2**	**2**	**2**	**2**							**2**	

1=addressed to small extent

2= addressed significantly

3= addressed strongly (major part of course)

Lab Experiments:

Exp. No.	List of Experiments
1	Familiarization of Linux environment - How to do Programming in C with Linux.
2	Familiarization of console VO and operators in C. a. Display "Hello World" b. Read two numbers, add them and display their sum c. Read the radius of a circle, calculate its area and display it d. Evaluate the arithmetic expression ((a -b/c *d+ e) * (f +g)) and display solution. Read the values of the variables from the user through console.
3	Write a program to a. Calculate simple and compound interest. b. Find the roots of quadratic equation.
4	Write a program to swap values of two variables with and without using third variable.
5	Write a program to find the largest of three numbers with and without ternary Operators.
6	Write a program to input name, marks of 5 subjects of a student and display the name of the student, the total marks scored, percentage scored and the class of result.
7	Read a Natural Number and check whether the number is a. prime or not b. Armstrong or not C. even or odd.
8	Write a program to compute grade of students using if else adder. The grades are assigned as followed: Marks Grade marks<50 F 50 marks< 60 C 60 marks<70 B 70 marks<80 B+ 80 marks<90 A 90 marks<100 A+
9	Write a program to check whether the entered year is leap year or not (a year is leap if it is divisible by 4 and divisible by 100 or 400).
10	Write a program to find whether a character is consonant or vowel using switch statement.
11	Find the factorial of a given Natural Number n using recursive and non-recursive functions.
12	Compute sum of the elements stored in an array using pointers and user defined function.

Course Code: CSBB 102	PC (YES/ NO)	PE (YES/ NO)	OE (YES/ NO)	AS (YES/ NO)	HM (YES/ NO)	ST-IS-PR (YES /NO)	AE (YES/ NO)
	YES	NO	NO	NO	NO	NO	NO

Type of course	Program Core	
Course Title	**INTRODUCTION TO COMPUTER SYSTEMS**	
Course Objectives:	1. To understand the role and functions of a computer system. 2. To understand the memory hierarchy, and role of operating system along with system software. 3. To understand number systems, Boolean logic, and analyze logic gates through experiments. 4. To understand the importance of data communication, and computer networks.	
Course Outcomes	**CO1:** Understand the basic data types of High Level Languages that are stored and processed by Computer Systems.	**L1, L2**
	CO2: Understand the dynamic behavior of memory organization, operating systems and the system software used in software development.	**L2, L3, L4**
	CO3: Apply the concept of number systems, Logic gates and Boolean Algebra through programming experiments.	**L3, L4, L5**
	CO4: Explain the basic mechanisms involved in data communication, and computer networking.	**L1, L2**

Semester	**Autumn: Yes**		**Spring:**		
I	**Lecture**	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	3	0	2	4	36
Prerequisite course code as per proposed course numbers					
Prerequisite credits					
Equivalent course codes as per proposed course and old course					
Overlap course codes as per proposed course numbers					

Text Books:		
1	Title	Computer Systems: A Programmer's Perspective
	Author	Randal Bryant, David O'Hallaron
	Publisher	Pearson
	Edition	3rd edition, 2015
Reference Book:		
1	Title	Computer Fundamentals

	Author	Anita Goel
	Publisher	Pearson
	Edition	1st edition, 2010
2	Title	Computer Fundamentals And Programming In C
	Author	Reema Thareja
	Publisher	Oxford University Press
	Edition	Second edition, 2016
3	Title	Fundamentals of Computers
	Author	E Balagurusamy
	Publisher	McGraw Hill Education
	Edition	

Content	**Unit 1** Introduction to Computers: Digital and Analog Computers; Characteristics of Computer; Evolution of Computers; Generations of Computer; Classification of Computer, Input and Output devices. **Unit 2** Computer Memory: Introduction, Memory Representation, Memory Hierarchy, CPU Registers, Impact of Cache Memory on System Performance, Programming Language and Program Development. **Unit 3** Computer Software: Introduction, Types of Software, System Software, Application Software, Operating System (Introduction, Objectives of Operating System, Types of OS, Functions of OS, Protection and Security, User Interface, Examples of Operating Systems). **Unit 4** Data Representation: Introduction, Number System, Conversion from Decimal to Binary, Octal, Hexadecimal, Conversion of Binary, Octal, Hexadecimal to Decimal, Conversion of Binary to Octal, Hexadecimal, Conversion of Octal, Hexadecimal to Binary, Boolean Logic, Logic Gates. **Unit 5** Data Communication and Computer Network: Introduction, Importance of Networking, Data Transmission Media, Data Transmission across Media, Data Transmission and Data Networking, Computer Network, Network Types, Network Topology, Communication Protocol, Network Devices, Wireless Networking.
Course Assessment	Continuous Evaluation 25%
	Mid Semester 25%
	End Semester 50%

Course Matrix (CO-PO-PSO Mapping)

COs	POs & PSOs													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	**3**												**1**	
CO2	**2**	**1**	**1**										**2**	**2**
CO3	**3**	**2**	**2**	**2**	**2**	**1**							**2**	**2**
CO4	**3**	**2**	**2**	**2**	**2**	**2**							**2**	

1=addressed to small extent

2= addressed significantly

3= addressed strongly (major part of course)

Lab Experiments:

Exp. No.	List of Experiments
1	To identify front panel, indicator, and switches along with rear side connection in a computer system.
2	To familiarize with a computer system layout and mark the position of motherboard, HDD, CD/DVD drives, and add on cards.
3	To create a document about describing about yourself using MS-WORD.
4	To create, open, rename, copy and delete the files in command prompt.
5	To execute external commands in MS-DOS.
6	To write a C program for converting a binary number to its decimal equivalent.
7	To write a C program for converting a decimal number to its binary equivalent.
8	To implement basic logic gates using C programming.
9	To implement universal gates using C programming.
10	To implement special gates XOR, XNOR using C programming.

Course Code: CSBB 151	PC (YES/ NO)	PE (YES/ NO)	OE (YES/ NO)	AS (YES/ NO)	HM (YES/ NO)	ST-IS-PR (YES/NO)	AE (YES/ NO)
	YES	NO	NO	NO	NO	NO	NO
Type of course	Program Core						
Course Title	**DATA STRUCTURES**						
Course Objectives:	This course demonstrates familiarity with major algorithms and data structures and analyzes performance of algorithms. It is used to choose the appropriate data structure and algorithm design method for a specified application and determine which algorithm or data structure to use in different scenarios						

Course Outcomes		
	CO1: Explain the fundamentals of data structures and algorithms	L1
	CO2: Build data structures for a given problem	L2
	CO3: Illustrate applications and use of tree data structures	L3
	CO4: Compare algorithms for graph data structures	L4
	CO5: Compare the basic algorithmic techniques and choose a suitable one for a given problem	L5
	CO6: Develop algorithms using various searching and sorting techniques	L6

Semester	**Autumn:**		**Spring: YES**		
II	**Lecture**	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	3	0	2	4	36
Prerequisite course code as per proposed course numbers	NIL				
Prerequisite credits	NIL				
Equivalent course codes as per proposed course and old course	NIL				
Overlap course codes as per proposed course numbers	NIL				

Text Books:

1	Title	Fundamentals of Data Structures
	Author	E. Horowitz, S. Sahni
	Publisher	Computer Science Press
	Edition	2nd Edition, 2008

Reference Book:

1	Title	Data Structures Using C
	Author	A. M. Tanenbaum, Y. Langsam, M. J. Augenstein

	Publisher	Pearson Education
	Edition	1990
2	Title	Data Structures Using C
	Author	E. Balagurusamy
	Publisher	TATA McGraw Hill
	Edition	2013
3	Title	Data Structure and Program Design
	Author	R.L. Kruse
	Publisher	Prentice Hall
	Edition	2nd Edition, 1996

Content	**Unit** – 1 Introduction: Dynamic aspects of operations on data, Characteristics of data structures, Creation and manipulation of data structures, Operations on data structures, Types of data structures – linear and nonlinear. Introduction to algorithm: Asymptotic notations, Analysis of algorithms: Time and Space complexity. **Unit** – 2 Arrays: Dynamic memory allocation, one-dimensional arrays, multidimensional arrays, operations on arrays, storage – Row major order, Column major order. Linked lists: types of linked lists – singly, doubly and circularly linked lists, operations on linked lists. **Unit** – 3 Stacks: Implementation of stacks– array and linked list, operations on stacks Applications of Stacks, Notations – infix, prefix and postfix, Conversion and evaluation of arithmetic expressions using Stacks. Queues: Implementation of queues– array and linked list, operations on queues, Types of queues – queue double ended queue and priority queue. **Unit** – 4 Trees: Binary tree, Binary search tree, Threaded binary tree, Height balanced trees, Tries, Heaps, Hash tables. Graph traversals: Breadth First Search, Depth First Search, Shortest path: Depth first search in directed and undirected graphs Union-find data structure and applications. Directed acyclic graphs; topological sort. **Unit** – 5 Searching: Linear search, Binary search and Hashing. Algorithms and data structures for sorting: Insertion Sort, Bubble sort, Selection Sort, Merge sort, Quick Sort, Heap sort, Radix sort, Bucket sort. Algorithm design techniques: Divide and conquer, Greedy approach, dynamic programming.
Course Assessment	Continuous Evaluation 25% Mid Semester 25% End Semester 50%

Course Matrix (CO-PO-PSO Mapping)

COs	POs & PSOs													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	2	3	2		2		3	1	1	2	2	3
CO2	3	2	2	3	3		2	2		1	1	2	2	3
CO3				3	3				**2**				2	3
CO4	3				**2**							1	2	3
CO5		2	2	3	3		2	3		1	1	2	2	3
CO6	**3**			3	3		**2**	2	3	1	1	2	2	3

1=addressed to small extent

2= addressed significantly

3= addressed strongly (major part of course)

Lab Experiments:

Exp. No.	List of Experiments
1.	Write a Program in C to Implement Stacks Using Arrays and Linked Lists
2.	Write a Program in C to Implement Queues Using Arrays and Linked Lists
3.	Write a program that uses functions to perform the following operations on singly linked list i) Creation ii) Insertion iii) Deletion iv) Traversal
4.	Write a program that uses functions to perform the following operations on doubly linked list i) Creation ii) Insertion iii) Deletion iv) Traversal.
5.	Write a program that uses functions to perform the following operations on circular linked List i) Creation ii) Insertion iii) Deletion iv) Traversal
6.	Write a program that uses both recursive and non recursive functions to perform the following searching operations for a Key value in a given list of integers: a) Linear search b) Binary search
7.	Write a program that implements the following sorting i) Bubble sort ii) Selection sort iii)Quick sort.
8.	Write a program that implements the following i) Insertion sort ii) Merge sort iii)Heap sort.
9.	Write a program to perform the following operations: a) Insert an element into a binary search tree. b) Delete an element from a binary search tree. c) Search for a key element in a binary search tree
10.	Write a program to implement the tree traversal methods

Course Code: CSLB 152	PC (YES / NO)	PE (YES / NO)	OE (YES / NO)	AS (YES / NO)	HM (YES/ NO)	ST-IS-PR (YES /NO)	AE (YES/ NO)
	YES	NO	NO	NO	NO	NO	NO

Type of course	Program Core	
Course Title	**DISCRETE MATHEMATICS**	
Course Objectives:	The purpose of this course is to understand and use discrete mathematics which is the backbone of computer science. In this course the students will learn various ways for describing sets, i.e., logic and proofs, identify induction hypotheses and prove elementary properties of modular arithmetic, and apply graph theory models of data structures to solve problems of connectivity and constraint satisfaction.	
Course Outcomes:	**CO1:** Illustrate the basics of discrete mathematics and predicate calculus	
	CO2: Explain set theory and relations.	
	CO3: Demonstrate the concepts of graph theory and experiment with trees to solve problems like minimum spanning tree and tree traversals.	
	CO4: Develop the concept of functions and recursive function theory	
	CO5: Illustrate different algebraic structures	

Semester	**Autumn:**		**Spring: Yes**		
II	**Lecture**	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	3	1	0	4	36
Prerequisite course code as per proposed course numbers	NIL				
Prerequisite credits	NIL				
Equivalent course codes as per proposed course and old course	NIL				
Overlap course codes as per proposed course numbers	NIL				

Text Books:		
1	Title	Discrete Mathematics and applications
	Author	K.H.Rosen
	Publisher	TataMcGraw Hill
	Edition	fifth edition 2003
Reference Book:		
2	Title	Elements of Discrete Mathematics
	Author	C.L.Liu
	Publisher	McGraw-Hill Book Company.

	Edition	Second edition 1985
3	Title	Discrete Mathematics for Computer Scientists and Mathematicians
	Author	J .L.Mott, A.Kandel, T.P .Baker
	Publisher	Prentice Hall of India
	Edition	Second edition 1986
4	Title	Logic and Discrete Mathematics
	Author	W.K.Grassmann and J.P.Tremblay
	Publisher	Pearson
	Edition	1995

Content	**Unit** – 1 (5 Hours) Mathematical reasoning; propositions; negation disjunction and conjunction; implication and equivalence; truth tables; predicates; quantifiers; natural deduction; rules of Inference; methods of proofs; use in program proving; resolution principle. **Unit**-2 (10 Hours) Set theory; Paradoxes in set theory; inductive definition of sets and proof by induction; Peono postulates; Relations; representation of relations by graphs; properties of relations;equivalence relations and partitions; Partial orderings; Posets; Linear and well-ordered sets **Unit** – 3 (7 Hours) Graph Theory; elements of graph theory, Euler graph, Hamiltonian path, trees, tree traversals, spanning trees. **Unit** – 4 (7 Hours) Functions; mappings; injection and surjections; composition of functions; inverse functions;special functions; Peono postulates; pigeonhole principle; recursive function theory. **Unit** – 5 (7 Hours) Definition and elementary properties of groups, semigroups, monoids, rings, fields, vector spaces and lattices. Elementary combinatorics; counting techniques; recurrence relation; generating functions.
Course Assessment	Continuous Evaluation 25% Mid Semester 25% End Semester 50%

Course Matrix (CO-PO Mapping)

COs	POs													
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO11	PO12	PSO1	PSO2
CO1	3	2	2										3	
CO2	3	3	3								2	3	3	2
CO3	3	3	3		3		3		3			3	3	3
CO4	3	2	3		2		2					3	3	
CO5	2	2	2								2	2	2	3

1=addressed to small extent

2= addressed significantly

3= addressed strongly (major part of course)

Course Code: CSLB 153	PC (YES / NO)	PE (YES / NO)	OE (YES / NO)	AS (YES / NO)	HM (YES/ NO)	ST-IS-PR (YES/NO)	AE (YES/ NO)
	YES	NO	NO	NO	NO	NO	NO
Type of course	Program Core						
Course Title	**SYSTEM PROGRAMMING**						
Course Objectives:	This course introduces basic understanding of the concepts of System Programming. Understanding of the Design, write, and test moderately complicated low-level programs using a systems programming language. Implement routines that read and write structured binary files such as word processing documents, index systems, or serialized hierarchical data. Understanding the various methods of Linkers and Loaders, interpreters and debugging methods to manage system memory.						
Course Outcomes:	**CO1:** Apply the knowledge of assembler and macro processors to convert assembly languageinto machine code.						L4, L2
	CO2: Analyse working phases of Compiler toundertake meaningful language translation.						L6, L3
	CO3: Evaluate Linkers, Loaders, Interpreters and debugging methods to manage system memory and provide a portable runtime environment.						L5
	CO4: Analyse the working of an operatingsystem and its components.						L6, L2
Semester	**Autumn:**			**Spring: Yes**			
	Lecture		**Tutorial**	**Practical**	**Credits**		**Total teaching Hours**
Contact Hours	3		0	0	3		
Prerequisite course code as per proposed course numbers							
Prerequisite credits	NIL						
Equivalent course codes as per proposed course andold course	NIL						
Overlap course	NIL						

codes as per proposed course numbers					
Text Books:					

1	Title	Systems Programming,
	Author	Donovan John J.,
	Publisher	New York, Tata Mc-Graw Hill
	Edition	2014
2	Title	Introduction to Systems Software,
	Author	Dhamdhere, D.M.,
	Publisher	Tata Mc-Graw Hill
	Edition	1996
Reference Book:		
1	Title	Principles of compiler Design
	Author	Aho A.V. and J.D. Ullman
	Publisher	Addison Wesley/Narosa
	Edition	
2	Title	System Software- An Introduction to System Programming
	Author	L.L. Beck
	Publisher	Addition Wesley
	Edition	3rd edition, 1996
Content	**Unit** – 1 **Introduction:** Evolution of the Components of a Programming System, Evolution of Operating Systems, Machine Structure, Machine Language and Assembly Language. **Unit** – 2 **Assemblers:** Design of Assembler, Table Processing: searching and sorting, Macro Language and the Macro Processor: Macro Instructions, Features of Macro facility, Implementation. **Unit** – 3 **Linkers and Loaders**: Concept of linking, Case study of Linker in x86 machines, various loading schemes, Design of an absolute loader, Design of a direct-linking loader. **Unit** – 4 **Compilers:** Statement of problem, Phases of the complier, Data structures, Recursion, Call and Return statements, Storage Classes-Use implementation, Block structure, Nonlocal GoTo's, Interrupts, Pointers. **Debuggers**: Introduction to various debugging techniques, Case Study: - Debugging in Turbo C++ IDE. **Unit** – 5 **Operating System**: I/O programming, Memory management, Processor management, Device management, Information management.	

Course Assessment	Continuous Evaluation 25%
	Mid Semester 25%
	End Semester 50%

Course Matrix (CO-PO-PSO Mapping)

COs	Pos													
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO11	PO12	PSO1	PSO2
CO1	3	3	-	-	-	-	-	-	-	-	-	-	3	-
CO2	3	3	2	3	-	-	-	-	-	-	-	-	3	-
CO3	3	3	-	3	-	-	-	-	-	-	-	-	3	3
CO4	2	2	3	3	-	-	-	-	-	-	-	-	2	3

1=addressed to small extent

2= addressed significantly

3= addressed strongly (major part of course)

Course Code: CSPB 154	**PC (YES / NO)**	**PE (YES / NO)**	**OE (YES/ NO)**	**AS (YES / NO)**	**HM (YES/ NO)**	**ST-IS-PR (YES/NO)**	**AE (YES/ NO)**
	YES	NO	NO	NO	NO	NO	NO
Type of course	Program Core						
Course Title	**INTRODUCTION TO HARDWARE**						
Course Objectives:	• To impact the knowledge of various hardware components of a computer. • To provide the skill of assembling the computer. • To impart the knowledge of various electronics components.						
Course Outcomes:	**CO1:** Identify various hardware components of a system.						L1
	CO2: To assemble the computer.						L2
	CO3: To test various electronic components using Digital Multimeter.						L1

Semester	**Autumn:**		**Spring: Yes**		
II	**Lecture**	**Tutorial**	**Practical**	**Credits**	**Total teaching Hours**
Contact Hours	0	0	2	1	
Prerequisite course code asper proposed course numbers	NIL				
Prerequisite credits	NIL				
Equivalent coursecodes as per proposed course andold course	NIL				
Overlap coursecodes as per proposed course numbers	NIL				
Course Assessment	Continuous Evaluation 25% Mid Semester 25% End Semester 50%				

Course Matrix (CO-PO Mapping)

COs	POs													
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO11	PO12	PSO1	PSO2
CO1	1		1		2	1							1	1
CO2	1		1		2	1			1				1	1
CO3	2		1		2	1							1	1

1=addressed to small extent

2= addressed significantly

3= addressed strongly (major part of course)

Lab Experiments:

Exp. No.	List of Experiments
1	To study the history of a Computer System.
2	Study of CPU
3	Study of RAM
4	Study of Processor
5	History of HDD and CDROM
6	Testing of Electronics components using a Digital Multimeter
7	Study of Motherboard
8	Simulation of electric circuit using any electric circuit simulator.
9	Study of peripheral devices I (Input- Keyboard, Mouse)
10	Study of peripheral devices II (Output- Printer, Monitor, Scanner)

Course Code: CSBB 202	PC (YES / NO)	PE (YES / NO)	OE (YES / NO	AS (YES / NO)	HM (YES/ NO)	ST-IS-PR (YES /NO)	AE (YES/ NO)
	YES	NO	NO	NO	NO	NO	NO
Type of course	Program Core						
Course Title	**DESIGN AND ANALYSIS OF ALGORITHMS**						
Course Objectives:	• To develop an understanding of the fundamental concepts of algorithms and data structures, including time and space complexity analysis. • To learn various techniques for designing efficient algorithms, such as divide-and-conquer, greedy algorithms, dynamic programming, and graph algorithms. • To learn how to use randomized algorithms, approximation algorithms, and NP-completeness to solve real-world problems and to evaluate the practical efficiency of algorithms. • To gain exposure to current research and developments in the field of algorithm design and analysis and to understand their practical implications.						
Course Outcomes:	**CO1:** Analyze the asymptotic time and space complexity of algorithms, and understand the trade-off between these complexities.						L4, L2
	CO2: Design and implement algorithms for solving various computational problems, using techniques such as divide-and-conquer, greedy algorithms, dynamic programming, and graph algorithms.						L6, L3
	CO3: Use randomized algorithms, approximation algorithms, and NP-completeness to solve real-world problems and to evaluate the practical efficiency of algorithms.						L5
	CO4: Develop strong problem-solving skills and a deep understanding of the principles of algorithm design and analysis.						L6, L2

Semester	**Autumn: Yes**		**Spring:**		
III	**Lecture**	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	3	0	2	4	36
Prerequisite course code as per proposed course numbers	NIL				
Prerequisite credits	NIL				
Equivalent course codes as per proposed course and old course	NIL				
Overlap course codes as per proposed course numbers	NIL				

Text Books:		
1	Title	Introduction to Algorithms
	Author	Cormen, Leiserson, Rivest

	Publisher	Prentice Hall of India
	Edition	3rd Edition 2010
Reference Book:		
1	Title	Fundamental of Computer algorithms.
	Author	Horowitz and Sahani
	Publisher	Universities Press
	Edition	Second edition 2008
2	Title	Computer Algorithms : Introduction to Design andAnalysis
	Author	Sara Baase and Allen Van Gelder
	Publisher	Pearson Education
	Edition	3rd Edition 1999
3	Title	Fundamental of Algorithms
	Author	Brassard Bratley
	Publisher	PHI
	Edition	1st Edition 1996
4	Title	Algorithms Design
	Author	M T Goodrich et. al.
	Publisher	John Wiley
	Edition	
5	Title	The Design and analysis of Algorithms
	Author	A V Aho et al
	Publisher	Pearson Education
	Edition	1st Edition 2002
7	Title	Algorithm Design
	Author	Jon. Kleinberg and E Tardos
	Publisher	Pearson Education
	Edition	1st Edition 2013

Content	**Unit** – 1 Introduction: Algorithms, Analysis of Algorithms, Design of Algorithms, Complexity of Algorithms, Asymptotic Notations, Growth of function, Recurrences and their solution methods. Sorting in polynomial Time: Insertion sort, Merge sort, Heap sort, and Quick sort Sorting in Linear Time: Counting sort, Radix Sort, Bucket Sort, Medians and order statistics. **Unit** – 2 Advanced Data Structure: Red Black Trees, Augmenting Data Structure, Binomial Heap, B-Tree, Fibonacci Heap, and Data Structure for Disjoint Sets, All kinds of Algorithms on these data structures, Dictionaries and priority Queues, mergeable heaps, concatenable queues.

	Unit – 3 Advanced Design and Analysis Techniques: Dynamic programming, Greedy Algorithm, Backtracking, Branch-and-Bound, Amortized Analysis. Graph Algorithms: Elementary Graph Algorithms, Breadth First Search, Depth First Search, Minimum Spanning Tree, Kruskal's Algorithms, Prim's Algorithms, Single Source Shortest Path, All pair Shortest Path, Maximum flow and Traveling Salesman Problem. **Unit** – 4 Dynamic Programming: Chained matrix multiplication, longest common subsequence. Divide and Conquer: Order Statistics – finding the median, exponentiation, matrix multiplication, LCS. Computational Geometry: Line segments, Optimal polygon triangulation. Approximate Algorithm: Travelling Salesman Problem, vertex-cover problem. **Unit** – 5 Primality testing, Integer factorization, Randomized algorithms, Probabilistic algorithms. String Matching algorithms: Rabin Karp, KMP, Boyer Moore. Introduction to problem classes – NP, NPC, NP-Hard.
Course Assessment	Continuous Evaluation 25% Mid Semester 25% End Semester 50%

Course Matrix (CO-PO Mapping)

COs	POs													
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO11	PO12	PSO1	PSO2
CO1	3			2								2	3	
CO2	3	3	3	2	3	2	2		3		2	3	3	2
CO3	3	3	3		3		3		3			3	3	3
CO4	3	2	3		2		2					3	3	

1=addressed to small extent

2= addressed significantly

3= addressed strongly (major part of course)

Lab Experiments:

Exp. No.	List of Experiments
1	Sort a given set of elements using the Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted. The elements can be read from a file or can be generated using the random number generator
2	Implement a Merge Sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted .The elements can be read from a file or can be generated using the random number generator.

3	A) Obtain the Topological ordering of vertices in a given digraph. B) Compute the transitive closure of a given directed graph using Warshall's algorithm
4	Implement 0/1 Knapsack problem using Dynamic Programming
5	From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijikstra's algorithm
6	Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.
7	A) Print all the nodes reachable from a given starting node in a digraph using BFS method. B) Check whether a given graph is connected or not using DFS method.
8	Find a subset of a given set S = {s1, s2,….., sN} of n positive integers whose sum is equal to a given positive integer d. For example, if S= {1, 2, 5, 6, 8} and d = 9 there are two solutions {1,2,6}and{1,8}.A suitable message is to be displayed if the given problem instance doesn't have a solution.
9	Implement any scheme to find the optimal solution for the Traveling Salesperson problem and then solve the same problem instance using any approximation algorithm and determine the error in the approximation.
10	Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.
11	Implement All-Pairs Shortest Paths Problem using Floyd's algorithm.
12	Implement N Queen's problem using Back Tracking.

Course Code: CSBB 203	PC (YES/ NO)	PE (YES/ NO)	OE (YES/ NO)	AS (YES/ NO)	HM (YES/ NO)	ST-IS-PR (YES /NO)	AE (YES/ NO)
	YES	NO	NO	NO	NO	NO	NO

Type of course	Program Core	
Course Title	**OPERATING SYSTEM**	
Course Objectives:	• To understand the role and functions of an operating system and its impact on the overall performance of a computer system. • To understand the concepts and techniques involved in process management, such as process creation, scheduling, interprocess communication, and synchronization. • To understand the concepts and techniques involved in memory management, such as virtual memory, swapping, paging, and segmentation. • To gain hands-on experience with the design and implementation of operating systems through programming projects and case studies.	
Course Outcomes	**CO1:** Understanding of the fundamental concepts, design principles, and implementation techniques of modern operating systems	**L2, L3**
	CO2: Ability to design, implement, and evaluate process management, memory management, file system management, and input/output management algorithms	**L5, L6**
	CO3: Ability to understand and implement distributed systems, such as client-server systems, distributed file systems, and distributed operating systems.	**L2, L3, L4**
	CO4: Hands-on experience with the design and implementation of operating systems through programming projects and case studies.	**L5, L6**

Semester	**Autumn: Yes**		**Spring:**		
III	**Lecture**	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	3	0	2	4	36
Prerequisite course code as per proposed course numbers					
Prerequisite credits					
Equivalent course codes as per proposed course and old course					
Overlap course codes as per proposed course numbers					

Text Books:		
1	Title	Operating System Concepts
	Author	Abraham Silberschatz, Peter B. Galvin, Greg Gagne
	Publisher	Addison-Wesley

	Edition	Sixth edition, 2003
Reference Book:		
1	Title	Modern Operating Systems
	Author	Andrew Tanenbaum
	Publisher	Prentice Hall
	Edition	
2	Title	Operating Systems
	Author	William Stallings
	Publisher	Prentice Hall
	Edition	
3	Title	An introduction to operating systems
	Author	Harvey M. Deitel
	Publisher	Addison-Wesley
	Edition	
4	Title	Operating Systems: Design and Implementation
	Author	Andrew Tanenbaum & Albert Woodhull
	Publisher	Prentice-Hall
	Edition	
5	Title	Operating System Design - The XINU Approach
	Author	Douglas Comer
	Publisher	Prentice-Hall
	Edition	
6	Title	Fundamentals of Operating Systems
	Author	A.M. Lister
	Publisher	Macmillan
	Edition	1979
Content	**Unit 1** Basics: Operating System Functionalities, Types of Operating Systems, ComputerArchitecture support to Operating Systems. **Unit 2** Process Management: Threads, Process Scheduling - Uniprocessor scheduling algorithms, Multiprocessor and Real-time scheduling algorithms, Process Synchronization - Peterson's Solution, Bakery. Algorithm, Hardware Support to Process Synchronization, Semaphores, Critical Regions, Monitors - Deadlock prevention, deadlock avoidance and Deadlock Detection and Recovery - BankersAlgorithm,. **Unit 3** Memory Management: Segmentation and space allocation, Basics of linking and loading, Demand Paging, Page replacement algorithms, Analysis of page allocation policies, Thrashing- Working Set.	

Content	**Unit 4** File Systems: Contiguous, Sequential and Indexed Allocation, File system interface, File System implementation, Case study of Unix File system, Mounting and Unmounting files systems, Network File systems. **Unit 5** I/O System: Disk Scheduling, Device drivers - block and character devices, streams, Character and Block device switch tables. Protection and Security - Accessibility and Capability Lists.
Course Assessment	Continuous Evaluation 25%
	Mid Semester 25%
	End Semester 50%

Course Matrix (CO-PO-PSO Mapping)

COs	POs & PSOs													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	**3**												**1**	
CO2	**2**	**1**	**1**										**2**	**2**
CO3	**3**	**2**	**2**	**2**	**2**	**1**							**2**	**2**
CO4	**3**	**2**	**2**	**2**	**2**	**2**							**2**	

1=addressed to small extent

2= addressed significantly

3= addressed strongly (major part of course)

Lab Experiments:

Exp. No.	List of Experiments
1	Basic of Unix Commands
2	Implementation of Process Related System Calls (Fork).
3	Implementation of System Calls (Open, Read, Write and Close) for File Managemen
4	Implementation of Process Synchronization
5	Implementation of Memory Management Using Address Translation
6	Implementation of FIFO Page Replacement Algorithms
7	Implementation of LRU Page Replacement Algorithms
8	Implementation of First Come First Serve and Shortest Job Fist Scheduling Algorithm
9	Implementation of Priority and Round Robin CPU Scheduling Algorithm
10	Implementation of Banker's Algorithm.
11	Implementation of Sleeping Barbar Problem in process synchronization
12	Implementation of Algorithm for Deadlock Detection

Course Code: CSBB 204	PC (YES/ NO)	PE (YES/ NO)	OE (YES/ NO)	AS (YES/ NO)	HM (YES/ NO)	ST-IS-PR (YES/NO)	AE (YES/ NO)
	YES	NO	NO	NO	NO	NO	NO
Type of course	Program Core						
Course Title	**DATABASE MANAGEMENT SYSTEMS**						
Course Objectives	• To understand the role and functions of a database management system and its impact on the overall performance of a computer system. • To understand the concepts and techniques involved in ER modeling. • To understand the SQL commands and relational algebraic expressions for query processing. • To gain hands-on experience with designing and implementing database management systems through programming projects and case studies.						

Course Outcomes		
	CO1: Learn the basic concepts of Database Systems	L2
	CO2: Model the real-world systems using Entity Relationship Diagrams and convert the ER model into a relational logical schema using various mapping algorithms	L3
	CO3: Make use of SQL commands and relational algebraic expressions for query processing	L4
	CO4: Simplify databases using normalization process based on identifie keys and functional dependencies and solve the atomicity, consistenc isolation, durability, transaction, and concurrency related issues of databases	L5

Semester	**Autumn: Yes**		**Spring**		
III	**Lecture**	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	3	0	2	4	36
Prerequisite course code as per proposed course numbers					
Prerequisite credits					
Equivalent course codes as per proposed course and old course					
Overlap course codes as per proposed course numbers					

Text Books:

1	Title	Fundamentals of Database Systems
	Author	R. Elmasri and S.B. Navathe
	Publisher	Pearson
	Edition	2016

Reference Book:

1	Title	Database Systems Concepts
	Author	H.f.Korth and Silberschatz
	Publisher	McGraw Hill
	Edition	
2	Title	Data Base Design
	Author	C.J. Date
	Publisher	Addison Wesley
	Edition	
3	Title	DBM and Design
	Author	Hansen and Hansen
	Publisher	PHI
	Edition	
4	Title	Database System
	Author	Hector Garcia-Molina, Jeff Ullman, and Jennifer Widom
	Publisher	Pearson
	Edition	2nd Edition

Content	**Unit 1** Introduction - General introduction to database systems; Database - DBMS disctinctio approaches to building a database, data models, database management system, thre schema architecture of a database, challenges in building a DBMS, various component of a DBMS. **Unit 2** Database design and ER Model:- Overview, ER-Model, Constraints, ER-Diagrams, ERD Issues, weak entity sets, Codd's rules, Relational Schemas, Introduction to UML Relational database model: Logical view of data, keys, integrity rules. Relationa Database design: features of good relational database design, atomic domain an Normalization (1NF, 2NF, 3NF, BCNF). **Unit 3** Relational algebra: introduction, Selection and projection, set operations, renaming Joins, Division, syntax, semantics. Operators, grouping and ungrouping, relation comparison. Calculus: Tuple relational calculus, Domain relational Calculus, calculus v algebra, computational capabilities. **Unit 4** SQL - Introduction, data definition in SQL, table, and key and foreign key definitions update behaviors. Querying in SQL - basic select-from-where block and its semantic nested queries- correlated and uncorrelated, notion of aggregation, aggregation functions group by and having clauses, embedded SQL. Data Storage and Indexes - fil organizations, primary, secondary index structures, various index structures - hash based, dynamic hashing techniques, multi-level indexes, and B+ trees. **Unit 5** Transaction management and Concurrency control: Transaction processing and Erro recovery - concepts of transaction processing, ACID properties, and serializability concurrency control, Lock based concurrency control (2PL, Deadlocks), Time stamping methods, optimistic methods, and database recovery management. Error recovery and logging, undo, redo, undo-redo logging and recovery methods.
Course Assessment	Continuous Evaluation 25%

	Mid Semester 25%
	End Semester 50%

Course Matrix (CO-PO-PSO Mapping)

COs	POs & PSOs														
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	**3**												**1**		
CO2	**2**	**1**	**1**										**2**	**2**	**1**
CO3	**3**	**2**	**2**	**2**	**2**	**1**							**2**	**2**	
CO4	**3**	**2**	**2**	**2**	**2**	**2**							**2**		

1=addressed to small extent

2= addressed significantly

3= addressed strongly (major part of course)

Lab Experiments:

Exp. No.	List of Experiments
1	Library Management system (File Handling)
2	Introduction to SQL • Installation of SQL-Server • SQL data definition • Constraints in SQL • Schema change Statement
3	Basic SQL Queries
4	Complex SQL Queries-1 • Nested Queries • Correlated Nested Queries • EXISTS Function in SQL • Aggregation Function
5	Complex SQL Queries-2 • Joined Tables • Aggregate Functions
6	Complex SQL Queries-3 • Grouping • EXISTS and UNIQUE functions • Aggregate Functions

7	Entity-Relationship Diagram from Case Study
8	Normalization of the Case Study
9	Webpage Connectivity with SQL Server Using XAMPP- 1
10	Webpage Connectivity with SQL Server Using XAMPP- 2
11	Mini DBMS Project
12	Mini DBMS Project

Course Code: CSBB 251	PC (YES/ NO)	PE (YES/ NO)	OE (YES/ NO)	AS (YES/ NO)	HM (YES/ NO)	ST-IS-PR (YES/NO)	AE (YES/ NO)
	YES	NO	NO	NO	NO	NO	NO
Type of course	Program Core						
Course Title	**COMPUTER ARCHITECTURE AND ORGANIZATION**						
Course objectives:	The purpose of this course is to have a thorough understanding of the basi structure and operation of a digital computer. Students will learn the basi operations involved in the execution of an instruction, interrupts and thei usage to implement I/O control and data transfers and identify the differen architectural design issues that can affect the performance of a computer such as RISC architecture, instruction set design, and addressing modes.						

POs		
	CO1: Identify functional units and illustrate register transfer operations.	**L1**
	CO2: Explain the internal organization of the computer and its instructions.	**L2**
	CO3: Understand fixed and floating point algorithms and apply micro program instructions.	**L3**
	CO4: Summarize the memory organization and pipelining concepts.	**L2**

Semester	**Autumn:**		**Spring: Yes**		
IV	**Lecture**	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	3	0	2	4	36
Prerequisite course code as per proposed course numbers					
Prerequisite credits					
Equivalent course codes as per proposed course and old course					
Overlap course codes as per proposed course numbers					

Text Books:		
1	Title	Computer Organization and Design - The Hardware/Software Interface
	Author	D. A. Patterson and J. L. Hennessy
	Publisher	Morgan Kaufmann
	Edition	2014
Reference Book:		
1	Title	Computer System Architecture
	Author	M. Morris Mano
	Publisher	Prentice Hall of India Pvt Ltd
	Edition	Third edition, 2002
2	Title	Computer Organization and Architecture - Designing for Performance

	Author	W. Stallings
	Publisher	Prentice Hall of India
	Edition	2002
3	Title	Computer Organization
	Author	C. Hamacher, Z. Vranesic and S. Zaky
	Publisher	McGrawHill
	Edition	2002
4.	Title	Computer Architecture and Organization
	Author	J .P. Hayes
	Publisher	McGraw-Hill
	Edition	1998

Unit - 1

Introduction: Function and structure of a computer Functional components of a Function and structure of a computer, Functional components of a compute Interconnection of components, Performance of a computer.

Unit -2

Representation of Instructions Representation of Instructions: Machine instruction Operands, Addressing : Machine instructions, Operands, Addressing modes, Instruction formats, Instruction sets, Instruction set architectures - CISC and RISC architectures.

Unit - 3

Processing Unit: Organization of a processor - Registers, ALU and Control unit, Dat path in a CPU, Instruction cycle, Organization of a control unit - Operations of a contro unit, Hardwired control unit, Microprogrammed control unit.

Unit – 4

Memory Subsystem: Semiconductor memories, Memory cells - SRAM and DRAM cells, Internal Organization of a memory chip, Organization of a memory unit, Erro correction memories, Interleaved memories, Cache memory unit - Concept of cache memory, Mapping methods, Organization of a cache memory unit, Fetch and write mechanisms, Memory management unit - Concept of virtual memory, Hardware suppor for memory management.

Unit – 5

Input/Output Subsystem: Access of I/O devices, I/O ports, I/O control mechanisms Program controlled I/O Interrupt controlled I/O and DMA controlled I/O I/O interface Program controlled I/O, Interrupt controlled I/O, and DMA controlled I/O, I/O interfaces - Serial port, Parallel port, PCI bus, SCSI bus, USB bus, I/O peripherals - Input devices Output devices, Secondary storage devices.

Continuous Evaluation 25%
Mid Semester 25%
End Semester 50%

Course Matrix (CO-PO-PSO Mapping)

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1	1	-	1	-	-	1	-	-	-	-	-	3	
CO2	2	1	-	1	-	-	1	-	-	-	-	-	3	2
CO3	3	2	-	1	-	-	2	-	-	-	2	-	3	3
CO4	3	2	-	1	-	-	2	-	-	-	1	-	3	

1=addressed to small extent

2= addressed significantly

3= addressed strongly (major part of course)

Lab Experiments:

Exp. No.	List of Experiments
1	To study and verify the truth table of logic gates.
2	Implement Half Adder and Full Adder using basic logic gates.
3	To simplify the given expression and to realize it using Basic gates and Universal gates
4	Implement Gray-to-Binary and Binary-to-Gray code conversion.
5	To implement 4 x 1 and 8 x 1 multiplexers
6	Verify the excitation table of various Flip Flops
7	To Design an 8-bit Arithmetic Logical Unit.
8	Design the control unit of a computer using either handwriting or microprogramming based on its register transfer language description.
9	To implement a simple instruction set computer with a control unit and a data path.
10	To design the data path of a computer from its register transfer language description.

Course Code: CSBB 252	**PC (YES/ NO)**	**PE (YES/ NO)**	**OE (YES/ NO)**	**AS (YES/ NO)**	**HM (YES/ NO)**	**ST-IS-PR (YES/NO)**	**AE (YES/ NO)**
	YES	NO	NO	NO	NO	NO	NO
Type of course	Program Core						
Course Title	**ARTIFICIAL INTELLIGENCE**						
Course Objectives:	Gain a comprehensive understanding of Artificial Intelligence, covering its historical development, problem-solving techniques, search strategies, logical reasoning, and planning methods, with a focus on practical applications, particularly in the field o robotics. Develop essential skills to tackle complex AI challenges effectively.						
Course Outcomes	**CO1:** Understand the basic concepts of AI.						**L1, L2**
	CO2: Apply search strategies to solve AI problems.						**L3**
	CO3: Apply knowledge representation and reasoning to solve real world AI Problems.						**L3**
	CO4: Explore machine learning concepts and algorithms for real world applications.						**L4**

Semester	**Autumn:**		**Spring: YES**		
IV	**Lecture**	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	3	0	2	4	36
Prerequisite course code as per proposed course numbers					
Prerequisite credits					
Equivalent course codes as per proposed course and old course					
Overlap course codes as per proposed course numbers					

Text Books:		
1	Title	Artificial intelligence : A Modern Approach,
	Author	Stuart Russell, Peter Norvig
	Publisher	Prentice Hall
	Edition	Fourth edition, 2020.
Reference Book:		
1	Title	Artificial Intelligence: A New Synthesis
	Author	Nils J. Nilsson
	Publisher	Morgan-Kaufmann, 1998.
	Edition	
2	Title	Heuristics: Intelligent Search Strategies for Computer Problem Solving

	Author	Judea Pearl
	Publisher	Addison-Wesley Publishing Company
	Edition	1984
Content	**UNIT 1** Introduction, History, Possible Approaches in AI, Automated Problem Solving Agent Intelligent Agent & Environment, Complex Problems and AI, Shannon number, Problem Representation in AI **UNIT 2** Search Strategies: Search introduction, Uninformed Search, Informed/Heuristic Search Beyond Classical Search, Local Search, Problem Reduction, Adversarial Search Constraint Satisfaction Problems **UNIT 3** Logic and Deduction: Logical Agents, Propositional logic and Predicate Logic, Inferencing By Resolution Refutation AI Planning: AI Planning, Robot introduction and types , Steps in Robot Motion Planning, Graph-based Planning, Graph Construction Methods and path planning in Configuration Space, Skeletonization, Collision Detection and Freespace Sampling Intruder Finding Problem, Probabilistic roadmaps(PRM)], Rapidly Exploring Random Trees (RRT) **UNIT 4** Quantifying Uncertainty, Basic of Probability, Probabilistic Reasoning, Bayes Ne Bayesian Network, Fuzzy Logic, Decisions Theory, Utility Function, Decision Network Markov Decision Process, Probabilistic Reasoning over time, Hidden Markov Model Kalman filter, Markov Chain Monte Carlo **UNIT 5** Reinforcement Learning, Learning Agent, Introduction to Machine Learning, Types of Machine Learning, Learning from experience: Reinforcement Learning, Background, Model based and Model free learning, TD and Q Learning, RL Applications, Learning from Example, Supervised learning : Introduction, Naive Bayes, Decision Tree Perceptrons, Neural Network, Introduction to Deep Learning. AI Applications and Ethics, Computer Vision and Robotics, natural language understanding, AI in Healthcare, Ethics of AI
Course Assessment	Continuous Evaluation 25%
	Mid Semester 25%
	End Semester 50%

Course Matrix (CO-PO-PSO Mapping)

COs	POs & PSOs													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	**3**												**2**	**2**

CO2	**2**	**3**	**3**	**3**	**3**								**3**	**3**
CO3	**2**	**2**	**3**	**3**	**3**								**3**	**3**
CO4	**2**	**2**	**3**	**3**	**3**								**3**	**3**

1=addressed to small extent

2= addressed significantly

3= addressed strongly (major part of course)

Lab Experiments:

Exp. No.	**List of Experiments**
1	Introduction to Prolog programming
2	Python Frameworks Tutorial (with Jupyter and Colab) and it's Data Structures
3	Searching in graph based problem space, exploring Uninformed search Techniques
4	Exploring Informed search Techniques (Vacuum world and Maze Problem)
5	Exploring Uninformed and Informed search Techniques (PACMAN Search Space)
6	Multi agent in a search space
7	Introduction Logical Agent and Knowledge representation using Prolog
8	Reasoning Under Uncertainty using Bayesian Learning
9	Reinforcement Learning using Q-Learning
10	Introduction to Machine Learning and Python libraries for Data Analysis (Pandas, NumPy, Matplotlib)

Course Code: CSBB 254	PC (YES/ NO)	PE (YES/ NO)	OE (YES/ NO)	AS (YES/ NO)	HM (YES/ NO)	ST-IS-PR (YES/NO)	AE (YES/ NO)
	YES	NO	NO	NO	NO	NO	NO
Type of course	Program Core						
Course Title	**SOFTWARE ENGINEERING**						
Course Objectives:	The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects. Topics include process models, software requirements, software design, software testing, software process/product metrics, risk management, quality management and UML diagrams						
Course Outcomes:	**CO1:** Analyze the basic knowledge in software engineering to learn the various software development process models.						L4, L2
	CO2: Develop the standard models about the software product that is to be engineered and the processes that provides a framework for the software engineering methodologies.						L6, L3
	CO3: Apply the knowledge of software engineering to analyse the progress of the projects with the issues raised and develop the process for software projects using real life experiments.						L5
	CO4: Evaluate the software project processes and components of the projects to identify the risks, manage the change to assure quality in software projects and also able to see the risks with the help of real time projects						L6, L2
Semester	**Autumn:**				**Spring: Yes**		
IV	**Lecture**		**Tutorial**		**Practical**	**Credits**	**Total teaching hours**
Contact Hours	3		0		2	4	36
Prerequisite course code as per proposed course numbers							
Prerequisite credits	NIL						
Equivalent course codes as per proposed course and old course	NIL						
Overlap course codes as per	NIL						

proposed course numbers					
Text Books:					
1	Title	Software Engineering, A practitioner's Approach			
	Author	Roger S. Pressman			
	Publisher	Mc Graw Hill International Edition.			
	Edition	6th edition			
2	Title	Software Engineering			
	Author	Sommerville			
	Publisher	Pearson Education			
	Edition	7th edition			
Reference Book:					
1	Title	Software Engineering, an Engineering approach			
	Author	James F. Peters, Witold Pedrycz			
	Publisher	John Wiley			
	Edition				
2	Title	Software Engineering principles and practice			
	Author	Waman S Jawadekar,			
	Publisher	The Mc Graw-Hill Companies.			
	Edition	2000			
Content	**Unit** – 1 **Introduction:** Definition of software and Software engineering, Need of Software engineering, Difference between Program and Product, Software development life cycle, Different life cycle models (waterfall, Iterative waterfall, Prototype, Evolutionary, Incremental and Spiral model), Agile software development and their characteristics, V-Model. Critical Comparisons of SDLC models. **Unit** – 2 **Software Requirements:** Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document. **Requirements engineering process:** Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management. **System models:** Context models, behavioral models, data models, object models, structured methods. **Unit** – 3 **Software Design:** Goals of good software design, Design strategies and methodologies, Data oriented software design, Structured Design: Structure chart, Coupling, Cohesion, Modular structure, Packaging, Object oriented design, Topdown and bottom-up approach, Design patterns, Structured Analysis: DFD, Data Dictionary, Software Measurement and Metrics: Various Size Oriented Measures: Halstead's software science, Function Point (FP) based measures, Cyclomatic Complexity Measures: Control flow				

	graphs. Development: Selecting a language, Coding guidelines, Writing code, Code documentation. **Unit** – 4 **Software Testing**: Testing process, Design of test cases, Functional Testing: Boundary value analysis, Equivalence class testing, Decision table testing, Cause effect graphing, Structural testing, Path testing, Data flow and mutation testing, Unit testing, Integration and system testing, Debugging, Alpha & beta testing, testing tools & standards **Unit** – 5 **Software Maintenance**: Management of maintenance, Maintenance process, Maintenance models, Regression testing, Reverse engineering, Software reengineering, Configuration management, documentation. **Quality Management**: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards. **Current trends in Software Engineering:** Software Engineering for projects and products. Introduction to Web Engineering and Agile process
Course Assessment	Continuous Evaluation 25%
	Mid Semester 25%
	End Semester 50%

Course Matrix (CO-PO-PSO Mapping)

COs	POs													
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO11	PO12	PSO1	PSO2
CO1	3	3	-	-	-	-	-	-	-	-	-	-	3	-
CO2	3	3	2	3	-	-	-	-	-	-	-	-	3	-
CO3	3	3	-	3	-	-	-	-	-	-	-	-	3	3
CO4	2	2	3	3	-	-	-	-	-	-	-	-	2	3

1=addressed to small extent

2= addressed significantly

3= addressed strongly (major part of course)

Lab Experiments:

Exp. No.	List of Experiments
1.	INTRODUCTION TO RAPTOR 1) Find the roots of a quadratic equation. 2) Print all the numbers until the given number.

	3) Find the largest among the given three numbers.
2.	1) Find whether a number is prime. 2) Find the factorial of a given number. 3) Generate pyramid pattern. 4) Find the minimum and maximum element in an array.
3.	To draw different levels of DFD.
4.	To draw an ER diagram on UPS system.
5.	To draw a (a) Use case diagram of ATM system. (b) Use case diagram of online shopping system (c) UML diagram of credit card processing
6.	Create a project of parking pass for students using ASANA software.
7.	Create a project of purchasing MATLAB software using ASANA software.
8.	To draw a Gantt chart and network diagram.
9.	To draw a structured chart.
10.	Development of design Document.
11.	Development of SRS document.
12.	To draw different levels of DFD.
13.	To draw a sequence diagram and collaboration diagrams.

Course Code: CSBB 311	PC (YES/ NO)	PE (YES/ NO)	OE (YES/ NO)	AS (YES/ NO)	HM (YES/ NO)	ST-IS-PR (YES/NO)	AE (YES/ NO)
	NO	YES	NO	NO	NO	NO	NO
Type of course	**Program Elective**						
Course Title	**MACHINE LEARNING**						
Course Objectives:	With the increased availability of data from varied sources, there has been increasing attention paid to the various data driven disciplines such as analytics and machine learning. This course aims to provide students with the knowledge of key concepts of machine learning from a mathematically well motivated perspective. The course aims to familiarize the students with the two broad categories of machine learning algorithms supervised and unsupervised.						
Course Outcomes:	**CO1:** Learn the basics and mathematical background of Machine learning.						L1, L2
	CO2: Data exploratory analysis before applying machine learning						L2, L3
	CO3: Compare machine learning techniques						L2, L3, L4
	CO4: Apply Machine learning in real life applications.						L4, L5, L6

Semester	**Autumn:**		**Spring: Yes**		
	Lecture	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	3	0	2	4	36
Prerequisite course code as per proposed course numbers	NIL				
Prerequisite credits	NIL				
Equivalent course codes as per proposed course and old course	NIL				
Overlap course codes as per	NIL				

proposed course numbers					
Text Books:					
1	Title	Introduction to Machine Learning			
	Author	Ethem ALPAYDIN			
	Publisher	The MIT Press			
	Edition	2004			
2	Title	Pattern recognition and machine learning			
	Author	Bishop, C. M.			
	Publisher	New York: Springer			
	Edition	2007			
Reference Book:					
1	Title	Machine Learning,			
	Author	Tom Mitchel			
	Publisher	McGraw Hill			
	Edition				
2	Title	Machine learning in action			
	Author	Harrington, Peter.			
	Publisher	Manning Publications Co			
	Edition	2002			
Content	**Unit 1:** Supervised Learning Machine learning basics, Artificial Neural Network, Classifying with k Nearest Neighbors, Splitting datasets one feature at a time: decision trees, Classifying with probability theory: naive Bayes, Support vector machines, Improving classification with the AdaBoost meta algorithm. **Unit** 2: Unsupervised Learning Grouping unlabeled items using k means clustering, Association analysis with the Apriori algorithm, Efficiently finding frequent itemsets with FP growth. **Unit**3: Reinforcement learning Markov decision process (MDP), Bellman equations, Value iteration and policy iteration, Linear quadratic regulation (LQR), Linear Quadratic Gaussian (LQG), Q learning, Value function approximation, Policy search, POMDPs. **Unit** 4: Forecasting and Learning Theory Predicting numeric values: regression, Logistic regression, Tree based regression. Bias/variance tradeoff, Union and Chernoff/Hoeffding bounds, Vapnik– Chervonenkis (VC) dimension, Worst case (online) learning, Practical advice on how to use learning algorithms. **Unit** 5: Additional Tools Dimensionality reduction: Feature Extraction Principal component analysis to simplify data, Simplifying data with the singular value decomposition, Feature Selection – Ranking methods, subset selection – forward and backward. Big Data and MapReduce				

Course Assessment	Continuous Evaluation 25%
	Mid Semester 25%
	End Semester 50%

Course Matrix (CO-PO-PSO Mapping)

COs	POs													
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO11	PO12	PSO1	PSO2
CO1	3	2											1	2
CO2	2	3	2	3	3								3	2
CO3	2	2	2	3	3								3	3
CO4	2	2	3	3	3								3	3

1=addressed to small extent

2= addressed significantly

3= addressed strongly (major part of course)

Lab Experiments:

Exp No.	List of Experiments
1	Python Frameworks Tutorial (with Jupyter and Colab) and it's Data Structures
2	Introduction to Python libraries for Data Analysis (Pandas, NumPy, Matplotlib)
3	Data Collection & Creation Using Web Scraping- Static and Dynamic Webpages
4	Exploratory Data Analytics and Feature Engineering
5	Regression Techniques: Linear and Logistic
6	Traditional Computational Techniques (Decision Tree, KNN)
7	Implementing Classifier with Probability Theory(naïve Bayes and Bayesian Networks)
8	Implementation of Perceptron for logic gates (AND, OR, NOT)
9	Implementing Support Vector Machine Classifier from scratch
10	Neural networks for Binary Classification
11	Introduction to Reinforcement Learning: Path finding bot problem

Course CODE: CSBB 313	PC (YES/ NO)	PE (YES/ NO)	OE (YES/ NO)	AS (YES/ NO)	HM (YES/ NO)	ST-IS-PR (YES/NO)	AE (YES/ NO)
	NO	YES	NO	NO	NO	NO	NO
Type of course	**Program Elective**						
Course Title	**DIGITAL IMAGE PROCESSING**						
Course objectives:	The course aims to cover techniques and tools for digital image processing, image transformation in spatial and frequency domains. It introduces image analysis techniques in the form of image segmentation. The course also aims to cover the processing of colored images. The course also aims to cover techniques and tools for digital image processing and to provide hands-on experience in applying these tools to process images. The students would be encouraged to develop the image processing tools from scratch, rather than using any image processing library functions. Students will also get an opportunity to familiarize with image processing platforms such as Open CV, MATLAB, etc.						
Course Outcomes:	**CO1:** Learn the basics and mathematical background of Machine learning					L1, L3	
	CO2: Data exploratory analysis before applying machine learning					L2	
	CO3: Utility of image compression techniques for storage and transmission purpose.					L3, L5	
	CO4: To learn about color imaging, color models, and color image processing.					L4	

Semester	**Autumn: YES**		**Spring: YES**		
	Lecture	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	3	0	2	4	36
Prerequisite course code as per proposed course numbers	NIL				
Prerequisite credits	NIL				
Equivalent course codes as per proposed course and old course	NIL				
Overlap course codes as per proposed course numbers	NIL				

Text Books:		
1	Title	Digital Image Processing
	Author	R.C. Gonzalez, R.E Woods

	Publisher	Pearson Education
	Edition	3rd Edition, 2008
Reference Book:		
1	Title	Digital Image Processing Using MATLAB
	Author	R.C. Gonzalez, R.E Woods, S. L. Eddins
	Publisher	PHI
	Edition	2003
2	Title	Image Processing, Analysis, and Machine Vision
	Author	M. Sonka, V. Hlavac, R. Boyle
	Publisher	Brooks/Cole
	Edition	3rd edition, 2007
3	Title	Digital Image Processing
	Author	W.K. Pratt
	Publisher	Wiley-Interscience
	Edition	4th Edition, 2007

Content

Unit-1

Introduction: Digital image representation, Fundamental steps in image processing, Components of Digital Image processing systems, Elements of visual perception, Image Formation model, Image Sampling and quantization, Relationship between pixels – neighborhood, adjacency connectivity, regions, boundaries and distance measures.

Unit-2

Image Enhancement: Enhancement by point processing, Sample intensity transformation, Histogram processing, Image subtraction, Image averaging, Spatial filtering- Smoothing Spatial filters, Sharpening Spatial filters, Frequency domain- Fourier Transform, Low-Pass, High-Pass, Laplacian, Homomorphic filtering.

Unit-3

Image Segmentation: Detection of discontinuities – point, line and edge detection, Edge linking and boundary detection, Thresholding,
Region-based segmentation – region growing, region splitting and merging, Use of motion in segmentation- Spatial techniques and Frequency domain techniques.

Unit-4

Image Compression: Coding redundancy, Interpixel redundancy, fidelity criteria, Image compression models, Error-free compression, Variable length coding, Bit-plane coding, Lossless predictive coding, Lossy compression, Image compression standards, Real-Time image transmission JPEG and MPEG.

	Unit-5 Color Image Processing: Color Models, Pseudo color Image Processing, Color Transformations, Smoothing and sharpening, Image Segmentation based on color.
Course Outcomes	• Introduce the students to the fundamental techniques and algorithms used for acquiring, processing and extracting useful information from digital images. • Analysis and study of methods used for image sampling and quantization, image transforms, image enhancement and restoration, image encoding, image analysis and pattern recognition • Learn how to apply the methods to solve real world problems in several areas including medical, remote sensing and surveillance and develop the insight necessary to use the tools of digital imageprocessing (DIP) to solve any new problem.
Course Assessment	Continuous Evaluation 25%
	Mid Semester 25%
	End Semester 50%

Course Matrix (CO-PO-PSO Mapping)

COs	POs & PSOs													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	**3**												**1**	
CO2	**2**	**1**	**1**										**2**	**2**
CO3	**3**	**2**	**2**	**2**	**2**	**1**							**2**	**2**

1=addressed to small extent
2= addressed significantly
3= addressed strongly (major part of course)

Lab Experiments:

Exp. No.	List of Experiments
1.	Simulation and Display of an Image, Negative of an Image(Binary & Gray Scale)
2.	Implementation of transformations namely, translation, rotation, scale and shear
3.	Implementation of Histogram, and Histogram Equalization

4.	Implementation of FFT(1-D & 2-D) of an image
5.	Implementation of Image Compression by DCT
6.	Implementation of Image Smoothening Filters(Mean and Median filtering of an Image)
7.	Implementation of image sharpening filters and Edge Detection using Gradient Filters
8.	Implementation of image restoring techniques
9.	Implementation of image segmentation techniques
10.	Program for morphological operation: erosion and dilation

Course Code: CSLB 315	PC (YES/ NO)	PE (YES/ NO)	OE (YES/ NO)	AS (YES/ NO)	HM (YES/ NO)	ST-IS-PR (YES/NO)	AE (YES/ NO)
	NO	YES	NO	NO	NO	NO	NO
Type of course	**Program Elective**						
Course Title	**OPTIMIZATION TECHNIQUES**						
Course objectives:	This course aims to cover the concepts of optimization methods and algorithms developed for solving various types of optimizationProblems. To apply the mathematical results and numerical techniques of Optimization theory to various Engineering and Analytics problems. Explain the theoretical workings of the graphical, simplex, and analytical methods for making effective decision on variables so as to optimize the objective function.						

Course Outcomes:		
	CO1: To understand the fundamentals of Linear Programmingand Dynamic Programming.	**L1,L3**
	CO2: Enumerate the fundamentals of Integer programming technique andapply different techniques to solve various optimization problems arisingfrom engineering areas.	**L1,L2**
	CO3: Identify appropriate optimization methods to solve complex problems involved in various industries.	**L1,L2,L4**
	CO4: To understand the graphical, simplex, and analytical methods formaking effective decisions.	**L2,L5**

Semester	**Autumn: YES**		**Spring: YES**		
	Lecture	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	3	1	0	4	36
Prerequisite course code as per proposed course numbers	NIL				
Prerequisite credits	NIL				

Equivalent course codes as per proposed course and old course	NIL				
Overlap course codes as per proposed course Numbers	NIL				

Text Books:		
1	Title	An Introduction to Optimization
	Author	Edwin K.P. Chong, Stanislaw H. Zak,
	Publisher	Wiley
	Edition	4th
Reference Book:		
1	Title	Convex Optimization
	Author	Stephen Boyd
	Publisher	LievenVandenberghe
	Edition	3rd
2	Title	Modern Optimization with R (Use R)
	Author	Paulo Cortez
	Publisher	Springer
	Edition	2014

Content	**Unit** 1 Preliminaries: Proofs, Vector Spaces and Matrices, Linear Transformations, Eigenvalues and Eigenvectors, Orthogonal Projections, Quadratic Forms, Matrix Norms, Concepts from Geometry, Elements of Calculus **Unit** 2 Unconstrained Optimization: Basics of Set Constrained and Unconstrained Optimization, One Dimensional Search Methods, Golden Section Search, Fibonacci Search, Newton's Method, Secant Method, Solving Ax = b **Unit 3** Linear Programming: Introduction to Linear Programming, Simplex Method, Duality **Unit 4** Nonlinear Constrained Optimization: Problems with Equality Constraints, Problems with Inequality Constraints, Karush Kuhn Tucker Condition, Convex Optimization Problems, **Unit 5** Algorithms for Constrained Optimization: Projections, Project gradient methods, Penalty methods.
Course Assessment	**Continuous Evaluation 25%**
	Mid Semester 25%
	End Semester 50%

Course Matrix (CO-PO-PSO Mapping)

COs	POs & PSOs													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	**2**												**1**	
CO2	**1**	**1**	**2**										**1**	**2**
CO3	**1**	**2**	**1**	**2**	**2**	**1**							**2**	**2**
CO4	**2**	**2**	**2**	**2**	**1**	**2**							**2**	

1=addressed to small extent
2= addressed significantly
3= addressed strongly (major part of course)

Course Code: CSBB 424	PC (YES/ NO)	PE (YES/ NO)	OE (YES/ NO)	AS (YES/ NO)	HM (YES/ NO)	ST-IS-PR (YES/NO)	AE (YES/ NO)
	NO	YES	NO	NO	NO	NO	NO

Type of course	**Program Elective**	
Course Title	**DEEP LEARNING AND APPLICATIONS**	
Course objectives:	The purpose of this course is to provide the students with the advance knowledge of Machine learning. It aims to enable the students to understand the design of various Deep Learning models and application	
COs	**CO1:** Solve problems in linear algebra, probability, optimization, and machine learning.	L1, L2, L3
	CO2: Implement deep learning models in Python using the PyTorch library and train them with real-world datasets.	L4, L5, L6
	CO3: Design convolutional networks for handwriting and object classification from images or video.	L4, L5, L6
	CO4: Design recurrent neural networks with attention mechanisms for natural language classification, generation, and translation.	L4, L5, L6

Semester	**Autumn: YES**		**Spring: YES**		
	Lecture	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	3	0	2	4	36
Prerequisite course code as per proposed course numbers	CSL501				
Prerequisite credits	NIL				
Equivalent course codes as per proposed course and old course	NIL				
Overlap course codes as per proposed course numbers	NIL				

Text Books:		
1	Title	Deep Learning
	Author	Ian Goodfellow and Yoshua Bengio and Aaron Courville.
	Publisher	MIT Press
	Edition	2016
Reference Book:		
1	Title	Machine Learning: An Algorithmic Perspective, Second Edition
	Author	Stephen Marsland
	Publisher	Chapman and Hall/CRC
	Edition	2nd
2	Title	Introduction to Probability For Data Science
	Author	Stanley H. Chan

	Publisher	Michigan Publishing
	Edition	May 2021
Content	**Unit** – 1 **Introduction:** Well posed learning problem, Types of Machine Learning, Applications, Linear Algebra, Probability and Information Theory, Numerical Computation **Unit** – 2 **Traditional Machine Learning Basics:** Linear Regression, Logistic, Regression, k Nearest Neighbors, Classifier with Probability Theory, Decision Trees, Random Forest, Support Vector Machine, **Artificial Neural Network:** Artificial Neuron, Perceptron, Stochastic Gradien Descent, and Back Propagation Neural Network, Neural Network Architecture, NN with One Hidden Layer, NN with One Hidden Layer and Multiple Outputs, Neura Network Hyper-parameters **Unit** – 3 **Deep Architecture:** need, applications, Hyper-parameters in Deep Neural Networks (Encoding, Layers, Loss function, Learning Rate, Momentum and Optimization Regularization and dropout, Batch Norms) , vanishing gradient problem, and way to mitigate it **Convolution Neural Network:** from Dense Layers to Convolutions, pooling layers CNN Architectures (AlexNet, VGG, NiN, GoogLeNet, ResNet, DensNet), Application in Image segmentation, Automated Object Detection models. **Unit** – 4 **Deep Sequence Models:** Sequence Modeling Problems, Motivation and Applications, Traditional Models: Recurrent Neural Networks, Back-propagation through time; Modern Recurrent Neural Networks: Gated Recurrent Units, Lon Short Term Memory (LSTM), Deep Recurrent Neural Networks, automatic imag captioning, video to text with LSTM models. Unit- 5 **Deep Unsupervised Learning:** Latent variable models, Autoencoders, Deep Generative Modeling: Variational Autoencoders, Generative Adversarial Networks (GANs), Recent Advance, Image generation with Generative adversarial networks, **Advance Topic in Deep Learning:** Transfer Learning: Need and motivation, Transfer Learning Process, Data Augmentation, Applications **Unit** –6 **Deep Reinforcement Learning:** Components of an RL - (Agent, Policy, Valu function, Model), MDP, DP, TDL, Q-Learning. SARSA Learning, Deep-Reinforcemen Learning Need and Applications, Types of Deep-RL : Deep Q-Network (DQN) , Polic Gradient [Advantage Actor-Critic (A2C/A3C), DDPG, PPO] , Alpha zero Future Trends in Deep Leaning, Attention models for computer vision tasks.
Course Assessment	Continuous Evaluation 25%
	Mid Semester 25%
	End Semester 50%

Course Matrix (CO-PO-PSO Mapping)

COs	POs & PSOs													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	3	3								3	3
CO2	2	2	3	3	3								3	3
CO3	2	2	3	3	3								3	3
CO4	3	2	3	3	3								3	3

1=addressed to small extent
2= addressed significantly
3= addressed strongly (major part of course)

Lab Experiments:

Exp. No.	List of Experiments
1	Python Frameworks Tutorial (with Jupyter and Colab) and it's Data Structures
2	Introduction to Python libraries for Data Analysis (Pandas, NumPy, Matplotlib)
3	Data Collection & Creation Using Web Scraping- Static and Dynamic Webpages
4	Exploratory Data Analytics and Feature Engineering
5	Regression Techniques: Linear and Logistic
6	Traditional Computational Techniques
7	Implementation of Perceptron for logic gates (AND, OR, NOT)
8	Neural networks for Binary Classification
9	Building CNN Image classifier using keras for image classification
10	Introduction to Sequence Models for Prediction
11	Financial Planning via Deep Reinforcement Learning

Course Offered to Other Departments

Course Code: CSBB 111	PC (YES/ NO)	PE (YES/ NO)	OE (YES/ NO)	AS (YES/ NO)	HM (YES/ NO)	ST-IS-PR (YES /NO)	AE (YES/ NO)
	NO	NO	YES	NO	NO	NO	NO

Type of course	Open Elective	
Course Title	**COMPUTER PROGRAMMING**	
Course Objectives:	1. To understand the computational model of Computer. 2. To understand the concepts of C programming. 3. To apply functions of C programming for solving problems. 4. To understand the concept of file management in C.	
Course Outcomes	**CO1:** Understand the basics of computer and various problem solving approaches.	**L1, L2**
	CO2: Understand the fundamentals of C programming.	**L1, L2**
	CO3: Apply functions, arrays, and structures for solving problem.	**L2, L3, L4**
	CO4: Understand the use of pointers and file management in C.	**L2, L3**

Semester	**Autumn: Yes**		**Spring:**		
	Lecture	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	2	0	2	3	24
Prerequisite course code as per proposed course numbers					
Prerequisite credits					
Equivalent course codes as per proposed course and old course					
Overlap course codes as per proposed course numbers					

Text Books:		
1	Title	Programming in ANSI C
	Author	E. Balagurusamy
	Publisher	TATA McGraw Hil1
	Edition	6 edition, 2012
Reference Book:		
2.	Title	Let Us C
	Author	Yashwant Kanetkar
	Publisher	Infinity Science Press
	Edition	13th edition, 2012
	Title	Schaum's Outline of Programming with C
	Author	Byron S Gottfried

	Publisher	TATA McGraw Hill
	Edition	2 d edition, 1996
3.	Title	The C Programming Language
	Author	Brian Kernighan & Dennis Ritchie
	Publisher	Prentice Hal
	Edition	2nd edition, 1988

Content	**Unit 1** Introduction and Characteristics of Computers, Applications, Notion of Algorithms, Development of Flowcharts, Number system: Introduction and type of Number System, Arithmetic Operations in Number System, Signed and Unsigned Number System. **Unit 2** Introduction to programming language, Characteristics of C Language, Identifiers and Keywords, Basic Data types - int, float double, char, Bool, Void, Constant and Variables, Declarations and Statements, Representation of Expressions, Classification of operators, precedence and association, type conversion and typecasting, formatted input and output. **Unit 3** Introduction to Decision Control Statements, Conditional statements- If-else, Switch-case constructs, Iterative statements, Loops- While, do-while, for. **Unit 4** Function - User defined functions, library functions, Parameter passing call by value, call by reference, recursion. **Unit 5** Arrays- Advantages and drawbacks, One dimensional, Multi-Dimensional Arrays and strings: Declaration, initialization, Accessing, Passing arrays and strings as parameters to functions. Pointers, Structures
Course Assessment	Continuous Evaluation 25%
	Mid Semester 25%
	End Semester 50%

Course Matrix (CO-PO-PSO Mapping)

COs	POs & PSOs													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	**3**												**1**	
CO2	**2**	**1**	**1**										**2**	**2**
CO3	**3**	**2**	**2**	**2**	**2**	**1**							**2**	**2**
CO4	**3**	**2**	**2**	**2**	**2**	**2**							**2**	

1=addressed to small extent

2= addressed significantly

3= addressed strongly (major part of course)

Lab Experiments:

Exp. No.	List of Experiments
1	Familiarization of Linux environment - How to do Programming in C with Linux.
2	Familiarization of console VO and operators in C. a. Display "Hello World" b. Read two numbers, add them and display their sum c. Read the radius of a circle, calculate its area and display it d. Evaluate the arithmetic expression ((a -b/c *d+ e) * (f +g)) and display solution. Read the values of the variables from the user through console.
3	Write a program to a. Calculate simple and compound interest. b. Find the roots of quadratic equation.
4	Write a program to swap values of two variables with and without using third variable.
5	Write a program to find the largest of three numbers with and without ternary Operators.
6	Write a program to input name, marks of 5 subjects of a student and display the name of the student, the total marks scored, percentage scored and the class of result.
7	Read a Natural Number and check whether the number is a. prime or not b. Armstrong or not C. even or odd.
8	Write a program to compute grade of students using if else adder. The grades are assigned as followed: Marks Grade marks<50 F 50 marks< 60 C 60 marks<70 B 70 marks<80 B+ 80 marks<90 A 90 marks<100 A+
9	Write a program to check whether the entered year is leap year or not (a year is leap if it is divisible by 4 and divisible by 100 or 400).
10	Write a program to find whether a character is consonant or vowel using switch statement.
11	Find the factorial of a given Natural Number n using recursive and non-recursive functions.
12	Compute sum of the elements stored in an array using pointers and user defined function.
13	Write a program to add two distances in feet and inches using structures
14	Write a program to read two complex numbers using structures and perform their addition, subtraction and display result.

Course Code: CSBB 351	**Open course (YES/NO)**	**HM Course (Y/N)**	**DC (Y/N)**	**DE (Y/N)**
	No	No	Yes	NO
Type of course	Core			
Course Title	**Compiler Design**			
Course Coordinator				
Course Objectives:	• To understand the structure and functions of a compiler. • To understand the various phases of compiler design. • To learn context-free grammar, compiler parsing techniques, and construction of abstract syntax trees. • To learn how to generate intermediate code by applying various code optimization strategies. • To gain hands-on experience designing and implementing a mini compiler through programming projects and case studies.			
Course Outcomes	**CO1:** Identify the phases of compilers for a programming language.			**L1, L2**
	CO2: Learn context-free grammars, compiler parsing techniques, construction of abstract syntax trees.			**L3**
	CO3: Generate syntax-directed translation rules for a given context-free grammar by examining S-attributed and L-attributed grammars and practice of compiler implementation and generate intermediate code by applying various code optimization strategies.			**L4**
	CO4: Build a domain-specific mini compiler.			**L5**

Semester	**Autumn: No**		**Spring: Yes**		
V	**Lecture**	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	3	0	2	4	60
Prerequisite course code as per proposed course numbers					
Prerequisite credits					

Equivalent course codes as per proposed course and old course					
Overlap course codes as per proposed course numbers					
Text Books:					
1.	Title	Compilers Principles, Techniques, and Tools			
	Author	Alfred Aho, Ravi Sethi, Jeffrey D Ullman			
	Publisher	Pearson			
	Edition	2014			
Reference Book:					
1.	Title	The Theory and Practice of Compiler Writing			
	Author	Tremblay, Sorenson			
	Publisher	Addison-Wesley			
	Edition	1992			
2.	Title	Compiler Design in C			
	Author	Holub			
	Publisher	PHI			
	Edition				

Content	**Unit 1** **Introduction**: Compilers Analysis of the source program, Phases of a compiler, Cousins of the Compiler, Grouping of Phases, Bootstrapping and Compiler construction tools, Symbol Table. **Lexical Analysis:** Role of Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, From Regular expression to Automata and Design of Lexical Analysis generator. **Unit 2** **Syntax Analysis**: Role of the parse, Writing Grammars, Context-Free Grammars, Ambiguous Grammars, Top-Down parsing, Recursive Descent Parsing, Predictive Parsing, Bottom-up parsing, Shift Reduce Parsing, Operator Precedence Parsing, LR Parsers, SLR Parser, Canonical LR Parser, LALR Parser. **Unit 3** **Syntax Directed Translation:** Syntax Directed Definitions, Application of SDT and SDT schemes. **Unit 4** **Intermediate Code Generation:** Directed acyclic graphs, three-address code Intermediate languages - Declarations, Assignment Statements, Boolean Expressions, Array references, Back patching. **Unit 5** **Code Generation and Optimization:** Issues, Basic Blocks, and Flow Graphs, DAG representation of Basic Blocks, Optimization of basic Blocks, Peephole Optimization, Principal Sources of Optimization, Loop Optimization, Global Data Flow Analysis.

List of Experiments	1. Identify the files created during the compilation and execution of a C program. 2. Create a program to generate a Lexical Analyzer to identify the following tokens: Keywords, Identifier, Operators, Digits, Literals, and Separators. 3. Read about the FLEX tool and its installation, and then create a program using FLEX to generate a Lexical Analyzer and identify the following tokens: Keywords, Identifier, Operators, Literals, and Separators. 4. Write a program using FLEX to count the number of Lines, Words, Capital Letters, Small Letters, Numbers, Digits, Special Character, Delimiter, Operator, Relational Operator, Total Characters. 5. Write a program using FLEX to read and validate a mathematical expression and display the result. The result should follow the BODMAS Rule. If the expression is invalid, then display it as invalid. 6. Mini Project: Domain/Application Specific Project: Create a lexical analyzer for the Mini Project 7. Perform the following: a. Write a C program to find First and Follow for all non-terminals. b. Explore YACC Tool. c. Download and Install the YACC Tool on your system and prepare an installation lab manual. 8. Perform the following: a. Write a YACC program to check if the entered statement is a valid arithmetic expression. b. Design a grammar for a declarative statement for the C program. Further, write a YACC program to check if the entered statement is a valid declarative statement according to the grammar generated. c. Design a grammar for a relational expression of C language. Further, write a YACC program to check if the entered statement is a valid relational expression according to the grammar generated. 9. Perform the following: a. Consider the example of a simple desk calculator. You are required to write YACC specifications for this grammar. Show your parsing sequence for the sample input string. b. Write a YACC program to recognize a series of expressions, each on a new line. Also, give a parse sequence for each of the expressions. c. For the grammar below, write a YACC program to compute the decimal value of an input string in binary. 10. Mini Project: Domain/Application Specific Project Create syntax analyzer for the Mini Project 11. Perform the following: a. Using YACC, write semantic actions to check if the parenthesis is balanced and count the number of matching parenthesis. P->(P) | a b. Using YACC, write semantic actions translating arithmetic expressions from infix into postfix notation. 12. Perform the following: a. Append semantic actions to the grammar of a simple desk calculator. b. Write a YACC program to evaluate sum, product, min, and max expressions. 13. Write a program to generate machine code (Assembly Code) from an abstract syntax tree generated by the parser.

Course Assessment	Continuous Evaluation 25%
	Mid Semester 25%
	End Semester 50%

Course Code: CSBB 352	**Open course (YES/NO)**	**HM Course (Y/N)**	**DC (Y/N)**	**DE (Y/N)**
	No	No	Yes	No
Type of course	Core			
Course Title	**Theory of App Development**			
Course Coordinator				
Course Objectives:	• To learn the basics of application development. • To get familiarized with various routinely used application development tools and techniques. • To know the development and utility of android applications. • To comprehend and adapt the android environment for application development. • To learn the frameworks and additional tools for development of applications aiming at improved user experience.			

Course Outcomes		
	CO1: Demonstrate understanding of the fundamentals of android environment.	**L2**
	CO2: Develop user interface and database for better user experience and efficient data access.	**L3**
	CO3: Applying development techniques for best user interface and experience in the android application.	**L4**
	CO4: Applying suitable connectivity controls to the android application.	**L4**

Semester	**Autumn: No**		**Spring: Yes**		
V	**Lecture**	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	2	0	2	3	
Prerequisite course code as per proposed course numbers					
Prerequisite credits					
Equivalent course codes as per proposed course and old course					
Overlap course codes as per proposed course numbers					

Text Books:		
1.	Title	Android App Development for Dummies
	Author	Michael Burton

	Publisher	Wiley
	Edition	Third edition
2.	Title	IOS App Development for Dummies
	Author	Jesse Feiler
	Publisher	Wiley
	Edition	First edition
3.	Title	Android Application Development - Black Book
	Author	Pradeep Kothari
	Publisher	Dreamtech Press
	Edition	First edition
Reference Book:		
1.	Title	Android Programming: The Big Nerd Ranch Guide
	Author	Bill Phillips and Brian Hardy
	Publisher	Big Nerd Ranch Guides
	Edition	Second edition
2.	Title	iOS Programming: The Big Nerd Ranch Guide
	Author	Christian Keur and Aaron Hillegass
	Publisher	Big Nerd Ranch Guides
	Edition	Fifth edition

Content	**Unit 1** **Introduction:** Fundamentals of Java for Android Application Development, Overview of Android, Developing Spectacular Android Applications, Fragments and Intents in Android, User Interface, Application Programming, Android Resources, App Widget. **Unit 2** **User Interface and Application Components:** Front-End Designing, User Input Handling, Data Storage, Status Bar, Pictures and Menus, Emailing and Networking in Android, Working with Location Services and Maps, Working with Graphics and Animation, Audio, Video and Camera. **Unit 3** **Extended Development, Publication and Connectivity :** Development for Tablets, Supporting Older Versions of Android, Application publication on Google Play Store, Monetizing and Distributing Android Applications. Working with Bluetooth, NFC, and Wi-Fi, Telephony and SMS and Hardware Sensors. **Unit 4** **User Experience Enhancement:** Memory management, Working with the Source Editor, Adding Outlets and Actions, and Adding Animation and Sound, to the application. The Trip Model: Implementing the Master View Controller, Working with Split View Controllers and the Master View, Finishing the Basic App Structure, Working with Web Views, Displaying Events Using a Page View Controller, Geocoding, and Location Selection.
List of Experiments	1. Installation of Android studio. 2. Development Of Hello World Application 3. Create an application that takes the name from a text box and shows hello message along with the name entered in text box, when the user clicks the OK button 4. Create a screen that has input boxes for User Name, Password, Address, Gender (radio buttons for male and female), Age (numeric), Date of Birth (Date Picket), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button (use any layout) 5. Design an android application to create page using Intent and one Button and pass the Values from one Activity to second Activity 6. Design an android application Send SMS using Intent 7. Create an android application using Fragments 8. Design an android application Using Radiobuttons 9. Design an android application for menu. 10. Create a user registration application that stores the user details in a database table.
Course	Continuous Evaluation 25%

Assessment	Mid Semester 25%
	End Semester 50%

Course Code: CSBB 424	**Open course (YES/NO)**	**HM Course (Y/N)**	**DC (Y/N)**	**DE (Y/N)**
	No	No	No	Yes
Type of course	Elective			
Course Title	**Deep Learning and Applications**			
Course Coordinator				
Course Objectives:	Provide students with advanced knowledge of Machine Learning and Deep Learning, enabling them to design, implement, and apply various models.			
Course Outcomes	**CO1:** Understand foundational concepts of deep learning, including neural networks, training methods, and optimization techniques.			**L1, L2, L3**
	CO2: Develop and evaluate convolutional neural networks (CNNs) for image classification and object detection.			**L4, L5, L6**
	CO3: Design recurrent neural networks (RNNs) and Transformer models for NLP and time-series analysis.			**L4, L5, L6**
	CO4: Implement advanced deep learning techniques for generative modeling, speech recognition, and gesture recognition.			**L4, L5, L6**

Semester	**Autumn: No**		**Spring: Yes**		
V	**Lecture**	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	3	0	2	4	60
Prerequisite course code as per proposed course numbers	Machine Learning Course				
Prerequisite credits					
Equivalent course codes as per proposed course and old course					
Overlap course codes as per proposed course numbers					

Text Books:		
1.	Title	Deep Learning
	Author	Ian Goodfellow and Yoshua Bengio and Aaron Courville
	Publisher	MIT Press
	Edition	2016

Reference Book:		
1.	Title	Machine Learning: An Algorithmic Perspective
	Author	Stephen Marsland
	Publisher	Chapman and Hall/CRC
	Edition	2014
2.	Title	Introduction to Probability for Data Science
	Author	Stanley H. Chan
	Publisher	Michigan Publishing
	Edition	2021

Content	**Unit 1** **Introduction:** Overview of Deep Learning and its comparison with Machine Learning. Artificial Neural Networks (ANNs): Structure, perceptrons, limitations, forward propagation, backpropagation, activation functions (e.g., ReLU, Softmax), loss functions, and optimization techniques (e.g., SGD, Adam). Regularization methods (L1/L2, Dropout, Batch Normalization), hyperparameter tuning, and implementation frameworks (e.g., TensorFlow, PyTorch). **Unit 2** **Convolutional Neural Networks (CNN):** Key concepts: convolution, padding, pooling, and CNN layers. Architectures: AlexNet, VGG, ResNet, MobileNet, and others. Applications: Image classification, transfer learning, object detection (e.g., YOLO, RCNN), and image segmentation (e.g., Mask RCNN). Regularization and data augmentation techniques. **Unit 3** **Recurrent Neural Networks (RNN):** RNN architecture: LSTM, GRU, and their challenges (e.g., vanishing gradients). Applications: NLP tasks (e.g., sentiment analysis, text generation). **Unit 4** **Transformers:** Transformers: Encoder-decoder architecture, self-attention, and multi-head attention. Applications: NLP tasks (e.g., translation, summarization, sentiment analysis) and Vision Transformers (ViTs). Pre-trained models: BERT, GPT, and their use cases. **Unit 5** **Advanced Topics:** Architecture and applications of GANs, Time Series Forecasting - ARIMA and SARIMA, LSTM, GRU, LLMs, Federated Learning, and Real-world implementations in speech and gesture recognition.

List of Experiments	1. MNIST Digit Classification: Train a simple neural network. 2. Image Classification with CNN: Use CNNs for CIFAR-10 classification. 3. Transfer Learning with Pre-trained CNN: Fine-tune models for custom datasets. 4. Object Detection with YOLO: Implement YOLO for real-time object detection. 5. Incremental Learning with CNNs: Update models with new data without retraining. 6. Instance Segmentation with Mask RCNN: Apply Mask RCNN for image segmentation. 7. Time Series Forecasting with LSTM: Predict trends using LSTM models. 8. Sentiment Analysis with LSTM: Analyze sentiment in text data. 9. Text Summarization with BERT: Perform extractive and abstractive summarization. 10. Generative Adversarial Networks (GANs): Generate images using GANs. 11. Style Transfer with GANs: Apply artistic style transfer. 12. Time Series Forecasting with SARIMA: Compare SARIMA and LSTM for forecasting. 13. Text Classification with BERT: Fine-tune BERT for text classification. 14. Speech Recognition: Implement a speech recognition system. 15. Hand Gesture Recognition: Use CNNs for gesture recognition.
Course Assessment	Continuous Evaluation 25%
	Mid Semester 25%
	End Semester 50%

Course Code: CSBB 314	**Open course (YES/NO)**	**HM Course (Y/N)**	**DC (Y/N)**	**DE (Y/N)**
	No	No	Yes	No
Type of course	Core			
Course Title	**Computer Vision**			
Course Coordinator				
Course Objectives:	Provide students with advanced knowledge of Computer Vision, Image Processing, Machine Learning and Deep Learning, enabling them to design, implement, and apply various models.			
Course Outcomes	**CO1:** Understand foundational concepts of deep learning, including neural networks, training methods, and optimization techniques.			**L1, L2, L3**
	CO2: Develop and evaluate convolutional neural networks (CNNs) for image classification and object detection.			**L4, L5, L6**
	CO3: Design recurrent neural networks (RNNs) and Transformer models for NLP and time-series analysis.			**L4, L5, L6**
	CO4: Implement advanced deep learning techniques for generative modeling, speech recognition, and gesture recognition.			**L4, L5, L6**

Semester	**Autumn: No**		**Spring: Yes**		
V	**Lecture**	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	3	0	2	4	60
Prerequisite course code as per proposed course numbers	Machine Learning Course				
Prerequisite credits					
Equivalent course codes as per proposed course and old course					
Overlap course codes as per proposed course numbers					

Text Books:

1.	Title	Computer Vision: Algorithms and Applications
	Author	Richard Szeliski
	Publisher	Springer

	Edition	2016
Reference Book:		
1.	Title	Computer Vision: a Modern Approach
	Author	David Forsyth and Jean Ponce
	Publisher	Pearson
	Edition	2015
2.	Title	Multiple View Geometry in Computer Vision
	Author	Richard Hartley and Andrew Zisserman
	Publisher	Cambridge University Press
	Edition	2014

Content	**Unit 1** **Introduction to computer vision** Overview of Computer Vision, Applications of Computer Vision in Robotics, Healthcare, and Autonomous Vehicles, Basic Image Formation and Representation, Image Filtering: Smoothing, Sharpening, and Histogram Equalization. **Unit II** **Image Processing and Feature Detection Fundamentals** Grayscale and Color Image Processing, Image Transformations: Fourier Transform, Discrete Cosine Transform (DCT), Edge Detection: Sobel, Canny, and Laplacian, Corner and Blob Detection (Harris Corner Detector, FAST, SIFT, and SURF), Image Matching and Transformation: Homography and RANSAC, Keypoint Descriptors **Unit III** **Machine Learning for Computer Vision Tasks** Classification: SVMs, KNN, Decision Trees, Introduction to Neural Networks and Convolutional Neural Networks (CNNs), Deep Learning Architectures for Vision (AlexNet, ResNet, VGG), Anomaly Detection in Images (e.g., Autoencoders), Applications: Face Recognition, Gesture Recognition, and Scene Understanding **Unit IV** **Convention computer vision and algorithms** Object detection and segmentation e.g. Edge, texture, region, detection of sliding windows: Feature extraction, e.g. linear binary pattern, principal component analysis, Gabor filters, bags of features, Matching and recognition e.g. Bayesian classifier, support vector machine, fusion, Image alignment and stitching. Principal Component Analysis (PCA) **Unit V** **Deep learning and Neural Networks for computer vision** Key components and basic architecture of deep neural network, Convolution neural network, Object detection using R-CNN, Segmentation using image-to image neural network, Temporal processing and recurrent neural network.

List of Experiments	1. To implement data visualization techniques such as box plot, scatter plot, and Q plot for comprehensive analysis of a given dataset. 2. To apply Gaussian, Median, and Bilateral filters to reduce noise in images while preserving important structural details. 3. To enhance and detect edges in images by applying Sobel, Prewitt, and Laplacian filters, emphasizing significant transitions in pixel intensity. 4. To classify images using a Support Vector Machine (SVM) model and assess performance through cross-validation and key metrics such as accuracy and confusion matrix. 5. To apply Convolutional Neural Networks (CNNs) for image classification and evaluate the model's performance using standard metrics such as accuracy and confusion matrix. 6. To implement K-Nearest Neighbors (KNN) for image classification and evaluate its effectiveness through cross-validation and performance metrics such as accuracy. 7. To apply the Fast Fourier Transform (FFT) to convert images into the frequency domain and filter out high-frequency noise, enhancing image quality. 8. To apply Principal Component Analysis (PCA) for reducing the dimensionality of image datasets while retaining the most significant features. 9. To extract texture features from images using Gabor filters, enhancing the understanding of surface patterns and texture-related information. 10. To perform object detection by leveraging Region-based Convolutional Neural Networks (R-CNN) for detecting and classifying objects in images. 11. To implement Long Short-Term Memory (LSTM) networks for sequential image classification and evaluate the model’s performance using cross-validation and accuracy metrics. 12. To classify images using a Decision Tree model and evaluate its performance using cross-validation and metrics such as accuracy, precision, and recall.
Course Assessment	Continuous Evaluation 25%
	Mid Semester 25%
	End Semester 50%

Course no: CSBB 316	Open course (YES/NO)	HM Course (Y/N)	DC (Y/N)	DE (Y/N)
	NO	NO	NO	Yes
Type of course	Elective			
Course Title	Information Storage and Retrieval			
Course Coordinator				
Course objectives:	This course will introduce the student to the fundamentals of information storage and retrieval (ISR) systems, including components, models, structures, information representation, vocabulary control, search strategies (query language/operation, indexing, and search), User interface and visualization and User dimension and evaluation. It will also cover some practical Web information retrieval (IR) systems.			

Course Outcomes			
	CO1	To identify the various components of an information storage and retrieval system. To understand the different models and structures an IR system takes.	L1, L2
	CO2	To understand the theoretical foundations of various IR methods.	L2
	CO3	To be able to design and implement IR systems using off-the-shelf software packages.	L3, L4
	CO4	To evaluate the factors that influence the performance of an IR system and to create solutions to the field's current problems.	L5, L6

Semester	**Autumn:**		**Spring:**		
	Lecture	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	3	0	2	4	36
Prerequisite course code asper proposed course numbers	NIL				
Prerequisite credits	NIL				
Equivalent course codes as per proposed course and oldcourse	NIL				
Overlap course codes as perproposed course numbers	NIL				
Text Books:					

1	Title	Information Storage and Retrieval
	Author	Manning, Raghavan, Schütze
	Publisher	Cambridge University Press
	Edition	2008
2	Title	Information Representation and retrieval in the digital age
	Author	Medford
	Publisher	Information Today Press
	Edition	2003
3	Title	Information Storage and Retrieval
	Author	R. R. Korfhage
	Publisher	R. R. Korfhage
	Edition	1997
Reference Books:		
1	Title	Text Information Retrieval Systems
	Author	Charles T. Meadow
	Publisher	Academic Press
	Edition	2007
2	Title	Visualization for Information Retrieval
	Author	Zhang
	Publisher	Springer
	Edition	2008
Content	**Unit – 1** **Introduction:** What is information retrieval, Significance of information retrieval and storage, Definition of information retrieval system, Objectives of information retrieval system, Function overview, Relationships between Digital library and IRS, Abstraction, Algorithm, Data structure, Measure of information systems, Logical organization, Physical organization, Components of information retrieval systems, Comparisons among different information systems **Unit–2** **Models of Information Retrieval:** Search and Browsing. Queries and matching. Classic Boolean model, Document space, measure, similarity. Classic vector model. Queries and matching for advanced models. Problems with classic Boolean model. Extended Boolean model. Fuzzy model, Other models and aspects of matching.	

	Unit – 4 **Text Analysis:** From text to index. Types of indexing. Zipf's law, TF*IDF. Stop lists and stemming, Document similarity, Multi-language retrieval. **Alternative retrieval techniques:** Citation processing, hypertext browsing, Information Visualization and its use for information access. Adaptive information visualization and adaptive navigation support. **Unit – 5** **Output presentation and visualization:** What to present, ranking, clustering, output exploration, visual interfaces for output exploration: GUIDO, VIBE, BIRD, TileBars, LyberWorld **Web IR and other modern problems of information retrieval:** Characterising the Web. Search engines. Meta-search. Search services. Agents and bots. Clustering and information exploration. Use of hyperlinks. Web recommenders and other adaptive Web-based information systems.
Course Assessment	Continuous Evaluation 25% Mid Semester 25% End Semester 50%

Course no: CSBB 408	**Open course (YES/NO)**	**HM Course (Y/N)**	**DC (Y/N)**	**DE (Y/N)**
	NO	NO	NO	Yes
Type of course	**Elective**			
Course Title	**Reinforcement Learning and Applications**			
Course Coordinator				
Course objectives:	The purpose of this course is to provide the students with the knowledge of current, advanced techniques and applications in RL. Learn how to define RL tasks and the core principals behind the RL, including policies, value functions. Given an application problem, decide if it should be formulated as a RL problem; if yes be able to define it formally, state what algorithm is best suited for addressing it.			
POs				

Semester	**Autumn: Yes**		**Spring: Yes**		
	Lecture	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	3	0	2	4	36
Prerequisite course code as per proposed course numbers	Machine Learning and Deep Learning				
Prerequisite credits	NIL				

Equivalent course codes as per proposed course and old course	NIL				
Overlap course codes as per proposed course numbers	NIL				

Text Books:		
1	Title	Reinforcement Learning: An Introduction
	Author	Richard S. Sutton and Andrew G. Barto
	Publisher	MIT Press
	Edition	2nd Edition.
Reference Book:		
1	Title	Reinforcement Learning: State-of-the-Art,
	Author	Marco Wiering and Martijn van Otterlo, Eds
	Publisher	Springer Science & Business Media
	Edition	2012

Content	**Unit** – 1 **Introduction**: Origin and history of Reinforcement Learning research. Its connections with other related fields and with different branches of ML. **Probability Basics**: - Axioms of probability, concepts of random variables, PMF, PDFs, CDFs, Expectation. joint and multiple random variables, joint, conditional and marginal distributions. Correlation and independence. **Unit** – 2 **Markov Decision Process**: Introduction to RL terminology, Markov property, Markov chains, Markov reward process (MRP). Bellman equations for MRPs. Markov decision process (MDP), state and action value functions, Bellman expectation equations, optimality of value functions and policies, Bellman

	optimality equations. **Unit** – 3 **Prediction and Control by Dynamic Programing**: Dynamic programing definition and formulation of planning in MDPs, principle of optimality, iterative policy evaluation, policy iteration, value iteration, convergence of policy evalutation and value iteration algorithms, DP extensions. Monte Carlo methods, model free RL, Monte Carlo control, On policy and off policy learning, Importance sampling. **Unit** – 4 **TD Methods**: Incremental Monte Carlo Methods for Model Free Prediction, Overview TD(0), TD(1) and TD(λ), k-step estimators, unified view of DP, MC and TD evaluation methods, TD Control methods - SARSA, Q-Learning and their variants. **Unit- 5** **Function Approximation and Policy Gradient Methods**: Getting started with the function approximation methods, Revisiting risk minimization, gradient descent from Machine Learning, Gradient MC and Semi-gradient TD(0) algorithms, Eligibility trace for function approximation, Afterstates, Control with function approximation, Least squares, Experience replay in deep Q-Networks. Deep-Reinforcement Learning Need and Applications, Types of Deep-RL : Deep Q-Network (DQN) , Policy Gradient [Advantage Actor-Critic (A2C/A3C), DDPG, PPO] , Alpha zero . Log-derivative trick, Naive REINFORCE algorithm, bias and variance in Reinforcement Learning, Reducing variance in policy gradient estimates, baselines, advantage function, actor-critic methods.
Course Assessment	Continuous Evaluation 25%
	Mid Semester 25%
	End Semester 50%

Course no: CSBB 415	**Open course (YES/NO)**	**HM Course (Y/N)**	**DC (Y/N)**	**DE (Y/N)**
Type of course	**Elective**			
Course Title	**Motion Planning for Robotics**			
Course objectives:	The course enables the students learn the basics of mobile robotics and helps them understand the mechanisms to plan and navigate different types of robots from a pre-specified source to a pre-specified goal using a variety of Artificial Intelligence techniques.			
POs	From this course the students will be able to appreciate intelligent robotic systems involving single and multiple robots, get a good grasp in Artificial Intelligence tools and techniques from an application point of view, and be able to plan for different robots in different conditions and constraints.			

Semester	**Autumn: Yes**		**Spring:**		
I	**Lecture**	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours					
Prerequisite course code as per proposed course numbers	NIL				
Prerequisite credits	NIL				
Equivalent course codes as per proposed course and old course	NIL				
Overlap course codes as per proposed course numbers	NIL				
Text Books:					
1	Title	Principles of Robot Motion: Theory, Algorithms, and Implementations			

	Author	H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, S. Thrun
	Publisher	MIT Press, Cambridge, MA
	Edition	2005
Reference Book:		
1	Title	Planning Algorithms
	Author	S. M. LaValle
	Publisher	Cambridge University Press, NY.
	Edition	2006
2	Title	On-Road Intelligent Vehicles: Motion Planning for Intelligent Transportation Systems,
	Author	R. Kala
	Publisher	Elsevier, Waltham, MA
	Edition	2016

Content	**Unit – 1: Introduction** Introduction to Robots and Robotics: Definitions of Robots, Robotics, Motivation, A Brief History of Robotics, A Robot System, Interdisciplinary Areas in Robots, DOF of Joints and Robotic System, Classification of Robots, Workspace of Manipulator Workspace of Manipulator, Resolution, Accuracy and Repeat Ability; Applications of Robots **Unit – 2 : Kinematics of Robotic Motion** Introduction to Robot Motion Planning, Basics of Serial robotic arms and Mobile robots, Transformations, Robot Kinematics: Properties of Rotation Matrix, Cylindrical and Spherical Coordinate, D-H Parameters Setting, Forward Kinematics, Inverse Kinematics, Trajectory Planning, Singularity Checking, Jacobian Matrix **Unit – 3 : Dynamics of Robotic Motion** Robot Dynamics: Earliest Bug Algorithms, Configuration Space, Manifold, Topology of Configuration Space, Dynamics (Inertia Tensor), Inverse Dynamics, Forward Dynamics, Inertia Tensor. **Unit – 4 : Motion Control** Control Scheme: Control Scheme, Sensors, Robot Vision, Robot Motion Planning, Intelligent Robot: Biped Walking. **Unit – 5 : Applications of Motions in Robots** Motion planning for Multi robotic systems, motion planning in 3D. Applications of robotics in active perception, medical robotics, autonomous vehicles, and other areas.
Course Outcomes:	1. Explain the foundational concepts of robotics such as robot definitions,classifications, degrees of freedom, workspace, and their applications in various fields (K2). 2. Illustrate the principles of robot kinematics including forward and inverse kinematics, trajectory planning, and transformations using D-H parameters (K2). 3. Apply the concepts of robot dynamics to analyze configuration space, inertia tensors, and motion behavior in robotic systems (K3). 4. Utilize control schemes and sensor integration to enable robot vision, motion planning, and intelligent functionalities like biped walking (K3). 5. Develop solutions for advanced robotic applications by implementing motion planning for multi-robot systems in 3D environments and exploring areas such as

	medical robotics and autonomous vehicles (K6).
Course Assessment	Continuous Evaluation 25%
	Mid Semester 25%
	End Semester 50%

Lab Experiments:

Exp. No.	**Experiments**
1.	Study the components of a robotic system, identify degrees of freedom (DOF), and classify robots based on structure and applications.
2.	Measure and analyze the workspace, resolution, accuracy, and repeatability of a robotic manipulator.
3.	Implement forward kinematics for a 2-DOF or 3-DOF robotic arm using D-H parameters.
4.	Solve inverse kinematics for a simple robotic arm and validate the results using simulation.
5.	Perform trajectory planning for a robotic arm to move between two predefined points in its workspace.
6.	Simulate the configuration space of a robotic arm in a 2D environment.
7.	Implement a basic PID control scheme for controlling the position of a robotic joint.
8.	Interface a proximity or IR sensor with a robotic system to detect obstacles and trigger actions.
9.	Develop a simple vision-based system to detect and track an object using a camera.
10.	Simulate multi-robot motion planning in a 2D environment to achieve collision-free navigation.

Course no: CSBB 412	Open course (YES/NO)	HM Course (Y/N)	DC (Y/N)	DE (Y/N)
Type of course	**Elective**			
Course Title	**Motion Analytics and Planning for Robotics**			
Course objectives:	The course enables the students learn the basics of mobile robotics and helps them understand the mechanisms to plan and navigate different types of robots from a pre-specified source to a pre-specified goal using a variety of Artificial Intelligence techniques.			
POs	From this course the students will be able to appreciate intelligent robotic systems involving single and multiple robots, get a good grasp in Artificial Intelligence tools and techniques from an application point of view, and be able to plan for different robots in different conditions and constraints.			

Semester	**Autumn: Yes**		**Spring:**		
I	**Lecture**	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours					
Prerequisite course code as per proposed course numbers	NIL				
Prerequisite credits	NIL				
Equivalent course codes as per proposed course and old course	NIL				
Overlap course codes as per proposed course numbers	NIL				

Text Books:		
1	Title	Principles of Robot Motion: Theory, Algorithms, and Implementations
	Author	H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W.

		Burgard, L. E. Kavraki, S. Thrun
	Publisher	MIT Press, Cambridge, MA
	Edition	2005
Reference Book:		
1	Title	Planning Algorithms,
	Author	S. M. LaValle
	Publisher	Cambridge University Press, NY.
	Edition	2006
2	Title	Robot Analysis- The mechanics of Serial and Parallel Manipulators
	Author	L. Tsai
	Publisher	John Wiley & Sons
	Edition	1999

Content	**Unit – 1 Motion analytics and Mathematics** Introduction to Motion analytics, Trigonometry and Vector, Mechanics, and Signal Processing **Unit 2: Introduction to Bio-Motion and Human Gait** **Anatomy of Human body, motion physiology, bio-mechanics and human gait,** Anthropometry in Bio-Motion, Walking and Gait Terminologies, Movement Analysis Methods Vision Based, Marker Based Motion Capture, Marker Less Motion Capture, Sensor Based, and Other Techniques **Unit 3: Kinematic and inverse kinematics** Robot Kinematics: Kinematic conventions, Mapping, free body diagram, Homogeneous transformations, Rotation matrix, Forward Kinematics Denavit - Hartenberg (DH) representation, inverse kinematics: solution of inverse problems, imaging measurement techniques, biomechanical models, force transducers and force plates, bone on bone force. **Unit 4: Model of Human Pose and Motion** Object detection, computer vision-classification, semantic segmentation, object detection, instance segmentation, traditional object detectors- SIFT, HOG, BOW, deep learning based object detectors: Landmark detection, sliding windows detection, bounding box prediction (YOLO algorithm), anchor boxes, evaluating object localization, human body representation, traditional method- latent variables, models-PCA, FA etc., discriminative model-regression, generative model- kalmann filter, partial filter. **Unit 5: Machine Learning for Motion Modelling** ML Approaches: Type of Machine Learning, Understanding Data and Feature Engineering in ML, Reasoning Under Uncertainty, Fuzzy Logic, Probabilistic Reasoning, Neural Network: Motion Graph, Inverse Kinematics, Latent Variable Model, Supervised Techniques, Unsupervised Techniques, Reinforcement Techniques.
Course Outcomes:	1. Explain the foundational concepts of robotics such as robot definitions, classifications, degrees of freedom, workspace, and their applications in various fields (K2). 2. Illustrate the principles of robot kinematics including forward and inverse kinematics, trajectory planning, and transformations using D-H parameters (K2). 3. Apply the concepts of robot dynamics to analyze configuration space, inertia tensors, and motion behavior in robotic systems (K3). 4. Utilize control schemes and sensor integration to enable robot vision, and image processing algorithms. 5. Develop solutions for advanced robotic applications by implementing motion planning for multi-robot systems in 3D environments and exploring machine learning and deep learning models.
Course	Continuous Evaluation 25%

Assessment	Mid Semester 25%
	End Semester 50%

Lab Experiments:

Exp. No.	Experiments
1.	Study the components of a robotic system, identify degrees of freedom (DOF), and classify robots based on structure and applications.
2.	Measure and analyze the workspace, resolution, accuracy, and repeatability of a robotic manipulator.
3.	Implement forward kinematics for a 2-DOF or 3-DOF robotic arm using D-H parameters.
4.	Solve inverse kinematics for a simple robotic arm and validate the results using simulation.
5.	Perform trajectory planning for a robotic arm to move between two predefined points in its workspace.
6.	Simulate the configuration space of a robotic arm in a 2D environment.
7.	Implement a basic PID control scheme for controlling the position of a robotic joint.
8.	Interface a proximity or IR sensor with a robotic system to detect obstacles and trigger actions.
9.	Develop a simple vision-based system to detect and track an object using a camera.
10.	Pose Estimation models- decision tree for body segment classification.

<table>
<tr><td rowspan="2">Course Code:

CSLB 315</td><td>Open course (YES/NO)</td><td>HM Course (YES/NO)</td><td>DC (YES/NO)</td><td>DE (YES/NO)</td></tr>
<tr><td>NO</td><td>NO</td><td>NO</td><td>YES</td></tr>
<tr><td>Type of course</td><td colspan="4">Elective</td></tr>
<tr><td>Course Title</td><td colspan="4">OPTIMIZATION TECHNIQUES</td></tr>
<tr><td>Course Objectives:</td><td colspan="4">This course aims to cover the concepts of optimization methods and algorithms developed for solving various types of optimization Problems.
To apply the mathematical results and numerical techniques of Optimization theory to various Engineering and Analytics problems.
Explain the theoretical workings of the graphical, simplex, and analytical methods for making effective decision on variables so as to optimize the objective function.</td></tr>
<tr><td rowspan="4">Course

Outcomes</td><td colspan="3">CO1: Understand the fundamentals of Linear Programming and Dynamic Programming</td><td colspan="2">L1, L3</td></tr>
<tr><td colspan="3">CO2: Enumerate the fundamentals of Integer programming technique and apply different techniques to solve various optimization problems arising from engineering areas.</td><td colspan="2">L1, L2</td></tr>
<tr><td colspan="3">CO3: Identify appropriate optimization methods to solve complex problems involved in various industries.</td><td colspan="2">L1, L2, L4</td></tr>
<tr><td colspan="3">CO4: To understand the graphical, simplex, and analytical methods for making effective decisions.</td><td colspan="2">L2, L5</td></tr>
<tr><td>Semester</td><td colspan="2">Autumn:</td><td colspan="3">Spring: YES</td></tr>
<tr><td>VI</td><td>Lecture</td><td>Tutorial</td><td>Practical</td><td>Credits</td><td>Total teaching hours</td></tr>
</table>

Contact Hours	3	1	0	4	36(L)+12(T)
Prerequisite course code as per proposed course numbers	NIL				
Prerequisite credits	NIL				
Equivalent course codes as per proposed course and old course	NIL				
Overlap course codes as per proposed course numbers	NIL				

Text Books:		
1	Title	An Introduction to Optimization
	Author	Edwin K.P. Chong, Stanislaw H. Zak
	Publisher	Wiley
	Edition	2017
Reference Book:		
1	Title	Convex Optimization
	Author	Stephen Boyd
	Publisher	LievenVandenberghe
	Edition	2004
2	Title	Modern Optimization with R (Use R)

	Author	Paulo Cortez
	Publisher	Springer
	Edition	2014
Content	**UNIT 1 Introduction to Optimization (8 hours)** Proofs, Vector Spaces and Matrices, Linear Transformations, Eigenvalues and Eigenvectors, Orthogonal Projections, Quadratic Forms, Matrix Norms, Concepts from Geometry, Elements of Calculus. **UNIT 2 Unconstrained Optimization (8 hours)** of Set Constrained and Unconstrained Optimization, One Dimensional Search Methods, Golden Section Search, Fibonacci Search, Newton's Method, Secant Method, Solving Ax = b. **UNIT 3 Linear Programming (6 hours)** Introduction to Linear Programming, Simplex Method, Duality. **UNIT 4 Nonlinear Constrained Optimization (8 hours)** Problems with Equality Constraints, Problems with Inequality Constraints, Karush Kuhn Tucker Condition, Convex Optimization Problems. **UNIT 5 Additional Tools (6 hours)** Algorithms for Constrained Optimization: Projections, Project gradient methods, Penalty methods.	
Course Assessment	Continuous Evaluation 25%	
	Mid Semester 25%	

	End Semester 50%

Course no: CSBB 317	PC (YES/NO)	PE (YES/NO)	IS-TP(YES/NO)	SEM (YES/NO)	TH-DIS (YES/NO)
	NO	YES	NO	NO	NO
Type of course	**Program Elective**				
Course Title	**Soft Computing**				
Course Coordinator					
Course objectives:	This course aims to introduce the fundamental principles, techniques, and applications of soft computing, along with the mathematical foundations of key topics such as Artificial Neural Networks, Fuzzy Logic, and Genetic Algorithms.				
Course Outcomes:	• Understand the fundamental principles of soft computing, including fuzzy logic, artificial neural networks (L2, L3). • Explain applications & operations of Fuzzy Logic in real life Problems (L4, L5). • Analyze and evaluate the performance of different soft computing models for specific applications. (L2, L3, L4). • Design and implement soft computing solutions using appropriate tools and techniques for practical and innovative problem-solving (L5, L6).				
POs					

	Autumn:		**Spring:**		
	Lecture	**Tutorial**	**Practical**	**Credits**	**Total teaching hours**
Contact Hours	3	0	2	4	36
Prerequisite course code as per proposed course numbers	NIL				
Prerequisite credits	NIL				

Equivalent course codes as per proposed course and old course		NIL				
Overlap course codes as per proposed course numbers		NIL				
Text Books:						
1.	Title	A comprehensive foundation. Neural Networks				
	Author	Simon Haykin				
	Publisher	Pearson				
	Edition	Second Edition, 2001				
Reference Book:						
1.	Title	Fuzzy logic with engineering applications				
	Author	Timothy J. Ross				
	Publisher	John Wiley & Sons				
	Edition	Third Edition, 2009				
2.	Title	An Introduction to Genetic Algorithms				
	Author	Melanie Mitchell				
	Publisher	Prentice-Hall				
	Edition	1998				
3.	Title	Genetic Algorithms in Search, Optimization, and Machine Learning				
	Author	D. E. Goldberg				
	Publisher	Addison-Wesley				

	Edition	1989
4.	Title	Understanding Neural Networks and Fuzzy Logic: Basic Concepts and Applications
	Author	S. V. Kartalopoulos
	Publisher	IEEE Press
	Edition	PHI, 2014
5.	Title	Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis & Applications
	Author	S. Rajasekaran & G. A. Vijayalakshmi Pai
	Publisher	PHI
	Edition	2003
6.	Title	Principles of Soft Computing
	Author	S. N. Sivanandam & S. N. Deepa
	Publisher	Wiley - India
	Edition	Second Edition, 2007
Content	**Unit I: Introduction** (6 Hours) Soft computing vs Hard computing, Applications of Soft computing, Basic mathematics of soft computing, Learning and statistical approach to regression and classification. **Unit II: Neural Networks** (8 Hours) Single layer perceptron, Learning rules, ADALINE Network, LMS algorithm, Multi layer perceptron, Radial basis function, Associative Memory Networks, Kohnen's self-organizing networks, Hopfield Network, Principal component analysis, Introduction to SVM, Binary classification, Regression by SVM: linear & nonlinear, Decomposing multiclass classification into binary classification. **Unit III: Fuzzy Logic** (8 Hours)	

	Introduction to Fuzzy logic, Probability vs Possibility Theory, Classical set and fuzzy set, fuzzy set operations, Criteria for Selecting appropriate aggregation Operators. Fuzzy relation, Fuzzy composition, Fuzzy Inference system, Fuzzification, Fuzzy rule based system, Defuzzification, Fuzzy Arithmetic, Fuzzy logic application, Predicate logic, Fuzzy Rule based Approximate Reasoning, Models of Neuro-fuzzy system (NFS), Interpretation of NFS layers, Adaptive N-F Inference system (ANFIS) Architecture **Unit IV: Metaheuristic Approaches** (8 Hours) Concept of "Genetics" and "Evolution" and its application to probabilistic search techniques, Basic GA framework and different GA architectures, GA operators: Encoding, Crossover, Selection, Mutation, Ant Colony Optimization, Particle Swarm Optimization **Unit V: Multi-objective Optimization Problem Solving** (6 Hours) Concept of multi-objective optimization problems (MOOPs) and issues of solving them, Multi-Objective Evolutionary Algorithm (MOEA), Non-Pareto approaches to solve MOOPs, Pareto-based approaches to solve MOOPs, Some applications with MOEAs. **List of Experiments:** 1. Form a perceptron net for basic logic gates with binary input and output. 2. Using Adaline network, generate XOR function with bipolar inputs and targets. 3. Design a feed forward neural network with back propagation from scratch to classify numbers using MNIST dataset. 4. Using Kohonen's Self-Organizing Map (SOM) cluster the MNIST dataset and visualizing the resulted clusters on a 2D map. 5. Design a SVM based classifier to classify two selected classes from Iris dataset. Also evaluate the decision boundary. 6. Use PCA as a dimensionality reduction technique on the Iris dataset to reduce dimensions from 4 to 2 and visualize class separation in 2D

	space. 7. Design fuzzy inference systems for an Automatic Washing Machine with 5 modes of operation. 8. Design and simulate a fuzzy inference system for temperature control of an Air Conditioner. 9. Using an Adaptive Neuro-Fuzzy Inference System (ANFIS) model, predict daily energy consumption based on historical time-series data of temperature and humidity over the Energy Consumption Dataset. 10. Solve the traveling salesman problem using a genetic algorithm to find the shortest possible route that visits all cities exactly once and returns to the starting city.
Course Assessment	**THEORY Evaluation**: • Continuous Evaluation: 25% • Mid Semester: 25% • End Semester: 50%
	LAB Evaluation: • Continuous Evaluation: 50% • End Term Evaluation: 50%
	Final Evaluation: 60% of Theory + 40% of Lab

CSBB 409	Open course (YES/NO)	HM Course (Y/N)	DC (Y/N)	DE (Y/N)
Type of course	YES	NO	NO	NO
Course Title	**Social Network Analysis**			
Course Coordinator				
Course objectives	Students who successfully complete this course will gain: a broad conceptual introduction to the modern theory and applications of complex networks, experience critiquing scientific papers, experience working with large, complex data sets, experience with technical writing and in class presentations.			
POs	After the course , students will be interested in contemporary complex network research. This foundation will be both analytical and computational. After taking the course the students will be able to investigate specific research tasks and real world applications that require a complex network approach or knowledge of the fundamentals of complex network science. Specifically the students will •learn techniques for analyzing real world networks empirically, •learn how to construct networks from real world data, •learn fundamentals of graph theory and network mathematics as well as the statistical physics approach to large scale networks, •learn the fundamentals for generating random network models on a computer, and •learn to investigate dynamical processes that evolve on networks.			

Semester	**Autumn:**		**Spring**		
VII/VIII	**Lecture**	**Tutorial**	**Practical**	**Credits**	**Total Teaching Hours**
Contact Hours	3	0	2	4	36
Prerequisite course code as per proposed course numbers					
Prerequisite credits					
Equivalent course codes as per proposed course and old course					
Overlap course codes as per proposed course numbers					

Text Books:		
1	Title	Networks: An Introduction
	Author	Mark Newman
	Publisher	Oxford University Press
	Edition	20-May-10
2	Title	Network Scince
	Author	Albert Laszlo Barabasi
	Publisher	Cambridge University Press
	Edition	2016
3..	Title	Networks, Crowds, and Markets: Reasoning About a Highly Connected World
	Author	David Easley and Jon Kleinberg
	Publisher	Cambridge University Press
	Edition	Sep-10
Reference Book:		
1.	Title	The Structure of Complex Networks Theory and Applications
	Author	Ernesto Estrada
	Publisher	Oxford University Press
	Edition	17-Dec-11
2.	Title	Dynamical Processes on Complex Networks
	Author	Alain Barrat, Marc Barthelemy, and Alessandro Vespignani
	Publisher	Cambridge University Press
	Edition	24-Nov-08

3.	Title	The Structure and Dynamics of Networks
	Author	Mark Newman, Albert-Laszlo Barabasi, & Duncan J. Watts
	Publisher	Princeton University Press
	Edition	April 17, 2006
4.	Title	Exploratory Social Network Analysis with Pajek
	Author	Nooy, Wouter de, Andrej Mrvar, and Vladimir Batagelj
	Publisher	Cambridge University Press
	Edition	10-Jan-05

Content	Unit – 1 **Basic Network Properties:** Empirical Study of Networks: Technological networks, Information networks, Social networks, Biological networks, Economic networks
	Fundamentals of Network Theory: Mathematics of networks, Measures and metrics, Structural properties of networks: Diameter, Clustering Coefficient, Degree distribution.
	Unit – 2 **Network Models**: Random graphs, Random graphs with general degree distributions, Small world, Power-law, Decentralized centrality in Small world. **Temporal Networks, Multilayer Networks**
	Node Centralities: Degree centrality, Closeness centrality, Betweenness centrality, Centralization , eigenvector centrality.
	Unit – 3 **Network communities and Link analysis:** Graph Partitioning, Edge betweenness, Modularity optimization and Spectral Clustering, Community Detection Algorithms, Overlapping communities in networks. Link Analysis: HITS and Page Rank.
	Unit – 4 **Diffusion and Epidemics on Networks:** Physical diffusion, Diffusion equation, Random walks on graph. Epidemics models SI, SIS, SIR. Modeling of infection propagation.
	Unit – 5 **Network Visualization:** Network X, UCINET, Gephi, Pajek, Graphviz.
Course Assessment	Continuous Evaluation 25% Mid Semester 25% End Semester 50%