DIT UNIVERSITY
Dehradun

Detailed Course Structure & Syllabus
of
B.Tech. – Computer Science & Engineering
with Specialization in
Chip Design

Approved by the Academic Council in its 22nd Meeting held on 06.03.2023

Introduction

The Ministry of Human Resource Development (MHRD), Govt. of India, has initiated development of a New Education Policy (NEP) to bring out comprehensive reforms in the Indian education system.

The University Grants Commission (UGC) has subsequently initiated several steps to foster academic excellence through introduction of paradigm shift in learning and teaching pedagogy, innovation and improvement in course curricula, examination and education system.

While a majority of education institutions have started following the semester-based system of education, it has been observed that this new system is still producing graduates who lack knowledge, values, skills and are not job ready professional. The reason for this lacking could be attributed to the rigidity of our program structures and lack of flexibility to have choices among core subject education, liberal arts, ability enhancement, skill development, etc., that is fundamental to overall development and employability of these graduates.

To make this possible, a fully flexible choice-based credit system (FFCBCS), a well-established internationally known system, is proposed. This fully flexible choice-based credit system allows students the flexibility to learn at their own pace, and register for both core subjects and a variety of courses from other areas, leading to holistic development of an individual. The FFCBCS will facilitate us to bench mark our programs with best international liberal arts based academic programs.

***Advantages of the FFCBCS structure**:*

- Shift in focus from the teacher-centric to student-centric education. Student can curve out their program structure by choosing minimum number of credits from well-defined baskets.
- Student may undertake as many credits as they can cope with.
- FFCBCS allows students to choose courses from various baskets of inter-disciplinary, intra-disciplinary, skill oriented, ability enhancing, and from other disciplines.

Features unique to DIT University FFCBCS structure:

1. A minimum of 150-160 credits has to be earned by a student to be eligible for an Under Graduate degree in Engineering. Each department will decide their total credits for each program, and it can vary across disciplines.

2. Courses are categorized into 11 baskets, and a student will have the option to choose coursesin most baskets and earn ***minimum number of credits*** required in each basket for the award of his/her degree. For each basket, Engineering departments have the flexibility to identify course(s) which will be a core requirement for their program.

3. In certain disciplines, students may choose a ***Specialization*** by earning 18 credits of Discipline Elective courses towards a particular area of that discipline (intra-disciplinary). In addition to this, brighter students will have the option to receive (a) a ***Certificate*** by earning *additional* 9 credits towards a particular area either inside or outside their discipline, or (b) ***Minor*** by earning additional 18 credits towards a particular area outside their discipline. Certificates and Minors can be earned througheither University courses, or with MOOCs from providers as identified by the University. Each department will design the structures and eligibility conditions for registration to its certificates or minor program, which may be reviewed annually, to keep the ***Certificates*** and ***Minors*** contemporary and relevant to latest changes.

4. An Academic Advisory Committee may be formed comprising all HoDs/ Programme Coordinator and one representative each from respective departments. Academic Advisory Committee will meet at the end of every semester after the completion of Board of Examination meeting to discuss and finalize course offerings by respective departments in the upcoming semester. Academic

Advisory Committee will be chaired by the Dean Academic Affairs/ Deans of respective Schools/ Competent Authority.

5. To provide sufficient flexibility and room during the program for additional ***Certificates***, ***Specializations, and Minors***, 8-week summer semesters (Summer 1, Summer 2, and Summer 3) may have to run. Summer semesters are critical for implementing a fully flexible system. Each department will decide *a priori* which courses to offer in the summer semester and get them finalized at the Academic Advisory Committee meeting.

6. Project based learning has to be incorporated as a core component of evaluation in each course, and depending on the level and type of the course, the project can be of several types - Study Oriented Project, Lab Oriented Project, Design Oriented Project, Computer Oriented Project, Projects of Organizational Aspects, Research Projects, or Entrepreneurship and Start Up Projects. A Capstone Project has been introduced in the 8th semester for all Bachelor of Technology students.

7. Courses under each basket may be updated on an annual basis.

8. Each student will be advised by a faculty advisor of his/her department for registration of courses from each basket in the beginning of semester, depending upon the availability of seats. A student advising centre may be formed where students will have access to department faculty advisers. Faculty advisers should have complete access to view individual student‘s academic transcript for advising purposes.

9. A student getting an F grade in a core course (departmental or otherwise) at the end of the semester will have to earn those credits by registering for the same course whenever it is offered in subsequent semesters. If the course is not a core course, the student may choose to register for any other course next semester in that basket as advised by the department faculty adviser. Additional fees for those number of credits may apply.

10. Students may opt for summer training/internships/industrial tours as advised by the department. However, these activities will not have credits.

Baskets of FFCBCS

11 Baskets of courses have been identified to provide student comprehensive exposure to a large number of areas, leading to the holistic development of an individual. These baskets are as follows:

1. **Language and Literature:** These include courses related to English or other popularlanguages worldwide, communication skills, and literature. These courses are of 3 credits each.

2. **Core Science:** These courses include science courses from the disciplines of Physics and Chemistry. These courses are of 5 credits each.

3. **Core Mathematics:** This basket includes courses from Mathematics department, crafted for Engineering students. These courses are of 4 credits each.

4. **Engineering Sciences:** This basket includes introductory courses from various disciplines of Engineering designed to provide the student solid foundation to the domain of engineering. These courses are of 4 credits each.

5. **Discipline Core:** This basket includes compulsory courses in the discipline in which the student is admitted to the University. These courses are of 4 credits each.

6. **Discipline Elective:** This basket provides students courses other than discipline core, and are normally in certain specialized areas. These courses are of 3 credits each.

7. **Humanities and Liberal Arts:** This basket includes liberal arts courses in various disciplines like psychology, management, economics, etc., and are of 3 credits each.

8. **Skill Enhancement:** Courses in this basket are primarily hands-on and aims to allow students acquire skills required in certain disciplines that are currently in high demand in the job market. Thesecourses are of 2 credits each.

9. **Ability Enhancement:** These courses aim to enhance knowledge and ability of an individual in certain required areas related to national and societal interest. Courses in this basket are of 2 credits each.
10. **Free Electives:** Student can register for any three courses outside their department of his/her choice. These courses can also be taken from MOOCs, and a minimum of 9 credits have to be taken bya student in this basket. These courses are of 3 credits each.
11. **Capstone Project:** Capstone project is a semester long multifaceted experimental/research assignment that serves as a culminating academic and intellectual experience for students, taken in the last semester of study. It is of 12 credits and may be done groups of not more than three students, and in three modes as follows:
    - **Mode A**: Project with a department faculty.
    - **Mode B**: Project as part of Industry Internship arranged only by the career and placement service of the University. Students securing this assignment on their own will not be allowed, unless the project is secured at a well-known industry, and duly approved by the department. The department's decision in all such cases will be final.
    - **Mode C**: Semester long project in an academic institute/lab of National/International Importance, secured by students on their own. The department's decision to allow in all such cases willbe final.

A separate rule booklet will be released for implementation of Capstone Project.

Structure of the B.Tech. FFCBCS Program in Computer Science and Engineering & IT

Basket/Area	Min Credits To be taken	Credit per course	Courses
Language and Literature (LL) Core: Professional Communication Elective: Choose any 1 more LL course	6	3	2
Core Sciences (CoS) Core: None Elective: Choose any 2 CoS Course	10	5	2
Core Mathematics (CM) Core: Engg. Maths 1, Engg. Maths 2, Probs and Stats Elective: None	12	4	3
Engineering Sciences (ES) Core: Prog. For Problem Solving, Data Structures, Semiconductor Device Fundamentals Elective: Choose any 2 more ES courses	20	4	5
Discipline Core (DC) Core: CO, DM, OOPJ, OS, DAA, CN, DBMS, AI, AE, VCT, MES, DID Elective: None	48	4	12
Discipline Elective (DE) Core: None Elective: Choose any 6 courses as per your Specialization	18	3	6
Humanities and Liberal Arts (HL) Core: None Elective: Choose any 3 HL Courses	9	3	3
Skill Enhancement Courses (SEC)* Core: None Elective: Choose any courses to complete credits	8	-	-
Ability Enhancement Courses (AEC)* Core: Entrepreneurship and startups, Env. Sc, Indian Constitution Elective: None	8	-	-
Free Electives (FE) Core: None Elective: Choose any 3 FE courses	9	3	3
Capstone Project (TP) Mode A: Project with a department faculty Mode B: Project as part of Industry Internship Mode C: Project in an academic institute/lab of National Importance. All Modes must be semester long	12	12	1
Total Credits	160		

* *Credits in SEC and AEC courses may vary.*

DIT University CSE/IT FFCBCS Program Structure

Basket/Area	Credits
Language and Literature (LL)	6
Core Sciences (CoS)	10
Core Mathematics (CM)	12
Engineering Sciences (ES)	20
Discipline Core (DC)	48
Discipline Elective (DE)	18
Humanities and Social Sciences (HSS)	9
Skill Enhancement Courses (SEC)	8
Ability Enhancement Courses (AEC)	8
Free Electives (FE)	9
Project (PRJ)	12
Total	**160**

Course Baskets: University FFCBCS Baskets (other than DC/DE) for B.Tech.Program.
A * against a course means it is a core course for all B.Tech. Students.

Course Code	FFCBCS Baskets (other than DC/DE)				
	Language and Literature (min 6 credits to be taken)	**Contact Hrs**			**Credits**
	Name of Courses	L	T	P	C
LAF181	Professional Communication*	2	0	2	3
LAF182	Indian English Literature	3	0	0	3
LAF183	English Language Teaching	3	0	0	3
LAF184	Corporate Communication and Soft Skills	2	0	2	3
	Core Sciences (min 10 credits to be taken)				
	Name of Courses	L	T	P	C
CHF101	Engineering Chemistry(CSE,IT,EE,ECE)	3	1	2	5
CHF102	Applied Engineering Chemistry (for ME/CE/PE)	3	1	2	5
PYF101	Wave & Optics and Introduction to Quantum Mechanics	3	1	2	5
PYF102	Introduction to Mechanics	3	1	2	5
PYF103	Electricity & Magnetism	3	1	2	5
PYF105	Engineering Physics* (Since 2022)	3	1	2	5
	Core Mathematics (min 8 credits to be taken)				
	Name of Courses	L	T	P	C
MAF101	Engineering Mathematics I*	3	1	0	4
MAF102	Engineering Mathematics II*	3	1	0	4
MAF201	Engineering Mathematics III (EE, ME, CE)	3	1	0	4
MAF202	Probability and Statistics (CSE, IT, ECE, PE)	3	1	0	4
	Engineering Sciences (min 20 credits to be taken)				
	Name of Courses	L	T	P	C
ECF101	Fundamental of Electronics Engineering.	3	0	2	4
EEF101	Basic Electrical Engineering	3	0	2	4
EEF143	Electrical and Electronics Engineering Practice (non EE/EECE)	3	0	2	4
MEF101	Thermodynamics	3	1	0	4
CSF101	Programming for Problem Solving*	3	0	2	4
CSF102	Data Structures*	3	0	2	4
CSF103	Semiconductor Device Fundamentals*	3	1	0	4
CSF104	Digital Logic and Computer Design	3	0	2	4
MEF102	Engineering Graphics	2	0	4	4
MEF103	Engineering Mechanics	2	1	2	4
MEF 106	Modern Manufacturing techniques	2	0	4	4
MEF201	Mechanical Engineering Materials	3	0	2	4
PEF204	Fluid Mechanics	3	0	2	4
EEF141	Electrical Engineering Material	3	0	2	4
ECF142	Fundamental of Semiconductor Electronics	3	1	0	4
ECF144	Digital Electronics and Applications	3	0	2	4
CEF101	Civil Engineering Materials	3	1	0	4

	Skill Enhancement (min 8 credits to be taken)				
	Name of Courses	L	T	P	C
CSF306	Technical Training 1	2	0	4	4
CSF307	Technical Training 2	2	0	4	4
	Ability Enhancement (min 8 credits to be taken)				
	Name of Courses	L	T	P	C
CHF201	Environmental Science*	2	0	0	2
LAF285	Indian Constitution*	2	0	0	2
MEF483	Entrepreneurship and Start-ups*	0	0	4	2
UCF201	Aptitude and Soft Skills	2	0	0	2
	Humanities and Liberal Arts (min 9 credits to be taken)				
	Name of Courses	L	T	P	C
LAF281	Introduction to Psychology	3	0	0	3
LAF282	Human Values	3	0	0	3
LAF283	Literature, Language & Society	3	0	0	3
LAF284	Principles of Management	3	0	0	3
LAF285	Indian Constitution	2	0	0	2
LAF286	Youth Psychology	3	0	0	3
LAF287	Sustainable Development	3	0	0	3
LAF381	Positive Psychology & Living	3	0	0	3
LAF382	Engineering Economics	3	0	0	3
LAF383	Introduction to Linguistics	2	0	2	3
LAF384	Creative Writing	3	0	0	3
LAF385	Health Psychology	3	0	0	3
LAF386	Ecology and Human Development	3	0	0	3
LAF481	Application of Psychology	3	0	0	3
LAF482	Intellectual Property Rights	3	0	0	3
LAF483	Science Technology and Society	3	0	0	3
LAF484	Education and Social Change	3	0	0	3
LAF485	Industrial Psychology	3	0	0	3
LAF486	Innovation and Entrepreneurship	3	0	0	3
	Free Electives (min 9 credits to be taken)				
	Name of Courses	L	T	P	C
ECF481	Analogue Electronics (ECE)	2	0	2	3
ECF482	Cellular Communication Network (ECE)	2	0	2	3
ECF381	Microcontroller (ECE)	2	0	2	3
ECF382	Bio Medical Instrumentation (ECE)	2	0	2	3
ECF483	Digital Image processing (ECE)	2	0	2	3
CSF381	Software Project Management	3	0	0	3
CSF345	Introduction to Data Science	3	0	0	3
CSF482	Introduction to Cyber security	3	0	0	3
MEF348	Robotics Engineering	3	0	0	3
MEF381	Composites materials	3	0	0	3

MFF444	Operation Research	2	1	0	3
MEF446	Product Design & Development	3	0	0	3
MEF481	Total Quality Management	3	0	0	3
MEF482	Renewable Energy Sources	3	0	0	3
MEF485	Solar Energy System	2	0	2	3
PEF381	Carbon Capture and Sequestration	3	0	0	3
PEF491	Polymer Technology	3	0	0	3
PEF492	Health, Safety and Environment in Industry	3	0	0	3
CEF281	Properties of Materials	3	0	0	3
CEF382	Disaster Preparedness Planning & Management	3	0	0	3
CEF481	Environmental Management & Sustainability	3	0	0	3
CEF482	Natural Dynamics	3	0	0	3
CEF483	GIS	3	0	0	3
CEF484	Resource Dynamics and Economic Implications	3	0	0	3
CEF343	Environmental Risk Assessment and Disaster Management	3	0	0	3
CEF348	Air & Water Pollution	3	0	0	3
CEF349	Remote sensing and Image processing	3	0	0	3
CHF366	Green Chemistry	3	0	0	3
MAF452	Optimization Techniques	3	0	0	3
	Project (12 credits)				
UCF439	Capstone Project	0	0	12	12

Course Baskets: B.Tech. CSE FFCBCS DC Basket and CSE DE specialized tracks Baskets.

Discipline Core (48 credits)						
			Contact Hrs			Credits
	Name of Courses	**Pre-requisite Courses**	L	T	P	C
CSF201	Computer Organization and Architecture	None	3	1	0	4
CSF202	Discrete Mathematics	None	3	1	0	4
CSF204	Operating Systems	CSF201	3	0	2	4
CSF205	Database Management Systems	CSF101	3	0	2	4
CSF207	Object Oriented Programming with Java	CSF101	3	0	2	4
CSF208	Analog Electronics Fundamentals	None	3	0	2	4
CSF209	Digital IC Design	CSF104/ECF144	3	0	2	4
CSF302	Design and Analysis of Algorithms	CSF102	3	0	2	4
CSF303	Computer Networks	CSF101	3	0	2	4
CSF304	Artificial Intelligence	CSF201	3	0	2	4
CSF310	VLSI Circuits Technologies	CSF208	3	0	2	4
CSF311	Microcontrollers and Embedded Systems	CSF201	3	0	2	4

Discipline Electives (min 18 credits to be taken)					
Chip Design					
	Name of Courses	L	T	P	C
CSF331	IC Fabrication Technologies	3	0	0	3
CSF332	CMOS Analog VLSI Design	2	0	2	3
CSF431	Emerging Technologies in Chip Design	3	0	0	3
CSF432	FPGA Systems	2	0	2	3
CSF442	Robotics	2	0	2	3
CSF444	Internet of Things	2	0	2	3

Abbreviations

1.	PSP	Problem Solving and Programming
2.	DS	Data Structures
3.	SDF	Semiconductor Device Fundamentals
4.	DLCD	Digital Logic and Computer Design
5.	COA	Computer Organization & Architecture
6.	OS	Operating System
7.	DAA	Design and Analysis of Algorithm
8.	AI	Artificial Intelligence
9.	DM	Discrete Mathematics
10.	OOPJ	Object Oriented Programming with Java
11.	DBMS	Database Management Systems
12.	CN	Computer Networks
13.	AEF	Analog Electronics Fundamentals
14.	VCT	VLSI Circuits Technologies
15.	MES	Microcontrollers and Embedded Systems
16.	DID	Digital IC Design
17.	CAVD	CMOS Analog VLSI Design
18.	IFT	IC Fabrication Technologies
19.	ETCD	Emerging Technologies in Chip Design
20.	FS	FPGA Systems
21.	IoT	Internet of Things

Flow of Actions for implementing FFCBCS every semester

After release of Final Exam results, Academic Advisory Committee meets to decide & finalizecourse offerings in each basket

Courses are created in SAP and in LMS with required number of seats

Registrar announces the date for Registration

Students get advised and registers for courses in the Student Advising

Centre
List of students gets added in LMS

Class Starts

1. Department offering the course	Computer Science and Engineering
2. Course Code	CSF101
3. Course Title	Programming for problem solving
4. Credits (L:T:P:C)	3:0:2:4
5. Contact Hours (L:T:P)	3:0:2
6. Prerequisites (if any)	None
7. Course Basket	Engineering Sciences

COURSE OUTLINE:

This course contains the fundamental concepts about the computer hardware and intends to provide to students about the knowledge of C language

COURSE OBJECTIVE:

The objective of the course is to make the students to understand the key hardware components in a modern computer system and as to how the software is mapped to the hardware. The student shall also be able to learn make the computer programs using C language by exploring the various features of C.

COURSE OUTCOMES:

At the end of the course, the student will be able to:

CO1. Develop simple algorithms for arithmetic and logical problems.

CO2. Implement conditional branching, iteration and recursion.

CO3. Describe a problem into functions and synthesize a complete program using divide and conquer approach.

CO4. Implement arrays, pointers and structures to formulate algorithms and programs.

CURRICULUM CONTENT

UNIT 1: Introduction to Computer, Programming & algorithms (8 L)

Introduction to components of a computer system (disks, memory, processor, where a program is stored and executed, operating system, compilers etc.)

Idea of Algorithm: steps to solve logical and numerical problems. Representation of Algorithm: Flowchart/Pseudocode with examples, From algorithms to programs; source code, variables (with data types) variables and memory locations, Syntax and Logical Errors in compilation, object and executable code.

UNIT 2: Arithmetic Expression, and Conditional statements, Loops, Expression: (7 L)

Arithmetic, Logical, Relational expressions and precedence.

Loops & Branching: Writing and evaluation of conditionals and consequent branching, Iteration and loops.

UNIT 3: Arrays & Functions (7 L)

Arrays: Arrays (1-D, 2-D), Character arrays and Strings.

Functions: functions (including using built in libraries), Parameter passing in functions, call by value, passing arrays to functions: idea of call by reference, Storage class.

Searching & Sorting: Searching, Basic Sorting Algorithms (Bubble sort)

UNIT 4: Recursion and Structure (8 L)

Recursion: Recursion, as a different way of solving problems. Example programs, such as Finding Factorial, Fibonacci series, Ackerman function etc.

Structure: Structures, Defining structures and Array of Structures.

UNIT 5: Pointers & File handling (7 L)

Pointers: Idea of pointers, Defining pointers, Use of Pointers in self-referential structures

File handling: different modes of opening a file in C, reading, writing from files.

TEXT BOOKS

1. Byron Gottfried, "Schaum's Outline of Programming with C", 2nd edition 2006 McGraw-Hill.
2. E. Balaguruswamy, "Programming in ANSI C", 8th Edition 2019, McGraw-Hill Education India.

REFERENCES

1. Brian W. Kernighan and Dennis M. Ritchie, "The C Programming Language", 2nd edition 1988, Prentice Hall of India.

LIST OF EXPERIMENTS:

S.NO.	EXPERIMENT NAME
1	Familiarization with programming environment.
2	Programming for Simple computational problems using arithmetic expressions.
3	Programming for Problems involving if-then-else structures.
4	Programming for Iterative problems e.g., sum of series.
5	Programming for 1-D Array manipulation.
6	Programming for Matrix problems, String operations.
7	Programming for Simple functions
8	Programming for Recursive functions.
9	Programming for Pointers and structures.
10	Programming for File operations

1.	School offering the course	School of Computing
2.	Course Code	**CSF102**
3.	Course Title	**Data Structures**
4.	Credits (L:T:P:C)	3:0:1:4
5.	Contact Hours (L:T:P)	3:0:2
6.	Prerequisites (if any)	CSF101 (Programming for problem solving)
7.	Course Basket	Engineering sciences

Course Summary: The course is a foundation level course and requires the knowledge of the C programming language. The course outlines the detailed architecture and implementation of basic data structures such as Stacks, Queues, Linked Lists, Trees, and Graphs. It also covers the time and space complexity analysis of different searching and sorting techniques. Some of the searching methods include Linear Search, Binary Search, and sorting mechanism includes Bubble sort, Insertion sort, Selection sort, Quick sort, Merge sort, and Heap Sort. The course also incorporates different hashing techniques, designing hash functions, hash table implementation, and collision resolution technique.

Course Objectives: The main objective of this course is to introduce the concept of data structure, how to choose a particular data structure, and how the choice of a data structure impacts the performance of programs. Other objectives include:
CO1: How to select the appropriate data structure model specific to some application.
CO2: Solve problems using data structures like Stacks, Queues, Linked Lists, Trees, Graphs, and writing programs for these solutions using C code.
CO3: Introduce the concept of algorithm writing, analyzing algorithms, converting pseudocode to appropriate C code, and showing how one solution is better than others by analyzing their computational complexities.

Course Outcomes: On successful completion of the course, students will be able to:
CO1: Develop an ability to read, write, and analyze the time and space complexity of any algorithms.
CO2: Describe the properties, behaviour, and implementation of basic data structures like Stacks, Queues, Linked List, Trees, and Graphs.
CO3: Convert pseudocode to its appropriate C code implementation.
CO4: Compare different searching and sorting techniques in terms of their memory usage and time consumption.
CO5: Design and implement different hash functions, analyze the collision effect, and hash table implementations.

Curriculum Content

Unit I: Introduction to Algorithms & Data Structure **(8 L)**
Introduction: Data types, Abstraction, Abstract Data Type (ADT), Concept of data structure, Types of data structures, Operations on Data Structures, Introduction to Algorithms, Writing Pseudocodes, Algorithm analysis, Complexity of algorithms and Time space trade-off, Searching: Linear and Binary Search Techniques and their complexity analysis.

Unit II: Arrays, Stacks, and Queues **(7 L)**
Arrays: Introduction to Array, Applications of Array, Operations on Arrays: Traverse, Insert, Delete etc. Stacks: Introduction to Stacks, Array representation of Stack, Operations on Stack: Push, Pop, etc. Applications of Stacks: Infix and Postfix Conversion, Evaluations of Infix and Postfix expressions. Queue: Introduction to Queue, Array representation and implementation of queues, Operations of Queue, Applications of Queue, Types of Queue: Circular Queue, Priority Queue, Double ended Queue. Operations on each type of Queue and their Applications.

Unit III: Linked Lists and Trees **(8 L)**
Linked Lists: Introduction to Dynamic Memory Allocation, Representation and Implementation of Single, Double, and Circular Linked Lists, Operations on Linked List: Insert, Delete, Traverse etc. Applications of Linked List, Linked List representation of Stack and Queue. Trees: Basic Tree terminologies, Types of Trees: Binary Tree, Binary Search Tree (BST), AVL Tree, B-Tree, and Heap. Representation and Implementations of different types of trees, Tree Traversal algorithms, Operation on trees: Insert, Delete, etc., Applications of Tress.

Unit IV: Graphs **(7 L)**
Graphs: Introduction to Graph and their Terminologies, Types of Graph, Representations of Graph, Graph traversal algorithms, Topological Sorting, Minimum Spanning Tree, Shortest Path Algorithms: Single Source Shortest Path like Bellman-Ford, Dijkstra and All Pair Shortest Path like Floyd-Warshall.

Unit V: Sorting & Hashing **(9 L)**
Sorting Algorithms and their Analysis: Selection Sort, Bubble sort, Insertion sort, Quick sort, Merge sort, Heap Sort. Performance Analysis and Comparison of all sorting techniques. Hashing: Hash Functions and its type, Hash Table construction, Collision Resolution, Universal Addressing, Open Hashing.

Text Books
1. Aaron M. Tenenbaum, Yedidyah, Langsam, Moshe J. Augenstein, Data Structures using C Pearson. 1st Edition. 2019
2. Schaum's outline series, Data structures with C, McGraw Hill Education; 1st edition (July 2017)

Reference Books
1. Horowitz and Sahani, "Fundamentals of Data Structures", Galgotia Publication,2nd Edition. 2008. 2. Robert Kruse, Data Structures and Program Design in C PHI.2nd Edition.2006.
3. Willam J. Collins, Data Structure and the Standard Template library –2003, T.M.H.1st Edition.
4. Kyle Loudon, Mastering Algorithms with C, O'Reily Publication, 1st Edition, 1999

List of Experiments:
1. Program in C for the implementation of Array for various operations.
2. Program in C for the creation of Stack for its various operation implementation.
3. Program in C for the creation of Queue for its various operation implementation.
4. Program in C for the creation of Link list for its various operation implementation.
5. Program in C for the creation of Circular Link list for its various operation implementation.
6. Program in C for the creation of Doubly Link list for its various operation implementation.
7. Program in C for the creation of Binary Search Tree for its various operation implementation.
8. Program in C for the Implementation of sorting Algorithms.
9. Program in C for the Implementation of basic Graph Algorithms.

1.	School offering the course	School of Computing
2.	Course Code	**CSF103**
3.	Course Title	**Semiconductor Device Fundamentals**
4.	Credits (L:T:P:C)	3:1:0:4
5.	Contact Hours (L:T:P)	3:1:0
6.	Prerequisites (if any)	None
7.	Course Basket	Engineering Science

Course Summary: The course covers fundamental solid-state physics concepts of a specific material class, namely semiconductors. Fundamental properties of semiconductors will be explored, as well as their device applications will be looked into.

Course Objective: To give knowledge about semiconductor physics and discus working and applications of basic devices, including p-n junctions, BJTs and FETs.

Course Outcome: At the end of the course, the student will be able to:
CO1: Apply the knowledge of basic semiconductor material physics
CO2: Explain the basic properties of semiconductors including the band gap, charge carrier concentration, doping and charge carrier injection/excitation.
CO3: Explain the working, design considerations and applications of various semiconducting devices including p-n junctions, BJTs and FETs.
CO4: Illustrate the qualitative knowledge of power electronic devices and become aware of the modern semiconductor devices.

Curriculum Content

Unit I: Basic Introduction (8 L)
Electrical conductivity: Classical free electron theory – assumptions, drift velocity, mobility and conductivity, drawbacks. Quantum free electron theory – Fermi energy, Fermi factor, carrier concentration. Band theory of solids – origin of energy bands, effective mass, distinction between metals, insulators and semiconductors.

Unit II: Theory of Semiconductors (10 L)
Intrinsic and extrinsic semiconductors, band structure of semiconductors, carrier concentration in intrinsic and extrinsic semiconductors, electrical conductivity and conduction mechanism in semiconductors, Fermi level in intrinsic and extrinsic semiconductors and its dependence on temperature and carrier concentration. Carrier generation – recombination, mobility, drift-diffusion current. Hall effect, Einstein relation, Continuity equation, Carrier injection, Diffusion length.

Unit III: Theory of PN Junctions (7 L)
PN junction formation, Diode and transistor: PN junction under thermal equilibrium, forward bias, reverse bias, carrier density, current, electric field, barrier potential. V-I characteristics, junction capacitance and voltage breakdown. Zener and Avalanche breakdown, Rectifying contact and Ohmic contact.

Unit IV: Transistors (7 L)
Fundamentals of Bipolar junction transistor, p-n-p and n-p-n transistors. JFET construction, amplifying and switching, Pinch off and saturation, Gate control, I-V characteristics. MOSFET fundamentals, Operation, MOS capacitor, Threshold voltage and its control, MOS C-V analysis and time dependent capacitance. Output and transfer characteristics of MOSFET, Heterojunctions – quantum wells.

Unit V: Modern Semiconductor Devices (7 L)
CCD – Introduction to nano devices, fundamentals of tunneling devices, design considerations, physics of tunneling devices, Power devices, operation and characteristics. Thyristor family. Power diodes. Power transistors.

Text Books:
1. S.M. Sze, Y. Li, K.K. Ng, Physics of Semiconductor Devices, John Wiley, 4^{th} Edition, 2021.
2. Ben G. Streetman and S.K. Banerjee, Solid State Electronic Devices, Pearson, 7^{th} Edition, 2018.

Reference Books:
1. D. Naeman, D. Biswas, Semiconductor Physics and Devices, McGraw-Hill, 4^{th} Edition, 2017.
2. C. Kittel, Introduction to Solid State Physics, Wiley, 2019.

1.	School offering the course	School of Computing
2.	Course Code	**CSF104**
3.	Course Title	**Digital Logic and Computer Design**
4.	Credits (L:T:P:C)	3:0:1:4
5.	Contact Hours (L:T:P)	3:0:2
6.	Prerequisites (if any)	None
7.	Course Basket	Engineering Science

Course Summary: This course covers the concepts of digital logic along with HDL. The course includes the fundamental concepts of Boolean algebra and its application for circuit analysis, multilevel gates networks, flip-lops, counters logic devices and synchronous and asynchronous sequential logic and digital integrated circuits.

Course Objective: This course aims to provide students with the understanding of the different digital logic circuits starting form logic gates to sequential circuits. The course also aims to introduce the technologies related to HDLs, so the students can construct, compile and execute Verilog HDL programs using provided software tools.

Course Outcome: At the end of the course, the student will be able to:
CO1: Understand and examine the structure of various number systems and its application in digital design.
CO2: Understand, analyze and design various combinational and sequential circuits.
CO3: Implement various Combinational and sequential circuits using verilog descriptions.
CO4: Synthesize complex digital circuits at several level of abstractions.

Curriculum Content

Unit I: Number Systems and Boolean Algebra (8 L)
Introduction to number system and Boolean algebra; Boolean identities, basic logic functions, standard forms of logic expressions, simplification of logic expressions. Logic families: Brief overview of Transistor as a switch; Logic gate characteristics – propagation delay, speed, noise margin, fan-out and power dissipation; Standard TTL and static CMOS gates.

Unit II: Combinational and Sequential logic (8 L)
Combinational logic: Arithmetic circuits, decoders, encoders, multiplexers, de-multiplexers, and their use in logic synthesis; Hazards in combinational circuits. Sequential logic circuits: Latches and Flip Flops (SR, D, JK, T); Timing in sequential circuits; Shift register; Counters – synchronous, asynchronous

Unit III: Introduction to Verilog (6 L)
Data types, operators and modules, Basics of gate level modelling, combinational and sequential logic circuit design using Verilog, basics of data flow modelling, memory, using a test bench for verification and simulation.

Unit IV: Finite State Machines (8 L)
Basic concepts and design; Moore and Mealy machines examples; State minimization/reduction, state assignment; Finite state machine design case studies and FSM circuit design examples in verilog and simulation.

Unit V: Memories and FPGAs (7 L)
ROM and RAM, PLA, PAL and FPGA; RTL based design projects and their implementation in FPGA using verilog.

Text Books:
1. M. Morris Mano, M.D. Ciletti, Digital Design: With an Introduction to the Verilog HDL, Pearson, 6th Edition, 2018.

Reference Books:
1. J. Cavanagh, Digital Design and Verilog HDL Fundamentals, CRC Press, 2017.
2. V. Taraate, Digital Logic Design Using Verilog: Coding and RTL Synthesis, Springer, 2016.

1.	School offering the course	School of Computing
2.	Course Code	**CSF201**
3.	Course Title	**Computer Organization & Architecture**
4.	Credits (L:T:P:C)	3:1:0:4
5.	Contact Hours (L: T:P)	3:1:0
6.	Prerequisites (if any)	CSF101
7.	Course Basket	Discipline Core

Course Summary

The course is proposed to teach the students the concepts of computer organization for several engineering computing systems. Students will develop the ability and confidence to use the fundamentals of computer organization as a tool in the engineering of digital systems.

Course Objectives

This course will facilitate the students to learn the fundamentals of computer organization and its relevance to classical and modern problems of computer design.

Course Outcomes

On successful completion of the course, students will be able to:
CO1: Understand the hardware components and concepts related to the control design, data representation and evaluation process of different arithmetic operations.
CO2: Understand the CPU organization, addressing modes and different types of instruction formats.
CO3: Understand input/output and memory organization.
CO4: Understand parallel processing and different types of multiprocessor's interconnection structures.

Curriculum Content

Unit I: Basic Structure of Computers & Register Transfer Language (8 L)
Basic Structure of Computers: Computer Types; Functional Units
Register Transfer and Micro operation: Register Transfer Language, Bus and Memory Transfers, Bus Architecture, Arithmetic, Logic, Shift Micro-operation, Design of ALU.

Unit II: Computer Arithmetic (8 L)
Introduction, Addition and Subtraction Algorithms, Multiplication and Division Algorithms, Floating Point Arithmetic Operation, IEEE Format for Floating Point Numbers.

Unit III: Control Design & Processor Organization (8 L)
Control Design: Execution of a Complete Instruction, Sequencing of Control Signals, Single and Multiple Bus Architecture, Hardwired Control Unit, Micro Programmed Control Unit
Processor Organization:
Accumulator Organization, General Register Organization, Stack Organization, Addressing Modes, Instruction Format, Data Transfer & Manipulations, Program Control.

Unit IV: Input-Output & Memory Organization (8 L)
Input-Output Organization: I/O Interface, Modes of Transfer, Interrupts & Interrupt Handling, Direct Memory Access, Input-Output Processor, Serial Communication.
Memory Organization: Memory Hierarchy, Main Memory (RAM and ROM Chips), Organization of Cache Memory (performance and mapping), Virtual Memory, Page Replacement Techniques.

Unit V: Parallel Processing & Multiprocessor (8 L)
CPU Performance: Processor Clock, Clock Rate, Cycle, Basic Performance Equation, and MIPS Rate.
Parallel Processing: Flynn's classification, Pipelining- Arithmetic Pipelining, Vector Processing, Array Processor, pipeline hazards.

Multiprocessor: Characteristic of Multiprocessor, Interconnection Structure, Cache Coherence.

Text Books:

1. John P. Hayes. Computer Architecture and Organization, 4th Edition, McGraw Hill, 2010.
2. M. Morris Mano. Computer System Architecture 3rd Ed, Pearson.
3. Carl Hamacher, ZvonkoVranesic, Safwatzaky. Computer Organization, 5th Edition.

Reference Books:

1. John L. Hennessey and David A. Patterson: Computer Architecture, A Quantitative Approach, 4th Edition, Elsevier, 2007.
2. Kai Hwang: Advanced Computer Architecture Parallelism, Scalability, Programmability, 2nd Edition, Tata Mc Graw Hill, 2010.

1.	School offering the course	School of Computing
2.	Course Code	**CSF202**
3.	Course Title	**Discrete Mathematics**
4.	Credits (L:T:P:C)	3:1:0:4
5.	Contact Hours (L:T:P)	3:1:0
6.	Prerequisites (if any)	None
7.	Course Basket	Discipline Core

Course Summary: This course covers elementary discrete mathematics for computer science and engineering. It emphasizes mathematical definitions and proofs as well as applicable methods. Topics include formal logic notation, proof methods; induction, well-ordering; sets, relations; elementary graph theory; integer congruence; asymptotic notation and growth of functions; permutations and combinations, and counting principles.

Course Objectives: The objectives of this course is to learn concepts of Discrete Mathematics and by applying the algorithms to solve the problems related to Recursion, combinatorial mathematics and problems on basic graph theory.

Course Outcomes: On successful completion of the course, students will be able to:
CO1: Apply the techniques to perform the operations on discrete structures such as sets, functions, relations, and sequences.
CO2: Identify the properties of Lattice by constructing the Hasse Diagram and demonstrate the proofs to solve problems using counting techniques.
CO3: Apply the properties of Algebric structures and design the propositional and predicate logic.
CO4: Apply the properties of Graph and Recurrence Relation to solve computational problems.

Curriculum Content

UNIT I: Introduction to Sets, Relations & Functions (7 L)
Set Theory: Introduction, Combination of sets, Multisets, ordered pairs, Set Identities.
Relations: Definition, Operations on relations, Properties of relations, Composite Relations, Equality of relations, Order of relations.
Functions: Definition, Classification of functions, Operations on functions, Recursively defined functions.
Natural Numbers: Introduction, Mathematical Induction.

UNIT II: Posets & Introduction to Boolean algebra (6 L)
Partial order sets: Definition, Partial order sets, Combination of partial order sets, Hasse diagram.
Lattices: Definition, Properties of lattices Bounded, Complemented and Complete Lattice
Combinatorics: Introduction, Counting Techniques, Pigeonhole Principle.

UNIT III: Groups & Rings (8 L)
Algebraic Structures: Definition, Groups, Subgroups and order, Cyclic Groups, Cosets, Lagrange's theorem, Normal Subgroups, Definition and elementary properties of Rings and Fields, Integers modulo n.

UNIT IV: Propositional logic, Predicate Logic & Introduction to Probability (8 L)
Propositional Logic: Proposition, well-formed formula, Truth tables, Tautology, Contradiction, Algebra of proposition, Theory of Inference, Natural Deduction.
Predicate Logic: First order predicate, well-formed formula of predicate, quantifiers, Inference theory of predicate logic.

UNIT V: Introduction to Graphs & Recurrence Relations (8 L)
Graphs: Definition and terminology, Representation of graphs, multigraphs, bipartite graphs, Planar graphs, Isomorphism and Homeomorphism of graphs, Euler and Hamiltonian paths, Graph coloring. Recurrence Relation & Generating function: Recursive definition of functions, Recursive algorithms, Method of solving recurrences.

Text Books:

1. Liu C.L., Elements of Discrete Mathematics, McGraw Hill Education. 4th edition, 2017.
2. Kolman B & Busby C.R., Discrete Mathematical Structure for Computer Science, Prentice Hall of India Ltd. 6th Edition 2008.
3. Deo N., Graph Theory, Prentice Hall of India 1974

Reference Books:

1. Trembley J.P. &Manohar R., Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw Hill.1st Indian Edition 2017.

1.	School offering the course	School of Computing
2.	Course Code	**CSF204**
3.	Course Title	**Operating Systems**
4.	Credits (L:T:P:C)	3:0:1:4
5.	Contact Hours (L:T:P)	3:0:2
6.	Prerequisites (if any)	CSF201
7.	Course Basket	Discipline Core

Course Summary: This course will introduce the core concepts of operating systems, such as processes and threads, scheduling, synchronization, memory management, file systems, input and output device management and security.

Course Objectives: This course is classified into two sections: a theory section that educates to students about the theories and principles that underlie modern operating systems, and a practical section that relates theoretical principles to operating system implementation. Theory section basically includes: Process and processor management, concurrency and synchronisation, memory management schemes, file system and secondary storage management, etc.

Course Outcomes: On successful completion of the course, students will be able to:
CO1: Describe the basic concepts of operating systems, including development and achievements, functionalities and objectives, structure and components.
CO2: Understand the general architecture & functioning of operating system such as processes, threads, files, Concurrency, IPC abstractions, shared memory regions, etc.
CO3: Analyze various algorithms e.g. Process scheduling and memory management algorithms.
CO4**:** Categorize the operating system's resource management techniques, deadlock management techniques, memory management techniques.

Curriculum Content

UNIT I: Introduction to Operating System **(8 L)**
Introduction: Components of a computer System, Operating system: User view & System view, Evolution of operating system, Single Processor & Multiprocessor systems, Real Time System, Distributed Systems, Multimedia Systems, Handheld Systems.
Operating System Structure: Operating System Services, User Operating System Interfaces: Command- Line and GUI, System Calls.

UNIT II: Management & Scheduling **(6 L)**
Process Management: Process Concept, Process States, Process Transition Diagram, Process Control Block (PCB).
CPU Scheduling: Scheduling Concepts, Performance Criteria, Scheduling Queues, Schedulers, Scheduling Algorithms: Preemptive &Non Preemptive: FCFS, SJF, Priority, Round-Robin.

UNIT III: Concurrent Processes & Deadlocks **(8 L)**
Concurrent Processes: Principle of Concurrency, Producer / Consumer Problem, Co-operating Processes, Race Condition, Critical Section Problem, Peterson's solution, Semaphores, Classical Problem in Concurrency- Dining Philosopher Problem; Inter Process Communication models and Schemes.
Deadlock: System Model, Deadlock Characterization, Prevention, Avoidance and Detection, Recovery from deadlock.

UNIT IV: Memory Management **(7 L)**
Memory Management: Bare machine, Resident monitor, Multiprogramming with fixed partition, Multiprogramming with variable partition, Multiple base register, Paging, Segmentation, Virtual memory concept, Demand paging, Performance, Paged replaced algorithm, Allocation of frames, Cache memory.

UNIT V: File System & I/O Management **(7 L)**

File System: Different types of files and their access methods, various allocation methods. I/O Management and Disk Scheduling: I/O Devices, Organization of I/O functions, Disk Structure, Disk Scheduling (FCFS, SSTF, SCAN, C-SCAN, LOOK).

Text Books:

1. Silberschatz, Galvin and Gagne, ―Operating Systems Concepts‖, Wiley, 9th Edition 2018.

Reference Books

1. Harvey M. Dietel, ― An Introduction to Operating System‖, Pearson Education 1st Edition 2009.
2. D.M. Dhamdhere, ―Operating Systems: A Concept based Approach‖, PHI. 3rd Edition.2017.

List of Experiments:

S.NO.	EXPERIMENT NAME
1	Implement the following algorithm FCFS, SJF, Round Robin, Priority in Linux.
2	Implement the concept of fork () system call using C programming in Linux environment only.
3	Implement the concept of threading in OS. Prefer threading in JAVA only.
4	Write a Java program to simulate producer-consumer problem using semaphores.
5	Write a Java program to simulate the concept of Dining Philosopher‘s problem.
6	Write a program using Linux to simulate Banker‘s algorithm.
7	Write a C program using Linux to simulate the following contiguous memory allocation techniques: a> Worst fit b> Best fit c> First fit.
8	Write a Java program to simulate the disk scheduling algorithms: a> FCFS b> SCAN c> C-scan
9	Write a C program using Linux to implement page replacement algorithms: a> FIFO b> LRU c> LFU
10	Write a C program to compare the Optimal page replacement algorithm with FIFO and LRU page replacement algorithms.

1.	School offering the course	School of Computing
2.	Course Code	**CSF205**
3.	Course Title	**Database Management Systems**
4.	Credits (L:T:P:C)	3:0:1:4
5.	Contact Hours (L:T:P)	3:0:2
6.	Prerequisites (if any)	CSF101
7.	Course Basket	Discipline Core

Course Summary: The students will learn the basic theory of databases. They will be able to design and develop a database using conceptual schema, logical schema, and physical schema and are expected to learn how to write database management system software. They will also learn some of the specialized databases.

Course Objectives: This course aims to educate students on the role of a well-structured relational database management system (RDBMS) in the efficient functioning of an organization. This course covers theory and practice in designing a relational database management system with an example of a current database product of MYSQL. Students also learn about the important concepts of database integrity, security, and availability with techniques like normalization, concurrency control, and recoverability control.

Course Outcomes: On successful completion of the course, students will be able to achieve the following:
CO1: Demonstrate the basic elements of a relational database management system.
CO2: Identify the data models for relevant problems.
CO3: Design entity-relationship and convert entity-relationship diagrams into RDBMS and formulate SQL queries.
CO4: Apply and create relational database design process with Normalization and De-normalization of data so that data redundancy, data inconsistency, and data loss problems may be resolved.

Curriculum Content

UNIT I: Introduction to Database System **(7 L)**
Introduction: Database System Applications, database System VS file System, Data Abstraction, Instances and Schemas, data Models: the ER Model, Relational Model & Other Models, Database Languages, database Users and Administrator, database System Structure, Storage Manager, the Query Processor, Two/Three-tier architecture.

UNIT II: E-R Modeling Data Base Design **(8 L)**
E-R model: Basic concepts, Design Issues, Mapping Constraints, Attributes and Entity sets, Relationships and Relationship sets, Keys, Entity-Relationship Diagram, Weak Entity Sets, Extended ER features.

UNIT III: Relational Model & SQL **(7 L)**
Relational Model: Structure of relational Databases, Relational Algebra, Relational Calculus, Extended Relational Algebra SQL: Form of Basic SQL Query, Nested Queries, Aggregative Operators, NULL values, Logical operators, Outer Joins, Complex Integrity Constraints in SQL.

UNIT IV: Database Design Concepts **(7 L)**
Database Design: Schema refinement, Different anomalies in designing a Database, Decompositions, Problem-related to decomposition, Functional Dependency, Normalization using functional dependencies, 1NF, 2NF, 3NF & BCNF, Lossless join decomposition, Dependency preserving Decomposition, Schema Refinement in Database Design, Multivalued Dependencies Closer properties of Multivalued dependency, Join dependency, 4NF, 5NF.

UNIT V: Transaction & Concurrency **(8 L)**
Transaction Management: Transaction-concepts, states, ACID property, schedule, serializability of schedules, concurrency control techniques - locking, timestamp, deadlock handling, recovery-log based recovery, shadow paging.

Text Books:

1. Raghurama Krishnan, Johannes Gehrke, Database Management Systems, TATA McGraw-Hill 3rd Edition,2014.
2. Silberschatz, Korth, Database System Concepts, McGraw hill, 6th edition,2013.
3. Elmasri Navate, Fundamentals of Database Systems, Pearson Education,7th edition 2016.

Reference Books:

1. Peter Rob & Carlos Coronel, Database Systems design, Implementation, and Management, Course Technology Inc, 7thEdition, 2006.
2. C.J. Date , Introduction to Database Systems, Pearson Education,8th edition,2012.
3. Bayross I., SQL, PL/SQL the Programming Language of Oracle, BPB Publications (2009) 4th ed.
4. HofferJ., Venkataraman, R. and Topi, H., Modern Database Management, Pearson (2016) 12th edition.

List of Experiments:

S.NO.	EXPERIMENT NAME
1	Implementation of Data Definition language in Query Language.
2	Implementation of Data Manipulation in Query Language.
3	Insertion & Updation of records in the database table
4	Implementation of GROUP functions (avg, count, max, min, Sum).
5	Execution of the various type of SET OPERATORS (Union, Intersect, Minus).
6	Apply the various types of Integrity Constraints on the table.
7	Creation of various types of JOINS.
8	Implementation of Views and Indices in database.
9	Implementation of the foreign key on the database.
10	Modify the database structure and drop the record with structure.

1.	School offering the course	School of Computing
2.	Course Code	**CSF207**
3.	Course Title	**Object Oriented Programming with Java**
4.	Credits (L: T:P:C)	3:0:1:4
5.	Contact Hours (L: T:P)	3:0:2
6.	Prerequisites (if any)	CSF101
7.	Course Basket	Discipline Core

Course Summary: This course is about the fundamentals of Object-Oriented Programming (OOP) Concepts and OOP-based software development methodology. The students are exposed to the concepts, fundamental syntax, and thought processes behind Object-Oriented Programming. By the end of the course, students will acquire the basic knowledge and skills necessary to implement object-oriented programming techniques in software development using Java.

Course Objectives: The objectives of this course are to learn object oriented programming paradigm using Java as programming language. Students will be exposed to fundamental concepts in java programming language, followed by object oriented paradigm and its building blocks.

Course Outcomes: On successful completion of the course, students will be able:
CO1: Define, understand, differentiate, and apply the Object oriented concepts using Java Programming language on real-time scenarios.
CO2: Use Exception handling and multithreading mechanisms to create efficient software applications.
CO3: Utilize modern tools and collection frameworks to create Java applications to solve real-world problems.
CO4: Design and develop GUI-based applications using applets and swings for internet and system-based applications.

Curriculum Content

Unit I: OOPS Concepts and Java Programming (9 L)
OOP concepts: Classes and objects, data abstraction, encapsulation, inheritance, benefits of inheritance, polymorphism, procedural and object oriented programming paradigm.
Java programming: History of java, comments data types, variables, constants, scope and life time of variables, operators, operator hierarchy, expressions, type conversion and casting, enumerated types, control flow statements, jump statements, simple java stand-alone programs, arrays, console input and output, formatting output, constructors ,methods, parameter passing, static fields and methods, access control, this reference, overloading methods and constructors, garbage collection, exploring string class.

UNIT II: **Inheritance, Interfaces and Packages (8 L)**
Inheritance: Inheritance hierarchies, super and subclasses, member access rules, super keyword, preventing inheritance: final classes and methods, the object class and its methods; Polymorphism: dynamic binding, method overriding, abstract classes and methods;
Interface: Interfaces VS Abstract classes, defining an interface, implement interfaces, accessing implementations through interface references, extending interface;
Packages: Defining, creating and accessing a package, understanding CLASSPATH, importing packages.

UNIT III: Collection Framework, File Handling (8L)
Interfaces: Collection - List - Set – Sorted Set - Enumeration - Iterator – List Iterator Other; Classes: Linked List – Array List - Vector – Hash Set – Tree Set – Hash table; Working with maps: Map interface; Map classes: Hash Map – Tree Map ; Utility classes– String Tokenizer, Formatter, Random, Scanner using javautil
File Handling: Streams, byte streams, character stream, text input/output, binary input/output, random access file operations, file management using file class:

UNIT IV: Exception Handling and Multithreading (8L)

Exception handling – Dealing with errors, benefits of exception handling, the classification of exceptions- exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, rethrowing exceptions, exception specification, built in exceptions, creating own exception sub classes, Guide lines for proper use of exceptions

Multithreading – Differences between multiple processes and multiple threads, thread states, creating threads, interrupting threads, thread priorities, synchronizing threads, interthread communication, thread groups, daemon threads.

UNIT V: GUI Programming and Event Handling (9 L)

GUI Programming: The AWT class hierarchy, introduction to swing, swings Vs AWT, hierarchy for swing components. Containers: JFrame, JApplet, JDialog, Jpanel, overview of some swing components: JButton, JLabel, JTextField, JTextArea, simple applications. Layout management: Layout manager types, border, grid and flow. Applets: Inheritance hierarchy for applets, differences between applets and applications, life cycle of an applet, passing parameters to applets.

Event Handling – Events, Event sources, Event classes, Event Listeners, Relationship between Event sources and Listeners, Delegation event model, Semantic and Low-level events, Examples: handling a button click, handling mouse and keyboard events, Adapter classes.

Text Books:

1. Intro to Java Programming (Comprehensive Version), by Y. Daniel Liang. Publisher: Pearson Education; Tenth edition (2018), ISBN-10: 935306578X, ISBN-13: 978-9353065782 .

Reference Books:

1. Java - The Complete Reference, by Herbert Schildt, Publisher: McGraw Hill Education; Tenth edition (2017), ISBN-10: 9789387432291, ISBN-13: 978-9387432291

List of Experiments:

S.NO.	EXPERIMENT NAME
1	WAP to calculate the Simple Interest and Input by the user.
2	WAP to create a class to find out the Area and perimeter of rectangle and box using super and this keyword.
3	WAP to design a class account using the inheritance and static that show all function of bank (withdrawal, deposit).
4	WAP to design a class using abstract Methods and Classes.
5	WAP to handle the Exception using try and multiple catch block.
6	WAP to Handle the user defined Exception using throw keyword.
7	WAP to design GUI application using AWT/Swing.
8	WAP to Implement the different layout (flow layout, Border Layout etc.)
9	Program to Implement push() and pop() on Stack using collection framework

1.	School offering the course	School of Computing
2.	Course Code	**CSF208**
3.	Course Title	**Analog Electronics Fundamentals**
4.	Credits (L:T:P:C)	3:0:1:4
5.	Contact Hours (L:T:P)	3:0:2
6.	Prerequisites (if any)	CSF103
7.	Course Basket	Discipline Core

Course Summary: The course covers fundamental concepts behind the working of different electronics devices like basic transistor amplifier circuits, oscillators.

Course Objective: To familiarize the student with the principal of operation, analysis and design of different electronic devices and circuits

Course Outcome: At the end of the course, the student will be able to:
CO1: Understand and analyze different types of diodes and their characteristics.
CO2: Determine operating point and various stability factors of transistor.
CO3: Design biasing circuits using diodes and transistors.
CO4: Design diode application circuits, amplifier circuits and oscillators employing BJT, FET devices.

Curriculum Content

Unit I: PN Junction Diode **(7 L)**
Qualitative Theory of P-N Junction, P-N Junction as a diode, diode equation, volt-ampere characteristics, temperature dependence of V-I characteristic, ideal versus practical –resistance levels (static and dynamic), transition and diffusion capacitances, diode equivalent circuits, load line analysis, breakdown mechanisms in semiconductor diodes, zener diode characteristics.

Unit II: Rectifiers and Filters **(7 L)**
P-N Junction as a rectifier, Half wave rectifier, full wave rectifier, Bridge rectifier, Harmonic components in a rectifier circuit, Inductor filter, Capacitor filter, L- section filter and comparison of various filter circuits, Voltage regulation using zener diode.

Unit III: Bipolar Junction Transistor **(8 L)**
Transistor construction, Input and Output characteristics of transistor in Common Base, Common Emitter, and Common collector configurations. BJT Specifications. BJT Hybrid Model, h-parameter representation of a transistor, Analysis of single stage transistor amplifier using h-parameters: voltage gain, current gain, Input impedance and Output impedance. Comparison of transistor configurations, Transistor as an amplifier.

Unit IV: Field Effect Transistors **(8 L)**
JFET (Construction, principal of Operation and Volt –Ampere characteristics). Pinch- off voltage-Small signal model of JFET. FET as Voltage variable resistor, Comparison of BJT and FET. MOSFET (Construction, principal of Operation and symbol), MOSFET characteristics in Enhancement and Depletion modes. FET Amplifiers, Effect of real surfaces; Work function difference, Interface charge, Threshold voltage and its control,

Unit V: Transistor Biasing and Stabilisation **(10 L)**
The Operating Point, D.C and A.C Load lines, Need for biasing, Biasing Stability, Self-Biasing or Emitter Bias, Stabilization against Variations in I_{co}, V_{BE}, and β, General Remarks on Collector-Current Stability, Bias Compensation, Thermal Runaway, Thermal Stability,

Text Books:

1. J. Millman, C.C. Halkias, C. Parikh, Integrated Electronics, McGraw Hill, 2nd Edition, 2017.
2. S. Salivahan, N.S. Kumar, Electronic Devices and Circuits, McGraw Hill, 5th Edition, 2022.

Reference Books:

1. A. Malvino, D.J. Bates, Electronic Principles, McGraw Hill, 7th Edition, 2017.
2. R. Boylstead, L. Nashelsky, Electronic Devices and Circuit Theory, Prentice Hall Publications, 11th Edition, 2015.

1.	School offering the course	School of Computing
2.	Course Code	**CSF209**
3.	Course Title	**Digital IC Design**
4.	Credits (L:T:P:C)	3:0:1:4
5.	Contact Hours (L:T:P)	3:0:2
6.	Prerequisites (if any)	CSF104/ECF144
7.	Course Basket	Discipline Core

Course Summary: VLSI design concepts are necessary for all levels of digital computer design from computer architecture down to digital logic gates. This course covers the design and implementation of digital circuits in a modern VLSI process technology. The lab component of the course will focus on using modern computer aided design (CAD) software to design, simulate, and lay out digital circuits.

Course Objective: This course is designed to provide students with in-depth analysis and design capability required for state-of-the-art low-power and high-performance digital integrated circuits.

Course Outcome: At the end of the course, the student will be able to:
CO1: Understand the structure of commercially available digital integrated circuit families.
CO2: Model complex digital systems at several levels of abstractions, behavioral, structural, and rapid system prototyping.
CO3: Analyze and design basic digital circuits with combinatorial and sequential logic circuits.
CO4: Understand the concepts of DFT and BIST.

Curriculum Content

Unit I: The Inverter and Interconnects (8 L)
The Inverter: Resistance and Capacitance and transient response. Dynamic, Short Circuit and Leakage power – Stacking Effect Interconnects: Capacitance, Resistance, Sheet Resistance, Skin depth, Resistance Models, lumped model (C and RC), Propagating delay and rise time, Elmore delay model, Example - Time constant of a RC-wire model.

Unit II: Combinational Circuit Design (7 L)
CMOS gates, Gate sizing, Capacitance estimation, Delay estimation, Logical effort, Path delay optimization, Buffer insertion, Circuit Families, Static CMOS, Ratioed gates, Cascade Voltage Switch Logic (CVSL) & Level Translators, Dynamic circuits, Pass Transistor circuits

Unit III: Sequential Circuits (7 L)
Sequencing Elements, Sequencing Methods; Flip flop, Latch, Delay definitions, Circuit Implementations of Latch/ Flop; Static, Dynamic, Max delay constraints Min delay constraints, Setup and Hold Time measurement. Timing analysis of latch/ flop-based systems.

Unit IV: Adders and Multipliers (8 L)
Adders, Basic terminology, Full adder circuit design, Inverting Adder, Carry Save Adder, Carry Select Adder, Carry Look Ahead Adder Multipliers, Basic Terminology, Booth and Modified Booth Encoding, 2s Complement Arithmetic, Array Multiplier, Carry Save Multipler, Signed multiplication and carry save implementation, Final Addition.

Unit V: DFT and BIST (8 L)
Design for Testability: Introduction, Testability Analysis, DFT Basics, Scan cell design, Scan Architecture Scan design rules, Scan design flow Fault Simulation, Test Generation, Built-In-Self-Test: Test pattern generation, Output response analysis.

Text Books:
1. J.M. Rabaey, A. Chandrakasan, B. Nikolic, Digital Integrated Circuit: A Design Perspective, Pearson, 2nd Edition, 2016.
2. J.F. Wakerlay, Digital Design Principles & Practices, Pearson Education, 5th Edition, 2018.

Reference Books:
1. S.M. Kang, Y. Leblebici, C. Kim, CMOS Digital Integrated Circuits: Analysis and Design, McGraw Hill, 4th Edition, 2016.

1.	School offering the course	School of Computing
2.	Course Code	**CSF302**
3.	Course Title	**Design and Analysis of Algorithms**
4.	Credits (L:T:P:C)	3:0:1:4
5.	Contact Hours (L:T:P)	3:0:2
6.	Prerequisites (if any)	CSF102
7.	Course Basket	Discipline Core

Course Summary: This course gives comprehensive introduction of computer algorithms with their time and space complexity. It provides example algorithms understanding of various categories like Divide & Conquer, Greedy, Dynamic Programming, Backtracking, and Branch & Bound. It introduces the problems that comes under category of P and NP.

Course Objectives: This course aims to provide the knowledge and understanding the various fundamental and advance data structures with their operational algorithms and complexity issues of algorithms. It aims to develop the ability to create algorithms for any task with best complexity.

Course Outcomes: After the study of this course student will be able to:

CO1: Understand and apply new algorithms.

CO2: Analyze complexity issues of algorithms

CO3: Create appropriate algorithm for performing any task.

CO4: Understand the existing and new algorithms in terms of P and NP Class problems.

Curriculum Content

Unit I **(6 L)**

Introduction: Algorithms, Performance Analysis: Space and Time Complexity, Asymptotic Notations- Big Oh, Omega, theta notations, finding complexity of the algorithm, Sorting: Insertion sort, Bubblesort, selection sort, count sort.

Unit II **(8 L)**

Recurrence relation and solution (substitution, recurrence tree and master method). Divide and Conquer: General method, binary search, quick sort, merge sort, heap sort.

Unit III **(8 L)**

Greedy Method: General method, Activity Selection, job scheduling with deadlines, fractional knapsack problem, Minimum cost spanning tree: Kruskal’s and Prim’s, single source shortest path, Huffman tree.

Unit IV **(9 L)**

Dynamic Programming: General Method, 0-1 Knapsack, Matrix chain multiplication, longestsubsequence, all pair shortest paths,

Backtracking: Travelling Salesman Problem, Graph Coloring, n-Queen Problem, Hamiltonian Cyclesand Sum of subsets.

Unit V **(6 L)**

Branch and Bound: Travelling Salesman Problem NP-Hard and NP-Complete problems: Basic Concepts, non-deterministic algorithms, NP-Hard and NP-Complete classes.

Text Books:
1. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, ―Introduction to Algorithms‖, MIT Press;3rd edition, 2010.
2. Ellis Horowitz, SatrajSahni and Rajasekharam, Fundamentals of Computer Algorithms, UniversitiesPress; Second edition, 2008.

Reference Books:
1. R.C.T.Lee, S.S.Tseng, R.C.Chang and T.Tsai, Introduction to Design and Analysis of Algorithms A strategic approach, McGraw-Hill Education (Asia), 2012.
2. Aho, Ullman and Hopcroft, Design and Analysis of algorithms, Pearson Education India; 1st edition2010.
3. Anany Levitin, Introduction to the Design and Analysis of Algorithm‖, Pearson Education India;2nd edition, 2008.

List of Experiments:

S.NO.	EXPERIMENT NAME
1	Program in C to Implement Insertion sort, selection sort
2	Program in C to Implement Quick Sort
3	Program in C to Implement Merge Sort
4	Program in C to Implement Binary Searching, Heap sort
5	Program in C to Implement Activity Selection problem
6	Program in C to Implement job scheduling with deadlines
7	Program in C to Implement fractional knapsack problem
8	Program in C to Implement single source shortest path (Djkstra Algorithm)
9	Program in C to Implement 0-1 Knapsack problem using Dynamic Programming
10	Program in C to Implement all pair shortest path

1.	School offering the course	School of Computing
2.	Course Code	**CSF303**
3.	Course Title	**Computer Networks**
4.	Credits (L:T:P:C)	3:0:1:4
5.	Contact Hours (L:T:P)	3:0:2
6.	Prerequisites (if any)	CSF201
7.	Course Basket	Discipline Core

COURSE SUMMARY

The course is a foundation level course and provides an in-depth description of computer networks. It begins by introducing the fundamentals of data communication and proceeds through the protocol layering architecture. It covers the physical layer by introducing the conversion of the analog and digital signals, transmission impairments, and transmission media. It also includes the data link layer and its services through protocols, network layer, IP address, delivery & forwarding packets, and network-layer protocols. Finally, it describes the transport layer & application layer that includes flow control, error control, congestion control, and application layer protocols like HTTP, FTP, SMTP, etc.

COURSE OBJECTIVES

The main objective of this course is to introduce you the fundamental concept of computer networks, how to build a network, what are the software & hardware requirements, how to analyze a network for performance and quality of service, and how two computers connected to a network communicate with each other.

COURSE OUTCOMES

On successful completion of the course, students will be able to achieve the following:

CO1: Develop an ability to describe what a computer network is and how data communication takes place between two computers connected to a network.
CO2: Understand the protocol layering architecture and the different functions of each layer.
CO3: Develop an ability to describe IPV4 addressing technique, including classful & classless address along with subnetting.
CO4: Develop an ability to analyze a network for their performance, quality of service, and throughput.

CURRICULUM CONTENT

Unit 1: Introduction to Computer Networks (8 L)

Data Communication and Network Fundamentals: Components of a Data Communication System, Data Flow, Computer Network and Internet, Network Topology, Network Models, Network Protocols, The Internet, History of Computer Network and the Internet.
Network Model and Layering Architecture: Network core: Packet Switch and Circuit Switch Network, A Network of Networks, Delay, Loss, and Throughput in Packet-Switched Networks, Protocol Layer and their Service Model: Layered Architecture, OSI and TCP/IP model.

Unit-2: Physical and Data Link Layer (7 L)

Physical Layer: Theoretical basis for communication, guided transmission media, wireless transmission, the public switched telephone networks, mobile telephone system, multiplexing: FDM, WDM, TDM, Transmission Media: Guided and Unguided Media,
Data Link Layer: Introduction to Data Link Layer, Services provided by the Data Link Layer, Error Detection and Correction Techniques, CRC, Checksum, Media Access Control: Random access protocol, Controlled Access Protocol, Ethernet and Ethernet Protocol

Unit 3: Network Layer (8 L)

Introduction to Network Layer, Packet switching at network layer, Network Layer Services, Logical Addressing, IPV4 addresses: classful and classless, Subnetting, Delivery and Forwarding of Packets:
Direct Delivery, Indirect Delivery, Next-hop method, Network Specific Method, Host Specific Method, forwarding with classful and classless addressing, Routing, Structure of a Router and switching techniques, Network Layer Protocols like ARP, RARP, ICMP etc. Unicast Routing Protocol: RIP, OSPF, BGP , Multicast Routing Protocol.

Unit 4: Transport Layer (7 L)

Introduction and Transport Layer Services: Process-to-Process Communication, Encapsulation and Decapsulation, Multiplexing and Demultiplexing, Flow Control, Error Control, Congestion Control, Connection-less and Connection-oriented services, Transport Layer Protocol: Simple protocol, Stop and-wait protocol, Go-back-N protocol, Selective-repeat protocol, TCP and UDP.

Unit 5: Application Layer (8 L)

Introduction to Application Layer, Application Architecture: Client-Sever, Peer-to-Peer, Process Communication, Client-Server communication Interface: Socket, IP, Using the services of Transport Layer, Application Layer Protocols: HTTP, FTP, SMTP, POP, IMAP, DNS.

TEXTBOOK(S)

1. Behrouz Forouzan, Data Communications, and Networking; McGraw Hill Education; 4th Edition (2017).
2. James F. Kurose and Keith W. Ross, Computer Networking: A Top-Down Approach Pearson Education; Sixth edition (2017)

REFERENCE BOOKS

1. Andrews S. Tanenbaum, David J Wetherall; Computer Networks; Pearson Education; 5th Edition, 2013
2. Peterson, Larry L., and Bruce S. Davie. *Computer networks: a systems approach*. Elsevier, 2007.

List of Experiments

S.NO.	EXPERIMENT NAME
1	Simulate a network having two communication node using Cisco packet Tracer.
2	Simulate a network having 4 communication nodes with one switch.
3	Simulate a network having Two subnet using 2 switch, one Router and 6 nodes using Cisco packet tracer
4	Simulate a network using Star Topology Using Cisco packet Tracer.
5	Simulate a network using Bus Topology Using Cisco packet Tracer.
6	Simulate a network using Ring Topology Using Cisco packet Tracer.
7	Simulate a network using Mesh Topology Using Cisco packet Trace.
8	Create a DHCP server using Cisco packet tracer
9	Implement Intra domain and Inter domain routing Protocol using Cisco Packet Tracer.
10	Implement of Bit Stuffing and CRC.

1.	School offering the course	School of Computing
2.	Course Code	**CSF304**
3.	Course Title	**Artificial Intelligence**
4.	Credits (L: T:P:C)	3:0:1:4
5.	Contact Hours (L: T:P)	3:0:2
6.	Prerequisites (if any)	CSF201
7.	Course Basket	Discipline Core

Course Summary: The course will start with a brief introduction to artificial Intelligence. This course includes basic AI search techniques like A*, BFS, DFS. Introduction to Prolog is also important part of the content. Knowledge Representation, Reasoning Planning and Learning being requirement for development of expert system is also part of this course.

Course Objectives: The course is proposed to teach concepts of Artificial Intelligence. The subject will provide the foundations for AI problem solving techniques and knowledge representation formalisms.

Course Outcomes: On successful completion of the course, students will be able to:
CO1: Identify and formulate appropriate AI methods for solving a problem.
CO2: Apply AI algorithms.
CO3: Compare different AI algorithms in terms of design issues, computational complexity and assumptions.
CO4: Utilize the concepts of AI for real world problem solving.

Curriculum Content

Unit I: Introduction **(6 L)**
Introduction- Definitions, History, Characteristics, Applications, Intelligent Agents, Agent Environment, Types of Intelligent Agents, Environment Types. Introduction to python, Basic syntax, Basic operations, Loops, Data types, Functions.

Unit II: Problem solving and search **(9 L)**
Problem solving techniques, Search Terminologies, Properties of Search Algorithms, Search Algorithms- Uninformed Search, Informed Search, Minimax Search, Constraint satisfaction problem.

Unit III: Knowledge Representation **(7 L)**
Knowledge Representation- Introduction, Approaches and Issues in Knowledge Representation, Propositional Logic and Inference, First-Order Logic and Inference, Unification and Resolution, Expert Systems.

Unit IV: Reasoning **(8 L)**
Reasoning- Introduction, Types of Reasoning, Probabilistic Reasoning, Probabilistic Graphical Models, Certainty factors and Rule Based Systems, Introduction to Fuzzy Reasoning.

Unit V: Planning and Learning **(7 L)**
Planning and Learning- Introduction to Planning, Types-Conditional, Continuous, Multi-Agent. Introduction to Learning, Categories of Learning, Inductive Learning, Supervised and Unsupervised & Reinforcement Learning, Basic Introduction to Neural Net Learning.

TEXT BOOKS:

1. Stuart Russell and Peter Norvig, “Artificial Intelligence: A Modern Approach”, Pearson Education India; 3rd edition (2015)

2. Elaine Rich, Kevin Knight and Shivashankar B. Nair, “Artificial Intelligence”, McGraw-Hill Education; 3rd edition (2017).

3. Nils J. Nilsson, “Artificial Intelligence-A New Synthesis", Morgan Kaufmann Publishers, Inc.; 1st edition (1998).

REFERENCES:

1. Dan W. Patterson, “Introduction to Artificial Intelligence and Expert Systems”, Pearson Education India; 1st edition (2015)

List of Experiments

S.NO.	EXPERIMENT NAME
1	Write a program to solve water jug problem.
2	Write a program to implement Breadth-first Search algorithm.
3	Write a program to implement Depth-first Search algorithm.
4	Write a program to implement Best-first Search Algorithm
5	Write a program to implement A* Search Algorithm
6	Write a program to print the root node using mini max algorithm.
7	Write a program to implement constraint Satisfaction problem.

1.	School offering the course	School of Computing
2.	Course Code	**CSF310**
3.	Course Title	**VLSI Circuits Technologies**
4.	Credits (L:T:P:C)	3:0:1:4
5.	Contact Hours (L: T:P)	3:0:2
6.	Prerequisites (if any)	CSF208
7.	Course Basket	Discipline Core

Course Summary: The course covers the principles and techniques of both MOS based digital and analog circuit design, connecting digital circuits, logic design, and analog components with the fundamental device physics, processing techniques and transistor level characteristics of Silicon integrated circuits, both in theoretical and practical aspects.

Course Objective: The course aims to help students understand the basic theory of MOS transistors and physics behind modelling the MOSFETs and CMOS circuits.

Course Outcome: At the end of the course, the student will be able to:
CO1: Understand MOS transistor theory.
CO2: Design digital systems using MOS circuits.
CO3: Learn Layout, Stick diagrams, and Switching characteristics of inverters.
CO4: Design combinational circuits.

Curriculum Content

Unit I: Introduction **(5 L)**
Introduction to VLSI technology, VLSI design flow, Digital Design Cycle, Physical Design Cycle.

Unit II: MOS Transistor **(7 L)**
MOS Structure and its operation, I-V Characteristics, Scaling Theory and Limitations of Scaling, channel length modulation, body effect, biasing of MOSFETs, capacitances in MOS, VLSI circuit and system representation.

Unit III: NMOS and CMOS Process Technology **(7 L)**
Evolution of ICs. Masking sequence of NMOS and CMOS Structures, Latch up in CMOS, Electrical Design Rules, Stick Diagram, Layout Design.

Unit IV: Combinational Circuits **(10 L)**
Logic gate characteristics, Design of MOS inverter with different loads, Determination of pull up and pull down ratio for an nMOS inverter driven by another nMOS inverter, Design of W/L, power dissipation, propagation delay, and noise margin analysis, CMOS Logic Circuits, CMOS logic Styles, Realization of simple gates.

Unit V: Operation of MOS Circuits **(8 L)**
Behavior of MOS Circuits at DC, MOS as an Amplifier, Calculation of the DC Bias Point, Voltage Gain, Transconductance, T-Equivalent Circuit Model, Biasing of Discrete MOS Amplifiers and Integrated Circuit MOS Amplifiers.

Text Books:
1. A.S. Sedra, K.C. Smith, Microelectronic Circuits: Theory and Applications, NY: Oxford Press, 7th Edition, 2017.
2. S.M. Kang, Y. Leblebici, C. Kim, CMOS Digital Integrated Circuits: Analysis and Design, McGraw Hill, 4th Edition, 2016.

Reference Books:

1. R. Gieger, P.Allen, N. Strader, VLSI Design Techniques for Analog and Digital Circuits, McGraw Hill, reprint, 2010.
2. N.H.E. West, K. Eshraghian, Principles of CMOS VLSI Design, Pearson Education, 2nd Edition, 2001.
3. J.P. Uyemura, Introduction to VLSI Circuits and Systems, John Wiley & Sons, 2002.

1.	School offering the course	School of Computing
2.	Course Code	**CSF311**
3.	Course Title	**Microcontrollers and Embedded Systems**
4.	Credits (L:T:P:C)	3:0:1:4
5.	Contact Hours (L: T:P)	3:0:2
6.	Prerequisites (if any)	CSF201
7.	Course Basket	Discipline Core

Course Summary: Microcontroller is the heart of the programmable devices. Embedded systems and Micro-controllers have also assumed a great significance in the electronic and consumer goods industry and are a very vital field. Students of electronics and related engineering branches often use microcontroller to introduce programmable control in their projects, automation and fault finding in industry. The subject aims expose students to the embedded systems besides giving them adequate knowledge of micro controllers.

Course Objective: The course aims to provide the students with basic knowledge of architectural features of ARM microcontroller that can be applied to program the ARM for different applications. The course will also introduce to the students the applicability of embedded systems and real time operating system used in embedded systems.

Course Outcome: At the end of the course, the student will be able to:
CO1: Understand basics of microprocessors and microcontrollers.
CO2: Describe the architectural features and instructions of ARM microcontroller
CO3: Learn to interface external devices and I/O with ARM microcontroller.
CO4: Comprehend the real time operating system used for the embedded system

Curriculum Content

Unit I: Basic Introduction (8 L)
Microprocessors versus Microcontrollers, ARM Embedded Systems: The RISC design philosophy, The ARM Design Philosophy, Embedded System Hardware, Embedded System Software. ARM Processor Fundamentals: Registers, Current Program Status Register, Pipeline, Exceptions, Interrupts, and the Vector Table, Core Extensions

Unit II: ARM Instruction Set (8 L)
Data Processing Instructions, Programme Instructions, Software Interrupt Instructions, Program Status Register Instructions, Coprocessor Instructions, Loading Constants ARM programming using Assembly language: Writing Assembly code, Profiling and cycle counting, instruction scheduling, Register Allocation, Conditional Execution, Looping Constructs

Unit III: Embedded System Components (8 L)
Embedded Vs General computing system, Classification of Embedded systems, Major applications and purpose of ES. Core of an Embedded System including all types of processor/controller, Memory, Sensors, Actuators, LED, 7 segment LED display, stepper motor, Keyboard, Push button switch, Communication Interface (onboard and external types), Embedded firmware

Unit IV: Embedded System Design Concepts (8 L)
Characteristics and Quality Attributes of Embedded Systems, Operational and non-operational quality attributes, Embedded Systems-Application and Domain specific, Hardware Software Co-Design and Program Modeling, embedded firmware design and development

Unit V: RTOS based Embedded System Design (8 L)
Operating System basics, Types of operating systems, Task, process and threads, Multiprocessing and Multitasking, Task scheduling, Task Communication, Task synchronization, Task synchronization issues-

Racing and Deadlock, Embedded system Development Environment, Disassembler/decompiler, simulator, emulator and debugging techniques.

Text Books:

1. K.V. Shibu, Introduction to Embedded Systems, McGraw Hill, 2nd Edition, 2017.
2. G.H. Raghunandan, Microcontroller (ARM) and Embedded Systems, Cengage Learning, 2020.
3. A.N. Sloss, D. Symes, C. Wright, ARM system developers guide, Elsevier, 2008.

Reference Books:

1. L.B. Das, Embedded Systems: An integrated approach, Pearson education, 2012.
2. D. Lacamera, Embedded Systems Architecture, Packt Publishing, 2018.
3. B. Amos, Hands-on RTOS with Microcontrollers, Packt Publishing, 2020.

Chip Design Specialized Track

S. No.	Course Code	Course Title	L T P C (Contact Hours)
1.	CSF331	IC Fabrication Technologies	3 0 0 3
2.	CSF332	CMOS Analog VLSI Design	2 0 2 3
3.	CSF431	Emerging Technologies in Chip Design	3 0 0 3
4.	CSF432	FPGA Systems	2 0 2 3
5.	CSF442	Robotics	2 0 2 3
6.	CSF444	Internet of Things	2 0 2 3

1.	School offering the course	School of Computing
2.	Course Code	**CSF331**
3.	Course Title	**IC Fabrication Technologies**
4.	Credits (L:T:P:C)	3:0:0:3
5.	Contact Hours (L: T:P)	3:0:0
6.	Prerequisites (if any)	None
7.	Course Basket	Discipline Elective

Course Summary: The semiconductor industry provides critical enabling technology for many products and fields and has seen periods of rapid growth. This course provides an overview of device and circuit design and the processing steps for semiconductor device fabrication. The course is intended to provide students with fundamental knowledge in device and integrated circuits (IC's) fabrication.

Course Objective: The objective of the course is to introduce the basic principles involved in IC fabrication and discuss processing technology. Fabrication of integrated circuits is a joint venture by electrical engineers, chemical engineers, materials scientists and physicists. This interdisciplinary course builds bridges across various disciplines.

Course Outcome: At the end of the course, the student will be able to:

CO1: Learn about the basic concepts and laws behind IC design and fabrication.

CO2: Acquire knowledge about crystal growth and wafer preparation techniques.

CO3: Learn about different fabrication process used in ICs industry.

CO4: Understand the various packaging techniques.

Curriculum Content

Unit I: Introduction **(6 L)**

History of semiconductor devices, Moore's law, feature size and minimum feature size trend, identifying the unit IC fabrication steps and Cleanroom technology.

Unit II: Crystal Wafer Growth **(10 L)**

Understanding the Silicon crystal structure. Various methods for raw material to Single crystal growth of Si, Dopant distribution, Crystal growth using Czochralski's method, Float Zone method and Bridgeman technique, Zone refining, characteristics and crystal evaluation, wafer preparation. Wafer Shaping operations, Slicing, polishing and etching, Crystalline defects and their effects.

Unit III: Fabrication Steps **(9 L)**

Oxidation: Oxidation techniques and systems, growth kinetics, various applications of SiO_2. **Epitaxy:** Importance of epitaxial layer growth, Epitaxial growth processes, equipment and doping and autodoping. **Diffusion:** Diffusion models and types, equipment and applications. **Subsequent process:** Ion-implantation, Photolithography, Electron beam and X-Ray lithography, Different printing techniques, Photo resist, dry and wet Etching, Metallization.

Unit IV: MOSFET Technology **(6 L)**

Design of junction diode, Transistor, FET and MOSFETs Polysilicon gates and Well Structures

Unit V: Packaging of ICs **(6 L)**

Mountings in packages using Dual-in-line (DIP) or TO packages. Packages using surface-mount-technology (SMT).

Text Books:
1. S.M. Sze, VLSI Technology, McGraw Hill, 2nd Edition, 2017.
2. D.A. Pucknel and K. Eshraghian, Basic VLSI Design, Prentice Hall, 3rd Edition.

Reference Books:
1. G.S. May, S.M. Sze, Fundamentals of Semiconductor Fabrication, Wiley, 2011.

1.	School offering the course	School of Computing
2.	Course Code	**CSF332**
3.	Course Title	**CMOS Analog VLSI Design**
4.	Credits (L:T:P:C)	2:0:1:3
5.	Contact Hours (L:T:P)	2:0:2
6.	Prerequisites (if any)	None
7.	Course Basket	Discipline Elective

Course Summary: Current system design practices in industry require the use of the intelligent macro model approach to study the input/output behavior of any subsystem. There is a greater need in the industry for application engineers who can judiciously use the integrated circuits (ICs) to design efficient systems. This course acquaints the students with general analog principles and design methodologies using practical devices and applications.

Course Objective: The objective of this course is to introduce to the students the basic building blocks of linear ICs and expose them to the design of analog systems using CMOS.

Course Outcome: At the end of the course, the student will be able to:
CO1: Learn the basics of analog VLSI design using CMOS.
CO2: Understand the basics of MOS amplifiers.
CO3: Understand the working and concepts related to CMOS Op-amps and differential amplifiers
CO4: Learn the basic concepts related to oscillators.

Curriculum Content

Unit I: Introduction to CMOS Analog VLSI Design **(4 L)**
Introduction to CMOS Analog VLSI Design, MOS Fundamentals, Analog MOS Models

Unit II: MOS Amplifiers **(7 L)**
Basic of MOS Amplifier, Cascode Amplifier, Types of MOSFET Amplifier, Differential Amplifiers, Current Sources, Frequency Response of Amplifier

Unit III: CMOS Op-Amps **(7 L)**
Basic of CMOS OPAMP, OPAMP Design Issues, OPAMP Design, Operational Transconductance Amplifier, OTA Operation Transconductance Amplifier and Application.

Unit IV: Differential Amplifier **(7 L)**
Fully Differential Amplifier, CMFB, Noise, Types of Noise, MOS Models for Noise, Noise Evaluation of circuits, Layout of Analog Circuit.

Unit V: Oscillators **(3 L)**
Oscillators, Ring Oscillators, VCO, LC Oscillators, Cross coupled oscillators, Single port oscillators, PLL.

Text Books:
1. B. Razavi, Design of Analog CMOS Integrated Circuits, McGraw Hill, 2nd Edition, 2017.
2. P. Allen and D. Holmberg, CMOS Analog Circuit Design, Oxford University Press, 3rd Edition, 2016.

Reference Books:
1. Paul R. Gray, Paul J. Hurst, S. Lewis and R. G. Meyer, Analysis and Design of Analog Integrated Circuits, Wiley India, 5th Edition, 2010.
2. David A. Johns, Ken Martin, "Analog Integrated Circuit Design", Wiley Student Edition, 2013.

1.	School offering the course	School of Computing
2.	Course Code	**CSF431**
3.	Course Title	**Emerging Technologies in Chip Design**
4.	Credits (L:T:P:C)	3:0:0:3
5.	Contact Hours (L: T:P)	3:0:0
6.	Prerequisites (if any)	None
7.	Course Basket	Discipline Elective

Course Summary: VLSI, along with embedded software development and hardware/board design, is at the heart of the chip design industry. The course will cover new-age technologies like open source processor architecture (RISC-V), Neuromorphic computing, Flexible electronics, Electric vehicles power technologies.

Course Objective: The objective of the course is to introduce the students to latest trends and technologies in electronics and chip design industry.

Course Outcome: At the end of the course, the student will be able to:

CO1: Understand the basic concepts of open source processor architecture.

CO2: Learn the basic concepts of flexible and printable electronics and understand the basic concepts for integration of devices on flexible platforms.

CO3: Understanding the latest trends in computing architectures that mimic biological synapse.
CO4: Learn the concepts of power semiconductors and their usage.

Curriculum Content

Unit I: Introduction to RISC-V architecture **(8 L)**
Basic introduction, History, Developing RISC-V.

Unit II: Flexible and printable electronics **(8 L)**
Introduction to Flexible & Printable electronics- Historical background - Materials, devices, systems, applications - Fabrication techniques - Unique aspects, status in the field and trends, Organic devices on flexible substrate, Sensors and biosensors.

Unit III: Neuromorphic computing **(8 L)**
Background, The neuron and the physical models, Characteristics, Challenges, Emerging Memory Devices for Neuromorphic Systems

Unit IV: New age power semiconductors and their utility **(8 L)**
Advanced materials, neural network chips for edge computing, Automotive chips.

Text Books:
Study material will be provided by instructors.

1.	School offering the course	School of Computing
2.	Course Code	**CSF432**
3.	Course Title	**FPGA Systems**
4.	Credits (L:T:P:C)	2:0:1:3
5.	Contact Hours (L: T:P)	2:0:2
6.	Prerequisites (if any)	None
7.	Course Basket	Discipline Elective

Course Summary: The semiconductor industry provides critical enabling technology for many products and fields and has seen periods of rapid growth. This course provides an overview of device and circuit design and the processing steps for semiconductor device fabrication. The course is intended to provide students with fundamental knowledge in device and integrated circuits (IC's) fabrication.

Course Objective: This course covers the different types of programming technologies and logic devices, and the architecture of different types of FPGA. The primary goal is to provide in depth understanding of system design. The course enables students to apply their knowledge for the design of advanced digital hardware systems with help of FPGA tools.

Course Outcome: At the end of the course, the student will be able to:

CO1: Explain the concept and basic structures of Field Programmable Gate Array (FPGA), and techniques to implement programmable logic circuits using typical FPGA design flow.

CO2: Analyze and break any given algorithm to sub-modules and implement the same on FPGA using HDLs and various IP cores.

CO3: Design and verify integrated circuits chips.

CO4: Design and implement different Field Programmable Gate Array (FPGA) architectures and their applications to real life.

Curriculum Content

Unit I: Introduction to Verilog
Data types, operators, Multiplexers, decoders, adders, Subtractor, de-multiplexers, flip-flops, counters, state machine, ALU with adder/Subtractor and shifter, multiply and divide hardware synthesis, memory, using a test bench for verification and simulation. Concept of state machine and implementation, simulation synthesis, verification, FSM Types, Implementation in HDLs, a simple computer design, other design examples and case studies

Unit II: PLDs and PALs
Types of Programmable Logic Devices, Combinational Logic Examples, PROM - Fixed AND Array and Programmable OR Array, Implementation of Functions using PROM, PLA - Programmable Logic Array (PLA) – Implementation Examples. PAL - Programmable Array Logic, Comparison of PROM, PLA and PAL, Implementation of a Function using PAL, Types of PAL Outputs, Device Examples.

Unit III: Overview of FPGA Technology
From discrete logic to FPGAs, flexibility and functionality, FPGA vs Programmable DSPs, FPGA technology - roadmap, clocking, data and sample rates, slices and configurable logic blocks, memory and registers, performance ratings, families DSP and FPGAs: FPGA elements for DSP algorithms, FPGA- Based System Design Examples.

Unit IV: Embedded systems & FPGAs
FPGA as a systems on chip platform, FPGA on-chip network standards, FPGAs as custom microcontroller and hybrid DSP microcontroller devices, Multiple cock domains on chip, Multi FPGA systems, multiple-clock domains, program and data memory, SRAM and DRAM.

Unit V: System Design Examples using FPGA Board

Design Applications using FPGA Board - Traffic Light Controller and Real Time CLO sck, XSV FPGA Board Features, Testing of FPGA Board, Setting the XSV Board CLO sck Oscillator Frequency, Downloading Configuration Bit Streams.

Text Books:

1. C. Unsalan, B. Tar, Digital System Design with FPGA: Implementation using Verilog and VHDL, McGraw Hill, 2017.
2. M. Morris Mano, M.D. Ciletti, Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog, Pearson, 6th Edition, 2018.

Reference Books:

1. J.P. Deschamps, Synthesis of Arithmetic Circuits: FPGA, ASIC and Embedded Systems, Wiley, 2006.
2. W. Wolf, FPGA based System Design, Prentice Hall, 2004

1. School offering the course	School of Computing
2. Course Code	**CSF442**
3. Course Title	**Robotics**
4. Credits (L:T:P:C)	2:0:1:3
5. Contact Hours (L:T:P)	2:0:2
6. Prerequisites (if any)	None
7. Course Basket	Discipline Elective

Course Summary: The course is proposed to teach the students the concepts of Robotics. Students will develop understanding of the different principles of sensors and methods of robot.

Course Objectives: The course will start with a brief introduction to robots and robotics. Various methods of robot teaching will be explained with some suitable examples. The working principles of various sensors used in robots will be explained in detail. The principles of motion planning algorithms will be explained in detail.

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

CO1: Learn how to keep robots in modern industries.

CO2: Learn how to apply robots in different areas (space, medical, manufacturing etc.).

CO3: Students will have brief idea of different components of robots system and their working principle.

CO4: Understand the notion of an agent, how agents are distinct from other software.

CO5: Understand the key issues in designing societies of agents that can effectively cooperate in order to solve problems, including an understanding of the key types of multi-agent interactions possible in such systems.

CO6: Understand the main application areas of agent-based solutions, and be able to develop a meaningful agent-based system using a contemporary agent development platform.

Curriculum Content

Unit I: Introduction to Robotics **(5 L)**

Components, Classification, Characteristics and Applications of Robots. Material transfer, Machine loading/unloading, Processing operation, Assembly and Inspection, Feature Application.

Unit II: Drive Systems, Actuators and Sensors: **(5 L)**

Actuators: Characteristics of Actuating Systems, Actuating Devices and Control. Sensors: Sensor Characteristics, Description of Different Sensors, Touch sensors, Tactile sensor, Proximity and range sensors, Robotic vision sensor, Force sensor, Light sensors, Pressure sensors.

Unit III: Basic Concepts of Artificial Intelligence **(5 L)**

Concepts of AI, AI Problems, techniques, Characteristics & Applications, AI versus Natural intelligence, Problem representation in AI, Problem-solution Techniques. Elements of Knowledge representation: Logic, Production Systems, Semantic Networks, Expert Systems. Defining the Problem as State Space Search, Production Systems, Production Systems, Issues in the Design of SearchPrograms, DFS & BFS Techniques

Unit 4: Intelligent Agents and Multi-agent System **(6 L)**

Design of intelligent agents - reasoning agents, agents as reactive systems; hybrid agents; layered agents, Multi-Agent Systems: Classifying multi-agent interactions-cooperative versus non-cooperative; zero-sum and other interactions; Cooperation, Prisoner's dilemma and Axelrod'sexperiments; Interactions between self-interested agents: auctions & voting systems: negotiation; Interactions between benevolent agents: cooperative distributed problem solving, partial global planning; coherence and coordination; Interaction languages and protocols: speech acts, KQML/KIF,the FIPA framework.

UNIT V: Multi-robot representations and Task Planning **(6 L)**
Task-Level Programming, Uncertainty, Configuration Space, Gross-Motion Planning, Grasp Planning,Fine Motion Planning, Task Planning Problem.**:** control architectures, simulation environments, and testbeds. Integration of assorted sensors (IR, Potentiometer, strain gages etc.), micro controllers and ROS(Robot Operating System) in a robotic system.

Text Books:
1. Robert J. Schilling, Fundamentals of Robotics Analysis and Control, PHI, 5th Edition, 2012.
2. Saeed B. Niku, Introduction to Robotics Analysis, Systems, Applications, Prentice Hall, 2nd Edition, 2014.

Reference Books:
1. Michael Wooldridge, An Introduction to Multi Agent Systems, Wiley, 2014
2. J.J. Craig, “Introduction to Robotics: Mechanics and Control, Addison- Wesley, 3rd edition, 2004.

1.	School offering the course	School of Computing
2.	Course Code	**CSF444**
3.	Course Title	**Internet of Things**
4.	Credits (L:T:P:C)	2:0:1:3
5.	Contact Hours (L:T:P)	2:0:2
6.	Prerequisites (if any)	None
7.	Course Basket	Discipline Elective

Course Summary: In this course students will be introduced to fundamental and architectural concepts of IoT systems, various kinds of communication using system-on-chip devices and building IoT prototypes. Students will learn how to create an end-to-end system by connecting to IoT cloud, perform IoT Analytics and understand cloud security.

Course Objectives: The objective of this course is to provide both conceptual and hands-on knowledge to students for IoT systems. Students will learn how to build and use end-to-end IoT systems, perform analytics on the data collected and understand security aspects of an IoT system.

Course Outcomes: On successful completion of the course, students will be able to:
CO1: Understand fundamental concepts and building blocks of an IoT system.
CO2: Understand and implement IoT prototypes using system-on-chip devices.
CO3: Understand and develop end-to-end systems by syncing with Cloud.
CO4: Understand security aspects of an IoT system.

Curriculum Content

Unit I **(3 L)**
Introduction, IoT Architecture, Sensing, Communication and Actuation, Hardware and Software setup.

Unit II **(15 L)**
GPIO pins setup and programming, Serial Communication in IoT, SPI and I2C in IoT.

Unit III **(3 L)**
Data transmission in Cloud, IoT Analytics and Visualization

Unit IV **(3 L)**
IoT Security, IoT Project execution and demonstration.

Text Books:
Instructors will provide reading materials.