DIT UNIVERSITY
Dehradun

Detailed Course Structure & Syllabus

of

B.Tech. – Information Technology
(Fully Flexible Choice Based Credit System)

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

Course Structure & Syllabus of B.Tech. – IT
Applicable for Batch: 2024-28

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 DisciplineElective 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 throughеither 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. **IT is not having any specialized track. Also, specialization certificate is not provided.**

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 largenumber 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 popular languages 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 andChemistry. These courses are of 5 credits each.
3. **Core Mathematics:** This basket includes courses from Mathematics department, crafted forEngineering 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.

Course Structure & Syllabus of B.Tech. – IT
Applicable for Batch: 2024-28

Structure of the B.Tech. FFCBCS Program in 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 Elective: Choose any 3 more ES courses	20	4	5
Discipline Core (DC) Core: CO, DM, OOPJ, OS, DBMS, DAA, CN, SE, TOC, AI, WTA, ECS Elective: None	48	4	12
Discipline Elective (DE) Core: None Elective: Choose any 6 courses	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.*

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

Course Structure & Syllabus of B.Tech. – IT
Applicable for Batch: 2024-28

DIT University 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**

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

Course Structure & Syllabus of B.Tech. – IT
Applicable for Batch: 2024-28

Course Baskets: University FFCBCS Baskets (other than DC/DE) for B.Tech.Programs.
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
MEF102	Engineering Graphics	2	0	4	4
MEF103	Engineering Mechanics	2	1	2	4
MEF106	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	0	2	4
ECF144	Digital Electronics and Applications	3	0	2	4
CEF101	Civil Engineering Materials	3	1	0	4

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

	Skill Enhancement (min 8 credits to be taken)				
	Name of Courses	L	T	P	C
CSFXXX	Technical Training 1	0	0	4	2
CSFXXX	Technical Training 2	0	0	4	2
CSFXXX	Value Added Training 1	0	0	4	2
CSFXXX	Value Added Training 2	0	0	4	2
SWAYXXX	MOOCS Courses (as advised by the departments)	2	0	0	0
	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 Structure & Syllabus of B.Tech. – IT
Applicable for Batch: 2024-28

Course Baskets: B.Tech. IT FFCBCS DC Basket and IT DE courses will be chosen from these 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
CSF207	Object Oriented Programming with Java	CSF101	3	0	2	4
CSF204	Operating Systems	CSF201	3	0	2	4
CSF205	Database Management Systems	CSF101	3	0	2	4
CSF315	Essentials of Cyber Security	None	3	0	2	4
CSF301	Software Engineering	CSF102	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
CSF309	Theory of Computation	CSF202	3	1	0	4
CSF416	Web Technologies and Applications	None	3	0	2	4
Discipline Electives (min 18 credits to be taken) Tentative List (* Pending Recommendation from BoS)						
	Name of Courses		L	T	P	C
CSF341	R Programming		2	0	2	3
CSF348	Mobile Application Programming using Android		2	0	2	3
CSF349	Cloud Computing		2	0	2	3
CSF355	Cyber Crime & Investigation		2	0	2	3
CSF345	Introduction to Data Science		2	0	2	3
CSF445	Mobile & Wireless Network Security		2	0	2	3
CSF361	Introduction to Blockchain Technologies		2	1	0	3
CSF364	Container Technologies		2	0	2	3
CSF346	Data Mining and Data Warehousing		2	0	2	3
CSF373	Server Side Engineering		2	0	2	3
CSF363	Blockchain Ecosystems & Governance		2	1	0	3
CSF446	Ethical Hacking & Digital Forensics		2	0	2	3

Course Structure & Syllabus of B.Tech. – IT
Applicable for Batch: 2024-28

Abbreviations

1	PSP	Problem Solving and Programming
2	DS	Data Structures
3	COA	Computer Organization & Architecture
4	OS	Operating System
5	DAA	Design and Analysis of Algorithm
6	AI	Artificial Intelligence
7	DM	Discrete Mathematics
8	DBMS	Data Base Management Systems
9	SE	Software Engineering
10	TOC	Theory of Computation
11	CN	Computer Networks
12	OOPJ	Object Oriented Programming with Java
13	ECS	Essentials of Cyber Security
14	WTA	Web Technologies and Applications

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

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

1. School offering the course	School of Computing
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
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	**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)	None
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 achieve the following:

CO1: This will help the students to be familiarized with the hardware components and concepts related to the control design, data representation and evaluation process of different arithmetic operations.

CO2: This will help the students to be familiarized with CPU organization addressing modes, different types of instruction formats.

CO3: The student will be able to learn the hardware components and concepts related to the input/output and memory organization.

CO4: Students will be able to get the theoretical concept of parallel processing and different types of multiprocessor's interconnection structures.

CURRICULUM CONTENT

Unit 1: 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 2: Computer Arithmetic (8 L)

Introduction, Addition and Subtraction Algorithms, Multiplication and Division Algorithms, Floating Point Arithmetic Operation, IEEE Format for Floating Point Numbers.

Unit 3 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-4 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- 5: 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.

REFERENCES :

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 achieve the following:

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 edition2017.
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	**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	**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 achieve the following:

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 eg. Process scheduling and memory management algorithms.

CO4: Categorize the operating system's resource management techniques, deadlock management techniques, memory management techniques.

CURRICULUM CONTENT

Unit 1: 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 2: 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 3: 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 4: 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- 5: 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.

REFERENCES:

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 System**
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 1: 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 2: 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 3 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-4 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- 5: 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

REFERENCES :

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.

School offering the course	School of Computing
2. Course Code	**CSF315**
3. Course Title	**Essentials of Cyber Security**
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	Discipline Core

COURSE OUTLINE:

This course aims to give an outline of cyber security. The course will equip students with a vibrant view of the existing cyber security landscape considering not only technical measures and defenses, but also the other theme areas including legal, management, crime, risk, social and human factors.

COURSE OBJECTIVE:

To understand the crucial necessity of cyber security in computer systems, networks and enlighten numerous threat scenarios. To understand the well-known cyber-attack events, clarify the attack scenarios, and enlighten mitigation techniques. To understand the variance between Systems Cyber Security, Network Cyber Security, and cryptography, crypto-protocols etc. To analyses the cyber threats to critical structures.

COURSE OUTCOMES:

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

CO1: Understand the cyber threat landscape, both in terms of recent developing issues and those issues which persist over time.

CO2: Outline the roles and effects of governments, commercial and other organisations, citizens and criminals in cyber security affairs.

CO3: Analyze the general values and policies that can be functional to systems to make them more vigorous to attack.

CO4: Choose key factors in cyber security from different corrective views including computer science, management, law, criminology

CURRICULUM CONTENT

UNIT-1: INTRODUCTION TO CYBER SECURITY (8L)

Overview of Cyber Security, Cyber Threats & Crime, Cyber Espionage, Internet Governance, Challenges and Constraints, necessity for a Comprehensive Cyber Security Policy, necessity for a Nodal Authority, necessity for an International convention on Cyberspace.

UNIT-2: SECURITY THREATS AND VULNERABILITIES (8L)

Overview, vulnerabilities in software, Intrusion, Physical Theft, Abuse of Privileges, Malware infection, System administration, Complex Network Architectures, Open Access to Organizational Data, Weak Authentication, Unprotected Broadband communications, Poor Cyber Security Awareness.

UNIT -3: SECURITY PRACTICES & SECURITY SAFEGUARDS (7L)

Security Practices: Security Management, Security Policy, Risk Management, Information Classification Process, Security Procedures and Guidelines, Business Continuity and Disaster Recovery.

Security Safeguards: Overview, Access control, Audit, Authentication, Biometrics, Cryptography, Deception, Denial of Service Filters, Ethical Hacking, Firewalls, Intrusion Detection Systems, Response, Scanning, Anti-Malware software.

UNIT -4: INTRUSION DETECTION &SECURING WEB (7L)

Intrusion detection and Prevention Techniques, Basic security for HTTP Applications and Services, Basic Security for SOAP Services, Identity Management and Web Services, Authorization Patterns, Security Considerations, Challenges, Network based Intrusion detection & Prevention Systems.

UNIT -5: SECURITY LAWS, STANDARDS & FORENSIC (7L)

Security Laws & Standards: Cyber Security Regulations, Roles of International Law, the state and Private Sector in Cyberspace, Cyber Security Standards. The INDIAN Cyberspace, National Cyber Security Policy,

Cyber Forensic: Digital Forensics Essentials: Principles of Digital Forensics and its challenges, Steps taken by computer forensic specialists, Benefits of Professional Forensic Methodology, Industry standards and reporting.

TEXT BOOKS

1. Micki Krause, Harold F. Tipton, ―Handbook of Information Security Management‖, Vol 1-3, CRC Press LLC, 2004.

REFERENCES

1. Bill Nelson, ―Computer Forensics and Investigations‖, Cengage Learning, India Edition, 2016.
2. Matt Bishop ―Computer Security Art and Science‖, Pearson/PHI, 2002.

1.	School offering the course	School of Computing
2.	Course Code	**CSF301**
3.	Course Title	**Software Engineering**
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

Software Engineering (SE) comprises the core principles consistent in software construction and maintenance: fundamental software processes and life-cycles, mathematical foundations of software engineering, requirements analysis, software engineering methodologies and standard notations, principles of software architecture and re-use, software quality frameworks and validation, software development, and maintenance environments and tools. An introduction to object-oriented software development process and design.

COURSE OBJECTIVES

It is providing the idea of decomposing the given problem into Analysis, Design, Implementation, Testing and Maintenance phases. To provide an idea of using various process models in the software industry according to given circumstances. Understanding of the role of project management including planning, scheduling, risk management, etc.

COURSE OUTCOMES

On successful completion of the course, students will be able to achieve the following:
CO1: Summarize the software requirement specifications and the SRS documents.
CO2: Examine the various design and development solutions with proper analysis.
CO3: Demonstrate the competence in communication, planning, analysis, design, construction, and development of software as per the requirements.
CO4: demonstrate the use of modern engineering tools necessary for software project management, time management and software reuse

CURRICULUM CONTENT

UNIT 1 **(7 L)**

Introduction to Software Engineering: Software Characteristics, Software Crisis, Software Engineering Processes, Software Development Life Cycle (SDLC) Models: Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models.

UNIT 2 **(8 L)**

Software Requirements: Functional and non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document.
Requirements engineering process: Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management.
System models: **Context Models, Behavioral models, Data models, Object models, structured methods**.

UNIT 3 **(8 L)**

Design Engineering: Design process and Design quality, Design concepts, the design model, pattern based software design.
Creating an architectural design: software architecture, Data design, Architectural styles and patterns, Architectural Design, assessing alternative architectural designs, mapping data flow into software architecture.
Modeling component-level design: Designing class-based components, **conducting component-level design, object constraint language, designing conventional components.**

UNIT 4 **(9 L)**

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, System testing, **the art of Debugging**.

Product metrics: Software Quality, Frame work for Product metrics, Metrics for Analysis Model, Metrics for Design Model, Metrics for source code, Metrics for testing, Metrics for maintenance. Software Measurement, Metrics for software quality

UNIT 5 **(8 L)**

Risk management: Reactive vs Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM.

Quality Management: Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews, Statistical Software Quality Assurance, Software reliability, **The ISO 9000 quality standards.**

TEXTBOOKS

1. R. S. Pressman, ―Software Engineering – A practitioner ‘s approach‖, McGraw Hill Education; 7 Edition (2009)
2. K.K. Aggarwal & Yogesh Singh, ―Software Engineering‖, New Age International, 2nd Ed. 2006.
3. Pankaj Jalote, Software Engineering, Wiley India, 2010

REFERENCE BOOKS

1. Rajib Mall, Fundamentals of Software Engineering, PHI Publication, 4th Edition, 2014.
2. Ian Sommerville, Software Engineering, Addison Wesley, 10th Edition, 2015
3. James Peter, W Pedrycz, ―Software Engineering‖, John Wiley & Sons,2000

List of Experiments

S.NO.	EXPERIMENT NAME
1	Development of problem statement.
2	Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.
3	Preparation of Software Configuration Management and Risk Management related documents.
4	Study and usage of any Design phase CASE tool
5	Performing the Design by using any Design phase CASE tools.
6	Develop test cases for unit testing and integration testing
7	Develop test cases for various white box and black box testing techniques.
	Sample Projects:
8	1. Passport automation System 2. Book Bank 3. Online Exam Registration
9	1. Stock Maintenance System 5. Online course reservation system 2. E-ticketing 3. Software Personnel Management System
10	1. Credit Card Processing 2. E-book management System. 3. Recruitment system

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 introduce 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, Bubble sort, 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, longest subsequence, all pair shortest paths,

Backtracking: Travelling Salesman Problem, Graph Coloring, n-Queen Problem, Hamiltonian Cycles and 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 Astrategic 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)	CSF101
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. Knowledge Representation, Reasoning Planning and Learning being requirement for the 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 achieve the following:
CO1: Identify and formulate appropriate AI methods for solving a problem.
CO2: Understand the informed and uninformed problem types and apply search strategies to solve them.
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	**CSF309**
3. Course Title	**Theory of Computation**
4. Credits (L:T:P:C)	3:1:0:4
5. Contact Hours (L:T:P)	3:1:0
6. Prerequisites (if any)	CSF202
7. Course Basket	Discipline Core

Course Summary

The course introduces fundamental concepts in the theory of computations and formal languages. This course contains types of languages and related grammars. This includes the detailed concepts of finite automaton, regular expression,context free grammars, pushdown automaton and Turing machines. It also includes introductory concepts of its applications into other area of computer science.

Course Objectives

This course will facilitate the students to learn the mathematical foundations of computation including automata theory; the theory of formal languages and grammars; the notions of algorithm, decidability, complexity, and computability. In this course students will able to learn formalization of the notions via formal languages. The perspective learners will able to understand the hierarchy of classes of problems or formal languages.

Course Outcomes

On successful completion of the course, students will be able to achieve the following:
CO1. Demonstrate the basic elements of computation and the knowledge of finite automata.
CO2. Construct the grammars corresponding to the learned automata.
CO3. Analyze and able to construct the pushdown automata & Turing machine for the application problems.
CO4. Predict the decision problems and learn about the undecidable problems.

CURRICULUM CONTENT

Unit 1: Introduction to Finite Automata. (8 L)

Introduction to Mathematical foundation for automata: Mathematical preliminaries, alphabets, strings, languages, states, transition, transition graph, generalized transition graph.
Finite Automata: Deterministic Finite Automata, Non-Deterministic Finite Automata, Non-Deterministic Finite Automata with ϵ transitions, minimization of DFA.

Unit 2: NFA & FA with output (7 L)

Conversions and Equivalence: Equivalence between NFA with and without ϵ transitions. NFA to DFA conversion.
Application of FA: Equivalence between two DFA's, Limitations of FSM; Application of finite automata, Finite Automata with output- Moore & Melay machine and its conversion.

Unit 3: Grammars & context Free Language (8 L)

Regular Languages: Regular sets; Regular expressions, Arden's theorem, Construction of finite Automata for a given regular expression, Pumping lemma for regular sets. Closure properties of regular sets. Grammar Formalism: right linear and left linear grammars; Equivalence between regular linear grammar and FA.

Context free grammar: Grammar for CFL, Derivation trees, sentential forms. Ambiguity in context free grammars; Normal forms: Chomsky normal form and Greibach normal form; Pumping Lemma for Context Free Languages, Closure property of CFL.

Unit 4: Pushdown Automata (8 L)

Push Down Automata: Push down automata, definition; Acceptance of CFL, Acceptance by final state and acceptance by empty state and its equivalence; Equivalence of CFL and PDA; Introduction to DCFL and DPDA

Unit 5: Turing Machine & Computational Decidability (8 L)

Turing Machine: Turing Machine, definition, model, Design of TM, Computable functions Church's hypothesis, Types of Turing machines, Universal Turing Machine, Halting problem.

Properties and Decision problems: Properties of recursive and recursively enumerable languages, unsolvable decision problem, undecidability of Post correspondence problem, Church Turing Thesis.

TEXT BOOKS:

1. Hopcroft H.E. and Ullman J. D ,"Introduction to Automata Theory Language and Computation",., Pearson Education.3rd Edition.2008.
2. J. C. Martin, "Introduction to Languages and the Theory of Computation", 3rd edition, Tata McGraw-Hill.2009.
3. K.L.P. Mishra, "Theory of Computer Science", PHI.3rd Edition 2014.

REFERENCES:

1. Lewis H.P. & Papadimitrou "Elements of Theory of Computation", C.H. Pearson, PHI.2nd Edition 2011.
2. Michael Sipser " Introduction to the Theory of Computation", Thomson India 2nd Edition(international) 2004.

1.	School offering the course	School of Computing
2.	Course Code	**CSF416**
3.	Course Title	**Web Technologies and Applications**
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	Discipline Core

COURSE SUMMARY

This course covers advanced modern web technology concepts using HTML, JavaScript, CSS. It also includes server-side programming using Servlets and JSP.

COURSE OBJECTIVES

The objectives of this course are to learn web designing and server programming technologies.

COURSE OUTCOMES

On successful completion of the course, students will be able to achieve the following:
CO1: Create a dynamic webpage by the use of java script and DHTML
CO2: Use server-side scripting with servlet/JSP to generate web pages dynamically using database connectivity.
CO3: Design the modern web pages using HTML and CSS features.
CO4: Develop the modern web applications using the client and server-side technologies and the web design fundamentals.

CURRICULUM CONTENT

Unit I **(8L)**

Introduction to HTML : HTML Common tags- Block Level and Inline Elements, Lists, Tables, Images, Forms, Frames; Cascading Style sheets, CSS Properties;
Java Script: Introduction to Java Script, Client-side scripting with Javascript, variable, functions, conditions, loops, Pop-up boxes.
Advance javascript: Objects in Java Script, JavaScript own objects, DOM and web browser environments, manipulation using DOM, Forms and validations.
DHTML: Combining HTML, CSS and JavaScript, Events and Buttons.

Unit II **(6L)**

JDBC: Data Base, Database Schema, A Brief Overview Of The JDBC Process, JDBC Driver Types, JDBC Packages, Database Connection, Associating The JDBC-ODBC Bridge With Database, Creating, Inserting, Updating And Deleting Data In Database Tables, Result Set, Metadata

Unit III **(8L)**

Web Servers and Servlets: Tomcat web server, Introduction to Servlets: Servlets, the Advantage of Servlets over "Traditional" CGI, Basic Servlet Structure, Simple Servlet Generating Plain Text, Compiling and Installing the Servlet, Invoking the Servlet, Lifecycle of a Servlet, The Servlet API, Reading Servlet parameters, Reading Initialization parameters, Context Parameters, Handling Http Request & Responses, Using Cookies-Session Tracking, Servlet with JDBC.

Unit IV **(8L)**

Introduction to JSP: The Problem with Servlet. The Anatomy of a JSP Page, JSP Processing, JSP Application Development: Generating Dynamic Content, Using Scripting Elements, Implicit JSP Objects, Declaring Variables and Methods , Sharing Data Between JSP pages, Users Passing Control and Data between Pages, JSP application design with JDBC, JSP Application Design with MVC.

Unit V **(6L)**

Introduction to Node JS: Node.js process model, Advantages of Node Js, Traditional web server model, Functions, buffer, and modules, Core modules, local modules, Types of elements, Modules export.

TEXTBOOK(S)

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

REFERENCE BOOKS

1. Java - The Complete Reference, by Herbert Schildt, Publisher: McGraw Hill Education; Tenth edition (2017), ISBN-10: 9789387432291, ISBN-13: 978-9387432291
2. Internet and World wide Web How to prorgram, P.J. Deitel and H.M. Deitel, Pearson.

List of Experiments

S.NO.	EXPERIMENT NAME
1	Create a calculator using HTML, CSS and JavaScript
2	Development of static website of an online Departmental Store. The website should be user friendly and should have the following pages: · Home page · Registration and user login · User profile page · Items catalog · Shopping cart · Payment by credit card · Order confirmation
3	Add validations to the above site for registration, user login, user profile and payment by credit card using Java Script.
4	Use JSP to converting the static web pages into dynamic web pages.
5	Create a servlet program to connect with the JSP page using HTTP protocol.
6	Create a student table and insert new students details and delete old student details.
7	Create database with User Information and Item information. The Item catalog should be dynamically loaded from the database.
8	Use the concept of session and cookies to create a dynamic login page.
9	Write a program to show how sharing of data is done using JSP pages.
10	Develop Simple Servlet Question Answer Application to demonstrate use of HttpServletRequest and HttpServletResponse interfaces.

1.	School offering the course	School of Computing
2.	Course Code	**CSF341**
3.	Course Title	**R Programming**
4.	Credits (L: T:P:C)	2:0:1:3
5.	Contact Hours (L: T:P)	2:0:2
6.	Prerequisites (if any)	NA
7.	Course Basket	DE

COURSE SUMMARY

In this course student will learn how to program in R and how to use R for effective data analysis. The course includes the installation and configuration of R programming a statistical programming environment, discuss generic programming language concepts and R data objects as they are implemented in a high-level statistical language. The course covers practical issues in statistical computing which includes programming in R, reading data into R, accessing R packages, writing R functions for f and Machine Learning algorithms.

COURSE OBJECTIVES

The objective of this course is to develop a broad perspective about the R programming and its applications to solve basic mathematical problems, statistical manipulations and scientific tasks such as data science and machine learning. R programming has its own built in functions to perform any specialized task. The course is intended to learn the basics of R software in this course.

COURSE OUTCOMES

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

CO1: Apply the basic functionalities of R programming to solve basic mathematical problems.

CO2: Apply the R programming for preprocessing the real-life datasets.

CO3: Understand and analyze the descriptive statistics for a given dataset.

CO4: Implement some classical machine learning models using R programming.

CURRICULUM CONTENT

UNIT 1 **(5L)**

What is R?, What is S? Basic Features of R, Limitations of R, R Framework setup, R packages, Use R like calculator, Reading and Writing data into R: combine or concatenate command, scan command, alternative commands for reading data, R constant and variables, operators and expression.

UNIT 2 **(5L)**

R data types and objects: Number and Text, Vector, Matrix, Factor, Array, List Data Frame, Manipulating Objects. Control structures, looping, scoping rules, Operations on Dates and Times, functions, debugging tools. R built-in packages and functions.

UNIT 3 **(5L)**

Dataset: Import/export bigger datafile (csv, text, excel, table, url, etc.), Identify and handle missing values, data formatting, Data Standardization, Data Normalization and Scaling, Data visualization, Binning, Multimedia datasets: text dataset, image dataset, audio dataset, video dataset.

UNIT 4 **(5L)**

Central tendency, Dispersion variance, standard deviation, shape skewness, kurtosis, percentiles, five-point summary, boxplots, histograms, bar plot, pie chart, scatter plot, two-way tables, covariance, correlation, Chi-Square test for two-way tables.

UNIT 5 **(6L)**

Introduction to machine learning, types of machine Learning, supervised learning using R- regression, decision tree, KNN, SVM, Unsupervised learning using R- Clustering: K-means, hierarchical, frequent itemset, dimensionality reduction.

TEXTBOOK(S)

1. R programming for data science. R. D. Peng, Leanpub, 2016.
2. Practical Data Science with R. Author(s): Nina Zumel, John Mount, Manning Shelter Island, 2014.

REFERENCES :

1. The R book, Crawley and Michael, John Wiley & Sons, 2012.
2. Beginning R: The statistical programming language. Mark Gardener, John Wiley & Sons, 2012.

1.	School offering the course	School of Computing
2.	**Course Code**	**CSF348**
3.	**Course Title**	**Mobile Application programming using Android**
4.	Credits (L:T:P:C)	2:0:1:3
5.	Contact Hours (L:T:P)	2:0:2
6.	Prerequisites (if any)	NA
7.	Course Basket	DE

Course Summary

This course deals with the internals of Android Operating System, GUI, various services, graphics design, database connectivity, network connectivity and integration of various APIs.

Course Objectives

The objective of this course is to teach mobile application programming to students using Android. Students will learn about the technologies and the tools used to develop Android mobile applications. Students will be introduced to the internals of the Android OS and mobile application development using the Android SDK.

Course Outcomes

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

CO1: Understand internals of the Android OS
CO2: Implement mobile application development using the Android SDK.
CO3: Implement GUI, Services, Database Connectivity and Web Service Integration
CO4: Understand network connectivity and integration of various APIs.

Curriculum Content

Unit 1: Overview of mobile applications, installing the development environment, Android Overview, architecture overview and Android development environment, Anatomy of an Android App, App lifecycle. **(3L)**

Unit 2: GUI development: XML for UI design, development tools, Activities, multiple activities, Activity lifecycle, Intents, MVC, GUI development, Lists, fragments, dialogs, Action Bar, 2D graphics and drawables. **(12L)**

Unit 3: Services and Broadcast Services, Database connectivity with SQLite, Web service integration using JSON, XML, SOAP and RESTful services. **(6L)**

Unit 4: Network connectivity, Integration with multiple APIs. **(3L)**

Textbook(s)

1. Head First Android Development: A Brain-Friendly Guide, by Dawn Griffiths and David Griffiths.

Reference Books

1. Android Programming: The Big Nerd Ranch Guide (Big Nerd Ranch Guides), 2017.

1.	School offering the course	School of Computing
2.	Course Code	**CSF349**
3.	Course Title	**Cloud Computing**
4.	Credits (L: T:P:C)	2: 0: 2 :3
5.	Contact Hours (L: T:P)	2:0:2
6.	Prerequisites (if any)	NA
7.	Course Basket	DE

COURSE SUMMARY

This course is designed to teach students the basic concepts and terminology of cloud computing. After establishing the definition of cloud computing, this course describes the various service delivery models of cloud computing architecture, and the ways in which clouds can be deployed as public, private, hybrid and community clouds. Students also learn about the security challenges that cloud deployments experience, and how these are addressed.

COURSE OBJECTIVES

This course is designed to provide knowledge about basic concepts of Cloud computing. Public cloud and its service and deployment models, private cloud and its need and challenges, Multi-cloud and business cloud, security threats in the cloud.

COURSE OUTCOMES

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

CO1: Elaborate cloud computing its service and deployment models.

CO2: Formulate the importance of virtualization, multi-tenancy in the cloud environment.

CO3: Define and examine different cloud computing services.

CO4: Categorize the different security threats and challenges faced by cloud provider, and Demonstrate the different types of business cloud and its uses.

CURRICULUM CONTENT

UNIT 1 **(6L)**

Overview of cloud computing and Distributed Computing: What is a cloud, Definition of cloud, Definition of cloud, characteristics of cloud, Traditional vs. Cloud Computing, Importance of Cloud Computing, Cloud service models (IaaS, PaaS & SaaS). Cloud deployment models (Public, Private, Hybrid and Community Cloud), Benefits and Challenges of Cloud Computing. Introduction, Examples of distributed computing, Concurrent Programming, Characteristics & Properties of Distributed Systems, client-server model, centralized vs distributed computing, Resource Sharing and the Web Challenges, security issues.

UNIT 2 **(7L)**

Private Cloud: Concept of Hypervisor, Basics of virtualization, Virtualization technologies, Server virtualization, VM migration techniques, Role of virtualization in Cloud Computing. Business cases for the need of Cloud computing environment, Concept of Private Cloud, Characteristics of Private Cloud, Private Cloud deployment models, Private Cloud Vendors, Virtual Private Cloud. Multitenancy, Types of tenancy, Application programming interfaces (API), Billing and metering of services.

UNIT 3 **(7L)**

Public Cloud: Concept of Public Cloud, Importance of Public Cloud, when to opt for Public Cloud, Public Cloud Service Models, and Public Cloud players. Infrastructure as a Service Offerings, IaaS Vendors, PaaS offerings, PaaS vendors, Software as a Service. Implementing public cloud AWS, Introduction, Service Offered, Creation of EC2 instance, Microsoft Azure: Introduction, Service Offered, Creation of DB instance. Implementing Security in public Cloud, Comparison of Public Cloud Vendors (AWS, Microsoft, Google, IBM, Salesforce).

UNIT 4 **(6L)**

Multi-Cloud: Concept of multi-cloud management, Challenges in managing heterogeneous clouds, benefits of multi-cloud management systems. Case study on Multi-Cloud Management System (Right Scale Cloud Management System). Business Clouds: Cloud Computing in Business, Various Biz Clouds focused on industry domains (Retail, Banking and Financial sector, Life Sciences, Social networking, Telecom, Education).

UNIT 5 **(7L)**

Cloud Security: Cloud security reference model, Principal security dangers/risks to cloud computing, Internal security breaches, Data corruption, Malicious Insiders, Data Loss or Leakage, Account or Service Hijacking, Unknown Risk Profile, Steps to reduce cloud security breaches, Identity management: Detection and forensics, Identity management: Detection and Identity management, Benefits of identity, Encryption techniques, Encryption & Encrypting data, Attacks on VM, Abuse and Nefarious Use of Cloud Computing.

TEXTBOOK(S)

1. R. Buyya, C. Vecchiola, S. T. Selvi, Matering Cloud Computing, Ed. Third reprint, 2013.
2. B. Sosinsky, Cloud computing Bible, Ed. Reprint Willy India Pvt. Ltd, 2014.
3. Carlin, Sean, and Kevin Curran. "Cloud computing security." Pervasive and Ubiquitous Technology Innovations for Ambient Intelligence Environments. IGI Global, 2013. 12-17.

REFERENCES :

1. M. Miller, Cloud Computing, Pearson education in South Asia, Ed. 9th 2014.
2. Buyya, Rajkumar, James Broberg, and Andrzej M. Goscinski, eds. Cloud computing: Principles and paradigms. John Wiley & Sons, 2010.

1. School offering the course	School of Computing
2. Course Code	**CSF355**
3. Course Title	**Cyber Crime & Investigation**
4. Credits(L:T:P:C)	3:0:0:3
5. Contact Hours(L:T:P)	3:0:0
6. Prerequisites(ifany)	NA
7. Course Basket	DE

COURSE OUTLINE:

This course provides an impression of cybercrime and the investigation practices put in place to respond to them. The course will emphasis on the types and extent of present cybercrimes, how the justice system responds to these crimes, the various legal protections afforded to computer users, the regulation and policies that govern cybercrime detection and prosecution, and related machineries.

COURSE OBJECTIVE:

To describe the nature and area of cybercrime. To grow knowledge of key incidents of cybercrime and their subsequent influence. To study and debate national and global digital law implementation efforts. To categorize and assess the precise technology that enables cybercrime and digital law enforcement. To assess the influence of cybercrime on information professions.

COURSE OUTCOMES:

Having successfully completed this course, students will be able to reveal facts and understanding of:

CO1: Analyse the essential concepts of cybercrime and forensics.

CO2: Distinguish the object and causes for cybercrime, detection and handling.

CO3. Understand the extents affected by cybercrime and investigation.

CO4: Demonstrate tools used in cyber forensic and apply their knowledge for report writing

CURRICULUM CONTENT

UNIT-I: PRINCIPLES AND CONCEPTS OF CYBER CRIMINOLOGY (8 L)

Crime, Offence, Misdemeanour, Cyber Space, Cyber Crime, Cyber Criminology, Information Security, Penetration Testing, Incident Response, GRC, Conventional crimes vs. Cyber Crimes, White Collar Crimes, Economic Offences, Organized Crimes, Terrorism, Crime and Media and other contemporary forms of crimes.

UNIT-II: PSYCHOLOGY OF CYBER CRIMINALS (7 L)

Types of Cyber Criminals, Modus Operandi of Cyber Criminals, Profiling of Cyber Criminals, Tools and Techniques adopted by Cyber Criminals, Psychological theories relating to cyber criminals, Causes of Cyber Crimes, Criminological Theories and Cyber Crime, Routine Activity Theory, Social Learning Theory, Differential Association Theory, Differential Opportunity Theory, Media and Crime and latest theories and other related theories.

UNIT-III: DIGITAL INVESTIGATION (7L)

Digital Evidence and Computer Crime, History and Terminology of Computer Crime Investigation, Technology and Law, The Investigative Process, Investigative Reconstruction, Motive and Technology, Digital Evidence in the Courtroom.

UNIT-IV: COMPUTER FORENSIC & UNDERSTANDING INFORMATION (7L)

Computer Forensic Fundamentals: Applying Forensic Science to computers, Computer Forensic Services, Benefits of Professional Forensic Methodology, Steps taken by computer forensic specialists. **Methods of storing data:** number systems, character codes, record structures, file formats and file signatures, Word processing and graphic file formats, Structure and Analysis of Optical Media Disk Formats, Recognition of file formats and internal buffers, Extraction of forensic artefacts, understanding the dimensions of other latest storage devices, SSD Devices.

UNIT-V: TYPES OF COMPUTER FORENSICS TOOLS AND TECHNOLOGY (7L)

Tools and Types of Military Computer Forensics Technology, Tools and Types of Law Enforcement Computer Forensic Technology, Tools and Types of Business Computer Forensic Technology.

TEXT BOOKS

1. SunitBelapure and Nina Godbole. ―Cyber Security: Understanding Cyber Crime, Computer Forensic and Legal Perspectives‖, Wiley India Pvt Ltd, ISBN: 978-81-265-2179, publish date 2013.
2. Bil Nelson, Amelia Philips and Christopher Steuart, ―Guide to Computer Forensics and Investigation‖, 4th Edition, Cengage Learning 2015.

REFERENCES

1. Thomas J Mowbray, ―Cybersecurity Managing Systems, Conducting Testing, and investigating Intrusions‖, copyright 2014 by John Wiley & sons, ISBN: 978-1-118-84965, 2014.
2. James Graham, Ryan Olson, Rick Howard, ―Cyber Security Essentials‖, CRC press, 15 Dec 2010.

1.	School offering the course	School of Computing
2.	Course Code	**CSF345**
3.	Course Title	**Introduction to Data Science**
4.	Credits (L: T:P:C)	2:0:1:3
5.	Contact Hours (L: T:P)	2:0:2
6.	Prerequisites (if any)	NA
7.	Course Basket	DE

COURSE SUMMARY

This course aims to provide a basic understanding of Data Science concepts. This course introduces students to the data science principles required to tackle real-world, data-rich problems in business.

COURSE OBJECTIVES

Data Science is the study of the generalizable extraction of knowledge from data. This course serves as an introduction to the data science principles required to tackle real-world, data-rich problems in business and academia, including: Data acquisition, cleaning, and aggregation, Exploratory data analysis and Visualization, Feature engineering, Model creation and validation, Basic statistical and mathematical foundations for data science

COURSE OUTCOMES

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

CO1: An understanding of problems solvable with data science and an ability to attack them from a statistical perspective.

CO2: An understanding of when to use supervised and unsupervised statistical learning methods on labeled and unlabeled data-rich problems

CO3: The ability to create data analytical pipelines and applications in Python. .

CO4: Apply the various tools needed to continue developing as a data scientist.

.

CURRICULUM CONTENT

Unit 1: Computer Science/Statistics/Linear Algebra Short Review (5 L)

What is data science? Brief review of prerequisite knowledge for studying data science. Basics of computer science; data structures/types, program control flow, and syntax in Python. Basics of statistics; probability and probability distributions. Basics of linear algebra; matrices, vectors using Python programming language.

Unit 2: Exploratory Data Analysis (Eda) And Visualization Design (5 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 3 Data Modeling: Supervised/Unsupervised Learning (5 L)

Two basic kinds of statistical models used for prediction. Supervised Learning algorithm: Linear Regression and Logistic Regression. Unsupervised Learning algorithm: K-Means clustering. Advanced supervised learning algorithms like linear support vector machines, decision trees, and random forest models for regression and classification. Advanced unsupervised learning algorithm like DBSCAN.

Unit-4 Data Modeling: Feature Selection, Engineering, And Data Pipelines (5 L)

Curse of dimensionality and Dimensionality reduction. Feature selection and feature extraction. Principal Component Analysis/Independent Component Analysis and regularization. Construct complete data pipelines, going from data ingestion, preprocessing to model construction and evaluation.

Unit- 5: Data Modeling: Model Evaluation And Project Presentations (6 L)

Exploration of more sophisticated model evaluation approaches like cross-validation and bootstrapping with the goal of making the model as generalizable as possible. Presentation of students' project and sharing learning experience.

TEXT BOOKS:

1. Cathy O'Neil and Rachel Schutt, Doing Data Science, Straight Talk from the Frontline, O'Reilly. 2014.
2. Jiawei Han, Micheline Kamber and Jian Pei Silberschatz, Korth, Data Mining: Concepts and Techniques, Third Edition. ISBN 0123814790. 2011

REFERENCES:

1. Mohammed J. Zaki and Wagner Miera Jr, Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press. 2014.

1. School offering the course	School of Computing
2. Course Code	**CSF445**
3. Course Title	**Mobile and Wireless Network Security**
4. Credits (L:T:P:C)	3:0:0:3
5. Contact Hours (L:T:P)	3:0:0
6. Prerequisites (if any)	NA
7. Course Basket	DE

COURSE OUTLINE:

This course will introduce students about Mobile and Wireless Networks, Vulnerabilities of Wired and Wireless Networks. It also includes overview of Fundamental Security Mechanisms, Hash functions, Electronic signatures and MAC, Cryptographic protocols. Topics would also include Wi-Fi Security Dedicated Architectures and Bluetooth Security.

COURSE OUTCOMES:

The purpose of this course is to provide In-depth knowledge about cellular design concepts and understanding of 3G Wireless network. It also provides an understanding of various security concerns and protocols in wireless networks (e.g., WiFi and mobile cellular networks) and mobile systems and applications.

COURSE OUTCOMES:

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

CO1: Understand the security threats, vulnerabilities in wireless and mobiles systems and their related mechanisms

CO2 Understand the strategies for developing secure mobile applications.

CO3: Select mobile security penetration tools for evaluating the robustness of mobile applications.

CO4: Understand various models, design principles and solutions used in wireless network security to obtain authentication.

CURRICULUM CONTENT

Unit 1: Introduction to Mobile and Wireless Networks (8L)

Introduction, Cellular network basic concepts and Applications, First generation (1G) mobile, Second generation (2G) mobile, Third generation (3G) mobile, IEEE wireless networks, WLAN: IEEE 802.11, WPAN: IEEE 802.15, WMAN: IEEE 802.16, WMAN mobile: IEEE 802.20, MIH: IEEE 802.21,
WRAN: IEEE 802.22, Macro mobility, Micro mobility, NEMO and MANET networks

Unit 2: Vulnerabilities of Wired and Wireless Networks (7L)

Introduction, Security in the digital age, Threats and risks to telecommunications systems, Homogenity vs. heterogeneity, The Internet and security

Unit 3 Fundamental Security Mechanisms (7L)

Basics on security, Symmetric and asymmetric cryptography, Hash functions, Electronic signatures and MAC, Cryptographic protocols, Secure communication protocols and VPN Implementation, Secure Socket Layer (SSL) and Transport Layer Security (TLS), IPsec protocol suite, Comparison between

SSL and IPsec security protocols, IPsec VPN and SSL VPN, Authentication, Access control, Firewalls, Intrusion detection

Unit-4 **(8L)**

Wi-Fi Security Dedicated Architectures: Introduction, Hot spot architecture: captive portals, Captive portal overview, Security analysis, Wireless intrusion detection systems: architecture, events, example; Wireless honeypots: design, requirements.

Wi-Fi Security: Introduction, Attacks on wireless networks, Passive attacks, Active attacks, TCP attacks, Trojan attack, Dictionary attacks, Security in the IEEE 802.11 standard, IEEE 802.11 security mechanisms, WEP (Wired Equivalent Privacy), WEP shortcomings, Attacks, Security in 802.1x, Authentication in wireless networks, RADIUS, EAP authentication procedures, PKI, Level 3 VPN, IPsec

Unit- 5: **(7L)**

Bluetooth Security: Introduction, Organization of Bluetooth nodes in the network, Bluetooth technical specification, Radio physical layer, Baseband, Link controller, Bluetooth device addressing, HCI layer, L2CAP layer, Bluetooth security, Bluetooth encoding, Attacks.

TEXT BOOKS

1. Jonathan Katz and Yehuda Lindell, Introduction to Modern Cryptography, Chapman & Hall/CRC Cryptography and Network Security Series, 2nd edition 2014.
2. Frank Adelstein, Sandeep K.S. Gupta, Golden G. Richard III, and Loren Schwiebert, Fundamentals of Mobile and Pervasive Computing, 2005.

REFERENCES

1. Levente Buttyán and Jean-Pierre Hubaux, Security and Cooperation in Wireless Networks, 2008.
2. James Kempf, Wireless Internet Security: Architectures and Protocols, 2008.
3. PatrickTraynor, Patrick McDaniel, and Thomas La Porta, Security for Telecommunications Networks, 2008.

1.	School offering the course	School of Computing
2.	Course Code	**CSF361**
3.	Course Title	**Introduction to Blockchain Technologies**
4.	Credits (L: T:P:C)	2:1:0:3
5.	Contact Hours (L: T:P)	2:1:0
6.	Prerequisites (if any)	NA
7.	Course Basket	DE

COURSE SUMMARY

The students will learn the basic components of blockchain, concept of distributed system, consensus mechanism and its important and basic of cryptocurrency.

COURSE OBJECTIVES

This course objective is to explain basic components of a blockchain (types, mechanics: transaction, block, block header, chain and terminology) its operations (processes, verification, validation, and consensus model) underlying algorithms, and essentials of trust to understand how blockchain systems (mainly Bitcoin and Ethereum) work.

COURSE OUTCOMES

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

CO1: Recognize foundational concepts of blockchain and learn about the decentralized peer-to-peer network.

CO2: Understand the formal definition of distributed consensus and apply these concepts on the blockchain.

CO3: Assess Blockchain applications in a structured manner.

CO4: Understand the meaning and properties of crypto economics: cryptography and economics.

CURRICULUM CONTENT

Unit 1: Distributed Systems: (6L)

Blockchain architecture, Basic components (blocks, nodes, etc.), Distinction between public and private blockchains, benefits and drawbacks, Fundamental traits and characteristics, Distributed Database, CAP theorem, the Byzantine Generals Problem and Fault Tolerance.

Unit 2: Cryptography in Blockchain: (6L)

Hadoop Distributed File System, Distributed Hash Table, ASIC resistance, Turing Complete. Blockchain Network, Mining Mechanism

Unit-3: Consensus: **(6L)**

Distributed Consensus, Merkle Patricia Tree, Gas Limit, Cryptography: Hash function, Digital Signature - ECDSA, Memory Hard Algorithm, Zero Knowledge Proof, Transactions and Fee.

Unit 4: Blockchain Design Principle: (6L)

Consensus, Security, and Operating Protocols, Blockchain Design Principle, Public and Private DLTs, Alternative Consensus Mechanisms to Bitcoin's Proof-of-Work, Proof-of-Stake, Proof-of-Burn, Voting-Based Consensus Algorithms, and Federated Consensus, Sybil Attack, Energy Utilization.

Unit-5 Crypto economics: (6L)

Property of crypto economics: cryptography and economics, Integration of cryptography and pseudo-anonymity in public blockchains, cryptoeconomics with respect to distributed systems fundamentals (liveness, safety, data availability).

TEXT BOOKS:

1. Kube N. Daniel Drescher: Blockchain basics: a non-technical introduction in 25 steps, 2018.

REFERENCES:

1. Warburg B, Wagner B, Serres T. Basics of Blockchain: A Guide for Building Literacy in the Economics, Technology, and Business of. Animal Venturs LLC; 2019..

Tutorial: Naive Blockchain construction, Memory Hard algorithm - Hashcash implementation, Direct Acyclic Graph.

1.	School offering the course	School of Computing
2.	Course Code	**CSF364**
3.	Course Title	**Container Technologies**
4.	Credits (L: T:P:C)	2: 0: 2 :3
5.	Contact Hours (L: T:P)	2:0:2
6.	Prerequisites (if any)	NA
7.	Course Basket	DE

COURSE SUMMARY

This course is designed to teach students the basic concepts and terminology of cloud computing. After establishing the definition of cloud computing. This course describes the basics of container technologies used in cloud computing, dockers, concept of Orchestration and kurbernetes

COURSE OBJECTIVES

This course provides the opportunity to learn concepts and design Containerization and build an Orchestration of containers. It also provides an ability to promote the cost effective light weight virtualization using container orchestration management tools and techniques.

COURSE OUTCOMES

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

CO1. Elaborate the container technology

CO2. Formulate and Design containers using Docker.

CO3. Categorize and demonstrate the concept of containerization using Docker files and Compose

CO4. Categorize and design an Orchestration of nodes.

CURRICULUM CONTENT

UNIT 1 **(6L)**

Introduction Container Technology: Containerization, History of Containers, Namespaces and C-groups, Containers vs Virtual Machines, Types of Containers. Docker: Overview, Installing Docker on Linux, Installation, Hub, Images, Containers, Features of Docker, Components of Docker.

UNIT 2 **(7L)**

Creating Containerized Services: Working with Containers, Architecture, Container & Hosts, Configuring, Containers & Shells, File, Building Files, Public Repositories, Managing Ports, Private Registries, Building a Web Server Docker File.

UNIT 3 **(7L)**

Managing Containers: Instruction Commands, Container Linking, Storage, Networking, Setting Node.js, Setting MongoDB, Setting NGINX, Toolbox, Setting ASP.Net, Docker Cloud, Logging, Docker – Compose, Docker - Continuous Integration.

UNIT 4 **(7L)**

Orchestration in Docker: Create and run multi-container applications using Docker Compose and manage clusters of Docker nodes using Docker Swarm. Topics: Docker Compose, Docker Swarm, Docker Service, Placement Rolling Update and Rollback Docker Stack, deploy a Multi-container Application using Compose, Running Docker in Swarm mode, deploying a Service in Swarm Scale, Services, Service Placement, Rolling Updates and Rollbacks Docker Stack.

UNIT 5 **(6L)**

Introduction to Kubernets: Understanding Kubernetes architecture, Introduction to Kubernetes objects, using basic Kubernetes objects, Using the kubectl command, Leveraging Kubernetes.

TEXT BOOKS:

1. Antonopoulos, Nick, and Lee Gillam. Cloud computing. London: Springer, 2010.
2. Comer, Douglas E. The Cloud Computing Book: The Future of Computing Explained. Chapman and Hall/CRC, 2021.
3. Raj, Pethuru, Jeeva S. Chelladhurai, and Vinod Singh. Learning Docker. Packt Publishing Ltd, 2015.
4. Luksa, Marko. Kubernetes in action. Simon and Schuster, 2017.

REFERENCES :

1. Foster, Ian, and Dennis B. Gannon. Cloud computing for science and engineering. MIT Press, 2017.
2. Chaudhary, Sanjay, Gaurav Somani, and Rajkumar Buyya, eds. Research advances in cloud computing. Springer Singapore, 2017.
3. Turnbull, James. The Docker Book: Containerization is the new virtualization. James Turnbull, 2014.
4. Sayfan, Gigi. Hands-On Microservices with Kubernetes: Build, deploy, and manage scalable microservices on Kubernetes. Packt Publishing Ltd, 2019.
5. Protechgurus, Dockers containers ultimate beginners guide, Independently published , August 18, 2018

1.	School offering the course	School of Computing
2.	Course Code	**CSF346**
3.	Course Title	**Data Mining and Data Warehousing**
4.	Credits (L: T:P:C)	2:0:1:3
5.	Contact Hours (L: T:P)	2:0:2
6.	Prerequisites (if any)	NA
7.	Course Basket	Discipline Elective

COURSE SUMMARY

This course aims to provide a basic understanding of Data ware housing and mining concepts, implementation of Data Mining algorithms. This course introduces students to enterprise data and the process and technologies to integrate data from a variety of sources.

COURSE OBJECTIVES

This course will cover the basic concepts of Data Warehouse and Data Mining techniques, Examine the types of the data to be mined and apply pre-processing methods on raw data. It also discover interesting patterns, analyze supervised and unsupervised models and estimate the accuracy of the algorithms

COURSE OUTCOMES

On successful completion of the course, students will be able to achieve the following:
CO1: Explain and evaluate the various data mining algorithms
CO2: Discover and measure interesting patterns from different kinds of databases.
CO3: Apply the techniques of clustering, classification, association finding,
CO4: Apply techniques for feature selection and visualization to real world data.

CURRICULUM CONTENT

Unit 1: Overview **(5 L)**

Motivation (for Data Mining), Data Mining-Definition & Functionalities. Data Warehousing: Overview, Definition, Delivery Process, Difference between Database System and Data Warehouse, Multi-Dimensional Data Model, Data Cubes, Stars, Snow Flakes, Fact Constellations, Concept hierarchy, Process Architecture, 3 Tier Architecture, Data Marting. ROLAP, MOLAP, HOLAP.

Unit 2: Data Pre-Processing **(5 L)**

Data Cleaning: Missing Values, Noisy Data, (Binning, Clustering, Regression, Inconsistent Data, Data Integration and Transformation. Data Reduction: Data Cube Aggregation, Dimensionality reduction, Data Compression, Numerosity Reduction, Clustering, Discretization and Concept hierarchy generation.

Unit 3 Concept Description **(5 L)**

Definition, Data Generalization, Analytical Characterization, Analysis of attribute relevance, Mining Class comparisons, Statistical measures in large Databases. Measuring Central Tendency, Measuring Dispersion of Data, Apriori Algorithm, Mining Multilevel Association rules from Transaction Databases

Unit-4 Classification **(5 L)**

What is Classification, Issues regarding Classification, Decision tree, Bayesian Classification, Classification by Back propagation.

Unit- 5: Cluster Analysis **(6 L)**

Data types in cluster analysis, Partitioning methods. Hierarchical Clustering- CURE and Chameleon, Density Based Methods-DBSCAN, OPTICS, Grid Based Methods STING, CLIQUE, Outlier Analysis

TEXT BOOKS:

1. Jiawei Han and Micheline Kamber, Data Mining Concepts and Techniques, Elsevier, Third Edition, 2012.

REFERENCES:

1. Margaret H.Dunham, Data-Mining: Introductory & Advanced Topics, Pearson Education, India, 3rd edition, 2012.

1.	School offering the course	School of Computing
2.	Course Code	**CSF373**
3.	Course Title	**Server-Side Engineering**
4.	Credits (L: T:P:C)	2:0:2:3
5.	Contact Hours (L: T:P)	2:0:2
6.	Prerequisites (if any)	None
7.	Course Basket	Discipline Elective

COURSE SUMMARY

The students will gain familiarity with what server-side programming is and what it can do

COURSE OBJECTIVES

Provide Opportunity to student to learn the concepts of MVC application and design a full fledge CRUD application.

COURSE OUTCOMES

On successful completion of the course, students will be able to achieve the following:
CO1: Understand and design an application using Maven.
CO2: Design an MVC application using Spring framework.
CO3: Design and develop a Restfull service.
CO4: Design a CRUD based application.

CURRICULUM CONTENT

UNIT 1: Maven (6 L)

What is Maven, Why command line, Dependency Resolution, Configurations, Installation Approach - Archetype, RAD,Setup Commands, Download from GIT, Life cycles, Phases and Goals, Profiles, Parent-Child Module, Dependency Plugins, Local Maven Repository Vs Project Centralise Repository.

UNIT 2: Spring Framework Overview (8 L)

Introduction to Spring, Installation, first application, Spring Container and Dependency Injection: Spring Container types, Working of Spring container, Dependency Injection by Constructor, Injecting string-based values. Bean Implementation: Introduction and Scope, creating source file, Implement Collections, Implement Java Based Configuration. **Aspect Oriented Programming**: Spring AOP, Implement Aspect Oriented Programming.

UNIT 3: Spring Data Access and Transaction Management (4 L)

Spring JDBC, JDBC Template, Spring Transaction. Develop Web Application using Spring: Spring Web MVC Overview, Advanced MVC Features, Development of Spring Web Application

UNIT 4: Spring ORM (12 L)

Hibernate + JPA, Introductions, DataSource Configurations, Object Relational, Mapping, About Maven Dependencies and Configs, Entity classes and Session Factory, CRUD Operations – APIs, Hibernate Configurations, Session Factory, Sessions, Mapping XML, Entities, Annotation Based, First/Second Level Caching , Transient, Persistent and Detached Objects

TEXTBOOK(S) :

1. Balaji Varanasi, Introducing Maven: A Build Tool for Today's JAVA Developers, Apress, 2nd Edition, 2019.
2. Craig Walls, Spring in Action, Manning Publications, 5th Edition, 2019.

REFERENCES:

1. Marten Deinum, Daniel Rubio, Josh Lang, Gary Mak, Spring 5 Recipes: A Problem-Soluiton Approach, Appress, 4th Edition, 2017.

1.	School offering the course	School of Computing
2.	Course Code	**CSF363**
3.	Course Title	**Blockchain Ecosystems & Governance**
4.	Credits (L: T:P:C)	2:1:0:3
5.	Contact Hours (L: T:P)	2:1:0
6.	Prerequisites (if any)	NA
7.	Course Basket	DE

COURSE SUMMARY

The students will learn the comparison between traditional and blockchain system. Some cases of blockchain implementation with some advantages and disadvantages, future of blockchain, etc.

COURSE OBJECTIVES

This course enables the students to understand the broader blockchain ecosystem, other blockchain platforms, application use cases, and challenges such as privacy and scalability

COURSE OUTCOMES

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

CO1- Understand permissioned blockchain, and concept of hyperledge and Blockchain as a service.

CO2- Define various use cases on realtime applications using blockchain and concept of cryptocurrency.

CO3- Understant the concepts of regulations and anonilty and legal aspects attached with technology.

CO4- Define and understand the future of blockchain globally and Indian scenario.

CURRICULUM CONTENT

UNIT 1 **(6L)**

Enterprise Blockchain: Real-World Applications: Permissioned Blockchains: The Linux Foundation's Hyperledger and Microsoft Azure's Blockchain as a Service, JP Morgan's Quorum, Ripple, and Tendermint.

UNIT 2 **(6L)**

Blockchain use Cases: Challenges and solutions Applications of blockchain, Business and industry use cases: cybersecurity, the integrity of information, E-Governance, Climate Change, Biodiversity, Energy, Internet of Things, Medical Record Management System, Sustainability and other contract enforcement mechanisms etc.

UNIT 3 **(6L)**

Scaling Blockchain (Cryptocurrency): Bitcoin as a payment method, comparison with traditional forms, Cryptocurrency: History, Distributed Ledger, Bitcoin protocols - Mining strategy and rewards, Vulnerability, Attacks, Namecoin, Vertical scaling (e.g. block size increases, Segregated Witness and the Lightning Network), Horizontal Scaling (e.g. sidechains, sharding).

.

UNIT-4 **(6L)**

Regulation and Anonymity: Anti-Money Laundering (AML) and Know Your Customer (KYC) Regulations, Anonymity goals, and government techniques for deanonymization of entities on blockchain, stakeholders, Roots of Bitcoin, Legal Aspects - Cryptocurrency Exchange, Black Market and Global Economy.

UNIT 5 **(6L)**

Blockchain Future: Global Status on Blockchain, Historic stance of the Indian Government, Current Scenario, Myths vs reality of blockchain technology, Understanding and working knowledge of the emerging blockchain technology**,** Do-main Name Service and future of Blockchain: venture capitalism, ICOs, and crowdfunding.

TEXT BOOKS:

1. Jun, M. Blockchain government - a next form of infrastructure for the twenty-first century. J. open innov. 4, 7 (2018). https://doi.org/10.1186/s40852-018-0086-3
2. Nijalingappa, Pradeep, and Mangesh Manikrao Ghonge, eds. Blockchain Technologies and Applications for Digital Governance. IGI Global, 2021.

REFERENCES:

1. Savirimuthu J. Blockchain and the law: The rule of code., 2019

1. School offering the course	School of Computing
2. Course Code	**CSF446**
3. Course Title	**Ethical Hacking and Digital Forensics**
4. Credits (L:T:P:C)	2:0:1:3
5. Contact Hours (L:T:P)	2:0:2
6. Prerequisites (if any)	NA
7. Course Basket	DE

COURSE OUTLINE:

This course will introduce students about Hacking windows, Network hacking, Password hacking, TCP / IP – Checksums, Dos attacks – SYN attacks, Smurf attacks, UDP flooding, DDOS Models. Firewalls, Packet filter firewalls, Packet Inspection firewalls, Application Proxy Firewalls. Batch File Programming, Fundamentals of Computer Fraud, Strategic Planning Process, Architecture strategies for computer fraud prevention, Penetrating testing process, Key Fraud Indicator selection process customized taxonomies, Computer Forensics, Accounting Forensics, Journal risk and control matrix, Misuse detection and Novelty detection

COURSE OBJECTIVE:

This course provides an introduction the concepts of Ethical Hacking and provides an understanding of Computer forensics fundamentals. This course will provide the opportunity to learn about different tool and techniques in Ethical Hacking and will analyse various computer forensics technologies and methods for data recovery.

COURSE OUTCOMES:

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

CO1: Identify and analyse the stages an ethical hacker requires in order to compromise a target system.

CO2: Understand the concepts of computer forensics fundamentals and types of computer forensics.

CO3: Evaluate security techniques used to protect system and user data.

CO4: Illustrate the methods for data recovery, evidence collection and data seizure.

CURRICULUM CONTENT

UNIT 1: **(6 L)**

Hacking windows – Network hacking – Web hacking – Password hacking. A study on various attacks – Input validation attacks – SQL injection attacks – Buffer overflow attacks - Privacy attacks.

UNIT 2: **(6 L)**

TCP / IP – Checksums – IP Spoofing port scanning, DNS Spoofing. Dos attacks – SYN attacks, Smurf attacks, UDP flooding, DDOS – Models. Firewalls – Packet filter firewalls, Packet Inspection firewalls – Application Proxy Firewalls. Batch File Programming.

UNIT 3: **(5 L)**

Fundamentals of Computer Fraud – Threat concepts – Framework for predicting inside attacks – Managing the threat – Strategic Planning Process.

UNIT-4 **(5 L)**

Architecture strategies for computer fraud prevention – Protection of Web sites – Intrusion detection system – NIDS, HIDS – Penetrating testing process – Web Services – Reducing transaction risks.

UNIT- 5: **(5 L)**

Key Fraud Indicator selection process customized taxonomies – Key fraud signature selection process – Accounting Forensics – Computer Forensics – Journaling and it requirements – Standardized logging criteria – Journal risk and control matrix – Neural networks – Misuse detection and Novelty detection.

TEXT BOOKS

1. Stuart McClure, Joel Scambray and Goerge Kurtz, ―Hacking Exposed 7: Network Security Secrets & Solutions‖, Tata McGraw Hill Publishers, 2010.
2. Bensmith, and Brian Komer, ―Microsoft Windows Security Resource Kit‖, Prentice Hall of India, 2010.

REFERENCES

1. Kenneth C.Brancik, ―Insider Computer Fraud‖ Auerbach Publications Taylor & Francis Group, 2008.
2. Ankit Fadia, ― Ethical Hacking‖ 2nd Edition Macmillan India Ltd, 2006 MTCF -202 Database S
3. Stuart McClure, Joel Scambray and Goerge Kurtz, ―Hacking Exposed Network Security Secrets & Solutions‖, 5th Edition, Tata McGraw Hill Publishers, 2010.