COURSE STRUCTURE

B.TECH.

COMPUTER SCIENCE & ENGINEERING

Specialization

in

Cloud Computing & Virtualization

Under

Choice Based Credit System (CBCS)

Credits Distributions

Sr. No.	Category	No. of Credits
1	Humanities and Social Sciences (HS)	25
2	Basic Sciences (BS)	19
3	Engineering Sciences (ES)	24
4	Professional Core (PC)	48
5	Professional Elective (PE)	31
6	Open Elective (OE)	16
7	Project Work (PW)	17
8	Mandatory Non Credit Courses (MNC) (4 Courses)	-
Total		**180**

First Semester

S. NO.	CODE	SUBJECT	TEACHING SCHEME			CREDITS	CONTACTS HRS/WK
			L	T	P		
1.	BMAS0101	Engineering Mathematics I	3	1	0	4	4
2.	BPHS0001	Engineering Physics	3	1	0	4	4
3.	BELH0001	English Language Skills for Communication – I	2	0	0	2	2
4.	BECG0001	Electronics Engineering	3	1	0	4	4
5.	BCSG1001	Python Programming	3	0	0	3	3
6.	BCSC0600	Introduction to Open Source Software & Open Standards	2	0	0	2	2
PRACTICALS							
1.	BPHS0801	Engineering Physics Lab	0	0	2	1	2
2.	BELH0801	English Language Lab – I	0	0	2	1	2
3.	BECG0800	Electronics Lab I	0	0	2	1	2
4.	BMEG0801	Engineering Drawing Lab	0	0	2	1	2
5.	BCSG1800	Python Programming Lab	0	0	2	1	2
		TOTAL	**16**	**3**	**10**	**24**	**29**

Second Semester

S. NO.	CODE	SUBJECT	TEACHING SCHEME			CREDITS	CONTACTS HRS/WK
			L	T	P		
1.	BMAS0102	Engineering Mathematics II	3	1	0	4	4
2.	BELH0002	English Language Skills for Communication – II	2	0	0	2	2
3.	BEEG0001	Electrical Engineering	3	1	0	4	4
4.	BMEG0001	Basic Mechanical Engineering	3	1	0	4	4
5.	BCSG0002	Computer Programming	3	0	0	3	3
6.	BCSC0601	Web Programming through PHP	3	0	0	3	3
PRACTICALS							
1.	BELH0802	English Language Lab – II	0	0	2	1	2
2.	BEEG0800	Electrical Engineering Lab	0	0	2	1	2
3.	BMEG0800	Engineering Workshop Practice Lab	0	0	2	1	2
4.	BCSG0801	Computer Programming Lab	0	0	2	1	2
5.	BCSC0800	Web Programming Lab	0	0	2	1	2
		TOTAL	**17**	**3**	**10**	**25**	**30**

Program Core

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE-REQUISITES
			L	T	P	J			
THEORY									
1.	BCSC0002	Object Oriented Programming	3	0	0	0	3	3	Programming
2.	BCSC0003	Database Management System	3	0	0	0	3	3	
3.	BCSC0004	Operating Systems	3	0	0	0	3	3	
4.	BCSC1005	Computer Organization	3	0	0	0	3	3	
5.	BCSC0006	Data Structures and Algorithms	3	1	0	0	4	4	Programming
6.	BCSC0007	Introduction to Microprocessors	3	0	0	0	3	3	Computer Organization
7.	BCSC0008	Computer Networks	3	1	0	0	4	4	
8.	BCSC0009	Software Engineering	3	0	0	0	3	3	
9.	BCSC1010	Discrete Mathematics	3	1	0	0	4	4	
10.	BCSC0011	Theory of Automata and Formal Language	3	1	0	0	4	4	
11.	BCSC0012	Design and Analysis of Algorithms	3	0	0	0	3	3	Programming, Data Structures
12.	BCSE0101	Digital Image Processing	3	0	0	0	3	3	Mathematics, Programming
PRACTICALS									
1.	BCSC0801	Object Oriented Programming Lab	0	0	2	0	1	2	Programming Lab
2,	BCSC0802	Database Management System Lab	0	0	2	0	1	2	
3.	BCSC0803	Operating Systems Lab	0	0	2	0	1	2	
4.	BCSC0804	Computer Organization Lab	0	0	2	0	1	2	
5.	BCSC0805	Data Structures and Algorithms Lab	0	0	2	0	1	2	Programming Lab
6.	BCSC0806	Microprocessors Lab	0	0	2	0	1	2	
7.	BCSC0807	Design and Analysis of Algorithms Lab	0	0	2	0	1	2	Programming, Data Structures
8.	BCSE0131	Digital Image Processing Lab	0	0	2	0	1	2	Programming
Total			**36**	**4**	**16**	**0**	**48**	**56**	

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACT S HR/WK	PRE- REQUISITES
			L	T	P	J			
LIST OF PROGRAMME ELECTIVES									
THEORY									
1.	BCSC0600	Introduction to Open Source Software & Open Standards	2	0	0	0	2	2	
2.	BCSC0601	Web Programming through PHP	3	0	0	0	3	3	
3.	BCSE1502	Introduction to Virtualization and Cloud Computing	3	0	0	0	3	3	
4.	BCSE0503	Cloud Computing Architecture & Deployment Models	3	0	0	0	3	3	
5.	BCSE0508	Cloud and Business Process Management	3	0	0	0	3	3	
6.	BCSE0509	Cloud Security, Backup & Disaster Recovery	3	0	0	0	3	3	
7.	BCSE0510	Container Orchestration and Infrastructure Automation	3	0	0	0	3	3	
8.	BCSE0511	DevOps	3	0	0	0	3	3	
9.	BCSE0701	Introduction To Machine Learning	3	0	0	0	3	3	
10.	BCSE0556	Hadoop & Big Data Analytics	3	0	0	0	3	3	
11.	BCSE0252	Full Stack Using Node JS	3	0	0	0	3	3	
12.	BCSE0053	Agile Software Development	3	0	0	0	3	3	
13.	BCSE0602	IT Network Security	3	0	0	0	3	3	
14.	BCSE0203	Internet of Things	3	0	0	0	3	3	
15.	BCSC0013	Compiler Design	3	1	0	0	4	4	
PRACTICALS									
1.	BCSC0800	Web Programming Lab	0	0	2	0	1	2	
2.	BCSE0531	Virtualization Lab	0	0	2	0	1	2	
3.	BCSE0532	Cloud Deployment Lab	0	0	2	0	1	2	
4.	BCSE0536	Cloud and Business Process Management Lab	0	0	2	0	1	2	
5.	BCSE0537	Cloud Security, Backup & Disaster Recovery Lab	0	0	2	0	1	2	
6.	BCSE0538	Container Orchestration and Infrastructure Automation Lab	0	0	2	0	1	2	
7.	BCSE0539	DevOps Lab	0	0	2	0	1	2	
8.	BCSE0731	Introduction To Machine Learning Lab	0	0	2	0	1	2	
9.	BCSE0585	Hadoop & Big Data Analytics Lab	0	0	2	0	1	2	
10.	BCSE0282	Full Stack Using Node JS Lab	0	0	2	0	1	2	
11.	BCSE0632	IT Network Security Lab	0	0	2	0	1	2	
12.	BCSE0232	Internet of Things Lab	0	0	2	0	1	2	

Projects

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR /WK	PRE- REQUISITES
			L	T	P	J			
1.	BCSJ0950	Mini Project – I	0	0	0	0	2	0	
2.	BCSJ0951	Mini Project – II	0	0	0	0	2	0	
3.	BCSJ0971	Project – Part I	0	0	0	0	3	0	
4.	BCSJ0972	Project – Part II	0	0	0	0	8	0	
5.	BCSJ0991	Industrial Training	0	0	0	0	2	0	
		TOTAL	**0**	**0**	**0**	**0**	**17**	**0**	

Mandatory Non Graded Course

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE- REQUISITES
			L	T	P	J			
THEORY									
1.	BCSM0001	Introduction to Cyber Security	2	0	0	0	0	2	
2.	BCHM0101	Disaster Management	2	0	0	0	0	2	
3.	MBAM0001	Basic Course in Entrepreneurship	2	0	0	0	0	2	
4.	MBAM0002	Leadership And Organizational Behavior	2	0	0	0	0	2	
5.	BCHM0202	Environmental Studies	2	0	0	0	2	2	
6.	BELM0001	Introduction to Bhagavad Gita	2	0	0	0	2	2	

Humanities and Social Sciences

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE- REQUISITES
			L	T	P	J			
THEORY									
1.	BELH0001	English Language Skills for Communication – I	2	0	0	0	2	2	
2.	BELH0002	English Language Skills for Communication – II	2	0	0	0	2	2	
3.	BELH0003	English for Professional Purposes – I	2	0	0	0	2	2	
4.	BELH0004	English for Professional Purposes – II	2	0	0	0	2	2	
5.	BELH0006	Ethics & Values	2	0	0	0	2	2	
6.	MBAH0001	Industrial Management	3	0	0	0	3	3	
PRACTICALS									
1.	BELH0801	English Language Lab – I	0	0	2	0	1	2	
2.	BELH0802	English Language Lab – II	0	0	2	0	1	2	
3.	BTDH0301	Soft Skills – I	0	0	2	0	1	2	
4.	BTDH0302	Soft Skills – II	0	0	2	0	1	2	
5.	BTDH0303	Soft Skills – III	0	0	8	0	4	8	
6.	BTDH0304	Soft Skills – IV	0	0	8	0	4	8	
TOTAL			**13**	**0**	**24**	**0**	**25**	**37**	

Basic Sciences

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACT S HR/WK	PRE- REQUISITES
			L	T	P	J			
THEORY									
1.	BMAS0101	Engineering Mathematics I	3	1	0	0	4	4	
2.	BMAS0102	Engineering Mathematics II	3	1	0	0	4	4	
3.	BMAS1103	Engineering Mathematics III	3	1	0	0	4	4	
4.	BPHS0001	Engineering Physics	3	1	0	0	4	4	
5.	BCHS0201	Environmental Studies	2	0	0	0	2	2	
PRACTICALS									
1.	BPHS0801	Engineering Physics Lab	0	0	2	0	1	2	
TOTAL			**17**	**5**	**4**	**0**	**24**	**26**	

Engineering Sciences

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE- REQUISITES
			L	T	P	J			
THEORY									
1.	BEEG0001	Basic Electrical Engineering	3	1	0	0	4	4	
2.	BECG0001	Electronics Engineering	3	1	0	0	4	4	
3.	BMEG0001	Basic Mechanical Engineering	3	1	0	0	4	4	
4.	BCSG1001	Python Programming	3	0	0	0	3	3	
5.	BCSG0002	Computer Programming	3	0	0	0	3	3	
PRACTICALS									
1.	BEEG0800	Electrical Engineering Lab	0	0	2	0	1	2	
2.	BECG0800	Electronics Lab I	0	0	2	0	1	2	
3.	BMEG0800	Engineering Workshop Practice Lab	0	0	2	0	1	2	
4.	BMEG0801	Engineering Drawing Lab	0	0	2	0	1	2	
5.	BCSG1800	Python Programming Lab	0	0	2	0	1	2	
6.	BCSG0801	Computer Programming Lab	0	0	2	0	1	2	
Total			**15**	**3**	**12**	**0**	**24**	**25**	

BCSG1001: PYTHON PROGRAMMING

Objective: *This course introduces the solving of mathematical problems using Python programming using Object Oriented concepts and its connectivity with database.*

Credits:05 **L-T-P-J:4-1-0-0**

Module No.	Content	Teaching Hours
I	Introduction to Python: Introduction and Basics; Setting up path Python Data Variables & Operators: Data Variables and its types, id () and type () functions, Coding Standards; **Control Structures:** if-else, elif, Nested if, Iteration Control structures, Break, Continue & Pass; **String Manipulation:** Accessing Strings, Basic Operations, String slices Function and Methods. **Lists:** Introduction, accessing list, Operations, Working with lists, Function and Methods. **Tuple:** Introduction, accessing tuples, Operations, Working, Functions and Methods. **Dictionaries:** Introduction, accessing values in dictionaries, Working with dictionaries, Properties, Functions.	22
II	**Functions:** Defining & Calling a function, Passing arguments to functions – Mutable & Immutable Data Types, Different types of arguments, Recursion, Scope of variables; **Modules and Packages:** User-defined modules and Standard Library: random, numpy, scipy, sys, Math Module, String Module, List Module, Date & Time Module, Regular Expressions: match, search, replace; **Input-Output:** Printing on screen, reading data from keyboard, Opening and closing file, Reading and writing files, Functions. **Exception Handling:** Exception, Exception Handling, except clause, try? finally clause, User Defined Exceptions. Basics of Python for Data Analysis, Introduction to series and data frames& Python using Pandas.	22

Text Books:

- Paul Barry: "Head First Python "O'Reilly Media, Inc.", 2010.

Reference Books:

- Bret Slatkin: "Effective Python: 59 Specific ways to write better Python", Addison Wesley, 2015.

Outcome: After completion of course, the student will be able to:

- CO1: Understand the basics of Python Programming.
- CO2: Apply the concepts of control structures and string manipulations of python programming.
- CO3: Understand the use of data structures available in Python List, Tuple and Dictionary.
- CO4: Experiment user-defined functions and access built-in functions.
- CO5: Experiment user-defined modules and access built-in modules- math, random, string, date, time, date time.
- CO6: Develop the programs using the concept of File Handling.
- CO7: Develop programs based on Exceptional Handling.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO2/PSO4
CO2	PO4/PSO1
CO3	PO5/PSO4
CO4	PO5,PO7/PSO1
CO5	PO2,PO8/PSO4
CO6	PO3,PO10/PSO2
CO7	PO5,PO9/PSO1

BCSG0002: COMPUTER PROGRAMMING

Objective: *To impart adequate knowledge on the need of problem solving techniques and develop programming skills to implements applications using the concepts of C Language. Also by learning the programming constructs they can easily switch over to any other language in future.*

Credits:05 **L-T-P-J:4-1-0-0**

Module No.	Content	Teaching Hours
I	**Generation of Programming Languages:** Low, Assembly, High and 4GL. **Language Processors:** Compiler, Interpreter, Assembler, Linker and Loader. **Algorithm:** Introduction, Features, Different Ways of stating Algorithms. **Flow Chart:** Introduction, Standard, Guidelines, Advantages and Limitations of using Flowcharts. **Basics of C:** Overview, Structure of a C program, Identifier, Keywords, Variables, Data types, Formatted Input and output. **Operators and Expression:** Assignment, Unary, Arithmetic, Relational, Logical, Bitwise, Conditional, Special operators and their precedence & Associativity. IEEE representation of data types like float & double, Lvalue and Rvalue **Type Conversion:** Type Promotion in expression, Conversion by Assignment, Truncation and Casting Arithmetic expression. **Decision and Case Control Structure:** if, if-else, nested if-else, Decisions using switch, switch versus if-else ladder, goto. **Loop Control Structure:** For loop, while loop, do-while loop, nesting of loops, break, and continue. **Arrays:** Introduction, one-dimensional and two-dimensional Array-Declaration, Initialization, Address Calculation. **Operations on Arrays:** Insertion, Deletion, Linear Search & Bubble Sort. **String:** Introduction, One dimensional and two dimensional Array-Declarations, Initialization **Operations on String:** Length, Copy, Reverse, Concatenate, Compare with & without built-in functions.	25
II	**Functions:** Declaration and Definition, Category of Functions, Parameter Passing Techniques – Call by Value, Passing Arrays to Functions. **Introduction to Storage Classes:** Auto, Static, Extern and Register. **Recursion:** Mechanics of Recursive Call, Implementation of Recursion, Recursion vs. Iteration. **The C Preprocessor:** Introduction, Macro Expansion and File Inclusion, Conditional Compilation and Miscellaneous Directives. **Pointers:** Declaration and Initialization of Pointer Variables, Accessing a Variable through its Pointer, Arrays and Pointers, Pointer and Strings, Pointer Arithmetic, Pointers to Pointers, Array of Pointers, Pointer to an Array, Two Dimensional Array and Pointers, Pointers to Functions, Dynamic Memory Allocation, void Pointer and Null Pointer. **User Defined Types:** enum, typedef, Union and Structure - Declaration, Initialization, Nested Structures, Arrays of Structures, Structure and Pointer, Passing Structure Through Function. Difference between Structures and Union. **File Handling:** Data and Information, File Concepts, File Organization, File Operations: Open, Read, and Close, Trouble in Opening a File. File Opening Modes, Working with Text Files. Random Access to Files of Records. Introduction to Command Line Arguments.	25

Text Books:

- Behrouz A. Forouzan and Richard F. Gilberg, "Computer Science – A Structured Programming Approach Using C", C Language Learning, 2007

Reference Books:

- Herbert Schildt, "C: The Complete Reference", 5th Edition, McGraw Hill Education
- K. N. King, "C Programming a Modern Approach", W. W. Norton, 2nd Edition, 2008.
- Kernighan and Ritche, "The C Programming Language", PHI, 2ndEdition, 2011.
- P. Dey and M. Ghosh, "Programming in C", Oxford University Press 2nd Edition, 2013.

Outcome: After completion of course, the student will be able to:

- CO1: Understand the basic concepts of problem solving skills.
- CO2: Apply the basic principles of programming in C language.
- CO3: Understand the concepts of arrays and strings in C language.
- CO4: Apply the concepts of functions to solve real world problems.
- CO5: Illustrate the concepts of recursion.
- CO6: Understand the concepts of pointers in C language.
- CO7: Understand the basic concepts of file handling.
- CO8: Develop algorithmic solutions to simple computational problems.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2,PO4,PO12/PSO1,PSO3
CO2	PO1,PO2,PO3,PO10/PSO1,PSO3
CO3	PO1,PO2,PO3,PO4/PSO1,PSO3
CO4	PO1,PO3, PO12/PSO1,PSO2
CO5	PO1,PO2,PO4 /PSO1,PSO3
CO6	PO1,PO2,PO3,PO4/PSO1,PSO2
CO7	PO1,PO3,PO6 /PSO1
CO8	PO1,PO2,PO4,PO10,PO12/PSO1,PSO3

BCSC0002: OBJECT ORIENTED PROGRAMMING

Objective: *This course introduces the Object-Oriented programming paradigm to students. It also teaches a student how to think objectively and model a Java program for solving real-world problems.*

CREDITS: 3 **L-T-P-J:3-0-0-0**

Module No.	Content	Teaching Hours
I	**Object-Oriented Programming:** Features of Object-Oriented Programming, Introduction to Object-Oriented Java Programming. **g Java Technology & Environment:** Understanding the compilation process of the JVM, JVM vs JDK vs JRE, Key Features of Java, Structure of a simple Java program. **Working with Java Primitive Data Types:** Strongly Typed nature of Java, Primitive Data Types in Java, The new 'var' keyword, Scope of a variable. **Accepting User Input in Java Programs:** using the Scanner class, using command line arguments. **Programming Constructs:** Sequence, Selection, Iteration & Transfer Statements, For-Each Loop. **Working with Java Arrays**: Declaring and Initializing One-Dimensional and Two-Dimensional Arrays in Java, Introduction to java. util. Arrays class. **The String API:** String Data Type, commonly used methods from the String API, String Tokenizer, String Builder & String Buffer. **Creating and Using Methods:** Signature of a method, Types of Methods, Overloading methods in a class, Static and Non-Static Methods. **Describing and Using Objects & Classes:** Declare the structure of a Java class, declaring members of a class (fields and methods), declaring and using Java Objects, lifecycle of an Object (creation, assignment, dereferencing and garbage collection), Constructors of a class, Overloading Constructors, Constructor chaining using 'this' and 'super' keyword. **Using Java Packages:** create and import Java packages and static imports, abstracting program logic to packages, creating executable main class, running the executable class inside a package. **Applying Encapsulation**: Using access modifiers with/in a class, principles of encapsulation. **Programming Abstractly Through Interfaces:** create and implement Interfaces for programs, private and default methods in Interfaces, declaring Abstract Classes, Constructors in Abstract Classes. Marker Interface, Functional Interfaces, Lambda Expressions in Java.	20
II	**Reusing Implementations using Inheritance:** Declaring Subclasses and Super classes, extend Abstract Classes, implementing Interfaces, exploring polymorphic behavior by overriding methods, Object Types vs Reference Types, differentiate overloading, overriding and hiding. **Exception Handling:** Exception Hierarchy, Need of Exception Handling, Checked Exceptions, Unchecked Exceptions and Errors, Try-Catch Blocks, Finally, Throw & Throws Keywords, creating and handling Custom Exceptions. **Threads in Java:** Life Cycle of a Thread, creating threads using Runnable and Thread, 'sleep ()', Thread Priorities. **Using Wrapper Classes:** Wrapper Classes in Java, Boxing-Unboxing-Auto Boxing-Auto Unboxing. **Generics & Collections:** Creating Generic classes, Generic Methods, Diamond Notation, Wildcards, Type Erasure, Collection Hierarchy, Base Interfaces, Lists, Sets and Maps. **The Stream API:** Introduction to the Stream API, using lambda expressions in Streams. **Regular Expressions:** Pattern and Matcher Class.	18

	JDBC: JDBC Drivers, Connecting to a MySQL Database, Driver Manager, Connection Interface, Statement Interface, Result Set Interface, Prepared Statements.	

Text Book:

- Herbert Schildt, "The Complete Reference, Java Eleventh Edition", Oracle Press.2019.

Reference Book:

- Cay S Hosrtmann, "Core Java Volume I—Fundamentals, Eleventh Edition", Pearson,2018.
- Rogers Cadenhead, "Sams Teach Yourself Java in 21 Days (Covers Java 11/12), 8th Edition", Pearson,2020.

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

- CO1: Understand the basics of Object-Oriented Programming paradigm.
- CO2: Construct the logical flow of programs by using the sequence, selection, iterations and transfer statements.
- CO3: Apply the concepts of Object-Oriented Programming to model programs in Classes, Abstract Classes, Interfaces and Enums, and simplify program function by dissecting it into methods.
- CO4: Understand accessibility of members in a program unit and create packages to prevent namespace collisions.
- CO5: Predict run-time errors in a program by examining program functioning.
- CO6: Show the parallel processing capabilities of a program using a multithreading concept.
- CO7: Experiment with the predefined classes and interfaces defined in the Collections Framework.
- CO8: Develop a program using JDBC connectivity to demonstrate data persistence.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO3/PSO1,PSO2
CO2	PO1,PO3/PSO1,PSO2
CO3	PO1,PO2/PSO1,PSO2
CO4	PO1/PSO2,PSO4
CO5	PO1,PO2,PO4/PSO4
CO6	PO1,PO2, PO3/PSO2
CO7	PO1,PO2,PO11/PSO2
CO8	PO1,PO2,PO3/PSO1,PSO2

BCSC0003: DATABASE MANAGEMENT SYSTEM

Objective: *The objective of the course is to enable students to understand and use a relational database& NoSQL system. Students learn how to design and create a good database.*

Credits: 03 **L-T-P-J: 3-0-0-0**

Module No.	Content	Teaching Hours
I	Introduction**:** An Overview of Database Management System, Database System Vs File System, Database System Concept and Architecture, Data Model Schema and Instances, Data Independence, Database Language and Interfaces (DDL, DML, DCL), Database Development Life Cycle (DDLC) with Case Studies. Data Modeling Using the Entity-Relationship Model**:** ER Model Concepts, Notation for ER Diagram, Mapping Constraints, Keys, Specialization, Generalization, Aggregation, Reduction of an ER Diagram to Tables, Extended ER Model. Relational Data Model and Language**:** Relational Data Model Concepts, Integrity Constraints, Entity Integrity, Referential Integrity, Keys Constraints, Domain Constraints, Relational Algebra **Database Design & Normalization I:** Functional Dependencies, Primary Key, Foreign Key, Candidate Key, Super Key, Normal Forms, First, Second, Third Normal Forms, BCNF, Non-Redundant Cover, Canonical Cover	20
II	**Database Design & Normalization II:** 4th Normal Form, 5th Normal Form, Lossless Join Decompositions, MVD and JDs, Inclusion Dependence. **File Organization:** Indexing, Structure of Index files and Types, Dense and Sparse Indexing **Transaction Processing Concept:** Transaction System, Testing of Serializability, Serializability of Schedules, Conflict & View Serializable Schedule, Recoverability, Recovery from Transaction Failures, Log Based Recovery, Deadlock Handling. **Concurrency Control Techniques:** Concurrency Control, Locking Techniques for Concurrency Control, 2PL, Time Stamping Protocols for Concurrency Control, Validation Based Protocol. **Distributed Database:** Introduction of Distributed Database, Data Fragmentation and Replication.	20

Text Books:

- Elmasri and Navathe, "Fundamentals of Database Systems", 6th Edition, Addison Wesley,2010.
- Sadalage, P. &Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Pearson Education,2012.

References Books:

- Date C J," An Introduction to Database Systems", 8th Edition, Addison Wesley.
- Korth, Silbertz and Sudarshan, "Database Concepts", 5th Edition, TMH,1998.
- Redmond, E. & Wilson, "Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement", 1st Edition.

Outcome: After the completion of the course, the student will:

- CO1: Understand the concept of database management systems and Relational database.
- CO2: Identify the various data model used in database design.
- CO3: Design conceptual models of a database using ER modeling for real life applications and construct queries in Relational Algebra.
- CO4: Create and populate a RDBMS for a real life application, with constraints and keys using SQL.
- CO5: Select the information from a database by formulating complex queries in SQL.
- CO6: Analyze the existing design of a database schema and apply concepts of normalization to design an optimal database.
- CO7: Discuss indexing mechanisms for efficient retrieval of information from a database.
- CO8: Discuss recovery system and be familiar with introduction to web database, distributed databases.

Mapping of Course Outcomes (COs) withProgram Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1 /PSO1
CO2	PO2, PO3/ PSO2
CO3	PO2,PO3,PO6,PO11/PSO1,PSO2,PSO4
CO4	PO1,PO3/PSO1
CO5	PO1,PO5/PSO1
CO6	PO2,PO3,PO9/ PSO2
CO7	PO1,PO11 /PSO1
CO8	PO1,PO3,PO12/ PSO2

BCSC0004: OPERATING SYSTEMS

Objective: *This course aims to introducing the concept of computer organization. In particular, it focuses on basic hardware architectural issues that affect the nature and performance of software.*

Credits:03 **L-T-P-J:3-0-0-0**

Module No.	Content	Teaching Hours
I	**Introduction:** Operating System and its Classification - Batch, Interactive, Multiprogramming, Time sharing, Real Time System, Multiprocessor Systems, Multithreaded Systems, System Protection, System Calls, Reentrant Kernels, Operating System Structure- Layered structure, Monolithic and Microkernel Systems, Operating System Components, Operating System Functions and Services. **Processes:** Process Concept, Process States, Process State Transition Diagram, Process Control Block (PCB), Process Scheduling Concepts, Threads and their management. **CPU Scheduling:** Scheduling Concepts, Performance Criteria, Scheduling Algorithms, Multiprocessor Scheduling. **Process Synchronization:** Principle of Concurrency, Implementation of concurrency through fork/join and parbegin/parend, Inter Process Communication models and Schemes, Producer / Consumer Problem, Critical Section Problem, Dekker's solution, Peterson's solution, Semaphores, Synchronization Hardware. **Classical Problem in Concurrency:** Dining Philosopher Problem, Readers Writers Problem.	20
II	**Deadlock:** System model, Deadlock characterization, Prevention, Avoidance and detection, Recovery from deadlock, Combined Approach. **Memory Management:** Multiprogramming with fixed partitions, Multiprogramming with variable partitions, Paging, Segmentation, Paged segmentation. **Virtual memory concepts:** Demand paging, Performance of demand paging, Page replacement algorithms, Thrashing, Locality of reference. **I/O Management and Disk Scheduling:** I/O devices, I/O subsystems, I/O buffering, Disk storage and disk scheduling. **File System:** File concept, File organization and access mechanism, File directories, File allocation methods, Free space management.	20

Text Books:

- Silberschatz, Galvin and Gagne, "Operating Systems Concepts",9th Edition, Wiley, 2012.

Reference Books:

- SibsankarHalder and Alex a Aravind," Operating Systems", 6th Edition, Pearson Education, 2009.
- Harvey M Dietel, "An Introduction to Operating System", 2nd Edition, Pearson Education, 2002.
- D M Dhamdhere, "Operating Systems: A Concept Based Approach", 2nd Edition, 2006.
- M. J. Bach, "Design of the Unix Operating System", PHI, 1986.

Outcome: After completion of course, the student will be able to:

- CO1: Understand the classification of operating system environment.
- CO2: Understand the basic of process management.
- CO3: Apply the concept of CPU process scheduling for the given scenarios.
- CO4: Illustrate theprocess synchronization and concurrency process in operating system.
- CO5: Analyze the occurrence of deadlock in operating system.
- CO6: Describe and analyze the memory management and its allocation policies.
- CO7: Understand the concepts of disk scheduling.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2,PO7/PSO1
CO2	PO1,PO2 /PSO1
CO3	PO1,PO4/PSO1,POS3
CO4	PO3,PO4,PO6/PSO3,PSO4
CO5	PO1,PO4/PSO1,PSO3
CO6	PO1,PO2/PSO1,PSO3
CO7	PO1,PO2,PO7/PSO1,PSO3

BCSC1005: COMPUTER ORGANIZATION

Objective: *This course aims to introducing the concept of computer organization. In particular, it focuses on basic hardware architectural issues that affect the nature and performance of software.*

Credits: 03 **L–T–P-J: 3–0–0-0**

Module No.	Content	Teaching Hours
I	**Basic Organization:** Basic organization of the computer and Block level description of the functional units, Number representation; 1's and 2's Complement, Integer Representation, Arithmetic Addition & Subtraction with overflow. fixed and floating-point number representation, IEEE standard floating point representation. Introduction to Combinational Circuit- half adder, full adder, binary adder/subtractor, carry look ahead adders. Multiplexer and Demultiplexer, Register, bus and memory transfer. **Central Processing Unit:** Addition and subtraction of signed numbers, Multiplication: Signed operand multiplication, Booths algorithm. Processor organization, general registers organization, stack organization, Three, Two, One & Zero address instruction. Addressing modes, Micro-operations (Arithmetic, Logical & Shift) and its applications.	20
II	**Multiprogramming and Multiprocessing:** Flynn's classification, Introduction to pipelined operation. Instruction types, formats, Instruction cycles. **Control Unit:** Execution of a complete instruction. Hardwired and micro programmed control unit. Unconditional and Conditional branching. Microinstruction with next address field, pre-fetching microinstructions, Concept of horizontal and vertical microprogramming. **Memory:** Basic concept of Memory and its hierarchy, RAM memories, 2D, 2 & 1/2D memory organization. ROM memories. Cache memories: concept and design issues, performance, address mapping and replacement. Virtual memory: concept and implementation. **Input/Output:** Peripheral devices, I/O interface, I/O ports, Interrupts: interrupt hardware, types of interrupts and exceptions. Buses, bus architecture, types of buses and bus arbitration. Modes of Data Transfer: Programmed I/O, interrupt initiated I/O and Direct Memory Access., I/O channels and processors. Standard communication interfaces.	20

Text Books:

- M. Mano, "Computer System Architecture", 3rd Edition, PHI,1996

Reference Books:

- D.W. Patterson, "Computer Organization and Design", 4th**Edition**, Elsevier Publication, 2008.
- William Stalling, "Computer Organization",8th Edition, PHI, 2011.
- V. CarlHamacher, Zaky, "Computer Organization",4th International Edition, TMH, 1996.
- John P Hays, "Computer Organization", 2nd Edition, TMH.
- Tannenbaum , "Structured Computer Organization", 5thEdition, PHI, 2005.
- P Pal Chaudhry, "Computer Organization & Design", 2ndEdition, PHI, 2002.

Outcome: After completion of the course, the student will be able to:

- CO1: Understand the basics of digital computer system.
- CO2: Demonstrate the principle of arithmetic operations on unsigned, signed integers and floating point numbers.
- CO3: Understand the concepts of Combinational and Sequential circuits and their applications.
- CO4: Understand the CPU architecture and organization.

- CO5: Explain the basic concepts of pipelining.
- CO6: Design the steps for the execution of the complete instruction for hardwired and micro-programmed control unit.
- CO7: Explain the function of memory hierarchy.
- CO8: Determine the interface of CPU with input/output devices and their modes of transfer.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO3/PSO1
CO2	PO1,PO3/PSO1
CO3	PO2,PO3,PO5/PSO2
CO4	PO2,PO3,PO4/PSO1,PSO3
CO5	PO2,PO3,PO4/PSO2
CO6	PO1,PO2,PO3/PSO1,PSO3
CO7	PO2,PO3,PO5/PSO2,PSO3
CO8	PO3,PO4/PSO1

BCSC0006: DATA STRUCTURES AND ALGORITHMS

Objective: *The objective of this course is that students will construct and application of various data structures and abstract data types including lists, stacks, queues, trees and graphs.*

Credits: 04 **L–T–P-J: 3–1–0-0**

Module No.	Content	Teaching Hours
I	**Introduction:** Basic Terminology, Elementary Data Organization, Properties of an Algorithm, Efficiency of an Algorithm, Time and Space Complexity, Asymptotic Notations – Big-Oh; Operations on Data Structure, Abstract Data Types (ADT). **Linked Lists:** Implementation of Singly Linked Lists, Doubly Linked List, Circular Linked List, Operations on a Linked List - Insertion, Deletion, Traversal; Generalized Linked List, Polynomial Representation and Addition. **Stacks:** Primitive Stack Operations - Push & Pop, Array and Linked Implementation of Stack in C, Application of Stack: Prefix and Postfix Expressions, Evaluation of Postfix Expression, conversion of Infix to Postfix expression, Recursion, Principles of Recursion, Tail Recursion, Removal of Recursion, use of stack in Recursion, Tower of Hanoi Problem. **Queues:** Operations on Queue - Add, Delete operations, Implementation of Queue Using Array and Linked List, Circular Queues, Deque and Priority Queue. **Trees:** Basic Terminology, Array Representation and Dynamic Representation; Complete Binary Tree, Algebraic Expressions, Extended Binary Trees, Tree Traversal Algorithms - Inorder, Preorder and Postorder; Threaded Binary Trees, Traversing Threaded Binary Trees.	20
II	**Search Trees:** Binary Search Trees (BST), Insertion and Deletion in BST, AVL Trees, Introduction to M-Way Search Trees, B Trees. **Searching:** Sequential Search, Binary Search. **Sorting:** Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Two Way Merge Sort, and Heap Sort. **Graphs:** Terminology, Adjacency Matrices, Adjacency List, Graph Traversal - Depth First Search and Breadth First Search; Spanning Trees, Minimum Cost Spanning Trees – Prim's and Kruskal's Algorithm; Shortest Path Algorithm – Bellman-Ford and Dijkstra's Algorithm. **Hashing & Indexing:** Hash Function, Collision Resolution Strategies. Primary Indices, Secondary Indices, Indexing and Hashing Comparisons.	20

Text Book:

- Aaron M. Tanenbaum, YedidyahLangsam and Moshe J. Augenstein, "Data Structures Using C and C++", 2nd Edition, PHI, 2009.

Reference Books:

- Horowitz and Sahani, "Fundamentals of Data Structures", 3rd Edition, W H Freeman & Co, 2004-05.
- Jean Paul Trembley and Paul G. Sorenson, "An Introduction to Data Structures with Applications", 2nd Edition, TMH, 2007.
- R. Kruse, "Data Structures and Program Design in C", 2nd Edition, Pearson Education, 2004.
- Lipschutz Schaum's Outline Series, "Data Structures", 12th Reprint, TMH, 2010.
- G A V Pai, "Data Structures and Algorithms", TMH, 2009.

Outcome: After completion of course, student will be able to:

- CO1: Understand the basic concepts of the data structure and algorithms.
- CO2: Understand the complexity representation in terms of Big Oh, Theta and Omega notations.
- CO3: Apply the associated operations in linear data structure like stack, Queue and link list.
- CO4: Apply the associated operations in Binary Search Tree, AVL Tree and M- Way Search Tree.
- CO5: Understand the basic algorithms such as heap sort, graph traversal, quick sort, AVL trees, and hashing.
- CO6: Select the appropriate data structure to solve the problem.
- CO7: Apply the shortest path algorithm to solve real life problem.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO1,PSO2
CO2	PO1, PO2/PSO1,PSO2
CO3	PO1/PSO1
CO4	PO1,PO4/PSO1
CO5	PO1,PO4/PSO3
CO6	PO2/PSO4
CO7	PO2/PSO4

BCSC0007: INTRODUCTION TO MICROPROCESSORS

Objective: *Objective of this subject is to introduce the basic concepts of microprocessor and assembly language programming. Identify and explain the operation of the components of typical microprocessor: the role of the ALU, registers, stack and the use of interrupts.*

Credits: 03 **L–T–P-J: 3–0–0-0**

Module No.	Content	Teaching Hours
I	**Introduction**: Microprocessors Evolution and Types, Basics of Pentium Microprocessor, Microprocessor Application, **8-Bit Microprocessor:**8085 Microprocessor and its Architecture, Addressing Modes, The 8085 Programming Model, Instruction Classification, Instruction Format, Overview of Instruction Set - Data Transfer Operation, Arithmetic Operation, Logic Operations and Branch Operations; Introduction to Assembly Language Program. **Programming Technique with Additional Instruction**: Looping, Counting, Indexing, Additional Data Transfer and 16-Bit Arithmetic Instruction, Counters and Time Delays, Stack and Subroutine.	20
II	**16 Bit Microprocessor:** Architecture of 8086 – Register Organization, Execution Unit, Bus Interface Unit, Signal Description, Physical Memory Organization, Mode of Operation, I/O Addressing Capabilities. **Peripheral Interfacing:** I/O Programming, Programmed I/O, Interrupt Driven I/O, DMA I/O, Memory-Mapped I/Os. **Peripheral Devices:** 8237 DMA Controller, 8255 Programmable Peripheral Interface, 8253/8254 Programmable Timer/Counter, 8259 Programmable Interrupt Controller.	18

Text Books:

- N Senthil Kumar, MSaravanan, and S Jeevananthan, "Microprocessors and Microcontrollers", Oxford University Press India, 2010.

Reference Books:

- Ramesh S. Gaonkar , "Microprocessor Architecture Programming and Applications with 8085", 4th Edition, Penram International Publishing, 2000.
- Ray A.K. Bhurchandi.K.M, "Advanced Microprocessor and Peripherals", TMH, 2002.
- D. V. Hall, "Microprocessors and Interfacing: Programming and Hardware", 2nd Edition, TMH, 1992.
- Y.C. Liu and G.A. Gibson, "Microcomputer Systems: The 8086/8088 Family Architecture Programming and Design", 2nd Edition, PHI, 2003.

Outcome: After the completion of the course, the student will be able to:

- CO1: Demonstrate the Microprocessor internal architecture and its operations.
- CO2: Develop programs based on 8085 microprocessor instruction set and addressing mode.
- CO3: Develop program using looping, counting, indexing, counter and time delays.
- CO4: Understand the concept of stack and subroutine for modular approach.
- CO5: Compare accepted standards and guidelines to select microprocessor (8085 & 8086) to meet performance requirements.
- CO6: Analyze the concept of interfacing the processor to external device with I/O programming & Interrupt Driven I/O.
- CO7: Understand the working of interfacing chips (8237, 8253/54, 8255 & 8259).

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2/PSO1
CO2	PO2,PO3/PSO1,PSO2
CO3	PO2,PO3/PSO1,PSO2
CO4	PO1,PO2,PO3/PSO1,PSO3
CO5	PO2,PO3,PO5/PSO1,PSO3
CO6	PO1,PO2/PSO3
CO7	PO1,PO2,PO4/PSO3

BCSC 0008: Computer Networks

Objective: *The objective is to understand fundamental underlying principles of computer networking, details and functionality of layered network architecture.*

Credits: 03 **Semester - IV** **L–T–P-J: 3–1–0-0**

Module No.	Content	Teaching Hours
I	**Introduction Concepts**: Goals and Applications of Networks, Network structure and architecture, The OSI reference model, services, Network Topology Design, Physical Layer Transmission Media, Line coding scheme, switching methods (circuit switching, Packet switching), TDM. **Medium Access sub layer**: Medium Access sub layer - Channel Allocations, LAN protocols - ALOHA protocols, CSMA, CSMA/CD, Overview of IEEE standards. **Data Link Layer**: Error detection and correction, Flow control (sliding window protocol)	20
II	**Network Layer**: Network Layer –IP addressing, subnet, CIDR, VLSM, Internetworking, Address mapping, routing. Connecting devices. **Transport Layer**: Transport Layer - Design issues, connection management, Flow control, TCP window management, congestion control-slow start algorithm. **Application Layer**: Data compression, Data Encryption, File Transfer, DNS, HTTP, SMTP, TELNET **Introduction to IPv6, transition from IPv4 to IPv6.**	20

Text Books:

- Forouzan B. A. , "Data Communication and Networking", 4th Edition, McGrawHill,2004.

References:

- Kurose, J.F. and Ross K.W., "Computer Networking: A Top-Down Approach Featuring the Internet", 3rd Edition, Addison-Wesley,2005.
- A.S. Tanenbaum, "Computer Networks", 2nd Edition, Prentice Hall India,2006.

Outcome: After the completion of the course, the student will be able to:

- CO1: Understand the concept of OSI and TCP/IP reference model.
- CO2: Understand the basics of data transmission at physical layer.
- CO3: Understand the channel allocation using ALOHA, CSMA and CSMA/CD.
- CO4: Apply error detection and correction technique to eliminate transmission error.
- CO5: Analyze the fixed and variable length address (IPv4) subneting for the given scenarios.
- CO6: Understand the design issues of the transport layer.
- CO7: Understand the mechanism of protocols at application layer such as FTP, HTTP, Telnet, DNS.
- CO8: Understand IPv6 addressing and differentiate it from IPv4.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO3,PO12/PSO1
CO2	PO1/PSO2
CO3	PO1,PO4/PSO1,PSO4
CO4	PO1,PO3/PSO1
CO5	PO1,PO3,PO4,PO6/PSO3
CO6	PO2,PO4/PSO1
CO7	PO5,PO12/PSO2
CO8	PO4,PO7/PSO4

BCSC0009: SOFTWARE ENGINEERING

Objective: *Be employed in industry, government, or entrepreneurial endeavors to demonstrate professional advancement through significant technical achievements and expanded leadership responsibility.*

L–T–P-J: 3–0–0-0

Credits: 03

Module No.	Content	Teaching Hours
I	**Introductory Concepts**: The evolving role of software – characteristics, components and applications. **Process Models:** Waterfall Model, Prototyping, Incremental, Spiral. **Agile software Development**: Introduction to Agile, Agile software development framework. **Software Requirement Specification**: Requirement Process, SRS Components, Requirement Specifications with Use Cases Diagram. **Software Project Planning:** Project Planning Objectives. **Software Metrics:** Size, Function Point, Staffing, Project Estimation Methods–COCOMO Model. **Function-Oriented Design**: Problem Partitioning, Abstraction, Top Down and Bottom Up Design. **Module-Level Concepts:** Coupling, Cohesion, Design Notation and Specification - Structure Charts; Structured Design Methodology - Data Flow Diagram, Sequence Diagram.	20
II	**OO Analysis and OO Design:** OO Concepts, Introduction to UML Design Patterns: Class Diagram, Activity Diagram, State Chart Diagram. **Coding**: Coding Process, Verification – Code Inspections, Software Metrics. **Testing Fundamentals**: Test Case Design, Black Box Testing Strategies, White Box Testing, Unit Testing, Integration Testing, System Testing. **Introduction to Automation Testing and Testing Tools:** Automated Testing Process, Framework for Automation Testing, Introduction to Automation Testing Tool. **Software Quality:** Models, ISO 9000 Certification for Software Industry, SEI Capability Maturity Model. **Software Maintenance**: Models Cost of Maintenance, Re-engineering, Reverse Engineering.	18

Text Books:

- R. S. Pressman , "Software Engineering: A Practitioners Approach", 7thEdition, McGraw Hill,2010.

Reference Books:

- K. K. Aggarwal and Yogesh Singh , "Software Engineering", 3rd Edition, New Age International Publishers,2008.
- Rajib Mall , "Fundamentals of Software Engineering", 3rd Edition, PHI Publication,2009.
- R.E Fairley , "Software Engineering", McGraw Hill,2004.
- Sommerville, "Software Engineering", 9th Edition, Pearson Education,2010.

Outcome: After the completion of the course, the student will be able to:

- CO1: Understand the basic concepts of software engineering.
- CO2: Apply software processes to solve real world problems.
- CO3: Estimate the cost, effort and schedule of software using COCOMO Model.
- CO4: Analyze the software design techniques (structure chart, SDM, sequence diagram).
- CO5: Understand the basic concepts of OO analysis and design.
- CO6: Develop the test cases to validate the software.
- CO7: Understand the basic models of software Quality and maintenance.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO7/PSO1
CO2	PO2,PO3/PSO4
CO3	PO2,PO11/PSO3
CO4	PO3,PO10/PSO4
CO5	PO3,PO7/PSO1
CO6	PO5,PO12/PSO2
CO7	PO4,PO9,PO12/PSO1

BCSC1010: DISCRETE MATHEMATICS

Objective: *The objective is to introduce students to language and methods of the area of Discrete Mathematics. The focus of the module is on basic mathematical concepts in discrete mathematics and on applications of discrete mathematics in computer science.*

Credits: 4 **L–T–P-J: 3–1–0-0**

Module No.	Content	Teaching Hours
I	**Sets, Relations and Functions:** Introduction to Set Theory, Venn diagrams, algebra of Sets, Inclusion-Exclusion Principle, Partitions, Relations, Properties and their types, Function and their types. Recurrence Relations and Generating Functions **Introduction to Counting Principle**: Permutation, Combination, Permutation with Repetition, Combination with Repetition, Pigeonhole Principle. **Posets & Lattices:** Definition, Properties of lattices – Bounded, Complemented, Modular and Complete lattice. **Natural Numbers:** Introduction, Mathematical Induction, Variants of Induction, Induction with Nonzero Base cases. Proof Methods, Proof by counter – example, Proof by contradiction.	20
II	**Propositional Logic** - Logical Connectives, Truth Tables, Normal Forms (Conjunctive and Disjunctive), Validity; **Predicate Logic** - Quantifiers, Inference Theory **Algebra**: Motivation of Algebraic Structures, Finite Groups, Subgroups and Group Homomorphism; Lagrange's Theorem; Commutative Rings and Elementary Properties; **Graph Theory:** Trees: Definition, Binary tree, Binary tree traversal, Binary search tree. Introduction to Graphs, , Operations on Graphs, Representation of graphs, Types: Planner, Directed, Complete, Bipartite Graph, Isomorphism, Euler Graph, Hamiltonian Graph, Connectivity.	20

Text Book:

- Kenneth H Rosen , "Discrete Mathematics and Its Applications", 7th edition, TMH,2012.

Reference Books:

- J.P. Tremblay , "Discrete Mathematical Structures with Applications to Computer Science", TMH, New Delhi,1997.
- V. Krishnamurthy , "Combinatorics: Theory and Applications", East-West Press, New Delhi,1986.
- Ralph P. Grimaldi, "Discrete and Combinatorial Mathematics- An Applied Introduction", 5th Edition, Pearson Education,2004.
- C.L. Liu, "Elements of Discrete Mathematics", 2nd Edition, TMH,2000.

Outcome: After the completion of the course, the student will be able to:

- CO1: Understand the notion of mathematical thinking and proofs to solve the problem.
- CO2: Apply the basics of discrete probability and number theory to solve the real world problem.
  CO3: Analyze basic discrete structures and algorithms using effectively algebraic techniques.
- CO4: Analyze mathematical concepts like sets, reasoning, relational algebra and graph theory to solve optimization problems.
- CO5. Analyze the validity of an argument using logical notation.
- CO6. Demonstrate the basic structures of proof techniques to write and evaluate the validity of arguments.
- CO7. Understand the basic principles of sets, set equalities and operations in sets.
- CO8. Apply counting principles to determine probabilities.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2/PSO1,PSO3
CO2	PO1,PO3/PSO4
CO3	PO2,PO3/PSO3
CO4	PO2,PO3/PSO3
CO5	PO1,PO2/ PSO3
CO6	PO1,PO3/PSO2,PSO3
CO7	PO1,PO2/PSO1
CO8	PO1,PO3/PSO1,PSO4

BCSC0011: THEORY OF AUTOMATA & FORMALLANGUAGES

Objective: *The objective of this course is that students will study and compare different models and views of the abstract notion of computation and its various aspects.*

Credits:04 **Semester V** **L-T-P-J:3-1-0-0**

Module No.	Content	Teaching Hours
I	**Introduction:** Alphabets, Strings and Languages; Automata and Grammars, Deterministic Finite Automata (DFA), Nondeterministic Finite Automata (NFA), Equivalence of NFA and DFA, Minimization of Finite Automata, Myhill-Nerode Theorem; FA with Output - Moore and Mealy machine, Applications and Limitations of FA. **Regular expression (RE):** Regular Expression to FA, DFA to Regular Expression, Arden Theorem, Non Regular Languages, Pumping Lemma for Regular Languages, Applications of Pumping Lemma, Closure Properties of Regular Languages. **Push Down Automata (PDA):** Introduction, Language of PDA, Acceptance by Final State, Acceptance by Empty Stack, Deterministic PDA.	20
II	**Context Free Grammar (CFG) and Context Free Languages (CFL):** Introduction, Derivation Trees, Ambiguity in Grammar, Ambiguous to Unambiguous CFG, Simplification of CFGs, Normal Forms for CFGs - CNF and GNF; Pumping lemma for CFLs, Equivalence of PDA and CFG. **Turing machines (TM):** Basic Model, Definition and Representation, Variants of Turing Machine and their equivalence, TM for Computing Integer Functions, Universal TM, Church's Thesis, Recursive and Recursively Enumerable Languages, Halting Problem, Introduction to Computational Complexity.	20

Text Books:

- K.L.P. Mishra and N. Chandrasekaran, "Theory of Computer Science: Automata, Languages and Computation", 3rd Edition, PHI,2006

Reference Books:

- Hopcroft, Ullman , "Introduction to Automata Theory, Languages and Computation", 3rd Edition, Pearson Education,2013.
- Martin J. C ," Introduction to Languages and Theory of Computations", 4th Edition, TMH,2011.

Outcome: After completion of course, the student will be able to:

- CO1: Understand the basic concepts of Context Free languages, Expression and Grammars.
- CO2: Analyze the conversion of NFA to DFA, Mealy to Moore and Moore to Mealy.
- CO3: Analyze the process to convert regular expression to DFA, DFA to regular expression, and minimization of DFA.
- CO4: Develop the PDA for the context free language and context free grammar.
- CO5: Analyze that the grammar is ambiguous or unambiguous.
- CO6: Apply the process to convert CFG to CNF and GNF.
- CO7: Understand the concept of Turing machine and its variants.
- CO8: Design the Turing machine for the real world application.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO1,PSO4
CO2	PO2,PO3/PSO3
CO3	PO2,PO3,PO9,PO12/PSO1,PSO3,PSO4
CO4	PO1,PO3,PO5,PO9/PSO3,PSO4
CO5	PO1,PO2,PO4/PSO3
CO6	PO2,PO3/PSO3
CO7	PO1,PO2/PSO1,PSO3
CO8	PO3,PO12/PSO1,PSO2,PSO3

BCSC0012: DESIGN & ANALYSIS OF ALGORITHMS

Objective: *The objective of this course is that students will construct and application of various data structures and concepts including Trees, Recursion & Dynamic programing.*

Credits:03 **L-T-P-J:3-0-0-0**

Module No.	Content	Teaching Hours
I	**Introduction:** Algorithms, analyzing algorithms, Complexity of algorithms, Growth of functions, Performance measurements, Sorting and order Statistics - Shell sort, Quick sort, Merge sort, Heap sort, Comparison of sorting algorithms, Sorting in linear time. **Advanced Data Structures:** Red-Black trees, B – trees, Binomial Heaps, Fibonacci Heaps. Divide and Conquer with examples such as Sorting, Matrix Multiplication, Convex hull and Searching.	20
II	Greedy methods with examples such as Optimal Reliability Allocation, Knapsack, Minimum Spanning trees – Prim's and Kruskal's algorithms, Single source shortest paths - Dijkstra's and Bellman Ford algorithms. Backtracking, Branch and Bound with examples such as Travelling Salesman Problem, Graph Coloring, n-Queen Problem, Hamiltonian Cycles and Sum of subsets Dynamic programming with examples such as Knapsack. All pair shortest paths – Warshal's and Floyd's algorithms, Resource allocation problem	20

Text Books:

- Thomas H. Coremen, Charles E. Leiserson and Ronald L. Rivest, Introduction to Algorithms, Third edition, Prentice Hall of India,2008.

Reference Books:

- Gilles Brassard Paul Bratley," Fundamentals of Algorithms", Prentice Hall,1996.
- Ellis Horowitz, SartajSahni, SanguthevarRajasekaran, "Fundamentals of Computer Algorithms", Orient Longman Pvt. Ltd,2008.
- Levitin , "An Introduction to Design and Analysis of Algorithms", Pearson,2008.

Outcome: After completion of course, student will be able to:

- CO1: Understanding of complexity representation in terms of Big Oh, Theta and Omega notations.
- CO2: Derive and solve recurrences describing the performance of divide-and-conquer algorithms (quick sort and merge sort).
- CO3: Compare and analyze different data structures (RB Tree, B Tree, Binomial Heaps, Fibonacci Heaps).
- CO4: Understand the major graph algorithms (DFS, BFS, Dijkstra's Bellman Ford) and their analyses.
- CO5: Understand the greedy paradigm and able to analyze when an algorithmic design situation calls for it. Synthesize greedy algorithms (Optimal Reliability Allocation, Minimum Spanning Trees, factorial Knapsack) and analyze them.
- CO6: Synthesize dynamic-programming algorithms (0/1 knapsack problem, Resource allocation problem, Warshal's and Floyd's algorithms) and analyze them.
- CO7: Understand the backtracking paradigm and able to analysis when an algorithmic design situation calls for it. Synthesize backtracking algorithms (N Queen Problem, TSP Problem, sum of subsets problem, Graph Coloring) and analyze them.
- CO8: Understand the branch and bound paradigm and able to analysis when an algorithmic design situation calls for it. Synthesize branch and bound algorithms (N Queen Problem, TSP Problem, Hamiltonian Cycles, Graph Coloring) and analyze them.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO3,PO4,PO12/PSO1,PSO3
CO2	PO1,PO3,PO4,PO5/PSO1,PSO3
CO3	PO1,PO3,PO6/PSO1,PSO3
CO4	PO1,PO2,PO3,/PSO1,PSO3
CO5	PO1,PO2/PSO1,PSO3
CO6	PO1,PO2,PO3, PO6/PSO1,PSO3
CO7	PO1,,PO4,PO12/PSO1,PSO3
CO8	PO1,PO2,PO3,PO4,PO12/PSO1,PSO2

BCSE0101: DIGITAL IMAGE PROCESSING

Objective: *The objective is to introduce students the Fundamentals of digital Image processing. Students should study the basic of image operations and understand image analysis algorithm. Students can have exposure to current applications in the field of digital image processing*

Credits: 03 **L–T–P-J: 3–0–0-0**

Module No.	Content	Teaching Hours
I	**Introduction and Fundamentals:** Motivation and Perspective, Applications, Components of Image Processing System, Element of Visual Perception, A Simple Image Model, Sampling and Quantization, Some Basic Relationships between Pixels. **Intensity Transformations and Spatial Filtering:** Introduction, Some Basic Intensity Transformation Functions, Histogram Processing, Histogram Equalization, Histogram Specification, Local Enhancement, Enhancement using Arithmetic/Logic Operations – Image Subtraction, Image Averaging, Basics of Spatial Filtering, Smoothing - Mean Filter, Order Statistics Filters, Sharpening – The Laplacian. **Filtering in the Frequency Domain:** Fourier Transform and the Frequency Domain, Basis of Filtering in Frequency Domain	20
II	**Morphological Image Processing:** Introduction, Logical Operations involving Binary Images, Dilation and Erosion, Opening and Closing, The Hit-or-Miss Transformation, Morphological Algorithms – Boundary Extraction, Region Filling, Extraction of Connected Components, Convex Hull, Thinning, Thickening. **Image Segmentation:** Point, Line & Edge detection, Thresholding, Region-based Segmentation, Region Extraction - Pixel Based Approach & Region Based Approach, Edge and Line Detection - Basic Edge Detection, Canny Edge Detection, Edge Linking - Hough Transform. **Representation & Description:** Representation - Boundary Following, Chain Codes; Boundary Descriptors – Shape Numbers.	20

Text Books:

- R.C.Gonzalez and R.E.Woods, "Digital Image Processing", Prentice Hall, 3rd Edition,2011.

Reference Books:

- BhabatoshChanda and D. DuttaMajumder, "Digital Image Processing and Analysis", PHI,2011.
- S. Sridhar , "Digital Image Processing", Oxford University Press,2011

Outcome: After completion of course, student will be able to:

- CO1: Understand mathematical formulation of an image, its processing steps and relationship between image pixels.
- CO2: Apply Image enhancement using intensity transformations and spatial filtering.
- CO3: Analyze image enhancement for frequency domain using Fourier transform.
- CO4: Formulate region of interest through morphological operations.
- CO5:Evaluate strongly co-related regions obtained through Segmentation using discontinuity and homogeneity based segmentation techniques
- CO6: Describe an object of an image using Shape Number and Boundary descriptors.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2,PO3/PSO1,PSO3
CO2	PO3,PO5,PO11,PO12/PSO3,PSO4
CO3	PO1,PO2,PO3,PO7/PSO2,PSO3
CO4	PO1,PO2,PO4/PSO1,PSO2,PSO4
CO5	PO4,PO5,PO3/PSO1,PSO2,PSO3
CO6	PO9,PO10,PO11,PO12/PSO1,PSO2,PSO4

BCSC0014: APPLIED DATABASE MANAGEMENT SYSTEM

Objective: *The objective of the course is to enable students to understand and use a relational database& NoSQL system. Students learn how to design and create a good database.*

Credits:04 **L-T-P-J:4-0-0-0**

Module No.	Content	Teaching Hours
I	**Introduction:** An Overview of Database Management System, Database System vs File System, Database System Concept and Architecture, Data Model Schema and Instances, Data Independence, Database Language and Interfaces (DDL, DML, DCL), Database Development Life Cycle (DDLC) with case studies. **Data Modeling Using the Entity-Relationship Model:** ER Model Concepts, Notation for ER Diagram, Mapping Constraints, Keys, Specialization, Generalization, Aggregation, Reduction of an ER Diagram to Tables, Extended ER Model. **Relational Data Model and Language:** Relational Data Model Concepts, Integrity Constraints, Entity Integrity, Referential Integrity, Keys Constraints, Domain Constraints, Relational Algebra **Database Design & Normalization:** Functional Dependencies, Primary Key, Foreign Key, Candidate Key, Super Key, Normal Forms, First, Second, Third Normal Forms, BCNF, 4th Normal Form, 5th Normal Form, Lossless Join Decompositions, Non Redundant Cover, Canonical Cover, MVD and JDs, Inclusion Dependence.	26
II	**Transaction Processing Concept:** Transaction System, Testing of Serializability, Serializability of Schedules, Conflict & View Serializable Schedule, Recoverability, Recovery from Transaction Failures, Log Based Recovery, Deadlock Handling. **Concurrency Control Techniques:** Concurrency Control, Locking Techniques for Concurrency Control, 2PL, Time Stamping Protocols for Concurrency Control, Validation Based Protocol. **Distributed Database:** Introduction of Distributed Database, Data Fragmentation and Replication. **NoSQL System:** RDBMS vs NoSQL, BASE properties, Key-value, Columnar, Document and Graph-Based database, Introduction of MongoDB, Cassandra, Neo4j and Risk. **Database Programming using Python**: Database connectivity, Retrieving Data from Database, Parameters Passing, Execute many Methods, Cursor Attributes, Invoke Stored Procedures, Invoke Stored Functions.	26

Text Books:

- Elmasri and Navathe, "Fundamentals of Database Systems", 6th Edition, Addison Wesley, 2010.
- Sadalage, P. & Fowler , "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Pearson Education, 2012.

References Books:

- Date C J," An Introduction to Database Systems", 8th Edition, Addison Wesley.
- Korth, Silbertz and Sudarshan, "Database Concepts", 5th Edition, TMH, 1998.
- Redmond, E. &Wilson, "Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement", 1st Edition.

Course Outcome: After completion of course, student will be able to:

- CO1: Understand the concept of database management systems and Relational database.
- CO2: Identify the various data model used in database design.
- CO3: Design conceptual models of a database using ER modeling for real life applications and construct queries in Relational Algebra.

- CO4: Create and populate a RDBMS for a real life application, with constraints and keys, using SQL.
- CO5: Select the information from a database by formulating complex queries in SQL.
- CO6: Analyze the existing design of a database schema and apply concepts of normalization to design an optimal database.
- CO7: Discuss recovery system and be familiar with introduction to web database, distributed databases.
- CO8: Explain the differences between RDBMS and No-SQL,BASE properties and No-SQL databases.
- CO9:Design and implement the database system with the fundamental concepts of DBMS using Python.

Mapping of Course Outcomes (COs) withProgram Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO1
CO2	PO2,PO3/PSO2
CO3	PO2,PO3,PO6,PO11/PSO1,PSO1,PSO2,PSO4
CO4	PO1,PO3/PSO1
CO5	PO1,PO5/PSO1
CO6	PO2,PO3/PSO2
CO7	PO1,PO3/PSO2
CO8	PO1,PO2,PO3/PSO1,PSO4
CO9	PO1,PO2,PO3,PO5/PSO1,PSO2,PSO4

BCSC0015: APPLIED DATA STRUCTURES AND ALGORITHMS

Objective: *The objective of this course is that students will construct and application of various data structures and abstract data types including lists, stacks, queues, trees and graphs.*

CREDITS: 05 **L-T-P-J: 4-1-0-0**

Module No.	Content	Teaching Hours
I	**Introduction**: Basic Terminologies, Elementary Data Organization with Arrays. **Algorithm**: Definition, Characteristics of an Algorithm, Time and Space Complexity. **Introduction to Asymptotic Notations**: Big-Oh, Big-Omega, Big-Theta **Operations on Data Structures**: Insertion, Deletion, Searching, Sorting, Traversing and Merging. **Abstract Data Types (ADT)**. **Linked Lists**: Implementation of Singly Linked Lists, Doubly Linked List, Circular Linked List, Operations on a Linked List - Insertion, Deletion, Traversal; Polynomial Representation and Addition. **Stacks**: Primitive Stack Operations - Push & Pop, Array and Linked List Implementation of Stack, Applications of Stack: Prefix and Postfix Expressions, Evaluation of Postfix Expression, conversion of Infix to Postfix expression. **Recursion**: Principles of Recursion, Head & Tail Recursion, Removal of Recursion, Use of stack in Recursion, Tower of Hanoi Problem, Nth Term of a Fibonacci Series. **Queues**: Operations on Queue – Enqueue & Dequeue operations, Implementation of Queue using Array and Linked List, Circular Queues, DEQueue. **Trees**: Basic Terminology, Array Representation and Linked Representation; Complete Binary Tree, Algebraic Expressions, Extended Binary Trees, Tree Traversal Algorithms - Inorder, Preorder and Postorder; **Search Trees**: Binary search trees, search efficiency, insertion and deletion operations, importance of balancing, AVL trees, searching, insertion and deletions in AVL trees, Tries, Red-Black Trees.	20
II	**Heaps**: Heaps as priority queues, heap implementation, insertion and deletion operations, binary heaps, binomial and Fibonacci heaps, heapsort, heaps in Huffman coding. **Hashing**: Search efficiency in lists and skip lists, hashing as a search structure, hash table, collision resolution. **Searching**: Sequential Search, Binary Search. **Sorting**: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Radix Sort, Bucket Sort. **Graphs**: Terminology, Adjacency Matrices, Adjacency List, Graph Traversal - Depth First Search and Breadth First Search; Spanning Trees, Minimum Cost Spanning Trees – Prim's and Kruskal's Algorithm; Shortest Path Algorithm – Bellman-Ford and Dijkstra's Algorithm. **Basic Algorithmic Techniques**: Greedy Algorithms, Divide & Conquer, Dynamic Programming. Backtracking. **Collections Framework:** Implementation of Collections Framework as Data Structures (Queue, Stack, List, Map, Set, Deque, PriorityQueue, Vector, Hashtable)	20

Text Books:

- Robert Lafore, "Data Structures And Algorithms in Java", 2nd Edition, Pearson SAMS,2003.

References Books:

- Elliot B. Koffman, Paul A. T. Wolfgang , "Data Structures: Abstraction and Design Using Java", 3rd Edition, Wiley,2016.

- Michael T. Goodrich, RobertoTamassia, Michael H. Goldwasser, "Data Structures and Algorithms in Java", 6th Edition, Wiley,2014.

- Horowitz and Sahani, "Fundamentals of Data Structures", 3rd Edition, W H Freeman & Co,2004-05.

Outcome: After completion of course, the student will be able to:

- CO1: Understand the basic concepts of the data structure and algorithms.
- CO2: Understand the complexity representation in terms of Big Oh, Theta and Omega notations.
- CO3: Apply the associated operations in linear data structure like stack, Queue and link list.
- CO4: Apply the associated operations in Binary Search Tree, AVL Tree and M- Way Search Tree.
- CO5: Understand the basic algorithms such as heap sort, graph traversal, quick sort, AVL trees, and hashing.
- CO6: Select the appropriate data structure to solve the problem.
- CO7: Apply the shortest path algorithm to solve real life problem.
- CO8: Understand the concepts of greedy approach, divide & conquer and dynamic programming.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2,PO4/PSO3
CO2	PO1,PO2,PO4/PSO3
CO3	PO1,PO2,PO3/PSO1,PSO4
CO4	PO1,PO2,PO3/PSO1,PSO2,PSO4
CO5	PO1,PO2,PO3,PO4,PO5/PSO1,PSO4
CO6	PO2,PO3/PSO1,PSO2
CO7	PO3,PO6,PO12/PSO1,PSO2
CO8	PO2,PO4/PSO3,PSO4

BCSG0800: PYTHON PROGRAMMING LAB

Objective: *This course introduces the solving of problems using Python programming using OO concepts and its connectivity with database.*

Credits:01 **L-T-P-J:0-0-2-0**

Module No.	Content	Lab Hours
I & II	Programs based on the concepts of: • Building Python Modules • Obtaining user Data • Printing desired output Programs based on the concepts of: • Conditional if statements • Nested if statements • Using else if and elif Programs based on the concepts of Iteration using different kinds of loops Usage of Data Structures • Strings • Lists • Tuples • Sets • Dictionary Program based on the concepts of User-defined modules and Standard Library (random, numpy, scipy, sys, Math Module, String Module, List Module). Program based on Input Output. Program based on exception Handling. Program based on Simple Data analysis. Program based on Pandas.	26

Text Books:

- Paul Barry: "Head First Python "O'Reilly Media, Inc.", 2010.

Reference Books:

- Bret Slatkin: "Effective Python: 59 Specific ways to write better Python", Addison Wesley, 2015.

Outcome: By the end of the course, students will learn to:

- CO1: Apply OO concepts using Python programming.
- CO2: Apply in-built packages defined in Python.
- CO3: Apply front-end as Python Programming to connect with any back-end.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO2/PSO1
CO2	PO3/PSO4
CO3	PO5/PSO2

BCSC0801: COMPUTER PROGRAMMING LAB

Objective: *The objective is to provide a comprehensive study of the C programming language. It stress the strengths of C, which provide students with the means of writing efficient, maintainable, and portable code.*

Credits:01 **L-T-P-J:0-0-2-0**

Module No.	Content	Lab Hours
I & II	• Mapping of flow chart, Algorithm, Language • Simple C-program execution • Programs based on various operators • Programs based on Decision and case Control Structure • Programs based on Loop Control Structure • Program based on special control statement ➢ break ➢ continue • Programs based on Array Insertion, Deletion, Linear Search & Bubble Sort • Programs based on String ➢Length, Copy, Reverse, Concatenate, Compare with & without built-in functions • Programs based on Functions. • Programs based on Storage Class. • Programs based on Recursion. • Programs based on Preprocessor. • Programs based on Pointers • Programs based on array • Programs based on string • Programs based on call by value and call by reference • Programs based on Dynamic Memory Allocation • Programs based on User Defined Data types ➢ Structure and Union ➢ Enum and Typedef • Programs based on File handling ➢ Opening a file ➢ Reading, writing and appending a file ➢ Closing file ➢ Random Access to Files of Records • Programs based on Command Line Argument.	52

Reference Books:

- Herbert Schildt, "C: The Complete Reference", 5th Edition, McGraw Hill Education
- K. N. King, "C Programming a Modern Approach", W. W. Norton, 2nd Edition, 2008.
- Kernighan and Ritche, "The C Programming Language", PHI, 2ndEdition, 2011.
- P. Dey and M. Ghosh, "Programming in C", Oxford University Press 2nd Edition, 2013.

Outcome: On Completion of this course, students are able to:

- CO1: Design programs involving decision structures, loops and functions.
- CO2: Understand the concepts of functions, recursion, pointers and file handling.
- CO3: Design programs involving structures, union and functions.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO3/PSO1,PSO2
CO2	PO3,PO4/PSO1
CO3	PO3/PSO2,PSO4

BCSC0801: OBJECT ORIENTED PROGRAMMING LAB

Objective: *The objective of this course is that students will study and learn Object Oriented Modeling and programming.*

Credits:01 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours
I& II	Programs in Java and python based on the concepts of: • Classes, Constructors, Polymorphism and Keyword Static. Programs based on the concepts of: • Inheritance, Multithreading Using Thread Class & Interface Runnable, String Handling, Generic Classes. Programs based on the concepts of: • Handling Database Connectivity. • Implementation of Collection Framework. Programs based on the concepts of: • Database Connectivity. • Retrieving Data from Database. • Parameters Passing, Execute many Method. • Cursor Attributes. • Invoke Stored Procedures. • Invoke Stored Functions.	24

Reference Books:

- Naughton, Schildt, "The Complete Reference JAVA2", 9th Edition, Oracle Press.
- Bhave&Patekar, "Programming with Java", Pearson Education
- Bret Slatkin: "Effective Python: 59 Specific ways to write better Python", Addison Wesley, 2015.

Outcome: After completion of course, the student will be able to:

- CO1: Implement object oriented language features.
- CO2: Design GUIs and Graphical programming.
- CO3: Design object oriented solutions for small systems involving database and event handling concepts.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2/PSO1
CO2	PO3,PO5/PSO2
CO3	PO3,PO5/PSO4

BCSC0802: DATABASE MANAGEMENT SYSTEM LAB

Objective: *The lab aims to develop an understanding of different applications and constructs of SQL, PL/SQL.*

Credits:01 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours
I & II	• Write the SQL queries for data definition and data manipulation language. • To implement various operations on a table. • To implement various functions in SQL. • To implement restrictions on the table. • To implement the concept of the grouping of Data. • To implement the concept of Joins in SQL. • To implement the concept of sub-queries. • To implement the concept of views, sequence. • To implement the concept of PL/SQL using a cursor. • To implement the concept of Procedure function and Triggers.	24

References Books:

- Date C J," An Introduction to Database Systems", 8th Edition, Addison Wesley.
- Korth, Silbertz and Sudarshan, "Database Concepts", 5th Edition, TMH, 1998.
- Majumdar& Bhattacharya, "Database Management System", TMH

Outcome: After the completion of the course, the student will be able to:

- CO1: Apply SQL queries for DML and DDL.
- CO2: Develop the SQL queries for real life scenarios.
- CO3: Implement the procedural language (PL/SQL) and Triggers.

.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2/PSO1,PSO4
CO2	PO1,PO2/PSO1,PSO4
CO3	PO2,PO3,PO5/PSO2,PSO3

BCSC0803: OPERATING SYSTEMS LAB

Objective: *The lab aims to develop understanding the operation of UNIX operating system.*

Credits:01 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours
I & II	• Implement the following basic commands (with options) used in UNIX/LINUX OS. • Write and implement the basic vi editor commands. • Shell scripts that use simple commands. • Decision based Shell scripts. • Shell scripts related to strings. • Shell scripts using pipes. • Shell scripts with loop statements. • Demonstration and solution for race condition. • Demonstration and use of System Calls. • Implement the basics of IPC in UNIX.	24

Reference Books:

- SibsankarHalder and Alex a Aravind," Operating Systems", 6th Edition, Pearson Education, 2009.
- Harvey M Dietel, "An Introduction to Operating System", 2nd Edition, Pearson Education, 2002.
- D M Dhamdhere, "Operating Systems: A Concept Based Approach", 2nd Edition, 2006.
- M. J. Bach. , "Design of the Unix Operating System", PHI, 1986.

Outcome: After completion of course, the student will be able to:

- CO1: Implement the basic operations on UNIX operating systems.
- CO2: Demonstrate the working of systems calls.
- CO3: Demonstrate message passing in Unix operating system.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO3,PO4/PSO1
CO2	PO1,PO2/PSO1
CO3	PO1,PO4,PO5/PSO1,PSO2

BCSC0804: COMPUTER ORGANIZATION LAB

Objective: *The aim of the lab is to better understand the design of sequential Circuits such as Flip-Flops, Registers, and Counters.*

Credits: 01 **L–T–P-J: 0–0–2-0**

Module No.	Content	Lab Hours
I & II	• Bread Board Implementation of Flip-Flops. • Experiments with clocked Flip-Flops. • Design of Counters. • Bread Board implementation of Counters &Shift Registers. • Implementation of Arithmetic Algorithms. • Bread Board implementation of Adder/Subtraction (Half, Full). • Bread Board implementation of Binary Adder. • Bread Board implementation of Seven Segment Display. • Small Project based on combinational and sequential circuit.	24

Reference Books:

- D.W. Patterson , "Computer Organization and Design", 4th **Edition**, Elsevier Publication, 2008.
- William Stalling , "Computer Organization",8th Edition, PHI, 2011.
- M. Mano , "Computer System Architecture", 3rd Edition, PHI.

Outcome: After the completion of the course, the student will be able to:

- CO1: Implement the Combinational and Sequential Circuit.
- CO2: Demonstrate the working of counter and shift register.
- CO3: Demonstrate the working of ALU and seven segment displays.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO2,PO3,PO5/PSO2
CO2	PO3,PO4/PSO2
CO3	PO3,PO5/PSO1,PSO2

BCSC0805: DATA STRUCTURES & ALGORITHMS LAB

Objective: *The objective of this course is that students will understand and implement simple data structures, able demonstrate different sorting and searching techniques. and will be familiar with graphs and their applications.*

Credits:01 **L-T-P-J:0-0-2-0**

Module	Content	Lab
I &II	• Program to implement various operations in a singly linked list. • Program to implement insertion, deletion and traversal in a doubly linked List. • Program to implement polynomial addition using linked list. • Program to demonstrate the various operations on stack. • Program to convert an infix expression into postfix expression. • Program to evaluate a given postfix expression. • Program to implement Tower of Hanoi problem using Recursion. • Program to demonstrate the implementation of various operations on linear and circular queue. • Program to demonstrate the implementation of insertion and traversals on a binary search tree. • Program to implement Dijkstra's Algorithm to find the shortest path between source and destination. • Program to search a given element as entered by the user using sequential and binary search to search a given element as entered by the user. • Implementation of various sorting algorithms like Selection Sort, Bubble Sort, Insertion Sort, Merge Sort, Quick Sort and Heap Sort.	24

Note: All Code must be done in Java as well as Python

Outcome: After completion of course, student will be able to:

- CO1: Demonstrate the associated operations in linear data structure like stack, Queue and link list.
- CO2: Demonstrate the associated operations in Binary Search Tree and Dijkstra's Algorithm.
- CO3: Implementation the sorting algorithms like Selection Sort, Bubble Sort, Insertion Sort, Merge Sort, Quick Sort and Heap Sort.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO1
CO2	PO4/PSO1,PSO3
CO3	PO2/PSO3,PSO4

BCSC0806: MICROPROCESSORS LAB

Objective: *The objective is to introduce the Architecture and programming of the microprocessor and learning about interfacing and various applications of microprocessor.*

Credits: 01 **L–T–P-J: 0–0–2-0**

Module No.	Content	Lab Hours
I & II	• To study 8085 microprocessor System. • To study 8086 microprocessor System. • To develop and run basic programs in 8085 ALP. • To develop and run programs in 8085 ALP related to the concept of looping, counting and indexing. • To perform interfacing of RAM chip to 8085/8086. • To perform interfacing of keyboard controller. • To perform interfacing of DMA controller. • To perform interfacing of UART/USART.	24

Reference Books:

- Ramesh S. Gaonkar, "Microprocessor Architecture Programming and Applications with 8085", 4th Edition, Penram International Publishing, 2000.
- D. V. Hall , "Microprocessors and Interfacing: Programming and Hardware", 2nd Edition, TMH, 1992.

Outcome: After completion of course, student will be able to:

- CO1: Demonstrate the arithmetic and logical operations using assembly language programming (8085).
- CO2: Demonstrate the memory operations using assembly language programming (8085).
- CO3: Demonstrate the interfacing of Keyboard, DMA and UART controller.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO3/PSO1,PSO2
CO2	PO1,PO2/PSO1,PSO2
CO3	PO1,PO3,PO5/ PSO2

BCSC0807: DESIGN & ANALYSIS OF ALGORITHMS LAB

Objective: *The objective of this course is that students will understand and implement simple data structures, able demonstrate different sorting and searching techniques. and will be familiar with graphs and their applications.*

Credits:01 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours

I& II	• Implementation of sorting algorithms:
	  ○ Insertion Sort
	  ○ Bubble Sort
	  ○ Selection Sort
	  ○ Divide and conquer approach:
	• Quick Sort
	• Merge Sort
	  ○ Heap Sort
	  ○ Counting Sort
	• Implementation of Searching Techniques:
	  ○ Linear Search
	  ○ Binary Search
	• Implementation of Matrix Multiplication
	• Implementation of Convex Hull
	• Implementation of Breadth First Search
	• Implementation of Depth First Search
	• Implementation of Greedy approaches:
	  ○ Optimal Reliability Allocation.
	  ○ Knapsack. Minimum
	  ○ Minimum Spanning trees:
	• Prim's and Kruskal's algorithms.
	  ○ Single source shortest paths –
	• Dijkstra's and Bellman Ford algorithms.
	• Implementation of Dynamic Programming:
	  ○ Longest Increasing Subsequence.
	  ○ Finding best path in maze.
	  ○ Matrix Chain Multiplication
	  ○ 0/1 Knapsack Problem
	  ○ Resource Allocation Problem
		32

Note: All Code must be done in Java as well as Python

Outcome: After completion of course, student will be able to:

- CO1: Implementation the sorting algorithms like Selection Sort, Bubble Sort, Insertion Sort, Merge Sort, Quick Sort and Heap Sort.
- CO2: Demonstrate and use the appropriate data structures for a given problem
- CO3: Implement the algorithms based on Greedy approach and Dynamic Programming.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2,PO4/PSO1,PSO2,PSO4
CO2	PO1,PO3,PO4/PSO1,PSO2,PSO3
CO3	PO2,PO3,PO5/PSO1,PSO2,PSO4

BCSE0131: DIGITAL IMAGE PROCESSING LAB

Objective: *The objective of this Lab is to develop hands-on experience to write programs using MATLAB/*Python *language for digital manipulation of images in both spatial and frequency domains.*

Credits: 01 **L–T–P-J: 0–0–2-0**

Module No.	Content	Lab Hours
I/II	• Basic commands to familiarize with MATLAB & perform the various Matrix operations. • Understanding image basic "image resize, image type conversion, extraction of color band, creating a synthesic image, pseudocolor image" • Perform various arithmetic operation (image addition, subtraction& complement) & logical operation (NOT, OR and XOR) on images • Perform various Image Enhancement operations: Image Negation function, Logarithmic Transformation, Power Law Transformation, Histogram Equalization, contrast stretching, plot histogram without using imhist function • Perform smoothing using linear (average filter) and order statistics filters (min, max & median) of varying sizes • Sharpen an image using Laplacian filter. • Perform various Fast Fourier transform (FFT) and frequency domain filtering on images using MATLAB. • Perform various Image Enhancement operation in frequency domain • Perform various Morphological operation dilation, erosion, internal & external boundary Extraction, Thinning,thickening of image &Perform Dilation, erosion, boundary Extraction without using direct function • Perform various thresholding segmentation (Simple, Multiple, and Adaptive thresholding) • Perform the various Edge Detection Operators (Ordinary, Roberts, Prewit, Sobeland Canny Operator) • Minor Application Assignment.	12*2=24

Reference Books:

- R.C. Gonzalez and R.E. Woods, "Digital Image Processing Using MATLAB", PHI, 2nd Edition,2010.
- Hands-On Image Processing with Python by SandipanDey, November 2018, Packt
- https://www.pyimagesearch.com/

Outcome: After studying the subject, the students will be able to:

- CO1: Implement digital image processing operations for image manipulation and Enhancement.
- CO2: Test the source code for Morphological and Segmentation algorithms.
- CO3: Apply image processing algorithms in real-life problems as Minor Application Assignment.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2,PO6,PO9/PSO1,PSO2
CO2	PO5,PO7,PO8,PO10/PSO1,PSO4
CO3	PO1,PO3,PO11,PO12/PSO3,PSO4

BCSC0808: APPLIED DATABASE MANAGEMENT SYSTEM LAB

Objective: *The lab aims to develop an understanding of different applications and constructs of SQL, PL/SQL and NoSQL databases.*

Credits:01 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours
I & II	• Write the SQL queries for data definition and data manipulation language. • To implement various operations on a table. • To implement various functions in SQL. • To implement restrictions on the table. • To implement the concept of the grouping of Data. • To implement the concept of Joins in SQL. • To implement the concept of sub-queries. • To implement the concept of views, sequence. • To implement the concept of PL/SQL using a cursor. • To implement the concept of Procedure function and Triggers. • Introduction to MongoDB and its Installation on Windows or Linux, Description of mongo Shell, create database and show database, Commands for MongoDB and To study operations in MongoDB – Insert, Query, Update, Delete and Projection • To implement Database connectivity using Python	24

References Books:

- Date C J," An Introduction to Database Systems", 8th Edition, Addison Wesley.
- Korth, Silbertz and Sudarshan, "Database Concepts", 5th Edition, TMH, 1998.
- Majumdar& Bhattacharya, "Database Management System", TMH
- Sadalage, P. & Fowler , "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Pearson Education, 2012.

Outcome: After the completion of the course, the student will be able to:

- CO1: Apply SQL queries for DML and DDL.
- CO2: Implement the procedural language (PL/SQL) and Triggers.
- CO3: Apply NoSQL queries in MongoDB.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2/PSO1,PSO4
CO2	PO2,PO3,PO5/PSO2,PSO3
CO3	PO5/PSO2

BCSC0809: APPLIED DATA STRUCTURES & ALGORITHMS LAB

Objective: *The objective of this course is that students will understand and implement simple data structures, able demonstrate different sorting and searching techniques. and will be familiar with graphs and their applications.*

Credits:01 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours
I&II	• Program to implement various operations in a singly linked list. • Program to implement insertion, deletion and traversal in a doubly linked List. • Program to demonstrate the various operations on stack. • Program to implement Tower of Hanoi problem using Recursion. • Program to demonstrate the implementation of various operations on linear and a • circular queue. • Program to implement Dijkstra's Algorithm to find the shortest path between source and • destination. • Program to search a given element as entered by the user using binary search (divide and • conquer approach) to search a given element as entered by the user. • Implementation of various sorting algorithms like Selection Sort, Bubble Sort, Insertion Sort, Merge Sort, Quick Sort, Heap Sort, Radix Sort, Bucket Sort. • Program to implement AVL Trees. • Program to implement Red-Black Trees. • Program to implement Binary Heaps. • Program to implement Priority Queues. • Program to implement Huffman Character Length Encoding. • Program to implement tree traversal using Backtracking. • Project to create a LogBook to generate a Calendar Display. • Project to create a Hangman game using String Lexical Analysis.	24

Note: All Code must be done in Java as well as Python

Outcome: After completion of course, student will be able to:

- CO1: Demonstrate the associated operations in linear data structure like stack, Queue and link list.
- CO2: Demonstrate the associated operations in Binary Search Tree AVL Tree, Red- Black, and Dijkstra's Algorithm.
- CO3: Implementation the sorting algorithms like Selection Sort, Bubble Sort, Insertion Sort, Merge Sort, Quick Sort and Heap Sort.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO1
CO2	PO4/PSO1,PSO3
CO3	PO2/PSO3,PSO4

Program Elective (Only for Specialization Programme)

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE- REQUISITES
			L	T	P	J			
Bouquet: Cloud Computing & Virtualization									
THEORY									
1.	BCSC0600	Introduction to Open Source Software & Open Standards	2	0	0	0	2	2	
2.	BCSC0601	Web Programming through PHP	3	0	0	0	3	3	
3.	BCSE1502	Introduction to Virtualization and Cloud Computing	3	0	0	0	3	3	
4.	BCSE0503	Cloud Computing Architecture & Deployment Models	3	0	0	0	3	3	
5.	BCSE0508	Cloud and Business Process Management	3	0	0	0	3	3	
6.	BCSE0509	Cloud Security, Backup & Disaster Recovery	3	0	0	0	3	3	
7.	BCSE0510	Container Orchestration and Infrastructure Automation	3	0	0	0	3	3	
8.	BCSE0511	DevOps	3	0	0	0	3	3	
9.	BCSE0701	Introduction To Machine Learning	3	0	0	0	3	3	
10.	BCSE0556	Hadoop & Big Data Analytics	3	0	0	0	3	3	
11.	BCSE0252	Full Stack Using Node JS	3	0	0	0	3	3	
12.	BCSE0053	Agile Software Development	3	0	0	0	3	3	
13.	BCSE0602	IT Network Security	3	0	0	0	3	3	
14.	BCSE0203	Internet of Things	3	0	0	0	3	3	
15.	BCSC0013	Compiler Design	3	1	0	0	4	4	
PRACTICALS									
1.	BCSC0800	Web Programming Lab	0	0	2	0	1	2	
2.	BCSE0531	Virtualization Lab	0	0	2	0	1	2	
3.	BCSE0532	Cloud Deployment Lab	0	0	2	0	1	2	
4.	BCSE0536	Cloud and Business Process Management Lab	0	0	2	0	1	2	
5.	BCSE0537	Cloud Security, Backup & Disaster Recovery Lab	0	0	2	0	1	2	
6.	BCSE0538	Container Orchestration and Infrastructure Automation Lab	0	0	2	0	1	2	
7.	BCSE0539	DevOps Lab	0	0	2	0	1	2	
8.	BCSE0731	Introduction To Machine Learning Lab	0	0	2	0	1	2	
9.	BCSE0585	Hadoop & Big Data Analytics Lab	0	0	2	0	1	2	
10.	BCSE0282	Full Stack Using Node JS Lab	0	0	2	0	1	2	
11.	BCSE0632	IT Network Security Lab	0	0	2	0	1	2	

12.	BCSE0232	Internet of Things Lab	0	0	2	0	1	2	

BCSE0501: INTRODUCTION TO IT INFRASTRUCTURE LANDSCAPE OVERVIEW

Objective: *The course enables students to understand the Database, Application and Middleware along with System Server hardware and Directory Services.*

Credits: 03 **Semester - III** **L–T–P-J: 3–0–0-0**

Module No.	Content	Teaching Hours
I	**Storage Overview:** Storage Networking Technology, Types Of Storage System, FC-AL (Fiber Channel Arbitrated Loop), Fabric, Storage Area Network, Zoning, Storage Virtualization, hybrid storage networking technologies (iSCSI, FCIP, FCoE), **Systems & Directory Services Overview:** Server Technology, Operating System, Virtualization, Hypervisor, I/o Virtualization, Partitioning, Server Deployment, Server Management Console ,Server Availability Concepts And Techniques, Server Workload. Directory Server Concepts, Directory, LDAP PROTOCOL, Overview of LDAP, LDAP Architecture, LDAP Models, LDAP Replication Topologies, LDAP Data Inter change Format (LDIF). **Database Overview:** Data Warehousing, And Data Marts, Data Mining (DM), Data Warehousing and Data Marts	20
II	**Network Security and Overview:** Network Overview, Network Topologies, Tree Topology, Firewalls, Switching Concepts, What Is Routing?, Virtual Lan's, Security Basics, Loss Of Privacy, Loss of Integrity, Security Technology, Active Audit , Secure Messaging, Data Security, Network Security. Guest hopping, attacks on the VM (delete the VM, attack on the control of the VM, code or file injection into the virtualized file structure), VM migration attack, Hyper jacking. **Application and Middleware Overview :** Introduction To Common Messaging System (MQ SERIES), Application Integration – Business Need, Middleware, Message Oriented Middleware ,Synchronous interaction, Asynchronous interaction, Coupling, Reliability ,Scalability, Availability, IBM WebSphere MQ, WebSphereMQ Objects, Web TierDeployment, Application Servers And Clustered Deployment, EMAIL ,Lotus Architecture, Lotus Domino Server Types, Lotus Notes Clients, Types of Certificates. **Server Architectures:** Stand-alone, blades, stateless, clustering, scaling, Limitation of traditional server deployments, Redundant Layer 2 and Layer 3 designs.	20

Text Book:

- Introduction to IT infrastructure Landscape (IBM ICE Publication)
- Avi Silberschatz, Henry F. Korth, S. Sudarshan , Database System Concepts, Sixth Edition, 2010.
- Vmware "VMware Security Hardening Guide" White Paper, June 2011.

Outcome: After completion of course, the student will be able to:

- CO 1: Understand the basics of Storage area Network and types of storage.
- CO2: Understand the concept of server technology ,Virtualization, Hypervisor.
- CO3: Understand LDIF architecture, LDAP Protocol and LDAP models.
- CO4: Explain the basics of Database ,Data warehousing and data mining.
- CO5: Explain the different topologies and security technology.
- CO6: Understand application integration, middleware and lotus architecture.
- CO7: Understand server architecture (blades, clustering, scaling, server deployments).

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO1
CO2	PO1/PSO1
CO3	PO1/PSO1
CO4	PO1/PSO1
CO5	PO1/PSO1
CO6	PO1/PSO1
CO7	PO1/PSO1

BCSE 1502: Introduction to Virtualization and Cloud Computing

Objective: *The course enables students to understand the virtualization technology, Applications along with cloud computing concepts and services.*

Credits: 03 **Semester - IV** **L–T–P-J: 3–0–0-0**

Module No.	Content	Teaching Hours
I	**Introduction to Virtualization** Overview of Virtualization: Need of Virtualization, traditional IT Infrastructure, shortcoming of physical infrastructure, Benefit of Virtualization, Comparison of traditional IT infrastructure with virtualized infrastructure, **Virtualization** Implementing Virtualization, Typical hardware / software server stack and its logical equivalence, Pre/post virtualization server stack, Types of virtualization, Area and technology based classification, History of virtualization, Time sharing system, IBM mainframe and PowerVM virtualization, Extending Virtualization to x86 and its hardware support, Impact of Virtualization: Cost and Manageability impact. **Server and Storage Virtualization** Types of Server Virtualization, Simulation, Hardware Assisted Virtualization, Hypervisors, Ring levels on x86 processors, Types of Hypervisors, IBM PowerVM Hypervisors, Common consideration in server Virtualization, Desktop Virtualization: Benefits Constraints and Types. Three major layers in Xen server, RAID Levels, DAS, NAS, SAN, Storage Virtualization overview: benefit and types, Features of logical layers, Host level storage Virtualization, Host based mirroring, Storage level Virtualization, Network based storage Virtualization. **Network and Application Virtualization** Network Virtualization overview: VPN, VLAN, Challenges in using application in traditional install, use and update model, Solution for challenges, Benefits of Application Virtualization.	20
II	**Introduction to Cloud Computing** Overview: Introduction to cloud computing, Service driven model, Advantage of cloud computing: Marketing point of view, Types of services, OS and Virtualization, VM, advantage of Virtualization, Virtualization and cloud and its overlapping, Business value, Business impact of cloud, Technological value of cloud, End user benefits, Change for provider and administrator, Pros and cons of cloud model, Anatomy of cloud, Solution component, Service catalog, User self-service portal, Service request management, Provisioning, Optimized infrastructure, Chargeback, Benefit of cloud, Delivery and Deployment Model, Different Cloud Architecture: Public, Private and Hybrid and its pros and cons, Delivery models. Cloud transformation roadmap, History of cloud, Clint-server, cluster, Grid models, Cloud vs Grid and their relationship, Cluster and Cloud, Utility computing and Evolution of Cloud Computing, Cloud Computing Milestones. **Cloud Deployment selection criteria** Pros and cons of each Deployment architecture of Cloud: Public, Private, Hybrid, Cloud deployment decision factors(Business IT Control, Business critical application, Data and transaction security, Compliance and audit, Balance of CAPEX and OPEX, Workload characteristics, Workload lifespan preferences, Industry segment- SME and Large enterprises, Data Freedom, Software characteristics, Time to deploy), Public Cloud: Factor Matrix, Advantage, Disadvantage, Public Cloud: Factor Matrix, Advantage and disadvantage, Hybrid Cloud: Factor matrix, Advantage, Disadvantage, Overview of Cloud delivery models, Infrastructure, IT Layers, IaaS Overview,	20

	Features, Cloud bursting, Multi tenancy, Resource pooling, PaaS: Overview, Component, Example, SaaS: Advantage, Example. **Implementation of Virtualization and Factors deciding Cloud workloads** Case study overview, customer IT landscape, Function of data center, Trigger for virtualization, Preparation for virtualization, Server selection, Server sizing, Server criticality, Provisioning, Proximity and Locality, Transition tool for virtualization, Cost savings, Cloud workload overview, Workload characterization, Factors, Suitable workload for cloud, Private cloud solution, Types of workload, Advantage, Mission critical workload, Mixed workload, Production only workload for hybrid cloud, Industry specific workload , Non suitable workload: Public, Private cloud, Possible workload by cloud.	

Text Books:

- R1. Introduction to Virtualization and Cloud Computing(IBM ICE Publication)
- R2. Cloud Computing Black Book (by : Kailash Jayaswal, Jagannath Kallakurchi, Donald J. Houde, Dr. Deven Shah)
- R3. Raj Kumar Buyya, James Broberg, Andrezei M.Goscinski , Cloud Computing: Principles and paradigms, 2011.

Outcome: After completion of course, the student will be able to:

- CO1: Understand the basics of virtualization technology and hypervisors.
- CO2: Analyze the trade-offs between deploying applications in the cloud and over the local infrastructure.
- CO3: Recognize real-world problem using cloud computing through group collaboration.
- CO4: Understand on-demand utility computing phenomenon of cloud computing.
- CO5: Understand the issues involved in cloud computing.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO3,PO5,PO7/PSO2
CO2	PO1,PO3,PO5,PO7/PSO4
CO3	PO1,PO3,PO7/PSO1
CO4	PO1,PO7/PSO1
CO5	PO1,PO3/PSO1
CO6	PO1,PO3,PO5,PO7/PSO4

BCSE0531: Virtualization Lab

Objective: *This course covers aims to explain various technologies related to Cloud Computing and their practical implementations, discuss different architectural models of cloud computing, the concepts of virtualization and cloud orchestration.*

Credits: 01 **Semester IV** **L–T–P: 0–0–2**

Module No.	Content	Teaching Hours
I&II	1. Introduction and Installation of VMware. 2. Installation of Ubuntu, CentOS on VMware. 3. Exercise on virtual Machine using QEMU. 4. Installation of windows, CentOS on QEMU. 5. Exercise on KVM on Ubuntu 6. Installation of windows, CentOS on KVM. 7. Exercise on KVM on CentOS. 8. Introduction and simulation with packet tracer. 9. Exercise on installation of VMware ESXi Server on VMware. 10. Creating and assigning instances using ESXi server on VMware. 11. Creating an EC2 instance on AWS 12. Configuration of db in AWS. 13. Creation of S3 bucket with single IAM user in AWS.	18

Outcome: After completion of the course, student will be able to:

- CO1: Design Virtual Machines over Type-2 Hypervisor & Test Client Server application over VMs created.
- CO2: Apply and analyze various kinds of networking in virtual environment.
- CO3: Apply and create various use cases of the key components of AWS.
- CO4: Understanding, installation of Type-1 hypervisor and assignment of instances on ESXi server.
-

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO3,PO5/PSO2
CO2	PO1,PO2,PO3,PO5/PSO1
CO3	PO1,PO3,PO5,PO7/PSO2
CO4	PO1,PO2,PO3,PO5/PSO2

BCSE 0503: CLOUD COMPUTING ARCHITECHURE & DEPLOYMENT MODELS

Objective: The objective is to study the architecture and deployment models to develop a private cloud using the open standards tools such as open stack.

Credits: 03 **Semester - V** **L–T–P-J: 3–0–0-0**

Module No.	Content	Teaching Hours
I	**Introduction:** Definition of cloud computing, Delivery Models, Conceptual reference model, Cloud Computing solution components. **Cloud computing Architecture:** The conceptual reference model, Service Deployment, Cloud service management, cloud taxonomy, IBM CCRA, Common cloud management platform. **Case Study:** IBM Smart Cloud Entry, VMware vClouddirector. **Cloud vendor selection:** SLA, Security and privacy, periodic update and maintenance, data location and Jurisdiction, Measurability, Pricing, Interoperability and lock in , Exit process, track record. **OpenStack:** Definition, Advantages, Releases, Architectural overview, Different components of Open Stack.	20
II	**OpenStack:** Open stack- Hypervisors, Network Services, Storage -Block Storage, Object Storage, Choosing Storage Backends, Commodity Storage Backend Technologies: swift, Ceph, Gluster, LVM, ZFS. **Advance concepts in Openstack:** Multiserver Openstack, Tenant model architecture, Cloud orchestration using OpenStack using OpenStack Heat and Ubuntu Juju. **Eucalyptus:** Introduction, Features and Functionality, Architecture, Basic and Advanced Components. Eucalyptus vs Openstack **OpenNebula:** Introduction, Features and Functionality, Architecture, Basic and Advanced Components. OpenNebula vs Openstack	20

Text Book:

- Cloud Computing Architecture & Deployment Models (IBM Publications-),2015.
- Raj Kumar Buyya, James Broberg, AndrezeiM.Goscinski , Cloud Computing: Principles and paradigms, 2011.

Reference Book:

- Rittinghouse, John, W, " Cloud computing ": Implementation, management and security.
- Barrie Sosinsky , "Cloud Computing Bible", Wiley, 2011.
- Bumgardner, V. C. , OpenStack in action. Manning PublicationsCompany, 2016.

Outcome: After completion of course, the student will be able to:

- CO1: Identify the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS and different clouds.
- CO2: Explain the components of Conceptual Reference Models like NISTand IBM CCRA.
- CO3: Understand the components and drafting of SLA.
- CO4: Explain the Architecture and Components of Openstack.
- CO5: Understand concepts of storages exist in the cloud environment like swift, Ceph, Gluster, LVM, ZFS.
- CO6: Understand the private cloud tools like OpenNebula and Eucalyptus.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO5/PSO1
CO2	PO1/PSO1
CO3	PO1/PSO1
CO4	PO1,PO2/PSO2
CO5	PO1/PSO1
CO6	PO1,PO2,PO5/PSO1

BCSE0532: Cloud Deployment LAB

Objective: *This course covers aims to explain various technologies related to Cloud Computing deployment models and their practical implementations, discuss different architectural models of cloud computing, the concepts of virtualization and cloud orchestration.*

Credits: 01 **Semester V** **L–T–P : 0–0–2**

Module No.	Content	Teaching Hours
I&II	Case Study of Public Cloud service provide: IBM Cloud Case Study of Public Cloud service provide: AWS Case Study of Public Cloud service provide: Azure Implementation of OpenStack using VMware Implementation of OpenStack on CentOS machine Deploying VM on Open Stack platform Building applications on VM in OpenStack platform	18

Outcome: After completion of Lab, student will be able to:

- CO1: Understand the Architecture and Components of IBM Cloud, AWS and Azure.
- CO2: Implement Openstack using VMware and CentOS machine.
- CO3: Implement of automation tools like Terraform and heat to create resources in AWS and Openstack environment.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO3,PO5,PO7/PSO1,PSO2
CO2	PO1,PO3,PO5/PSO1,PSO2
CO3	PO1,PO3,PO5,PO7/PSO1,PSO2

BCSE 0508: CLOUD AND BUSINESS PROCESS MANAGEMENT

Objective: *This course introduces the cloud monitoring, Service Management, System Administration and Business Process in Cloud Computing.*

Credits: 03 **Semester V** **L–T–P-J: 3–0–0-0**

Module No.	Content	Hours
I	**Service management & Service administration in Cloud:** Cloud Architecture, Aneka Cloud Architecture, General Architecture of Cloud Workflow Systems, Cloud Provisioning, Cloud-based Infrastructure Service Provisioning, Cloud Usage Monitor, Monitoring Agent, Resource Agent, Polling Agent, Key Benefits and features, Cloud monitoring and features, Metering and Billing, Smart Metering-Architecture, **Cloud Computing Services:** Cloud Computing Administration, Cloud Computing Solution for Enterprise, Public and Private Cloud for Enterprise. **Cloud Growth Planning & Managing Security and Resiliency:** Forecasting requirements for cloud managed resources. The IBM Cloud Computing Reference Architecture(CCRA), High Availability and Interoperability, **Operational view for cloud management**: IBM Smart Cloud, Integrated Infrastructure for Service Providers, Storefront, IBM Ecosystem Support, Cloud service provider deployment scenarios. **Service Catalog Management**: Service Catalog-Value to the Business, Basic Troubleshooting Techniques, Configuration Management –Principles.	20
II	**Cloud computing BPM:** Market Benefits Business Process Management Life Cycle Business model, Business process modeling tools. **Introduction to BPM:** Motivation and Definitions. Business Process-Process Designer. Administration and Stakeholders. Classification of Business Processes. Goals, Structure, and Organization. **PC Health Check:** What slows down a computer?, Running a PC Health Check, Patching and Updates, Tools for patch management, Patching demands proper due Diligence, Patch Management **BPM Life Cycle Methodology**: Business Process Management Activities BPM suites Practice BPM technology Managing end-to-end, customer-facing processes Establishing a common language for business IT alignment. **Business Process Management Overview**: Business process management overview. Overview of process modeling, Building processes in IBM BPM Reusing items in Process Designer. Creating a business process definition (BPD) Building services. **Creating User Interfaces**: Creating user interfaces. Coaches. Developing reusable Coach Views Templates. Stock controls. **Dashboards and Reports Business value**: Solution overview. IBM Solution for Collaborative Lifecycle Management. Designing process interactions for business users	20

Reference Books:

- IBM Student Guide on "Managing the Cloud".
- IBM Student Guide on "Business Process Management".
- Rajkumar Buyya "Mastering Cloud Computing".

Outcome: After completion of course, the student will be able to:

- CO1: Understand basics of cloud Storage systems.
- CO2: Explain the technologies and approaches for the business related issues.
- CO3: Understand the operation view and service catalog of cloud management.
- CO4: Understand the concepts of VPM cloud computing.
- CO5: Designprocess interactions interface for business users.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO1
CO2	PO1/PSO1
CO3	PO1/PSO1
CO4	PO1/PSO1
CO5	PO1,PO3/PSO1

CSE0536: CLOUD AND BUSINESS PROCESS MANAGEMENT LAB

Objective: *This course introduces the Management of Cloud in terms of IT service Management and monitors the performance of data centers with the help of cloud computing simulators.*

Credits: 01 **Semester V** **L–T–P-J: 0–0–2-0**

Module No.	Content	Lab Hours
I&II	Performance evaluation of services over Cloud. Management of cloud resources using Cloud Analyst. Simulation of large scale Cloud computing data centers with Cloud Analyst. Creating a Super User on Linux OS and learn access control. Creating a Warehouse Application in SalesForce.com. Stop and start Amazon EC2 instances at regular intervals using Lambda. Installation &Hands on practice on CloudSim Create smart AWS diagrams using CloudCraft for a business process. Building an Amazon CloudWatch dashboard outside of the AWS Management Console. Create an IAM User with Full Access to Amazon S3 and CloudWatch Logs	24

Reference Books:

- IBM Student Guide on "Managing the Cloud".
- IBM Student Guide on "Business Process Management"
- A. Srinivasan, "Cloud Computing: A Practical Approach for Learning and Implementation"

Outcome: After completion of course, the student will be able to:

- CO1: Analyze different cloud programming platforms and tools.
- CO2: Develop scalable applications using AWS features.
- CO3: Demonstrate the working of CloudSim.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2,PO3/PSO1
CO2	PO1,PO2,PO3,Po5/PSO1,PSO2
CO3	PO1,PO3,PO5/PSO1

BCSE 0509: CLOUD SECURITY, BACKUP AND DISASTER MANAGEMENT

Objective: *The course objectives is to introduce about the needs and application of security, backup and disaster management in cloud computing.*

Credits: 03 **Semester V** **L–T–P-J: 3–0–0-0**

Module No.	Content	Hours
I	**Cloud Computing Software Security Fundamentals:** Introduction to Cryptography, Cloud Information Security Objectives, Confidentiality, Integrity, and Availability, **Cloud Security Services:** Authentication, Authorization, Auditing, Accountability. Relevant Cloud Security Design Principles, Secure Cloud Software Requirements, Approaches to Cloud Software Requirements Engineering, Secure Cloud Software Testing. **Cloud Computing Security Challenges:** Security Policy Implementation, Policy Types, Regulatory Policies, Informative Policies, Computer Security Incident Response Team, Virtualization Security Management, Virtual Threats, Hypervisor Risks, Increased Denial of Service Risk, VM Security Recommendations, Best Practice Security Techniques, VM-Specific Security Techniques, Hardening the Virtual Machine, Securing VM Remote Access. **Infrastructure Security**: The Network Level, The Host Level, The Application Level **Data Security And Storage:** Aspects of Data Security, Data Security Mitigation, Provider Data and Its Security **Identity And Access Management**: Trust Boundaries and IAM, Why IAM?, IAM Challenges, IAM Definitions, IAM Architecture and Practice, CloudWatch, Cloud Trail	20
II	**Backup:** Recovery Objectives: RPO, RTO, Types of Backup, Architecture of Backup. **Data Preparation for Backup**: Data Compression, Data Deduplication, Archive, Data Protection in a virtualized environment, Cloud based Data Protection. **Fundamentals of Availability:** Introduction, Reliability, Serviceability, Need of Availability. **High Availability:** Components that affect Availability & need for High Availability, Availability Levels and how it is achieved, Single system, fault tolerant, HA clustering & components, Types of HA Solutions, HA Clustering Advantages. **HA Criteria and Applications:** Network layer HA, Hardware combination & HA Possibilities, Applications & Operating system layer. Hardware layer: Storage, HA for Virtual Environment, Components of Virtual Machine and HA on Virtual Machines. **Fundamental of Disaster Recovery:** Disaster Recovery, Types of Disasters, Business Continuity (BC) & Disaster Recovery (DR), Importance of Disaster Recovery, DR Terminologies, Disaster Recovery Planning, Phases of Planning, DR Technology Tree, Virtualization.	19

Reference Books:

- IBM Book , “Cloud Backup & DR”, 2016.
- Robin M Hostings , “Planning Cloud-Based Disaster Recovery for Digital Assets” : The Innovative Librarian's Guide, 2017.
- Bryan Strawser, “Rethinking Disaster Recovery”: The Impact of Cloud Computing (Bryghtpath LLC White Papers Book 2), 2016.
- IBM Book, “Cloud Security”, 2016.
- William Stallings. “Cryptography and network security”, 7/E. Pearson Education India, 2019.
- Chris Doston, ,” Practical Cloud Security”, F/E, O'Reilly, 2019
- Tim Mather, Subra Kumaraswamy, and Shahed Latif , “Cloud Security and Privacy”, 3/E O'Reilly, 2019.

Outcome: After completion of course, the student will be able to:

- CO1: Understand the concepts of Cloud Security.
- CO2: Identify the known threats, risks, vulnerabilities and privacy issues associated with Cloud based services.
- CO3: Evaluate the security controls necessary to ensure confidentiality, integrity and availability in cloud computing.
- CO4: Design & develop backup strategies for cloud.
- CO5: Understand the concept of High availability.
- CO6: Apply the concept of disaster recovery planning in cloud computing.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO1
CO2	PO1,PO3,PO5/PSO4
CO3	PO1,PO2,PO5/PSO4
CO4	PO1,PO2,PO3/PSO1
CO5	PO1/PSO1
CO6	PO1/PSO1

BCSE0537: CLOUD SECURITY, BACKUP & DISASTER RECOVERY LAB

Objective: *To cover the basic security prespective in cloud computing*

Credits: 01 **Semester VI** **L–T–P: 0–0–2**

Module No.	Content	Teaching Hours
I&II	1. Write a program to perform encryption and decryption 2. using the Ceaser Cipher algorithm. 3. Write a program to perform encryption and decryption 4. using the Substitution Cipher algorithm. 5. Write a program to implement RSA Algoithm 6. Write a Java program to implement DiffieHellman Key exchange Algoithm 7. To implement MFA in AWS 8. To create multiple users in CSP using IAM service roles. Assign different users different priveldges. 9. To implement Protect instances with AWS Virtual Private Cloud (VPC) 10. To Create outer firewall instance fwout 11. Configure routing tables on machines in the DMZ system 12. To implement a Brute fore Attack 13. To perform a SQL Injection attack 14. To implement Space Steganography 15. To implement Text Steganography **16.** To implement Image Steganography	12

Outcome:

- CO1: Understand the basics of Steganography techniques.
- CO2: Analyzing the effects of various attacks.
- CO3: Install the peripherals in a Cloud scenario with respect to Systems and Network Security.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2,PO3/PSO1,PSO2
CO2	PO1,PO2,PO3/PSO4
CO3	PO1,PO3,PO5/PSO2

BCSE0510:CONTAINER ORCHESTRATION & INFRASTRUCTURE AUTOMATION

Objective: *This course introduces the Docker, Kubernetes and micro-services in Deploy applications to production environment in Cloud Computing.*

Credits: 03 **Semester VI** **L–T–P-J: 3–0–0-0**

Module No.	Content	Hours
I	**Docker Fundamentals:** What is Docker? What isn't Docker? Server virtualization, Advantages of Docker, Is Docker a virtual machine? How does it help the developers? Why use Docker as a developer? Docker uses containers, why organizations embrace Docker containers? Containerdriven workloads, Substitutes for hypervisors, Utilizing micro services, Virtualization and containerization comparison, Convergence of containerization and virtualization, Containerization innovations, Overview of Docker editions, Installation of the Docker engine, Docker terminology, Docker community, Docker community edition, Docker enterprise edition, Build Kit features, Docker compose Container, Docker example, Let's understand Docker, Docker startup Docker and DevOps tools, Docker architecture Docker host, Understanding microservices, microservices architecture Docker components overview, Dockerize your applications, Docker for IBM Cloud, Containerization vs virtualization, Application containerization advantages and disadvantages, Running your own Docker container, Docker desktop for windows platform, Checking docker version, Running your first NGINX application, Docker repository,Docker tags, Docker TAG examples, Docker images, Managing containers and images, Docker volumes, Docker volumerelated commands, Docker volume drivers, Networking in Docker, Docker network types, Creating and inspecting a network, Summarize Docker Networking, Build a continuous Docker integration pipeline, CI workflow, Open network ports and protocols for communication, GitHub settings, Web-hooks and services.	20
II	**Configuring the Jenkins**: Test job configuration, Shell commands, Automation and orchestration, Key concepts in orchestration, Popular orchestra platforms: Swarm Docker, Kubernetes, Apache mesos, Container orchestration survey, Container adoption, Defining the functionality of container orchestration, Defining containers as a service function, Products/services used for container management and orchestration, Reducing container control by using configuration management tools, Orchestration primary method, Top orchestration products, Expected top orchestration products, Service discovery tools, Planning tools, Cluster management **What is Kubernetes?:** Clusters and architecture, Docker file instructions: CMD, Benefits, Kubernetes and DevOps, Kubernetes vs Docker, Kubernetes and IBM, Kubernetes architecture, Decentralized approach, Dynamic grouping, Kubernetes structure, Essential characteristics for manageability, API discovery and translation, Life of a request, API server internals, Bakery as a foundation, Defining the model of a bakery, Bakery standards, Bakery interactions, Bakery model practices discovery and implementation, Global bakery through automation, Cluster applications in a public cloud for Kubernetes, Kubernetes NWay active redundancy model, Top orchestration products, Configuration management and orchestration, Styles of workload orchestration, Container orchestration: Herding computational cattle, Emerging containers as a service marketplace, What is continuous delivery, DevOps culture, Docker, Jenkins and continuous delivery, Automation perspective, NexGen CI: Jenkins on docker, NexGen CD: Orchestrating docker, Blend of Jenkins with workflow, Container orchestration, Where in the system stack does the container orchestration fit?Tools available for container orchestration, Comparison to tracking containers without using an orchestration system, Continuous Integration/Continuous Deployment (CI/CD)	20

Reference Books:

- IBM Student Guide on "Container Orchestration and Infrastructure Automation".
- RajkumarBuyya "Mastering Cloud Computing".

Outcome: By the end of the class, students will learn to:

- CO1:Understand to solve problems that demonstrate how to use containers for your applications.
- CO2:Running containers in production and how to solve problems of advanced orchestration such as high availability, service discovery, and reconciliation in Cloud Computing.
- CO3:Evaluate whether Docker is an appropriate containerization platform for you.
- Describe how the components of Docker containers support compute container

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	
CO2	
CO3	

BCSE0538:CONTAINER ORCHESTRATION & INFRASTRCTURE AUTOMATION LAB

Objective: *This course introduces the hands on practice with Docker, Kubernetes and Jenkins open source containerization platform in cloud computing.*

Credits: 01 **Semester VI** **L–T–P: 0–0–2**

Module No.	Content	Lab Hours
I	Installation of Docker Get familiar with Docker basic Commands Docker images and docker containers How to run Jenkins on Docker Container? How to create Docker Compose File? How to create Docker Volume? How to create Docker Swarm? Hands-on practice on Kubernetes Create first Kubernetes Application	24

Reference Books:

- IBM Student Guide on "Container Orchestration & Infrastructure Automation Lab".
- A. Srinivasan, "Cloud Computing: A Practical Approach for Learning and Implementation"

Outcome: By the end of the class, students will learn to:

- Understand different cloud programming platforms and tools.
- Understand how to use Docker Containers.
- To develop scalable applications using Kubernetes.
- Understand the basic concepts of Docker images and Docker Repository.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	
CO2	
CO3	

BCSE0511: DEVOPS

Objective: *This course introduces the knowledge of DevOps Ecosystem in the Software Development.*

Credits: 03 **L–T–P–J: 3–0–2–0**

Module No.	Content	Hours
I	**Introduction to Devops**: Definition, Working and benefits of DevOps, DevOps history; **DevOps Phases**: Continuous Development, Continuous Integration, Continuous Testing, Continuous Deployment, Continuous Monitoring; **Agile Methodology**: Introduction to Agile Methodology, Traditional Waterfall Model V/s Agile Model, Agile V/s DevOps; **Business needs for DevOps**: DevOps Teams V/s Other Teams; **DevOps adoption**: Four DevOps adoption paths, Impact on Developers, Impact on Operations, Monolithic and Microservices Development. **DevOps Principles**: Development of DevOps team Culture, DevOps principles and DevOps roles, DevOps practices:	20
II	**Configuration Management**: Git workflow, Continuous Integration, Automated testing, Continuous Deployment, Continuous monitoring; **DevOps Lifecycle**. **Common Tools for DevOps**: Introduction to Docker, Introduction to Kubernetes, Introduction to Ansible, Introduction to Jenkins. **Testing, Automated Deployment and Monitoring**: Introduction to Testing, Verification and Validation, Black Box Testing V/s White Box Testing, DevOps monitoring using open source tools, Continuous Testing using DevOps. **DevOps with emerging technologies**: BigData, IoT	19

Text Books:

- DevOps- Student Guide, IBM

Reference Books:

- DevOps for Dummies, IBM Limited Edition – Sanjeev Sharma, John Wiley and Sons, Inc., 2014
- What is DevOps? – By Mike Loukides, O'Reilly Media, Inc., 2012

Outcome: *By the end of the class, students will learn to:*

- *Explain the benefits of DevOps Methodology with respect to traditional Software Development Methodology.*
- *Identify difference between DevOps and Agile Software Development methodology.*
- *Explain the concepts of DevOps while being Agile.*
- *Explain the Continuous Development, Continuous Integration, Continuous Testing and Continuous Delivery of Software.*
- *Work with the tools for DevOps*

Relate DevOps with the emerging technologies like BigData and IoT.

BCSE0701: INTRODUCTION TO MACHINE LEARNING

Objective: *The objective of this course to introduce basis process of machine learning, mathematical modeling of the supervised and unsupervised machine learning methods and to utilize combined voting of the different machine learning methods for solving real-world problems using machine learning. approach.*

L-T-P-J:0-0-2-0

Credits: 03

Module No.	Content	Teaching Hours
I	**Introduction:** Introduction to machine learning, Applications, and motivation, programming approach vs. machine learning approach in Artificial Intelligence, components of a learning problem (such as data, model, and error functions), basic learner, types of learning, features and feature vector, process of learning (training), testing, bias and variance error. **Python for Data Science**-Numpy, Pandas for preprocessing, Matplotlib and Jupyter Notebook. **Data Preprocessing**- Importing the Libraries, Importing the dataset, data imputation, Encoding Categorical Data, Splitting the dataset into Training and Test set, Feature Scaling. **Forecasting and Learning Theory:** Predicting numerical values: linear and non-linear regression. (Implementation on any real-world dataset e.g. Boston Housing), Regression model using Gradient Descendent. **Validation:** True and sample error, over-fitting, role of cross validation, regularization, bias-variance analysis. **Performance-Measures:** Types-of-errors, accuracy, confusion-matrix, precision-recall. **Project:** Estimation of diabeties using regression with gradient descendent.	21
II	**Dimensionality Reduction:** Feature Selection vs. feature extraction, Principal Component Analysis (PCA), Singular Value Decomposition. **Supervised Learning:** support vector machine, decision tree, Naïve Bayes classifier. **Unsupervised Learning:** clustering, Hierarchical clustering **Ensemble Learning**: Introduction, Bagging, Boosting, Improving classification, Ada-Boost algorithm. **Machine learning Approach in NLP-** Introduction to NLP libraries e.g. spacy, NLTK. Text classification using spacy, sentiment classification using spacy on IMDB dataset. **Introduction of CNN-** Difference between ANN and CNN, libraries to implement CNN and designing an application of image processing using CNN.	21

Text Books:

- Alpaydin, E. . Introduction to machine learning. MIT press, 2009.
- Bishop, C. M. . Pattern recognition and machine learning (information science and statistics) springer-verlag new york. Inc. Secaucus, NJ, USA, 2006.

Reference Books:

- Harrington, P. . Machine learning in action. Shelter Island, NY: Manning Publications Co , 2012.

Outcome: After completion of Lab, student will be able to:

- CO1: Apply the basic concepts of machine learning including bias-variance tradeoff.
- CO2: Apply the concepts of regression.
- CO3: Conceptualize supervise and re-enforcement learning for classification.
- CO4: Formulate the ensemble methods for improving classification.
- CO5: Apply ANN with optimization in machine learning.
- CO6: Design and develop projects based on machine learning.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO2,PO3/PSO1
CO3	PO4/PSO1,PSO3
CO4	PO2,PO3/PSO4
CO5	PO2,PO4/PSO4
CO6	PO1/PSO2,PSO4

BCSE0731: MACHINE LEARNING LAB

Credits: 02 **L–T–P-J: 2–0–0-0**

Module No.	Content	Teaching Hours
I & II	• Introduction to python, numpy and pandas • Implementation of Simple linear regression • Implementation of Multiple linear regression • Implementation of logistic regression • Implementation of regression models with regularization • Implementation of dimensionality reduction using PCA • Implementation of SVM • Implementation of Decision tree on real word data set • Implementation of Naïve Bayes Classifier • Implementation of k-means clustering • Implementation of ANN	20

Text Book:

- Alpaydin, E., ". Introduction to machine learning", MIT press, 2009.
- Bishop, C. M. ,"Pattern recognition and machine learning" , (information science and statistics) springer-verlag new york. Inc. Secaucus, NJ, USA, 2006.

Reference Books:

- Harrington, P., " Machine learning in action. Shelter Island" , NY: Manning Publications Co., 2012.

Outcome: After completion of Lab, student will be able to:

- CO1: Implement the basic concepts of machine learning including bias-variance tradeoff.
- CO2: Analyze data using regression and re-sampling methods.
- CO3: Perform supervise learning for classification.
- CO4: Apply and perform dimensionality reduction.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2/PSO3
CO2	PO2,PO3/PSO1,PSO3
CO3	PO1,PO4/PSO3,PSO4
CO4	PO4/PSO1,PSO3

BCSE0556: HADOOP AND BIG DATA ANALYTICS

Objective: *Exposure to any object oriented programming language (such as Java) and basic operational knowledge of any RDBMS (such as MySQL)*

Credits: 03 L-T–P-J: 3–0–0-0

Module No.	Content	Teaching Hours
I	**Big Data technology Landscape**: Big data growth story, big data sources, Types of Digital Data (Structured, Semi-Structured, Unstructured),Concept, importance and characteristics of data, Challenges with big data, Big data stack, Big Data 1.0, 2.0 and 3.0, Traditional BI vs. Big Data Environment, Big data Process, NoSQL Databases, NoSQL Vs. RDBMS, New SQL, Introduction to Data Science/Scientist **HADOOP 1.0:** Introducing Hadoop 1.0, Limitations of RDBMS, Hadoop Components, High Level Architecture of Hadoop , History of Hadoop, Special Features of Hadoop, Introduction to HDFS 1.0, Architecture, Daemons, Working with HDFS Command, Introduction to Map-Reduce 1.0, Architecture, Daemons **HADOOP 2.0:** Introducing Hadoop 2.0, Limitations of 1.0, Introduction to HDFS 2.0, Architecture, Daemons, Introduction to Map-Reduce 2.0, YARN, Architecture, Daemons, Word Count Example using Java, Introduction to Hadoop 3.0, Difference among Hadoop1.0, Hadoop2.0, Hadoop3.0 **Apache Spark:** Introduction, Introduction to Scala & functional programming, Spark Concepts: Main Primitives, RDD Fault Tolerance, Spark Operations, Job Execution, Spark Built-in libraries **Spark Streaming:** Real-time big data processing with Spark Streaming, the working of Spark Streaming and applications of Spark Streaming, Sliding Window Analytics	13
II	**Introduction to Cassandra DB**: Features of Cassandra, CQL Data Types, CQLSH: CRUD, Counter, TTL, List, Set, Map, Tracing, Import Export csv files **Introduction to HBase:** What is HBase? HBase Architecture, HBase Components, Data model, HBase Storage Hierarchy, Cross-Datacenter Replication, Auto Sharding and Distribution, Bloom Filter and Fold, Store, and Shift **HADOOP Ecosystem and Flume:** Introduction to Hadoop Ecosystem, Sqoop, Zookeeper, Kafka **Introduction to HIVE:** Hive Architecture, Hive Data types, Hive Collection Types, Hive File Formats, Hive Query Language, Hive Partitions, Bucketing, Views, RCFile Implementation, Hive User Defined Function, SerDe, UDF **Introduction to Pig:** History and Anatomy of Pig, Pig on Hadoop, Use Case for Pig, Pig Primitive Data Types, Pig Latin Overview, Execution Modes of Pig, Field, Tuple, Bag, User Defined Function, Parameters in Pig, Piggy Bank, Word count example using Pig, Pig vs Hive, When to use Pig.	14

Text Book:

- IBM Technologies ICE "Big Data Analytics Student Guide"
- Seema Acharya and Subhashini Chellappan, "Big Data and Analytics", 1st Edition, 2015, Wiley, India.
- Jure Leskovec, Anand Rajaraman, Jeff Ullman, "Mining of Massive Datasets", 2nd Edition, 2014, Cambridge University Press.

Reference Books:

- Chuck Lam, "Hadoop in Action", 2nd Edition , Manning Publications, 2014.

Outcome: After completion of course, student will be able to:

- CO1: Understand the concept and challenges of big data

CO2: Apply the existing technology to collect, manage, store, query, and analyze the big data.

- CO3: Implement job scheduling and resource management of the cluster using Hadoop and Yarn.
- CO4: Apply data summarization, query, and analysis over the big data with the help of pig and hive.
- CO5: Design the regression model, cluster and decision tree over the real big data.
- CO6: Implement different real life applications by using large-scale analytics tools.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO3
CO2	PO3/PSO4
CO3	PO2,PO3,PO11/PSO2,PSO4
CO4	PO5,PO2/PSO4
CO5	PO2,PO3/PSO2
CO6	PO3/PSO1

BCSE0585: HADOOP & BIG DATA ANALYTICS LAB

Objective: *This course introduces students to R, a widely used statistical programming language. Students will learn to manipulate data objects, produce graphics, analyse data using common statistical methods, and generate reproducible statistical reports. Student will also learn data mangling.*

Credits: 01 L-T–P-J: 0–0–2-0

Module No.	Content	Lab Hours
I&II	**Module 1: Introduction to R** • Introduction and installation of R and RStudio • Data types, vectors, multidimensional array. • R utilities function • Correlation and Linear Regression • Logistics Regression **Module 2: Hands-On MongoDB, Cassandra** • Installation of VM-Ware and Cloudera • Hands-On Mongo DB: CRUD, Where, Aggregation • Hands-On Mongo DB: Projection, Aggregation • Hands-On Cassandra DB: CRUD, Projection **Module 3: Hands-On MapReduce** • HDFS and its commands • Word-Count program using Map Reduce • Hands-On HIVE and Pig	24

Reference Books:

- Paul Teetor. R Cookbook: Proven recipes for data analysis, statistics, and graphics. O'Reilly Media, Inc.,2011.
- Norman Matloff. Theart of R programming: A tour of statistical software design. No Starch Press, 2011.
- Winston Chang. R graphics cookbook. O'Reilly Media, Inc., 2012.
- Hadley Wickham and Garrett Grolemund. R for data science. 2016.
- Phil Spector. Data manipulation with R. Springer Science& Business Media,2008.

Outcome: After completion of course, student will be able to:
Use R-Studio, read R documentation, and write R scripts.

- Import, export and manipulate data.
- Analyse the data using data analytics latest tools based on HDFS like Pig, Hive
- Implement the aggregation projection on data set using Cassandra, MongoDB.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO5/PSO2
CO2	PO2, PO4/ PSO2
CO3	PO2, PO5/PSO3
CO4	PO3, PO5/PSO4

BCSE0252: FULL STACK USING NODE JS

Objective: *The objective is to provide a comprehensive study of the Backend. It stresses the strengths of Web (Full Stack), which provide students with the means of writing efficient, maintainable, and portable Website.*

Credits: 03 **L–T–P:J: 3–0–0-0**

Module No.	Content	Teaching Hours
I	**MongoDB:** Introduction to MongoDB, MongoDB Environment, MongoDB Create Database, MongoDB Drop Database, MongoDB Create Collection, MongoDB Drop Collection, MongoDB Read Operations, MongoDB Write Operations, MongoDB Data Modelling, MongoDB Administration. **Angular JS:** Introduction to AngularJS, AngularJS Expressions, AngularJS Modules, AngularJS Data Binding, AngularJS Scopes, AngularJS Directives & Events, AngularJS Controllers, AngularJS Filters, AngularJS Services, AngularJS HTTP, AngularJS Tables, AngularJS Select, Fetching Data from MySQL, AngularJS Validation, AngularJS API. **Express Framework:** Introduction to Express Framework, Introduction to Nodejs, what is Nodejs, Getting Started with Express, Express Routing,	20
II	**Express Framework:** Implementing MVC in Express, Middleware, Using Template Engines, Error Handling, API Handling, Debugging, Developing Template Engines, Using Process Managers, Security & Deployment. **Node.js:** Introduction to Node JS, Setup Dev Environment, Node Core, Node Modules, Creating Web server, File System, Debugging Node JS Application, Automation and Deployment, Events & Database connectivity. **React.js:** Welcome to Starting with React, React Components, React State and Props, React Event Handling, Routing in React React flux, &. Styling React	20

Text Books:

- Node.js, MongoDB and Angular Web Development: The definitive guide to using the MEAN stack to build web applications by Brad Dayley, Brendan Dayley, Caleb Dayley
- Express.js: Guide Book on Web framework for Node.js by Rick L.
- Introduction to React by Cory Gackenheimer,

Outcome: After completion of course, student will be able to:

- CO1: Apply programming concepts using Node.Js.
- CO2: Develop web application using MongoDB and AngularJs.
- CO3: Develop web application based on MongoDB.
- CO4: Understand project management and code.
- CO5: Develop RESTful and MVC based web application.
- CO6: Understand the basic concepts of React.js.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1/PSO1
CO2	PO3/PSO2
CO3	PO3/PSO3
CO4	PO2/PSO3
CO5	PO3/PSO2
CO6	PO2/PSO1

BCSE0282: FULL STACK USING NODE JS LAB

Objective: *The objective is to provide a comprehensive study of the Backend. It stresses the strengths of Web (Full Stack), which provide students with the means of writing efficient, maintainable, and portable Website.*

Credits: 01 **L–T–P-J: 0–0–2-0**

Module No.	Content	Lab Hours
I/II	• Installing and Managing MogoDB. • Create & Manage Database. • Create & Manage collections • Implementation of Modeling. • Create your first AngularJS application in Visual Studio. • Build a shopping cart using AngularJS • Implementation AngularJS Expressions • Implementation AngularJS Modules • Implementation AngularJS Events • Implementation AngularJS Filters & Services • Node JS- Setup Dev Environment. • Express Routing • Implementing MVC in Express. • Implementing Template Engines. • Implementing Node Modules, • Implementing React Components, • Implementing React Event.	12*2=24

Reference Books:

- Node.js, MongoDB and Angular Web Development: The definitive guide to using the MEAN stack to build web applications by Brad Dayley, Brendan Dayley, Caleb Dayley
- Express.js: Guide Book on Web framework for Node.js by Rick L.
- Introduction to React by Cory Gackenheimer,

Outcome: After studying the subject, the students will be able to:

- CO1: Implement web application using MongoDB and Angular.Js.
- CO2: Develop web application using NodeJs.
- CO3: Develop RESTful and MVC based web application.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO5/PSO2
CO2	PO3/PSO2
CO3	PO5/PSO4

BCSE0053: AGILE SOFTWARE DEVELOPMENT

Prerequisite: *Exposure of Software Engineering Principles*

Credits: 03 **L–T–P-J: 3–0–0-0**

Module No.	Content	Teaching Hours
I	**Fundamentals of Agile:** The Genesis of Agile, Introduction and background, Agile Manifesto and Principles, Overview of Scrum, Extreme Programming, Feature Driven development, Lean Software Development, Agile project management, Design and development practices in Agile projects, Test Driven Development, Continuous Integration, Refactoring, Pair Programming, Simple Design, User Stories, Agile Testing, Agile Tools **Agile Scrum Framework:** Introduction to Scrum, Project phases, Agile Estimation, Planning game, Product backlog, Sprint backlog, Iteration planning, User story definition, Characteristics and content of user stories, Acceptance tests and Verifying stories, Project velocity, Burn down chart, Sprint planning and retrospective, Daily scrum, Scrum roles – Product Owner, Scrum Master, Scrum Team, Scrum case study, Tools for Agile project management **Agile Software Design:** Agile design practices, Role of design Principles including Single Responsibility Principle, Open Closed Principle, Liskov Substitution Principle, Interface Segregation Principles, Dependency Inversion Principle in Agile Design,	26
II	**Agile Software Development:** Need and significance of Refactoring, Refactoring Techniques, Continuous Integration, Automated build tools, Version control. Current researches in Agile software development **Agile Testing:** The Agile lifecycle and its impact on testing, Test-Driven Development (TDD), xUnit framework and tools for TDD, testing user stories - acceptance tests and scenarios, Planning and managing testing cycle, Exploratory testing, Risk based testing, Regression tests, Test Automation, Tools to support the Agiletester **Industry Trends:** Market scenario and adoption of Agile, Agile ALM, Roles in an Agile project, Agile applicability, Agile in Distributed teams, Business benefits, Challenges in Agile, Risks and Mitigation, Agile projects on Cloud, Balancing Agility with Discipline, Agile rapid development technologies	26

Text Book:

- Ken Schawber& Mike Beedle, Agile Software Development with Scrum, Pearson, 2008

Reference Books:

- KenSchawber & MikeBeedle, AgileSoftware Development with Scrum,Pearson,2008
- RobertC.Martin , Agile Software Development ,Principles, Patterns and Practices, Prentice Hall,2002
- Lisa Crispin & JanetGregory, AgileTesting: A Practical Guide for Testers and Agile Teams, Addison Wesley,2008
- Alistair Cockburn, Agile Software Development: The Cooperative Game ,Addison Wesley ,2006

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

- CO1: UnderstandthesignificanceofAgileMethodologiesinsoftwaredevelopment.
- CO2: Compare and contrast the different agile methods.
- CO3: Determine the suitability of agile methods for a particular Project.
- CO4:Evaluate how well a project is following agile principles, and assist the project to become moreagile (where appropriate).
- CO5: Understand the relationship between the customer and the development team in agile projects and the responsibilities of both communities.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO3,PO5,PO7,PO10/PSO1,PSO4
CO2	PO2,PO4,PO9/PSO2,PSO4
CO3	PO2,PO6,PO11/PSO1,PSO4
CO4	PO1,PO2,PO4/PSO1,PSO3
CO5	PO8,PO9,PO10,PO11/PSO2

BCSE0203: INTERNET OF THINGS

Objective: *To Implement Data and Knowledge Management and use of Devices in IoT Technology.*

Credits: 03 **L–T–P-J: 3–0–0-0**

Module No.	Content	Teaching Hours
I	**Introduction to IoT and Sensor:** Introduction to IoT- Sensing, Actuation, Logical design of IoT, Functional blocks/pillars/components of IoT, Communication models, IoT& M2M: Machine to Machine, Difference between IoT and M2M, **Introduction to Sensors:** About sensor, Properties of Sensors Basic physical principles of sensing, Categorization of Sensor, PIR Sensor, Temperature Sensor, Ultrasonic Sensor, IR Sensor, MQ2/MQ3 **Implementing IoT:** Introduction to Arduino Programming. Integration of Sensors and Actuators with Arduino. Implementation of IoT with Arduino, Node MCU. Mini project Statement using Node MCU.	20
II	**IoT Over Network:** IOT Networking Protocols: TCP/IP, 6LowPan, Routing Protocols, Thread. **Communication Protocol:** CoAP, SMCP, SMTP, HTTP, HTTPS, MQTT, MQTT-S SDN for IoT, Data Handling and Analytics. Cloud Computing, Fog Computing, Introduction to different IoT Tools. Implementation of IoT with Raspberry Pi.	20

Books for reference:

1. "The Internet of Things: Enabling Technologies, Platforms, and Use Cases", by Pethuru Raj and Anupama C. Raman (CRC Press)
2. "Internet of Things: A Hands-on Approach", by ArshdeepBahga and Vijay Madisetti (Universities Press)

Outcome:

- CO1: Understand the concepts of Internet of Things.
- CO2: Understand difference between Sensors and Actuators and their working principles.
- CO3: Design IoT applications using different sensors and actuators.
- CO4: Understand different protocols used in IoT over network.
- CO5: Understand different communication protocols.
- CO6: Explain the concept of cloud computing and fog computing.
- CO7: Implement IoT application using Raspberry Pi.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO3/PSO1
CO2	PO2 /PSO4
CO3	PO5/PSO2
CO4	PO11/PSO2
CO5	PO1,PO3/PSO4
CO6	PO2/PSO1
CO7	PO5/PSO3

BCSE0232: INTERNET OF THINGS LAB

Objective: *Coordinate and help to increase and optimize the utilization of results and value creation in the area of IoT.*

Credits: 01 **L–T–P-J: 0–0–2-0**

Module No.	Content	Lab Hours
I/II/III	• WAP to interface and blink the LED using Arduino UNO. • WAP to interface for different sensors (Like DHT11, temperature, IR, Ultrasonic etc) to Arduino UNO. • WAP to interface temperature sensor to ESP8266. Turn on the LED if temperature value met threshold value. • WAP to interface in between Bluetooth module and Arduino UNO. • Write a python program for Gateway to store sensor data on local MySQL database. • WAP to transmit the data wirelessly for longer distance using multi-hop technique. • Configure the gateway as local MQTT broker (Mosquito), configure one ESP8266 as sender (Publisher), and receive the data on the Smartphone (MQTT Dashboard).	12*2=24

Text Books:

- Upskill Learning , "ESP8266: Programming NodeMCU Using Arduino IDE - Get Started With ESP8266 (Internet Of Things, IOT, Projects In Internet Of Things, Internet Of Things for Beginners, NodeMCU Programming, ESP8266",2018.

Outcome: After completion of course, student will be able to:

- CO1: Design IoT applications using different sensors and actuators.
- CO2: Design IoT applications in different domain and be able to analyze their performance.
- CO3: Implement basic IoT applications on embedded platform.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO3,PO5/PSO1
CO2	PO4/PSO1
CO3	PO1/PSO4

BCSC0013: COMPILER DESIGN

Objective: *The course objective is to introduce the major concept areas of language translation and compiler design and to enrich the knowledge in various phases of compiler ant its use, code optimization techniques, machine code generation, and use of symbol table.*

Credits:04 **L–T–P-J: 3–1–0-0**

Module No.	Content	Teaching Hours
I	**Introduction to Compiler:** Phases and passes, bootstrapping, Optimization of DFA-Based Pattern Matchers implementation of lexical analyzers, lexical analyzer generator, LEX-compiler, YACC, Context free grammars, derivation and parse trees, capabilities of CFG. **Basic Parsing Techniques:** Parsers, Shift reduce parsing, operator precedence parsing, top down parsing, predictive parsers Automatic Construction of efficient Parsers: LR parsers, the canonical Collection of LR(0) items, constructing SLR parsing tables. **Advance Parser:** Constructing Canonical LR parsing tables, Constructing LALR parsing tables, using ambiguous grammars, an automatic parser generator, implementation of LR parsing tables.	20
II	**Syntax-directed Translation:** Syntax-directed Translation schemes, Implementation of Syntax directed Translators, Intermediate code, postfix notation, Parse trees & syntax trees, three address code, quadruple & triples, translation of assignment statements, Boolean expressions, statements that alter the flow of control, postfix translation, translation with a top down parser. More about translation: Array Reference, Cases: in arithmetic expressions, procedures call, declarations and case statements. **Symbol Tables:** Data structure for symbols tables, representing scope information. Run-Time Administration: Implementation of simple stack allocation scheme, Storage allocation in block structured language. **Error Detection & Recovery:** Lexical Phase errors, Syntactic phase errors, semantic errors. **Code optimization:** Machine-Independent Optimizations, Loop optimization, DAG representation of basic blocks, Value numbers and algebraic laws, Global Data-Flow analysis. **Code Generation:** Design Issues, Target Language. Addresses in the Target Code, Basic Blocks and Flow Graphs, Optimization of Basic Blocks, Code Generator.	20

Text Book:

- Aho, Sethi& Ullman , "Compilers: Principles, Techniques and Tools", 2nd Edition, Pearson Education, 2008.

Reference Books:

- V Raghvan, "Principles of Compiler Design", 2nd Edition, TMH, 2010.
- Kenneth Louden , "Compiler Construction", 1st Edition, Cengage Learning, 1997.
- Charles Fischer and Ricard LeBlanc , "Crafting a Compiler with C", Pearson Education, 2005

Outcome: After the completion of the course, the student will be able to:

- CO1: Understand basics of Compilers and its phases.
- CO2: Design top-down and bottom-up parsers and will be able to solve problems related to predictive parser, Shift reduce parsing, compute FIRST and FOLLOW sets, LR (0), LR (1) and LALR sets of items and parse table for a given grammar.

- CO3: Demonstrate the ability to write syntax directed translations of simple statements and understand the working of procedure calls.
- CO4: Demonstrate the ability to write intermediate code for a given high level programming language (preferably C or FORTRAN) and be able to represent the intermediate code as Quadruples, Triples and Indirect Triples
- CO5: Identify the basic blocks for three address code, draw flow graphs and represent directed acyclic graphs for the identified basic blocks.
- CO6: Write the target optimized code (assembly code) for the given three-address code.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO2,PO3,PO4/PSO1,PSO2,PSO3
CO2	PO1,PO2,PO3,PO4,PO5/PSO1,PSO2,PSO3
CO3	PO1,PO2,PO3,PO4/PSO1,PSO2,PSO3
CO4	PO1,PO2,PO3,PO5/PSO1,PSO2,PSO3
CO5	PO1,PO2,PO3,PO4,PO5/PSO1,PSO2,PSO3
CO6	PO1,PO2,PO3,PO4,PSO5/PSO1,PSO2,PSO3