COURSE STRUCTURE

B.TECH.

COMPUTER SCIENCE & ENGINEERING

Under

Choice Based Credit System (CBCS)

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.	BCHS0101/ BPHS0001	Engineering Chemistry/ Engineering Physics	3	1	0	4	4
3.	BELH0001	English Language Skills for Communication I	2	0	0	2	2
4.	BEEG0001/ BECG0001	Basic Electrical Engineering / Electronics Engineering	3	1	0	4	4
5.	BCSC0001	Computer Programming	4	1	0	5	5
PRACTICALS							
6.	BCHS0801/ BPHS0801	Engineering Chemistry Lab / Engineering Physics Lab	0	0	2	1	2
7.	BELH0801	English Language Lab I	0	0	2	1	2
8.	BEEG0800/ BECG0800	Electrical Engineering Lab/ Electronics Lab I	0	0	2	1	2
9.	BMEG0801	Engineering Drawing Lab	0	0	2	1	2
10.	BCSC0800	Computer Programming Lab	0	0	2	1	2
		TOTAL	**15**	**4**	**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.	BPHS0001/ BCHS0101/	Engineering Physics/ Engineering Chemistry	3	1	0	4	4
3.	BELH0002	English Language Skills for Communication II	2	0	0	2	2
4.	BECG0001/ BEEG0001	Electronics Engineering/ Electrical Engineering	3	1	0	4	4
5.	BMEG0001	Basic Mechanical Engineering	3	1	0	4	4
6.	BCSG0001	Python Programming	4	1	0	5	5
PRACTICALS							
7.	BPHS0801/ BCHS0801	Engineering Physics Lab/ Engineering Chemistry Lab	0	0	2	1	2
8.	BELH0802	English Language Lab II	0	0	2	1	2
9.	BECG0800/ BEEG0800	Electronics Lab I/ Electrical Engineering Lab	0	0	2	1	2
10.	BMEG0800	Engineering Workshop Practice Lab	0	0	2	1	2
11.	BCSG0800	Python Programming Lab	0	0	2	1	2
		TOTAL	**18**	**5**	**10**	**28**	**33**

Program Core

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE-REQUISITES
			L	T	P	J			
THEORY									
1.	BCSG0001	Python Programming	4	1	0	0	5	5	
2.	BCSC0001	Computer Programming	4	1	0	0	5	5	
3.	BCSC0002	Object Oriented Programming	3	0	0	0	3	3	Programming
4.	BCSC0003	Database Management System	3	0	0	0	3	3	
5.	BCSC0004	Operating Systems	3	0	0	0	3	3	
6.	BCSC0005	Computer Organization	3	0	0	0	3	3	
7.	BCSC0006	Data Structures and Algorithms	3	1	0	0	4	4	Programming
8.	BCSC0007	Introduction to Microprocessors	3	0	0	0	3	3	Computer Organization
9.	BCSC0008	Computer Networks	3	1	0	0	4	4	
10.	BCSC0009	Software Engineering	3	0	0	0	3	3	
11.	BCSC0010	Discrete Mathematics	3	1	0	0	4	4	
12.	BCSC0011	Theory of Automata and Formal Language	3	1	0	0	4	4	
13.	BCSC0012	Design and Analysis of Algorithms	3	0	0	0	3	3	Programming , Data Structures
PRACTICALS									
1.	BCSG0800	Python Programming Lab	0	0	2	0	1	2	
2,	BCSC0800	Computer Programming Lab	0	0	2	0	1	2	
3.	BCSC0801	Object Oriented Programming Lab	0	0	2	0	1	2	Programming Lab
4.	BCSC0802	Database Management System Lab	0	0	2	0	1	2	
5.	BCSC0803	Operating Systems Lab	0	0	2	0	1	2	
6.	BCSC0804	Computer Organization Lab	0	0	2	0	1	2	
7.	BCSC0805	Data Structures and Algorithms Lab	0	0	2	0	1	2	Programming Lab
8.	BCSC0806	Microprocessors Lab	0	0	2	0	1	2	
9.	BCSC0807	Design and Analysis of Algorithms Lab	0	0	2	0	1	2	Programming , Data Structures
Total			**41**	**6**	**18**	**0**	**56**	**63**	

Program Elective

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE- REQUISITES
			L	T	P	J			
Bouquet: Computer Network and Security									
THEORY									
1.	BCSE0001	Network Programming and Management	3	0	0	0	3	3	Computer Networks
2.	BCSE0002	Principles of Mobile Computing	3	1	0	0	4	4	Computer Networks
3.	BCSE0003	Ad Hoc Networks	3	0	0	0	3	3	Computer Networks
4.	BCSE0004	Cryptography and Network Security	3	0	0	0	3	3	Computer Networks
5.	BCSE0005	Cyber security and Digital Forensics	3	0	0	0	3	3	Computer Networks
6.	BCSE0006	Information Coding Techniques	3	0	0	0	3	3	Computer Networks
PRACTICALS									
7.	BCSE0031	Network Programming and Management Lab	0	0	2	0	1	2	Computer Networks
8.	BCSE0032	Cryptography and Network Security Lab	0	0	2	0	1	2	Computer Networks
9.	BCSE0033	Information Coding Techniques Lab	0	0	2	0	1	2	Computer Networks
PROJECTS									
10.	BCSE0041	Information Coding Techniques Project	0	0	0	-	2	0	Computer Networks

Head of the Department
Computer Engineering & Applications
Institute of Engineering & Technology
GLA University, Mathura

Program Elective

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE-REQUISITES
			L	T	P	J			
Bouquet: Software Engineering									
THEORY									
1.	BCSE0051	Software Quality Engineering	3	0	0	0	3	3	Software Engineering
2.	BCSE0052	Service Oriented Architecture	3	0	0	0	3	3	Software Engineering
3.	BCSE0053	Agile Software Development	3	0	0	0	3	3	Software Engineering
4.	BCSE0054	Software Project Management	3	0	0	0	3	3	Software Engineering
5.	BCSE0055	Software Testing	3	0	0	0	3	3	Software Engineering
6.	BCSE0056	Compiler Design	3	1	0	0	4	4	Theory of Automata & Formal Language

Program Elective

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE- REQUISITES
			L	T	P	J			
Bouquet: Image Processing and Intelligent System									
Theory									
1.	BCSE0101	Digital Image Processing	3	0	0	0	3	3	Mathematics, Programming
2.	BCSE0102	Computer Graphics And Multimedia	3	1	0	0	4	4	Mathematics, Programming
3.	BCSE0103	Soft Computing	3	0	0	0	3	3	Discrete Mathematics
4.	BCSE0104	Artificial Intelligence	3	0	0	0	3	3	Data Structures
5.	BCSE0105	Machine Learning	3	0	0	0	3	3	Mathematics, Programming
6.	BCSE0106	Machine Learning And Its Applications	3	0	0	0	3	3	Mathematics, Programming
Labs									
1.	BCSE0131	Digital Image Processing Lab	0	0	2	0	1	2	Programming
2.	BCSE0132	Soft Computing Lab	0	0	2	0	1	2	Programming
4.	BCSE0133	Machine Learning Lab	0	0	2	0	1	2	Programming
Projects									
1.	BCSE0141	Machine Learning Project	0	0	0	-	2	-	Programming

Program Elective

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE-REQUISITES
			L	T	P	J			
Bouquet: Advanced Data Processing									
Theory									
1.	BCSE0151	Advanced Concepts in Database Systems	3	0	0	0	3	3	DBMS
2.	BCSE0152	Data Mining and Warehousing	3	0	0	0	3	3	DBMS
3.	BCSE0153	Business Intelligence	3	0	0	0	3	3	DMW
4.	BCSE0154	Information Retrieval System	3	0	0	0	3	3	DATA STRUCTURE
5.	BCSE0155	Distributed and Parallel Databases	3	0	0	0	3	3	DBMS
6.	BCSE0156	Natural Language Processing	3	0	0	0	3	3	TAFL/Compiler Design
7.	BCSE0157	Introduction to Big Data Analytics	3	0	0	0	3	3	DBMS
8.	BCSE0158	Big Data Analytics	3	0	0	0	3	3	
Labs									
1.	BCSE0181	Data Mining and Warehousing Lab	0	0	2	0	1	2	
2.	BCSE0182	Business Intelligence Lab	0	0	2	0	1	2	
3.	BCSE0183	Big Data Analytics Lab	0	0	2	0	1	2	
Projects									
1.	BCSE0191	Business Intelligence Project	0	0	0	-	2	-	
2.	BCSE0192	Big Data Analytics Project	0	0	0	-	2	-	Programming

Program Elective

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE- REQUISITES
			L	T	P	J			
Bouquet: High Performance Computing									
Theory									
1.	BCSE0201	Advanced Computer Architecture	3	0	0	0	3	3	Computer Organization
2.	BCSE0202	Embedded System	3	0	0	0	3	3	Microprocessors
3.	BCSE0203	Internet of Things	3	0	0	0	3	3	Microprocessors
4.	BCSE0204	Internet of Things And Its Applications	3	0	0	0	3	3	
5.	BCSE0205	Distributed System	3	0	0	0	3	3	CN /OS
6.	BCSE0206	Parallel Algorithms	3	0	0	0	3	3	CO & Algorithms
7.	BCSE0207	Cloud Computing	3	0	0	0	3	3	
8.	BCSE0208	Cloud Computing and Virtualization	3	0	0	0	3	3	
Labs									
1.	BCSE0231	Embedded System Lab	0	0	2	0	1	2	
2.	BCSE0232	Internet of Things Lab	0	0	2	0	1	2	
3.	BCSE0233	Parallel Algorithms Lab	0	0	2	0	1	2	
4.	BCSE0234	Cloud Computing lab	0	0	2	0	1	2	
Projects									
1.	BCSE0241	Internet of Things Project	0	0	0	-	2	-	
2.	BCSE0242	Cloud Computing And Virtualization Project	0	0	0	-	2	-	

Program Elective

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE- REQUISITES
			L	T	P	J			
Bouquet: Development Tools and Technologies									
Theory									
1.	BCSE0251	Full Stack Using Scripting Technologies	3	0	0	0	3	3	
2.	BCSE0252	Full Stack Using Node JS	3	0	0	0	3	3	
3.	BCSE0253	Full Stack Using C#.net	3	0	0	0	3	3	
4.	BCSE0254	PHP - Scripting Language	3	0	0	0	3	3	
5.	BCSE0255	Digital Marketing And Transformation	3	0	0	0	3	3	
Labs									
1.	BCSE0281	Full Stack Using Scripting Technologies Lab	0	0	2	0	1	2	
2.	BCSE0282	Full Stack Using Node JS Lab	0	0	2	0	1	2	
3.	BCSE0283	Full Stack Using C#.net Lab	0	0	2	0	1	2	
4.	BCSE0284	PHP - Scripting Language Lab	0	0	2	0	1	2	
Projects									
1.	BCSE0291	Full Stack Using Scripting Technologies Project	0	0	0	0	2	-	
2.	BCSE0292	Full Stack Using Node JS Project	0	0	0	0	2	-	
3.	BCSE0293	Full Stack Using C#.net Project	0	0	0	0	2	-	
4.	BCSE0294	PHP - Scripting Language Project	0	0	0	0	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	
		TOTAL	**8**	**0**	**0**	**0**	**0**	**8**	

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.	MBAC0005	Industrial Management	3	0	0	0	3	3	
Practicals									
7.	BELH0801	English Language Lab – I	0	0	2	0	1	2	
8.	BELH0802	English Language Lab – II	0	0	2	0	1	2	
9.	BTDH0301	Soft Skills – I	0	0	2	0	1	2	
10.	BTDH0302	Soft Skills – II	0	0	2	0	1	2	
11.	BTDH0303	Soft Skills – III	0	0	8	0	4	8	
12.	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.	BMAS0103	Engineering Mathematics III	3	1	0	0	4	4	
4.	BCHS0101	Engineering Chemistry	3	1	0	0	4	4	
5.	BPHS0001	Engineering Physics	3	1	0	0	4	4	
6.	BCHS0201	Environmental Studies	2	0	0	0	2	2	
PRACTICALS									
7.	BCHS0801	Engineering Chemistry Lab	0	0	2	0	1	2	
8.	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.	BCSG0001	Python Programming	4	1	0	0	5	5	
5.	BCSC0001	Computer Programming	4	1	0	0	5	5	
PRACTICALS									
6.	BEEG0800	Electrical Engineering Lab	0	0	2	0	1	2	
7.	BECG0800	Electronics Lab I	0	0	2	0	1	2	
8.	BMEG0800	Engineering Workshop Practice Lab	0	0	2	0	1	2	
9.	BMEG0801	Engineering Drawing Lab	0	0	2	0	1	2	
10.	BCSG0800	Python Programming Lab	0	0	2	0	1	2	
11	BCSC0800	Computer Programming Lab	0	0	2	0	1	2	

Open Elective (Offer to other Departments)

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE- REQUISITES
			L	T	P	J			
THEORY									
1.	BCSO0001	Data Structures and Applications	3	0	0	0	3	3	Programming
2.	BCSO0002	Introduction To Object Oriented Programming	3	0	0	0	3	3	Programming
3.	BCSO0003	Essentials of Information Technology	3	0	0	0	3	3	Object Oriented Programming
4.	BCSO0004	Elements of Soft Computing	3	0	0	0	3	3	
5.	BCSO0005	Fundamentals of Computer	2	0	0	0	2	2	
6.	BCSO0006	Introduction to Programming	2	0	0	0	2	2	
PRACTICALS									
7.	BCSO0070	Data Structures and Applications Lab	0	0	2	0	1	2	Programming
8.	BCSO0071	Introduction To Object Oriented Programming Lab	0	0	2	0	1	2	Programming
9.	BCSO0072	Essentials of Information Technology Lab	0	0	2	0	1	2	Object Oriented Programming
10.	BCSO0073	Elements of Soft Computing Lab	0	0	2	0	1	2	
11.	BCSO0074	Fundamentals of Computer Lab	0	0	2	0	1	2	
12.	BCSO0075	Programming Lab	0	0	2	0	1	2	

BCSG0001: PYTHON PROGRAMMING

Objective: *This course introduces the solving of mathematical problems using Python programming using OO 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 dataframes& 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 to solve problems with smaller Lines of Code using Python as compared to other programming languages
- CO2: Apply OO concepts while programming in Python
- CO3: Demonstrate packages defined in Python
- CO4: Work with Python using GUI

BCSC0001: 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: Familiar with the concepts related to structured programming constructs
- CO2: Design an algorithmic solution for a given problem and as well C program for a given algorithm.
- CO3: Demonstrate their knowledge of, and ability to apply, programming fundamentals in different programming languages.
- CO4: Apply knowledge of computing and mathematics appropriate to the discipline; specifically, to include the application of mathematics, science and engineering to solve and reason about computational problems.
- CO5: Apply the concepts and design principles relating to: data structures, computer architecture and organization, programming languages, operating systems, and networks.

BCSC1002: 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:3-0-0**

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, StringTokenizer, StringBuilder & StringBuffer. **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 Superclasses, extend Abstract Classes, implementing Interfaces, exploring polymorphic behaviour 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. **JDBC:** JDBC Drivers, Connecting to a MySQL Database, DriverManager, Connection Interface, Statement Interface, Result Set Interface, Prepared Statements.	18

Text Book:

- Herbert Schildt (2019), “The Complete Reference, Java Eleventh Edition”, Oracle Press.

Reference Book:

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

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

- CO1: Understand and implement Abstraction, Encapsulation, Inheritance and Polymorphism in Java programs.
- CO2: Demonstrate Exception Handling abilities to create robust programs.
- CO3: Implement a database in Java programs.
- CO4: Use standard libraries in Java.
- CO5: Design programs that resemble real-world scenarios.

Head of the Department
Computer Engineering & Applications
Institute of Engineering & Technology
GLA University, Mathura

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

References Books:

- Date C J," *An Introduction to Database Systems*", 8th Edition, Addison Wesley.
- Korth, Silbertz and Sudarshan (1998), "*Database Concepts*", 5th Edition, TMH.
- 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: Explain the features of database management systems and Relational database.
- CO2: Design conceptual models of a database using ER modeling for real life applications and construct queries in Relational Algebra.
- CO3: Create and populate a RDBMS for a real life application, with constraints and keys,using SQL.
- CO4: Retrieve any type of information from a database by formulating complex queries in SQL.
- CO5: Analyze the existing design of a database schema and apply concepts of normalization to design an optimal database.
- CO6: Build indexing mechanisms for efficient retrieval of information from a database.

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 (2012), "*Operating Systems Concepts*",9th Edition, Wiley.

Reference Books:

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

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

- CO1: Describe the important computer system resources and the role of operating system in their management policies and algorithms.
- CO2: Understand the process management policies and scheduling of processes by CPU.
- CO3: Evaluate the requirement for process synchronization and coordination handled by operating system.
- CO4: Describe and analyze the memory management and its allocation policies.
- CO5: Identify use and evaluate the storage management policies with respect to different storage management technologies.
- CO6: Identify the need to create the special purpose operating system.

BCSC0005: 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	**PREAMBLE:** Subject Introduction, Basic organization of the computer and block level description of the functional units, Number Representation, Fixed and floating-point Number Representation-Arithmetic Addition/subtraction, overflow, IEEE standard for floating point representation, **Basic Organization:** Introduction to combinational circuit - Half Adder, Full Adder, carry look ahead adder, Multiplexor/ De multiplexer and Decoder/Encoder, Introduction to sequential circuit- Flip-Flops, Synchronous and Asynchronous Counters, Register, Bus and memory Transfer Language. **Arithmetic Operations:** Addition and subtraction of signed numbers, Hardware implementation of Method, Multiplication: Signed operand multiplication, Booths algorithm, Hardware implementation of Algorithms, Array Multiplier. **Processor Organization:** General register organization, Single Accumulator and Stack organization, Addressing Modes, Types of Computer Instructions – one, two, three & four address, Instruction Cycle, Instruction Formats.	20
II	**Micro-operations:** Arithmetic, Logical & Shift Micro operations with some applications. **Multiprogramming and Multiprocessing:** Introduction to pipelined operation. **Hardwired & Microprogrammed Unit:** Execution of a complete instruction & Branch Instructions, Hardwired control Unit, Micro programmed control Unit, Micro-Instructions, Microinstruction with Next Address field, Pre-Fetching Microinstructions, Concept of Horizontal and Vertical Microprogramming. **Memory:** Basic concept and Hierarchy, RAM memories, 2D, 2 & 1/2D Memory Organization, ROM Memories, Cache Memories: Concept and Design issues performance, Address mapping and Replacement, Auxiliary memories: Magnetic disk, Magnetic tape and Optical disks, 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, Direct Memory Access, I/O channels and Processors, Standard communication interfaces.	**20**

Text Books:

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

Reference Books:

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

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

- CO1: Understand the organization of the modern computer system hardware.
- CO2: Analyze the performance of component, able to calculate the effective address of different operands, arithmetic operations of positive and negative numbers.
- CO3: Understand the Basic hardware architectural issues that affect the nature and performance of software.
- CO4: Categorize memory organization and explain the function of each element of a memory hierarchy.
- CO5: Identify and compare different methods for computer I/O mechanisms.

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.	14

Text Book:

- Aaron M. Tanenbaum, Yedidyah Langsam and Moshe J. Augenstein (2009), "*Data Structures Using C and C++*", 2nd Edition, PHI.

Reference Books:

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

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

- CO1: Understand the concepts of data structure and algorithms.
- CO2: Implement of stack using array and linked list and its applications.

- CO3: Implement of Linked list and solve the problem of polynomials addition.
- CO4: Implement of Queue using array and linked list.
- CO5: Implement hierarchical based solution using tree traversal algorithms.
- CO6: Understand the representation of Binary |Search Tree and its operations.
- CO7: Determine shortest path algorithm from source node to destination node using graph.
- CO8: Select an appropriate sorting algorithm for a given data set with respect to their time complexity.
- CO9: Index and retrieve items in a database using hash functions with minimum collision.

Head of the Department
Computer Engineering & Applications
Institute of Engineering & Technology
GLA University, Mathura

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, M Saravanan, and S Jeevananthan (2010), "*Microprocessors and Microcontrollers*", Oxford University Press India.

Reference Books:

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

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

- CO1: Demonstrate the Microprocessor internal architecture and its operations within the area of manufacturing and performance.
- CO2: Apply knowledge for programming proficiency using addressing modes and programming technique with instruction set.
- CO3: Apply knowledge to calculate execution time of program or parts of program, and to design or to modify software with counters and time delays.
- CO4: Compare accepted standards and guidelines to select microprocessor (8085 & 8086) to meet performance requirements & understand the concept of Parallelism.
- CO5: Analyze the concept of h/w & s/w in order to interface the processor to external device with I/O programming & Interrupt Driven I/O.
- CO6: Develop microprocessor-based model by interfacing chips (8237, 8253/54, 8255 & 8259).

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. (2004), "Data Communication and Networking", 4th Edition, McGrawHill.

References:

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

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

- CO1: Describe the functions of each layer in OSI and TCP/IP model.
- CO2: Describe the functions of data link layer and explain the protocols.
- CO3: Explain the types of transmission media with real time applications.
- CO4: Classify the routing protocols and analyze how to assign the IP addresses for the given network.
- CO5: Explain the functions of Application layer and Presentation layer paradigms and Protocols.
- CO6: Describe the Session layer design issues and Transport layer services.
- CO7: Understand IPv6 addressing and differentiate it from IPv4.

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 (2010), "Software Engineering: A Practitioners Approach", 7thEdition, McGraw Hill.

Reference Books:

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

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

- CO1: Understand application of software Processes and apply software processes for the construction of SRS using requirement engineering.
- CO2: Estimate the cost, effort and schedule of software using COCOMO Model.
- CO3: Analyze and apply design techniques (structure chart, SDM, sequence diagram) in designing software.
- CO4: Analyze software using software metrics and understand coding process.
- CO5: Apply different testing techniques and tools necessary for software testing.
- Co6: Apply different maintenance models for maintenance of software and understand the concept of software Quality.

BCSC0010: 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, Proof Techniques, 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. **Probability Theory**: Introduction to Probability Theory, Conditional Probability, Total Probability, Bayes' Theorem.	20

II	**Propositional Logic -** Logical Connectives, Truth Tables, Normal Forms (Conjunctive and Disjunctive), Validity;
	Predicate Logic - Quantifiers, Inference Theory, Methods of Proof: Direct, Indirect, Mathematical Induction.
	Algebra: Motivation of Algebraic Structures, Finite Groups, Subgroups and Group Homomorphism; Lagrange's Theorem; Commutative Rings and Elementary Properties;
	Graph Theory: Introduction to Graphs, Types: Planner, Directed, Complete, Bipartite Graph, Isomorphism , Euler Graph, Hamiltonian Graph, Operations on Graphs, Representation of graphs, Connectivity.
		20

Text Book:

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

Reference Books:

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

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

- CO1: Understand modeling of computer science problems in the discrete structures like sets, relations; their basic properties and able to perform operations on them.
- CO2: Solve a specific class of problems using recurrence relations, the mathematical analysis of the time and memory required to perform the algorithm using counting techniques.
- CO3: Apply combinatorial analysis to identify objects in a system.
- CO4: Apply mathematical reasoning to read, comprehends, and constructs mathematical arguments as basic tool for natural language understanding.
- CO5: Understand and apply various proof techniques for determining validity of arguments.
- CO6: Understand the abstract mathematical structures such as sets, relations, graphs and trees to represent discrete objects and the relationship between them.

BCSC0011: THEORY OF AUTOMATA & FORMAL LANGUAGES

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:3-1-0**

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 (2006), "Theory of Computer Science: Automata, Languages and Computation", 3rd Edition, PHI.

Reference Books:

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

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

- CO1: Understand, design, construct, analyze and interpret Regular languages, Expression and Grammars.
- CO2: Design different types of Finite Automata and Machines as Acceptor, Verifier and Translator.
- CO3: Understand, design, analyze and interpret Context Free languages, Expression and Grammars.
- CO4: Design different types of Push down Automata as Simple Parser.
- CO5: Design different types of Turing Machines as Acceptor, Verifier, Translator and Basic computing machine.
- CO6: Compare, understand and analyze different languages, grammars, Automata and Machines and appreciate their power and convert Automata to Programs and Functions

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(2008), Introduction to Algorithms, Third edition, Prentice Hall of India.

Reference Books:

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

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

- CO1: Analyze worst-case running times of algorithms using asymptotic analysis.
- CO2: Apply the sorting algorithms to solve real life applications.
- CO3: Compare between different data structures. Pick an appropriate data structure for a design situation.
- CO4: Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it.
- CO5: Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it.
- CO6: Describe the greedy paradigm and explain when an algorithmic design situation calls for it.
- CO7: Explain the major graph algorithms and their analyses.

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: Solve problems with smaller Lines of Code using Python.
CO2: Apply OO concepts while programming in Python.
CO3: Apply in-built packages defined in Python.
CO4: Apply front-end as Python Programming to connect with any back-end.

BCSC0800: 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: Write, compile and debug programs in C language.
- CO2: Use different data types in a computer program.
- CO3: Design programs involving decision structures, loops and functions.
- CO4: Understand the concepts of functions, recursion, pointers and file handling.
- CO5: Write, compile and debug programs in C language.
- CO6: Use different data types in a computer program.
- CO7: Design programs involving structures, union and functions.

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, Executemany 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.

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 (1998), *"Database Concepts"*, 5th Edition, TMH.
- Majumdar& Bhattacharya, *"Database Management System"*, TMH

Outcome: After the completion of the course, the student will be able to:
CO1: Ability to create database tables
CO2: Ability to formulate SQL queries based on the problems given
CO3: Ability to apply PL/SQL.
CO4: Ability to create NoSQL databases.
CO5: Ability to connect database using Python program.

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 (2009),” Operating *Systems”*, 6th Edition, Pearson Education.
- Harvey M Dietel (2002), “*An Introduction to Operating System*”, 2nd Edition, Pearson Education.
- D M Dhamdhere (2006), “*Operating Systems: A Concept Based Approach*”, 2nd Edition.
- M. J. Bach. (1986), “*Design of the Unix Operating System*”, PHI.

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

- CO1: Implement the various operations on UNIX operating systems.
- CO2: Understand the working of systems calls.
- CO3: Understand and solve message passing in Unix operating system.

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 (2008), "*Computer Organization and Design*", 4thEdition, Elsevier Publication.
- William Stalling (2011), "*Computer Organization*",8th Edition, PHI
- M. Mano (1996), "*Computer System Architecture*", 3rd Edition, PHI.

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

- CO1: Design any sequential circuit for logical operations.
- CO2: Design any combinational circuit for logical operations.

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		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: Identify, implement and use the appropriate data structures for a given problem
- CO2: Apply algorithmic skills for computing and engineering practice.
- CO3:Apply design and development principles of data structures and algorithms in the construction of software systems.

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 (2000), “*Microprocessor Architecture Programming and Applications with 8085*”, 4th Edition, Penram International Publishing.
- D. V. Hall (1992), “*Microprocessors and Interfacing: Programming and Hardware*”, 2nd Edition, TMH.

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

- CO1: Understand the concepts of advanced microprocessors.
- CO2: Understand various interfacing circuits necessary for various applications.

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: Identify, implement and use the appropriate data structures for a given problem
- CO2: Apply algorithmic skills for computing and engineering practice.
- CO3: Apply design and development principles of data structures and algorithms in the construction of software systems.

Head of the Department
Computer Engineering & Applications
Institute of Engineering & Technology
GLA University, Mathura

SYLLABUS

OF

PROGRAM ELECTIVE

BOUQUET: COMPUTER NETWORK & SECURITY

DEPARTMENT OF COMPUTER ENGINEERING & APPLICATIONS

Under

Choice Based Credit System (CBCS)

Program Elective

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE- REQUISITES
			L	T	P	J			
Bouquet 1:Computer Network & Security									
Theory									
1.	BCSE0001	Network Programming and Management	3	0	0	0	3	3	Computer Networks
2.	BCSE0002	Principles of Mobile Computing	3	1	0	0	4	4	Computer Networks
3.	BCSE0003	Ad Hoc Networks	3	0	0	0	3	3	Computer Networks
4.	BCSE0004	Cryptography & Network Security	3	0	0	0	3	3	Computer Networks
5.	BCSE0005	Cyber security and Digital Forensics	3	0	0	0	3	3	
6.	BCSE0006	Information Coding Techniques	3	0	0	0	3	3	Computer Networks
Labs									
1.	BCSE0070	Network Programming and Management Lab	0	0	2	0	1	2	Computer Networks
2.	BCSE0071	Cryptography & Network Security Lab	0	0	2	0	1	2	Computer Networks
3.	BCSE0072	Cyber security and Digital Forensics Lab	0	0	2	0	1	2	
4.	BCSE0073	Information Coding Techniques Lab	0	0	2	0	1	2	Computer Networks
Projects									
1.	BCSE0081	Cyber security and Digital Forensics Project	0	0	0	-	2	0	

BCSE0001: NETWORK PROGRAMMING AND MANAGEMENT

Objective: To learn the basics of socket programming using TCP Sockets and UDP sockets. To develop knowledge of threads for developing high performance scalable applications. To learn about raw sockets.To understand simple network management protocols & practical issues

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

Module No.	Content	Teaching
I	**Introduction to Socket Programming-**Overview of TCP/IP Protocols, Introduction to Sockets, Socket address Structures, Byte ordering functions, address conversion functions, Elementary TCP Sockets, socket, connect, bind, listen, accept, read, write, close functions, Iterative Server, Concurrent Server. **Application development:** TCP Echo Server, TCP Echo Client, Posix Signal handling, Server with multiple clients, boundary conditions: Server process Crashes, Server host Crashes, Server Crashes and reboots, Server Shutdown, I/O multiplexing, I/O Models, select function, shutdown function TCP echo Server (with multiplexing) - poll function - TCP echo Client (with Multiplexing).	20
II	**Socket options:** Socket options, get socket and set socket functions - generic socket options - IP socket options -ICMP socket options - TCP socket options. **Elementary UDP sockets:** Elementary UDP sockets - UDP echo Server - UDP echo Client - Multiplexing TCP and UDP sockets - Domain name system - gethostbyname function - Ipv6 support in DNS - gethostbyadr function - getservbyname and getservbyport functions. **Advanced sockets**: Ipv4 and Ipv6 interoperability - threaded servers - thread creation andtermination - TCP echoserver using threads - Mutexes - condition variables - raw sockets - raw socket creation - raw socket output - raw socket input - ping program - trace route program.	20

Text Books:

- Unix Network Programming Volume 1: The Sockets Networking API by W. Richard Stevens, Bill Fenner and Andrew M. Rudoff, Pearson Education

Reference Books:

- The Definitive Guide to Linux Network Programming by Keir Davis, John Turner and Nathan Yocom,
- TCP/IP Sockets in C: Practical Guide for Programmers by Michael J. Donahoo and Kenneth L. Calvert, Morgan Kaufmann

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

- CO1: Understand Socket Programming.
- CO2: Know about the UDP Sockets.
- CO3: Understand the concepts of threads.
- CO4: Understand about raw sockets.
- CO5: Understand the concepts simple network management protocols.

BCSE0070: NETWORK PROGRAMMING AND MANAGEMENT LAB

Objective: To understand the use of client/server architecture in application development, to understand and use elementary socket system calls, advanced socket system calls and how to use TCP and UDP based sockets to implement network routing algorithms and application layer protocols.

Credits: 01 **L–T–P-J: 0–0–2-0**

Module No.	Content	Teaching Hours
I & II	• Understanding and using of commands like ifconfig, netstat, ping, arp, telnet, ftp, finger, traceroute, whoisetc. • Usage of elementary socket system calls (socket (), bind (), listen (), accept (), connect (), send (), recv (), send to (), recvfrom ()). • Implementation of Connection oriented concurrent service (TCP). • Implementation of Connectionless Iterative time service (UDP). • Implementation of Select system call. • Implementation of gesockopt (), setsockopt () system calls. • Implementation of getpeername () system call. • Implementation of remote command execution using socket system calls. • Implementation of Distance Vector Routing Algorithm. • Implementation of SMTP. • Implementation of FTP. • Implementation of HTTP.	24

Reference Books:

- W. Richard Stevens, “Unix Network Programming”, Prentice Hall, Pearson Education, 2009.
- Douglas E. Comer, “Hands-on Networking with Internet Technologies”, Pearson Education.

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

- CO1: Use network programming concepts to develop and implement distributed applications.
- CO2: Develop and implement next generation protocols required for emerging applications.
- CO3: Model and evaluate performance of networking systems.

BCSE0002: PRINCIPLES OF MOBILE COMPUTING

Objective: To learn the cellular concepts and to know about the radio wave propagation along with various wireless techniques.

Credits: 04 **L–T–P-J: 3–1–0-0**

Module No.	Content	Hours
I	**Introduction to Mobile Communications and Computing:** Introduction to Mobile Computing, novel applications, limitations, and architecture - GSM Mobile services, System architecture, Radio interface, Protocols, Localization and calling, Handover, Security, and New data services. **Introduction to wireless communication**: Evolution of mobile communications, mobile radio systems- Examples, trends in cellular radio and personal communications. Cellular Concept: Frequency reuse, channel assignment, hand off, Interference and system capacity, tracking and grade of service, Improving Coverage and capacity in Cellular systems. **Wireless radio propagation:** Free space propagation model, reflection, diffraction, scattering, link budget design, Outdoor Propagation models, Indoor propagation models.	20
II	**Wireless radio propagation:** Small scale Multipath propagation, Impulse model, Small scale Multipath measurements, parameters of Mobile multipath channels, types of small scale fading, statistical models for multipath fading channels., Interference, DSSS, FHSS. **Medium access control:** (Wireless) Medium Access Control: Motivation for a specialized MAC (Hidden and exposed terminals, Near and far terminals), SDMA, FDMA, TDMA, CDMA. **Wireless LAN standards:** Wireless LAN, IEEE 802.11, Architecture, services MAC Physical layer, IEEE 802.11a, 802.11b standards, HIPERLAN, Bluetooth.	20

Text Books:

- Mobile Communications, 2nd Edition by Jochen Schiller, Pearson Education
- Handbook of Wireless Networks and Mobile Computing Edited by Ivan Stojmenović, John Wiley & Sons, Inc.

Reference Books:

- Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML by Reza B'Far, Cambridge University Press
- Fundamentals of Mobile and Pervasive Computing by Frank Adelstein, Sandeep KS Gupta, Golden Richard III and Loren Schwiebert, McGraw-Hill Professional
- 802.11 Wireless Networks: The Definitive Guide, 2nd Edition by Matthew Gast, O'Reilly Media

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

- To be familiar with cellular concepts
- To be familiar with radio wave propagation concept
- To be aware of wireless techniques

BCSE0003: AD HOC NETWORKS

Objective: This course is offered for those who are interested in understanding and building systems support mechanisms for mobile computing systems including client-server web/database/file systems, and mobile ad hoc and sensor networks for achieving the goal of anytime, anywhere computing in wireless mobile environments.

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

Module No.	Content	Teaching Hours
I	**Introduction to Ad Hoc Wireless Networks:** Characteristics of MANETS, Applications of MANETS, Challenges. **Routing in MANETS:** Topology based versus position-based approaches, Topology based routing protocols, and position-based routing, other routing protocols. **Data Transmission in MANETS:** The broadcast storm, Multicasting, Geocasting. **TCP Over Ad Hoc Networks:** TCP protocol overview, TCP and MANETS, Solutions for TCP over Ad Hoc networks. **Basics of Wireless Sensors and Applications:** The Mica Mote, Sensing and Communication Range, Design Issues, Energy Consumption, Clustering of Sensors, Applications.	20
II	**Data Retrieval in Sensor Networks:** Classification of WSNs, MAC Layer, Routing Layer, High Level Application layer Support, Adapting to the Inherent Dynamic Nature of WSNs. **Introduction:** Basic principles and challenges, past and ongoing VANET activities. Cooperative Vehicular Safety Applications Enabling technologies, cooperative system architecture, safety applications. Vehicular Mobility Modeling Random models. MAC Layer of Vehicular Communication Networks Proposed MAC approaches and standards, IEEE 802.11p. **VANET Routing protocols:** Opportunistic packet forwarding, topology-based routing, geographic routing. Standards and Regulations Protocol Stack, DSRC regulations and standards.	20

Text Books:

- Ad Hoc and Sensor Networks: Theory and Applications, Carlos de Morais Cordeiro and Dharma Prakash Agrawal, World Scientific Publications / Cambridge University Press,2006.
- Wireless Sensor Networks: An Information Processing Approach, Feng Zhao, Leonidas Guibas, Elsevier Science Imprint, Morgan Kauffman Publishers, 2005

Reference Books:

- Ad Hoc Wireless Networks: Architectures and Protocols, C. Siva Ram Murthy and B. S. Manoj, Pearson Education, 2004.
- Guide to Wireless Sensor Networks, Sudip Misra, Isaac Woungang, and Subhas Chandra Misra, Springer International Edition, 2012.
- Wireless Mesh Networking, Thomas Krag and Sebastin Buettrich, O'Reilly Publishers,2007.
- Wireless Sensor Networks – Principles and Practice, Fei Hu, Xiaojun Cao, An Auerbach book, CRC Press, 2010.
- Wireless Ad hoc Mobile Wireless Networks-Principles, Protocols and Applications, Subir Kumar Sarkar, et al., Auerbach Publications, Taylor & Francis Group, 2008.
- Wireless Ad hoc Networking, Shih-Lin Wu, Yu-Chee Tseng, Auerbach Publications, 2007
- Wireless Ad hoc and Sensor Networks–Protocols, Performance and Control, JagannathanSarangapani, CRC Press, 2007.

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

- CO1: Understand the concept of mobile ad hoc networks, design and implementation issues and available solutions.

- CO2: Demonstrate the routing mechanisms and three classes of approaches: proactive, on-demand, and hybrid.
- CO3: Explain sensor networks and their characteristics. This includes design of MAC layer protocols, understanding of power management, query processing, and sensor databases.

BCSE0004: CRYPTOGRAPHY & NETWORK SECURITY

Objective: *This Course focuses towards the introduction of network security using various cryptographic algorithms and understanding network security applications and practical applications that have been implemented and are in use to provide email and web security*.

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

Module No.	Content	Teaching Hours
I	**Introduction:** Introduction to Security Attacks, Services and Mechanism, Classical Encryption Techniques-Substitution Ciphers and Transposition Ciphers, Steganography, Stream and Block Ciphers, Cryptanalysis. **Symmetric Key Cryptosystems:** Block Cipher Principles, Shannon's Theory of Confusion and Diffusion, Data Encryption Standard(DES), Strength of DES, Triple DES, Advance Encryption Standard (AES), Linear and Differential Cryptanalysis, Block Ciphers Modes of Operation. **Introduction to Number Theory:** Modular Arithmetic, Prime and Relative Prime Numbers, Primitive Roots, Fermat's and Euler's Theorem, Extended Euclidean Algorithm, Chinese Remainder Theorem. **Algebraic Structures:** Introduction to Group, Ring & Field of the Form GF(P). Asymmetric Key Cryptosystems: RSA Cryptosystem, Attacks on RSA, Security of RSA, Discrete Logarithm Problem, Elgamal Encryption Algorithm. **Hash Functions and Macs:** Authentication Functions, Message Authentication Code, Hash Functions, Birthday Attacks, Security of Hash Functions, Secure Hash Algorithm (SHA-512).	20
II	**Digital Signature:** Digital Signatures, RSA Digital Signature Scheme, Elgamal Digital Signature Techniques, Digital Signature Standards (DSS). **Key Management:** Symmetric Key Distribution, Diffie Hellman Key Exchange Algorithm. **Public Key Distribution:** X.509 Certificates, Public Key Infrastructure. **Authentication Applications and E-Mail Security:** Kerberos, Pretty Good Privacy (PGP), S/MIME. **IP Security and Web Security:** IP Sec Architecture, Authentication Header, Encapsulating Security Payloads, Combining Security Associations, Key Management, Introduction to Secure Socket Layer, Transport Layer Security, Secure Electronic Transaction (SET). **System Security:** Introductory Idea of Intrusion, Intrusion Detection, Malicious Programs, Firewalls.	22

Text Books:

- W. Stallings (2010), "Cryptography and Network Security: Principles and Practices",5th Edition, Pearson Education.

Reference Books:

- B. A. Forouzan (2003), "Cryptography & Network Security", 3rd Edition, Tata McGraw Hill.
- Wenbo Mao (2003), "Modern Cryptography: Theory and Practice", Prentice Hall.
- Douglas Stinson, "Cryptography Theory and Practice", 2nd Edition, Chapman& Hall/CRC.

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

- CO1: Understands various encryptions and decryption techniques.
- CO2: Learn protocols used in Web Security and Transport layer Security.
- CO3: Define the terms vulnerability, threat and attack and classify different attacks.
- CO4: Discuss network security designs using available secure solutions.

BCSE0071: CRYPTOGRAPHY & NETWORK SECURITY LAB

Objective: The objective of this lab is that to understand the principles of encryption algorithms, conventional and public key cryptography practically with real time applications.

Credits: 01 **L–T–P-J: 0–0–2-0**

Module No.	Content	Hours
I & II	• Write a program in ‘C’ to implement Additive and VignereCipher. • Write a program in ‘C’ to implement Autokey Cipher. • Write a program in ‘C’ to find out the Multiplicative inverse of a given number. • Write a program in ‘C’ to implement RSA Cryptosystem. • Write a program in ‘C’ to implement Elgamal Cryptosystem. • Write a program in ‘C’ to implement Rabin Miller Primality Test. • Write a program in ‘C’ to find out the Primitive roots. • Write a program in ‘C’ to implement Euclidean Algorithm. • Write a program in ‘C’ to implement Extended Euclidean Algorithm. • Write a program in ‘C’ to implement Diffie-Hellman key exchange Algorithm. • Write a program in ‘C’ to implement Random Number Generator.	24

Textbooks:

- W. Stallings (2010), "Cryptography and Network Security: Principles and Practices",5th Edition, Pearson Education.

Reference Books:

- B. A. Forouzan (2003), "Cryptography & Network Security", 3rd Edition, Tata McGraw Hill.
- Douglas Stinson, "Cryptography Theory and Practice", 2nd Edition, Chapman& Hall/CRC.
- K. N. King, “C Programming a Modern Approach”, W. W. Norton, 2nd Edition, 2008.
- Kernighan and Ritche, “The C Programming Language”, PHI, 2nd Edition, 2011.
- P. DeyandM.Ghosh, “ProgramminginC”, OxfordUniversityPress1stEdition,2000.

Outcome: After completion of this course students will be able to:

- CO1: Know the methods of conventional encryption.
- CO2: Understand the concepts of public key encryption and number theory
- CO3: Understand various applications of cryptography and security issues practically.

BCSE0005: CYBER SECURITY AND DIGITAL FORENSICS

Objective: To give knowledge of constitutional and case law to search and capture digital evidence, determine the most effective and appropriate forensic response strategies to digital evidence, and provide effective proof in a case involving digital evidence.

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

Module No.	Content	Teaching Hours
I	**Introduction:** Introduction and Overview of Cyber Crime, Nature and Scope of Cyber Crime, Types of Cyber Crime: Social Engineering, Categories of Cyber Crime, Property Cyber Crime. **Cyber Security issues:** Unauthorized Access to Computers, Computer Intrusions, White Collar Crimes, Viruses and Malicious Code, Security Engineering, Network Security, Information Security, Web Security, Database Security, Malware Security, Biometric Security, Security in Cloud Computing and Mobile Computing. Software Piracy, Intellectual Property, Digital laws and legislation, Law Enforcement Roles and Responses. **Investigation** Introduction to Cyber Crime Investigation - Investigation Tools Discovery - Digital Evidence Collection, Evidence Preservation, E-Mail Investigation	24
II	Tracking - IP Tracking, E-Mail Recovery, Hands on Case Studies, Encryption and Decryption Methods, Search and Seizure of Computers, Recovering Deleted Evidences, Password Cracking. **Digital forensics** Introduction to Digital Forensics, Forensic Software and Hardware, Analysis and Advanced Tools, Forensic Technology and Practices, Forensic Ballistics and Photography, Face, Iris and Fingerprint Recognition, Audio Video Analysis, Windows System Forensics, Linux System Forensics, Network Forensics. **Laws and acts** Laws and Ethics, Digital Evidence Controls, Evidence Handling Procedures, Basics of Indian Evidence ACT IPC and CrPC, Electronic Communication Privacy ACT, Legal Policies.	24

Text Books:

- Guide to Computer Forensics and Investigations 6th Edition by Bill Nelson, Amelia Phillips and Christopher Steuart, Cengage Publication
- Incident Response & Computer Forensics, Second edition by Chris Prosise and Kevin Mandia, McGraw-Hill Education

Reference Books:

- Computer Forensics and Digital Investigation with EnCase Forensic v7 1st Edition by Suzanne Widup, McGraw-Hill Education
- Forensic Computer Crime Investigation by Thomas A. Johnson, CRC Press
- Software Forensics: Collecting Evidence from the Scene of a Digital Crime 1st Edition by Robert Slade, McGraw-Hill Education

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

- To have various ideas about cybercrime.
- To have knowledge of the various issues of cybercrime.
- To investigate and find the cybercrime.
- To identify the cybercrime.
- To have clear idea of the various laws and acts.

BCSE0072: CYBER SECURITY AND DIGITAL FORENSICS LAB

Objective:

Credits: 01 **L–T–P-J: 0–0–2-0**

Module No.	Content	Hours
I & II	• Working on footprinting and Reconnaissance of a network and server. • Enumeration for network resources and enumerating local machines. • Scanning Networks and finding vulnerabilities and exploit cases. • Working on System Hacking and bypassing first line of security (passwords /Encryption) • Password hacking techniques like Brute force and Rainbow Attack. • Working with viruses and malware. • Packet Sniffing and network traffic analysis using wireshark. • Manipulating person mindset using Social Engineering and its detection techniques. • Attacking web applications and servers using various Denial-of-Service attacks. • Vulnerability Assessment of web applications using Nikto / Nessus/WPScanner • Vulnerability Assessment of wireless networks using Fluxion / Aircrack-ng	24

Textbooks:

"Gray Hat Hacking the Ethical Hackers Handbook, 3rd Edition" by Allen Harper and Shon Harris

Reference Books:

- "The Unrevealed Secrets of Hacking and Cracking – Hack Before You Get Cracked" by Prateek Shukla and Navneet Mehra
- "How to Unblock Everything on the Internet" by Ankit Fadia
- "The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, 2ed" by Dafydd Stuttard and Marcus Pinto

Outcome: After completion of this course students will be able to:

- To have knowledge of the various issues of cybercrime.
- To investigate and find the cybercrime.
- To identify the cybercrime.

BCSE0006: INFORMATION CODING TECHNIQUES

Objective: To introduce the fundamental concepts of information theory: data compaction, data compression, data transmission, error detection and correction.

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

Module No.	Content	Teaching Hours
I	**Information entropy fundamentals** Uncertainty, Information and Entropy - Source Coding Theorem - Huffman coding –Shannon Fano coding - Discrete Memory less channels - channel capacity - channel coding Theorem - Channel Capacity Theorem. **Data and voice coding** Differential Pulse Code Modulation - Adaptive Differential Pulse Code Modulation - Adaptive sub band coding - Delta Modulation - Adaptive Delta Modulation - Coding of speech signal at low bit rates (Vocoders, LPC). **Error control coding** Linear Block codes - Syndrome Decoding - Minimum distance consideration - cyclic codes - Generator Polynomial - Parity check polynomial - Encoder for cyclic codes - calculation of syndrome - Convolutional codes.	20
II	**Compression techniques - text** Principles - Text compression - Static Huffman Coding - Dynamic Huffman coding - Arithmetic coding. **Compression techniques - multimedia** Image Compression - Graphics Interchange format - Tagged Image File Format - Digitized documents - Introduction to JPEG standards. **Audio and video coding** Linear Predictive coding - code excited LPC - Perceptual coding, MPEG audio coders – Dolby audio coders - Video compression - Principles - Introduction to H.261 & MPEG Video standards.	20

Text Books:

- Communication Systems, 5th Edition by Simon Haykin and Michael Moher, Willey
- Multimedia Communications: Applications, Networks, Protocols and Standards by Fred Haskell, Pearson Education India

Reference Books:

- Multimedia Communications: Directions and Innovations, 1st Edition, by Jerry Gibson, Academic Press
- Information Theory and Network Coding by Raymond W. Yeung, Springer
- Fundamentals of Information Theory and Coding Design by Roberto Togneri and Christopher J.S deSilva, Chapman and Hall/CRC

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

CO1: Understand about information entropy

CO2: Know the fundamentals of pulse code and delta modulation technique.

CO3: Know the methods for framing the code word.

CO4: Know the fundamentals of data & voice coding

CO5: Know the fundamentals of video coding.

BCSE0073: INFORMATION CODING TECHNIQUES LAB

Objective: To understand and implement the fundamental concepts of information theory: data compaction, data compression, data transmission, error detection and correction

Credits: 01 **L–T–P-J: 0–0–2-0**

Module No.	Content	Hours
I & II	• Write a program for determination of various entropies and mutual information of a given channel. Test various types of channel such as a) Noise free channel. b) Error free channel c) Binary symmetric channel d) Noisy channel Compare channel capacity of above channels. • Write a program for generation and evaluation of variable length source coding (Any two) a) Shannon Fanocoding and decoding b) Huffman Coding and decoding c) Lempel Ziv Coding and decoding • Write a Program for coding & decoding of Linear block codes. • Write a Program for coding & decoding of Cyclic codes. • Write a program for coding and decoding of convolutional codes. • Write a program for coding and decoding of BCH and RS codes. • Write a program to study performance of a coded and uncoded communication system (Calculate the error probability). • Write a simulation program to implement source coding and channel coding for transmitting a text file. • Implementation of any compression algorithm for either audio, image or video data.	24

Reference Books:

- Ranjan Bose, “Information Theory coding and Cryptography”, McGraw-Hill Publication, 2ndEdition
- C Moreira, P G Farrell, “Essentials of Error-Control Coding”, Wiley Student Edition
- BernadSklar, “Digital Communication Fundamentals & applications”, 2ndEd. Pearson Education.
- Shu lin and Daniel j, Cistellojr., “Error control Coding” Pearson, 2nd Edition.
- Todd Moon, “Error Correction Coding: Mathematical Methods and Algorithms”, Wiley Publication
- Khalid Sayood, “Introduction to data compression”, Morgan Kaufmann Publishers

Outcome: By the end of the class, students will learn to:
CO1: Know the methods for framing the code word.
CO2: Know the fundamentals of data & voice coding
CO3: Know the fundamentals of video coding.

BCSE0086: CYBERSECURITY AND DIGITAL FORENSICS PROJECT

Objective: To provide students with a comprehensive overview of collecting, investigating, preserving, and presenting evidence of cyber crime left in digital storage devices.

Credits: 02 **L–T–P-J: 0–0–0-0**

1. **Data Forensics:**
   Objective: Find and analyze hidden data from slack, swap and dead spaces on small devices (PDAs, phones, USB sticks, etc.)
2. **Honeypots:**
   Objective: Configure a honeypot and analyze the collected target data with and from honeypots.
3. **Malware analysis:**
   Objective: Analyze of malware using static and dynamic/behavioral methods and use it for malware detection, mitigation, the development of countermeasures
4. **Dynamic Binary Instrumentation**:
   Objective: Use of software testing measures such as code coverage, Function call hooking, control flow analysis, dynamic binary instrumentation helps ensure that an adequate slice of the program's set of possible behaviors has been observed.
5. **Steganalysis**:
   Objective: Perform Steganalysis on a JPEG Image Using Variation Techniques. Find and analyze hidden data from the image.
6. **Mobile Forensics**:
   Objective: Investigate into a Factory Reset of Android Device. Find and analyze hidden data from the device.
7. **Network Forensics**:
   Objective: Find different logs on the network and analyze to close the defined problem.
8. **Capturing volatile and non-volatile information**:
   Objective: Capture information from different files systems. Simple and extended attributes, erased data, file reassembly.

SYLLABUS

OF

PROGRAM ELECTIVE

BOUQUET: SOFTWARE ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING & APPLICATIONS

Under

Choice Based Credit System (CBCS)

Program Elective

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE- REQUISITES
			L	T	P	J			
Bouquet: Software Engineering									
THEORY									
1.	BCSE0051	Software Quality Engineering	3	0	0	0	3	3	Software Engineering
2.	BCSE0052	Service Oriented Architecture	3	0	0	0	3	3	Software Engineering
3.	BCSE0053	Agile Software Development	3	0	0	0	3	3	Software Engineering
4.	BCSE0054	Software Project Management	3	0	0	0	3	3	Software Engineering
5.	BCSE0055	Software Testing	3	0	0	0	3	3	Software Engineering
6.	BCSE0056	Compiler Design	3	1	0	0	4	4	TAFL

BCSE0051: SOFTWARE QUALITY ENGINEERING

Prerequisite: *Exposure to basic concepts of software engineering*

Credits: 03 L–T–P: 3–0–0

Module No.	Content	Teaching Hours
I	**Software Quality:** Software Quality Attributes and Specification, Total Quality Management, ISO 9126 Quality Standards; Defects, Faults, Failures, Defect Rate and Reliability, Defect Prevention, Reduction, and Containment, Overview of Different Types of Software Review. **Software Quality Metrics:** Product Quality Metrics- Defect Density, Customer Problems Metric, Customer Satisfaction Metrics, Function Points; In-Process Quality Metrics- Defect Arrival Pattern, Phase-Based Defect Removal Pattern, Defect Removal Effectiveness; Metrics for Software Maintenance- Backlog Management Index, Fix Response Time, Fix Quality, Software Quality Indicators. **Software Quality Assurance:** Quality Planning and Control, Quality Improvement Process, Evolution of Software Quality Assurance (SQA), Major SQA Activities/Issues, Zero Defect Software.	20
II	**Software Testing:** Functional Testing - Boundary Value Testing, Equivalence Class Testing, Decision Table Based Testing, Cause Effect Graphing Techniques; Structural Testing- Path Testing, Data Flow Testing. **Test Selection & Minimization for Regression Testing:** Regression Testing, Regression Test Process, Initial Smoke or Sanity Test, Selection of Regression Tests, Classifying Test Cases, Methodology for Selecting Test Cases, Resetting the Test Cases for Regression Testing; Introduction to Ad-Hoc Testing. **Testing Web Applications:** Web testing, Functional Testing, User Interface Testing, Usability Testing, Configuration and Compatibility Testing, Security Testing, Performance Testing, Database Testing, Post-Deployment Testing, Web Metrics; Introduction to Automated Test Data Generation.	19

Text Book:

- Stephen H. Kan (2000), “Metrics and Models in Software Quality Engineering”, 2nd Edition, Pearson Education.
- Yogesh Singh (2011), “Software Testing”, Cambridge University Press.

Reference Books:

- Jeff Tian (2005), “Software Quality Engineering (SQE)”, Wiley-Interscience.
- S. Desikan and G. Ramesh (2008), “Software Testing: Principles and Practices”, Pearson Education.
- Aditya P. Mathur (2011), “Fundamentals of Software Testing”, Pearson Education.
- NareshChauhan (2010), “Software Testing: Principles and Practices”, 1st Edition, Oxford University Press.
- Naik and Tripathy (2008), “Software Testing and Quality Assurance”, Wiley India.

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

CO1: Critically evaluate alternative standards, models and techniques aimed for quality software to achieve the satisfaction of the client as well as developer.

CO2: Propose the innovative solutions for the software quality assurance and measurement problems in the context of various software development environments.

CO3: Evaluate software quality assurance issues in software development and propose the quality framework for an organization.

CO4: Write the software test scenarios, test cases, test plans and various metrics of different applications.

CO5: Perform the testing of web based projects in team.

CO6: Apply the suitable methodology to perform regression testing in projects.

BCSE0052: SERVICE ORIENTED ARCHITECTURE

Prerequisite: *Exposure of distributed systems*

Credits: 03 **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**Introduction:** Introduction to Web 1.0, 2.0, Web Services (WS), Characteristics of Web Services, WS Modeling Web Service Activities, WS Management, WS Composition, Service Descriptions; Introduction to Service Oriented Architecture (SOA), Characteristics Of SOA, Principles of SOA, Comparison of Service and Object Orientation, SOA Interaction Cycle (SIC), Comparing SOA to Client Server and Distributed Internet Architectures, Components of SOA; Patterns Coordination, Atomic Transactions, Business Activities, Orchestration, Choreography, Service Layer Abstraction, Application Service Layer, Business Service Layer, Orchestration Service Layer. **Service Oriented Analysis:** Business Centric SOA, Deriving Business Services, Service Oriented Design, WSDL Basics, SOAP Basics, Messaging with SOAP, Message Exchange.	20
II	**SOA Composition, Guidelines:** Entity Centric Business Service Design, Application Service Design, Task Centric Business Service Design, SOA Platforms - SOA Support in J2EE, Java API for XML Based Web Services (JAX,WS). **WS Integration concepts:** Concepts of Enterprise Service Bus (ESB), Web Services Interoperability Technologies (WSIT), SOA Support in .NET, Common Language Runtime, ASP.NET Web Forms, ASP.NET Web Services, Web Services Enhancements (WSE), Concepts of Business Process Execution Language (BPEL).	19

Text Book:

- Thomas Erl (2008), “SOA Principles of Service Design” The Prentice Hall Service Oriented Computing Series, Prentice Hall India.

Reference Books:

- Newcomer, Lomow (2005), “Understanding SOA with Web Services”, Pearson Education.
- Sandeep Chatterjee, James Webber (2005), “Developing Enterprise Web Services: An Architect’s Guide”, Pearson Education.
- Munindar P. Singh, Michael N. Huhns (2010), “Service-Oriented Computing: Semantics, Processes Agents”, Wiley.
- Dan Woods and Thomas Mattern (2006), “Enterprise SOA Designing IT for Business Innovation”, 1st Edition, O’Reilly.
-

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

CO1: Manage a modern medium scale software development project using SOA principles.
CO2: Suggest recommendations for designing services.
CO3: Appreciate the impact of SOA on IT infrastructure—in particular, how business processes and transactions are handled differently in a SOA environment as compared to traditional application architecture.

BCSE0053: AGILE SOFTWARE DEVELOPMENT

Prerequisite: *Exposure of Software Engineering Principles*

Credits: 03 L–T–P: 3–0–0

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 Agile tester **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:

- Ken Schawber & Mike Beedle, Agile Software Development with Scrum, Pearson, 2008
- Robert C. Martin, Agile Software Development, Principles, Patterns and Practices, Prentice Hall,2002
- Lisa Crispin & Janet Gregory, Agile Testing: 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: Recognize the significance of Agile Methodologies in software development.
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 more agile (where appropriate).
CO5: Understand the relationship between the customer and the development team in agile projects and the responsibilities of both communities.

BCSE0054: SOFTWARE PROJECT MANAGEMENT

Prerequisite: *Exposure of Software Engineering Principles*

Credits: 03 **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**Introduction to Project Management:** The characteristics of software projects, Objectives of project management: time, cost and quality, Basics of Project Management, Feasibility Study, Cost-benefit Analysis, Planning, Project Execution, Project and Product Life Cycles, Project Management Knowledge areas, Project Management Tools & Techniques. **Project Management & Planning:** System view of project management, Understanding organizations, stakeholder's management, project phases & project life cycles. Introduction to Agile software, why planning is necessary, Iterative steps for planning, Project Plan documentation methods, Software Requirement Specification. **Measurement and Control:** Measurements for project monitoring, what and when to measure, Plan versus Control, managing the plan, Deadline Effect. Reviews, feedback and reporting mechanisms, revisiting the plan. Project Scope Management: Scope Planning & Scope management plans, Function point calculation, Scope definitions & project scope statement, Work Breakdown Structure (WBS)	20
II	**Time Management:** Project time management, activities sequencing, network diagrams, activity recourse estimation, activity duration estimation, schedule development, Gantt Charts, Critical path method, Program evaluation & review technique (PERT) and CPM. **Project Cost management:** Basis principles of cost management, Cost estimating, type of cost estimate, cost estimate tools & techniques, COCOMO, Putnam/ SLIM model Estimating by Analogy, cost budgeting, cost control, earned value management, project portfolio management **Project Quality Management:** Quality Planning, quality Assurance, Quality control, Tool &techniques for quality control, Pareto Analysis, Six Sigma, CMM **Project Communication Management**: Communication Planning, Performance reporting, managing stakeholders, improving project communication **Project risk management**: Risk Management planning, common sources of risk, risk identification, risk register, qualitative risk analysis, using probability impact matrixes, risk monitoring & control.	19

Text Book:

- Bob Hughes and Mike Cotterell, Software Project Management, Tata McGraw Hill (2009).
- Daniel Galin, "Software Quality Assurance: from Theory to Implementation", Addison-Wesley, 2003.

Reference Books:

- Roger Pressman, A practitioner's Guide to Software Engineering, Tata McGraw Hill (2014)
- Andrew Stellman; Jennifer Greene, Applied Software Project Management, O'Reilly Media, Inc. (2005).
- Ramesh Gopalaswamy, "Managing and global Software Projects", Tata McGraw Hill Tenth Reprint, 2011.

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

CO1: Apply basic concepts related to software project planning, scope and feasibility.
CO2: Analyze various project estimation techniques, especially size estimation (FP), effort estimation (COCOMO models), schedule estimation (GANTT charts), and cost estimation.
CO3: Understand Time Management, Project Communication Management and Project Risk Management.

BCSE0055: SOFTWARE TESTING

Prerequisite: *Exposure to basic concepts of software engineering*

Credits: 03 **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**Testing Basics** Testing as an Engineering Activity – Role of Process in Software Quality – Testing as a Process – Basic Definitions – Software Testing Principles – The Tester's Role in A Software Development Organization – Origins of Defects – Defect Classes – The Defect Repository and Test Design – Defect Examples – Developer / Tester Support for Developing a Defect Repository. **Test Case Design** Introduction To Testing Design Strategies – The Smarter Tester – Test Case Design Strategies – Using Black Box Approach To Test Case Design – Random Testing – Equivalence Class Partitioning – Boundary Value Analysis – Other Black Box Test Design Approaches – Black Box Testing And Cots – Using White Box Approach To Test Design – Test Adequacy Criteria – Coverage And Control Flow Graphs – Covering Code Logic – Paths – Their Role In White Box Based Test Design – Additional White Box Test Design Approaches – Evaluating Test Adequacy Criteria. **Levels of Testing** The Need for Levels of Testing – Unit Test – Unit Test Planning – Designing The Unit Tests – The Class as A Testable Unit – The Test Harness – Running The Unit Tests and Recording Results – Integration Tests – Designing Integration Tests – Integration Test Planning – System Test – The Different Types – Regression Testing – Alpha, Beta and Acceptance Tests.	20
II	**Test Management** Basic Concepts – Testing, Debugging Goals, Policies – Test Planning – Test Plan Components – Test Plan Attachments – Locating Test Items – Reporting Test Results – The Role Of Three Groups In Test Planning And Policy Development – Process And The Engineering Disciplines – Introducing The Test Specialist – Skills Needed By A Test Specialist – Building A Testing Group. **System Testing** Integration Testing, System Testing, Interaction Testing, Performance Testing, Mutation Testing, Regression Testing, Error Seeding **Object Oriented Testing** Issues In Object Oriented Testing, Test Case Design By Object Oriented Software, Fault Based Testing, Test Cases And Class Hierarchy, Scenario Based Test Design, Testing Surface Structure And Deep Structure, Class Testing: Random Testing For Object Oriented Classes, Partition Testing At The Class Level; Inter Class Test Case Design: Multiple Class Testing, Tests Derived From Behavior Models, Test Case Generation Using Uml Diagrams, Gui Testing, Object Oriented System Testing.	19

Text Book:

- SrinivasanDesikan, Gopalaswamy Ramesh, "Software Testing: Principles andPractices", Pearson 2012
- C. J. Paul, Software testing: A craftsmen's approach, CRC Press, 2nd Ed, 2002.
- R. Gopalswamy, Software testing, Pearson, 2005.

Reference Books:

- Aditya P. Mathur, "Foundations of Software Testing", Pearson, 2008
- Paul Ammann, Jeff Offutt, "Introduction to Software Testing", CambridgeUniversity Press, 2008
- G. J. Myers, the art of software testing, Wiley Interscience New York, 2005.
- R. S. Pressman, Software Engineering a Practitioner's approach, McGraw Hill, 4th Ed, 1982.

- R. Mall, Fundamentals of Software Engineering, Prentice Hall of India, 2nd Ed, 2003.

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

CO1: Create test cases for different levels and types of software.
CO2: Develop the basic skills in software testing by implementing various strategies of software testing in their project.
CO3: Control and monitor testing activity.

Head of the Department
Computer Engineering & Applications
Institute of Engineering & Technology
GLA University, Mathura

BCSE0056: 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 (2008), "Compilers: Principles, Techniques and Tools", 2nd Edition, Pearson Education.*

Reference Books:

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

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.

SYLLABUS

OF

PROGRAM ELECTIVE

BOUQUET: IMAGE PROCESSING AND INTELLIGENT SYSTEM

DEPARTMENT OF COMPUTER ENGINEERING & APPLICATIONS

Under

Choice Based Credit System (CBCS)

Program Elective

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE- REQUISITES
			L	T	P	J			
Bouquet: Image Processing and Intelligent System									
Theory									
1.	BCSE0101	Digital Image Processing	3	0	0	0	3	3	Mathematics, Programming
2.	BCSE0102	Computer Graphics and Multimedia	3	1	0	0	4	4	Mathematics, Programming
3.	BCSE0103	Soft Computing	3	0	0	0	3	3	Discrete Mathematics
4.	BCSE0104	Artificial Intelligence	3	0	0	0	3	3	Data Structures
5.	BCSE0105	Machine Learning	3	0	0	0	3	3	Mathematics, Programming
6.	BCSE0106	Machine Learning and Its Applications	3	0	0	0	3	3	Mathematics, Programming
Labs									
1.	BCSE0131	Digital Image Processing Lab	0	0	2	0	1	2	Programming
2.	BCSE0132	Soft Computing Lab	0	0	2	0	1	2	Programming
4.	BCSE0133	Machine Learning Lab	0	0	2	0	1	2	Programming
Projects									
1.	BCSE0141	Machine Learning Project	0	0	0	-	2	-	Programming

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: 3–0–0**

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 (2011), “Digital Image Processing”, Prentice Hall, 3rd Edition,

Reference Books:

- BhabatoshChanda and D. DuttaMajumder (2011), “Digital Image Processing and Analysis”, PHI.
- S. Sridhar (2011), “Digital Image Processing”, Oxford University Press.

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: Represent and describe an object of an image using Shape Number and Boundary descriptors.

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: 0–0–2**

Module No.	Content	Lab Hours
I/II/III	• 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, Sobel and Canny Operator) • Minor Application Assignment.	12*2=24

Reference Books:

- R.C. Gonzalez and R.E. Woods (2010), "Digital Image Processing Using MATLAB", PHI, 2ndEdition,
- 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.

BCSE0102: COMPUTER GRAPHICS & MULTIMEDIA

Objective:

Credits: 04 **L–T–P: 3–1–0**

Module No.	Content	Teaching Hours
I	**Line clipping algorithms**: Cohen Sutherland line clipping algorithm, Liang Barsky Algorithms. Polygon clipping algorithms: Sutherland Hodgeman, Weiler Atherton algorithms. **2DTransformation:**Translation, Rotation, Scaling, Mirror Images, transformations about an arbitrary point. **3D- Transformation:**Translation, Rotation, Scaling, Rotation about an arbitrary axis, reflection about arbitrary planes. **Projection:** parallel and perspective projections; orthographic Projections, Axonometric, oblique, Multiple Views, Isometric Projection, Perspective Projections (one, two and three vanishing points). **Quadric surfaces**: spline representation, spline specification. Bezier curves, B-splines.	20
II	**Multimedia:** Multimedia Architecture, Multimedia File formats. **Compression** – lossy and lossless compression, **Text compression-** Run length encoding (RLE), Shannon- Fano Algorithm, Huffman Algorithm. Dictionary Methods-LZW Algorithms. **Image Compression-** JPEG, MPEG. **Audio Compression**; WAVE, MPEG-1/2 Audio Layers. MIDI-File format, MIDI and digital audio. **Illumination and Color Models:**basic illumination models- Ambient light, Diffuse and Specular, Shading- Flat, Gouraud, Phong. C**olor Models:** RGB color model, YIQ color model, CMY color models. **Graphics Standards:** GKS, PHIGS.	20

Text Books:

- D.Hearn and M.P.Baker, Computer Graphics, PHI, 1996.

Reference Books:

- W.K. Gilloi , Interactive Computer Graphics, PHI.
- D.F. Rogers, Procedural Elements for Computer Graphics, TMH .
- J.D. Foley and A.D. Van, Fundamentals of Interactive Computer Graphic
- Rogers and Adam, Mathematical Elements for Computer Graphics, TMH
- R.Steinmetz and K. Nahstedt, Multimedia: Computing, Communications and Applications PHI,PTR,1995
- Salomon, D., Motta, G. Handbook of Data Compression. (2010) Springer.

Outcome: At the end of the course, students will be able to understand:
CO1: Understanding and demonstration of basic concepts of 2D and 3D Geometrical Transformations.
CO2: Understanding the concept of vewing, viewport, window and curves and surfaces.
CO3: Implementation of clipping on different objects as line, point and polygon.
CO4: Identify the projections of a given object in 2D environment.
CO5: Understanding the concepts of Illumination and Shading.
CO6: Discuss and solve simple problems in the basic representation and handling of multimedia data (images, audio and animation).
CO7: Demonstration and implementation of compression techniques of Text, image, Audio, video.

BCSE0103: SOFT COMPUTING

Objective: *Students will get an insight of the intelligent computational approaches. Providing students, the mathematical background to carry out optimization.*

Credits: 03 **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I.	**Neural Networks:**Introduction to Soft Computing & Neural Computing, Fundamentals of Artificial Neural Network(ANN), Models of ANN, ANN models: Rosenblatt's Perception, McCulloch & Pitts Model, Single Layer Perceptron, Learning Methods in Perceptron, Linearly Separable Task and XOR Problem, Multi-Layer Perceptron, Back Propagation Learning Algorithm, Associative Memory: Hopfield network, Auto Associative Memory, Bidirectional Hetro-Associative Memory, ADALINE, MADALINE Network, Applications of Neural Network. **Fuzzy Logic:** Introduction to Fuzzy Sets & Crisp Sets, Fuzzy Membership and Fuzzy Operations, Properties of Fuzzy Sets- Linguistic Hedges, Fuzzy Logic – T-norms and other aggregation operators, Crisp Relations.	20
II.	**Fuzzy Logic:** Fuzzy Relations, Fuzzy System, Crisp Logic, Propositional Logic and its Laws, Inference in Propositional Logic, Fuzzy Logic, Inference in Fuzzy Logic (GMP and GMT), Fuzzy Rule Based System, Fuzzyfication&Defuzzification, Applications of Fuzzy Logic. **Genetic Algorithm(GA):**Introduction to GA, Search optimization Method, Evolutionary Algorithm Working Principle, Biological Background of GA, Working Principles of GA, Encoding(Binary, Value, Permutation, Tree), Operators of GA(Random Population, Reproduction or Selection), Crossover and Mutation, Basics of Genetic Algorithm with Examples, Introduction to Genetic programming	20

Text Books:

- S. Rajsekaran& G.A. VijayalakshmiPai(2003), "Neural Networks, Fuzzy Logic and Genetic Algorithm: Synthesis and Applications", 4th Edition, Prentice Hall of India.

Reference Books:

- Timothy J Ross (2016),"Fuzzy Logic with Engineering Applications", 3rd Edition, John Wiley and Sons.
- David E. Goldberg (2009),"Genetic Algorithm in Search Optimization and Machine Learning "Adission-Wesley.
- Karray (2009), "Soft Computing and Intelligent Systems Design: Theory, Tools and Applications", 1st Edition, Pearson Education.

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

CO1: Understand basics of Soft Computing including Artificial Neural Networks, Fuzzy Logic and Genetic Algorithms.

CO2: Demonstrate the ability to develop some familiarity with current research problems and research methods in Soft Computing by working on a research or design project.

CO3: Understanding about the fundamental theory and concepts of neural networks, neuro modeling, several neural networks paradigms and its applications.

CO4: Design and implement the concepts of knowledge using fuzzy inference systems and other machine intelligence applications.

CO5: Identify an evolutionary computing paradigm known as genetic algorithms and its applications to engineering optimization problems.

BCSE0132: SOFT COMPUTING LAB

Credits: 01 **L–T–P: 0–0–2**

Module No.	Content	Lab Hours
I/II/III	• Plot different types of activation functions. • Program to create a perceptron network using command 'newp' • Program for single perceptron • To implement AND function using ADALINE with bipolar inputs and outputs. • To construct and test auto associative network for input vector using HEBB rule • Create a multilayer perceptron network and write a Program for Backpropagation network. • Program for fuzzy set with properties and operations • Program for composition on Fuzzy and Crisp relations • Program to find a relation using Max-Min Composition, enter the two vectors whose relation is to be find • Method of Defuzzification • Write a program for design an inference system using membership function • Write a program for all types of crossover methods. • Write a program for complete Genetic Algorithm cycle.	12*2=24

Text Books:

- N P Padhy (2015), "Soft Computing: With MATLAB Programming", Oxford Publication

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

CO1: Understanding about the existing software tools to solve real problems using a soft computing approach

CO2: Implementation various soft computing approaches for a given problem.

CO3: Evaluation and comparison of solutions by various soft computing approaches.

CO4: Design and implement the concepts of knowledge using fuzzy inference systems and other machine intelligence applications.

CO5: Design and implement the concepts of knowledge using fuzzy inference systems and other machine intelligence applications

BCSE0104: ARTIFICIAL INTELLIGENCE

Objective: *The objective of the course is to present an overview of artificial intelligence (AI) principles and approaches.*

Credits: 03 **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**Introduction:** Introduction to Artificial Intelligence, Foundations of Artificial Intelligence, Applications of Artificial Intelligence, Turing Test. **Introduction to Intelligent Agents:** Agents and Environment, Concept of rationality. Nature of environments, Knowledge based Agents, Structure of Intelligent Agents. **Problem solving:**Searching for solutions, Uninformed search strategies, Informed search strategies, Heuristic versus Solution Guaranteed Algorithm, Hill Climbing, Simulated Annealing, Best First Search (A*), Problem Reduction (AO*).	14
II	**Knowledge & Reasoning:** Propositional logic, First order predicate logic, Horn Clause, Inference in First order logic, Unification, Forward& Backward chaining, Resolution, Probabilistic reasoning, Bayesian Networks. **Knowledge Representation:** Semantic nets,Partition net, Minskey Frames, Case Grammar Theory, Transition Nets , Augmented Transition Net	13
III	**Game Playing:** Adversarial Search, Search for games, Alpha - Beta pruning, Min-Max algorithm, Example of Games. **Introduction to Expert System:** Rule based System, Architecture of expert system, description of DENDRAL and MYCIN, Domain exploration, Meta Knowledge, Fuzzy Expert System, Self-Explaining System. **Introduction to Machine Learning:** Introduction, Supervised and unsupervised learning, Decision trees, Introduction to Statistical learning models, Reinforcement learning.	14

Text Book:

- *S. Russell and P. Norvig, "Artificial Intelligence: AModern Approach", 3rd* Edition*Prentice Hall*

Reference Books:

- *Elaine Rich and Kevin Knight, "Artificial Intelligence", McGraw-Hill*
- *Dan W. Patterson, "Artificial Intelligence and Expert Systems", Prentice Hall of India,*
- *E Charniak and D McDermott, "Introduction to Artificial Intelligence", Pearson Education*

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

- CO1: Understand basic concept of artificial intelligence and intelligent Agent.
- CO2: Differentiate between informed and uninformed search.
- CO3: Apply First Order Predicate Logic in reasoning.
- CO4: Understand Knowledge Representation.
- CO5: Explain decision tree and statistical learning.
- CO6: Understand rule and its application in Expert Systems.

BCSE0105: MACHINE LEARNING

Objective*:*

Credits: 03 **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**Introduction:** Machine Learning basics, Hypothesis space and inductive bias, training and test set, and cross-validation. **Introduction to Statistical Learning:** Bayesian Method. **Machine Learning:** Supervised (Regression, Classification) vs. Unsupervised (Clustering) Learning. **Data Preprocessing:** Imputation, Outlier management, One hot encoding, Dimensionality Reduction- feature extraction, Principal Component Analysis (PCA), Singular Value Decomposition **Supervised Learning:** Regression- Linear regression, Polynomial regression, Classification- Logistic regression, k-nearest neighbor classifier,	20
III	**Supervised Learning:** Decision tree classifier, Naïve Bayes classifier Support Vector Machine (SVM)Classifier, **Unsupervised Learning:** k-means clustering, Hierarchical clustering **Underfitting vs Overfitting:** Regularization and Bias/Variance. **Ensemble methods:** Bagging, Boosting, Improving classification with Ada-Boost algorithm.	20

Text Book:

- *Tom M. Mitchell (2013), Machine Learning. Tata McGraw-Hill Education.*
- *Alpaydin, E. (2009). Introduction to machine learning. MIT press.*

Reference Books:

- *Harrington, P. (2012). Machine learning in action. Shelter Island, NY: Manning Publications Co.*
- *Bishop, C. M. (2006). Pattern recognition and machine learning (information science and statistics) springer-verlag new york. Inc. Secaucus, NJ, USA.*

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

CO1: Understand the basic concepts of machine learning including bias-variance tradeoff.

CO2: Explain the concepts of regression and re-sampling methods.

CO3: Conceptualize supervise and re-enforcement learning for classification.

CO4: Understand the ensemble methods for improving classification.

CO5: Identify the ways of feature extraction, reduction and selection.

CO6: Understanding the applications of machine learning algorithms.

BCSE0106: MACHINE LEARNING AND ITS APPLICATIONS

Objective*:*

Credits: 03 L–T–P: 3–0–0

Module No.	Content	Teaching Hours
I	**Introduction:** Machine Learning basics, Applications, Type of Learning Algorithm **Hypothesis space and inductive bias**: Concept-Learning, Hypotheses Representation,Find-S Algorithm **Machine Learning:Training, Testing, Evaluation:** Cross-validation, Confusion Matrix, Precision, Recall, F1-score, ROC curve, Overfitting, Bias, Variance Trade off **Supervised Learning:**Regression- Linear regression, Polynomial regression, Gradient Descent Algorithm **Data Preprocessing:** Imputation, Outlier management, One hot encoding, Feature selection, Filter and Wrapper approach **Supervised Learning:**Classification- Logistic regression, k-nearest neighbor classifier, Decision tree classifier, Naïve Bayes classifier,	20
II	**Supervised Learning:**Support vector machine classifier **Feature Extraction:** Principal Component Analysis (PCA) **Unsupervised Learning:** k-means clustering, Hierarchical clustering **Ensemble methods:** Bagging, Boosting,Improving classification with Random forest, Ada-Boost algorithm. **Artificial Neural Network -** Introduction, Model Representation, Gradient Descent, Stochastic Gradient Descent, Multilayer Perceptron, Multiclass Representation, Backpropagation Algorithm. **Deep Learning:** Introduction, Convolutional neural network and Recurrent neural network	20

Text Book:

- *Tom M. Mitchell (2013), Machine Learning. Tata McGraw-Hill Education.*
- *Alpaydin, E. (2009). Introduction to machine learning. MIT press.*

Reference Books:

- *Harrington, P. (2012). Machine learning in action. Shelter Island, NY: Manning Publications Co.*
- *Bishop, C. M. (2006). Pattern recognition and machine learning (information science and statistics) springer-verlag new york. Inc. Secaucus, NJ, USA.*

Outcome: After completion of Lab, student will be able to:
CO1: Understand the basic concepts of machine learning including bias-variance tradeoff.
CO2: Explain the concepts of regression and re-sampling methods.
CO3: Conceptualize supervise and re-enforcement learning for classification.
CO4: Understand the ensemble methods for improving classification.
CO5: Identify the ways of feature extraction, reduction and selection.
CO6: Understanding the applications of machine learning algorithms.

BCSE0133: MACHINE LEARNING LAB

Credits: 01 **L–T–P: 0–0–2**

S.No	PROGRAM
1	Estimate parameters of a model based on Linear Regression method using a given set of training data set.
2	Estimate parameters of a model based on maximum likelihood estimation method using a given set of training data set.
3	Compute weights of ANN based on back propagation method using a given training dataset.
4	Compute probability of a person to be diabetic based on a given dataset of diabetic persons using Naïve Bayesian classifier.
5	Classify a person as male or female based on a given dataset using naïve Bayesian Classifier, and calculate accuracy, precision, and recall for your data set.
6	Write a program to implement *k*-Nearest Neighbour method to classify the iris data set. Print both correct and wrong predictions. Use Java/Python ML library classes
7	Predicts whether the bank should approve the loan of an applicant, based on his profile using Ensemble learning method.
8	Apply Ensemble learning to cluster a set of data stored in a .CSV file. Use the same dataset for clustering using *k*-Means method. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.
9	The stock prediction data is used to predict, whether the stock will go up or down. Perform the task of feature selection with the help of wrapper method.
10	Identify principal components of Big Mart sales data using Principal component analysis (PCA). Also plot the result of PCA, and give inferences.

Outcome: After studying the subject, the students will be able to:
CO1: Able to use different digital image processing algorithms.
CO2: Design code and test digital image processing applications using MATLAB/Python language.
CO3: Apply image processing algorithms in real-life problems.

BCSE0141: MACHINE LEARNING PROJECT

Credits: 02 **L–T–P: 0–0––**

1. Iris Flowers Classification ML Project: Learn about Supervised Machine Learning Algorithms

Objective: The goal of this machine learning project is to classify the flowers into among the three species – virginica, setosa, or versicolor based on length and width of petals and sepals.

2. BigMart Sales Prediction ML Project: Learn about Unsupervised Machine Learning Algorithms

Objective: The goal of the BigMart sales prediction ML project is to build a regression model to predict the sales of each of 1559 products for the following year in each of the 10 different BigMart outlets.

3. Social Media Sentiment Analysis using Twitter Dataset

Objective: Working with the twitter dataset will help you understand the challenges associated with social media data mining and also learn about classifiers in depth.

4. Sales Forecasting using Walmart Dataset

Objective: The goal of this machine learning project is to forecast sales for each department in each outlet to help them make better data driven decisions for channel optimization and inventory planning.

5. Learn to build Recommender Systems with MovieLens Dataset

Objective: The goal of this machine learning project is to start working with this dataset by building a world-cloud visualization of movie titles to build a movie recommender system.

6. Stock Prices Predictor

Objective: Stock prices predictor is a system that learns about the performance of a company and predicts future stock prices.

7. Boston Housing Price Prediction ML Project

Objective: The goal of this machine learning project is to predict the selling price of a new home by applying basic machine learning concepts on the housing prices data.

8. MNIST Handwritten Digit Classification

Objective: The goal of this machine learning project is study, analyze and recognize elements in the images.

9. Human Activity Recognition using Smartphone Dataset

Objective: The goal of this machine learning project is to build a classification model that can precisely identify human fitness activities. Working on this machine learning project will help you understand how to solve multi-classification problems.

10. Use classification and clustering techniques to deal with the Turkiye Student Evaluation Data Set

Objective: This dataset is based on an evaluation form filled out by students for different courses. It has different attributes including attendance, difficulty, score for each evaluation question, among others. This is an unsupervised learning problem.

11. Predict height and weight from Heights and Weights dataset.

Objective: The goal of this project is to learn the concept of Regression.

12. Text Mining Data Set

Objective: The goal of this project is classify the documents according to their labels.

13. Predict the income class of any country population

Objective: The goal of this project is to work with imbalanced classification problem.

14. Classify the type of sound from the audio.

Objective: The goal of this project is to introduce the student to audio processing in the usual classification scenario.

SYLLABUS

OF

PROGRAM ELECTIVE

BOUQUET: ADVANCED DATA PROCESSING

DEPARTMENT OF COMPUTER ENGINEERING & APPLICATIONS

Under

Choice Based Credit System (CBCS)

Program Elective

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE-REQUISITES
			L	T	P	J			
Bouquet: Advanced Data Processing									
Theory									
1.	BCSE0151	Advanced Concepts in Database Systems	3	0	0	0	3	3	DBMS
2.	BCSE0152	Data Mining and Warehousing	3	0	0	0	3	3	DBMS
3.	BCSE0153	Business Intelligence	3	0	0	0	3	3	DMW
4.	BCSE0154	Information Retrieval System	3	0	0	0	3	3	DATA STRUCTURE
5.	BCSE0155	Distributed and Parallel Databases	3	0	0	0	3	3	DBMS
6.	BCSE0156	Natural Language Processing	3	0	0	0	3	3	TAFL/Compiler Design
7.	BCSE0157	Introduction to Big Data Analytics	3	0	0	0	3	3	DBMS
8.	BCSE0158	Big Data Analytics	3	0	0	0	3	3	
Labs									
1.	BCSE0181	Data Mining and Warehousing Lab	0	0	2	0	1	2	
2.	BCSE0182	Business Intelligence Lab	0	0	2	0	1	2	
3.	BCSE0183	Big Data Analytics Lab	0	0	2	0	1	2	
Projects									
1.	BCSE0191	Business Intelligence Project	0	0	0	-	2	-	
2.	BCSE0192	Big Data Analytics Project	0	0	0	-	2	-	Programming

BCSE 0151: ADVANCED CONCEPTS IN DATABASE SYSTEMS

Objective: *This course aims to give students in depth information about query processing and optimization, advanced query processing techniques, and advanced databases.*

Credits: 03 **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**Query Processing and Optimization:** Concept of High-level Query Processing, Algorithms for Executing Query Operations. Heuristics for Query Optimizations, Estimations of Query Processing Cost, Query Evaluation Plans, Pipelined Evaluations, Query Processing using Join, Group by and Sub Queries.	14
II	**Advanced Query Processing:** Advanced Query Processing using Records, Cursors, Stored Procedures. Advanced PL/SQL Functions: Parameters in Procedure and Functions, Exception Handling, Types of PL/SQL Triggers: Row level Triggers, Statement level Triggers, Cyclic Cascading in Triggers.	13
III	**Enhanced Data Model for Advanced Applications:** Introduction to Temporal Database Concepts, Spatial and Multimedia Databases, Active Database System, Deductive Databases, Main Memory Database, And Evolutionary Database. **Introduction to Expert Database and Fuzzy Database System:** Expert Databases - Use of Rules of Deduction in Databases, Recursive Rules. Fuzzy Databases – Fuzzy Set & Fuzzy Logic, Use of Fuzzy Techniques to Define Inexact and Incomplete Databases.	13

Text Books:

- Elmasri, Navathe (2016), “Fundamentals of Database Systems”, 7th Edition, Pearson.

Reference Books:

- Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Wisdom (2008), “Database Systems: The Complete Book",2ND Edition, Pearson.
- Korth, Silbertz, Sudarshan (2011), “Database Concepts”, 6thEdition, McGraw Hill.
- Ramakrishnan, Gehrke (2007), “Database Management System”, 3rd.Edition, McGraw Hill.

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

- CO 1: Design high level query using Join, Group by and sub queries.
- CO 2: Understand query optimization and query evaluation using different techniques.
- CO 3: Implement advanced queries as series of primitive operations.
- CO 4: Implement PL/SQL triggers in different applications.
- CO 5: Understand functioning of advance databases.
- CO 6: Apply expert databases in designing fuzzy logic based applications.

BCSE 0152: DATA MINING & WAREHOUSING

Credits: 03 **L–T–P: 3–0–0**

Objective: *The Objective of this course is to introduce the basic concepts of Data Warehouse and Data Mining techniques.*

Module No.	Content	Teaching Hours
I	**Data Warehousing**: Overview, Difference between Database System and Data Warehouse, Multi-dimensional Data Model: Concept Hierarchy, Three-Tier Architecture, Meta Repository, Data Warehouse & OLAP Technology, Types of OLAP Servers. Data Cubes Computations & Data Generalization. **Data Pre Processing**: Data Cleaning, Data Integration and Data Transformation, Data Reduction **Mapping the Data Warehouse** to a Multiprocessor Architecture, Multi Dimensional Data Model. **Introduction**: Basics of Data Mining, Issues and Applications of Data Mining Techniques. Mining frequent Patterns: Basic Concepts of Association Rules Mining, Apriori Algorithm, FP-Growth. Multilevel Association Rules, Multi-Dimensional Association Rules.	13
II	**Classification and Predictions**: Classification & Prediction, Issues Regarding Classification and Prediction, Decision Tree, Bayesian Classification, Back Propagation, Neural Network, Nearest Neighbour Classifiers, Support Vector Machines, Prediction. **Data Mining Cluster Analysis**: Data Types in Cluster Analysis, Categories of Clustering Methods, Partitioning Methods. **Hierarchical Clustering-** CURE and Chameleon. **Density Based Methods-**DBSCAN, OPTICS. Grid Based Methods STING, CLIQUE. **Model Based Method** –Statistical Approach, Outlier Analysis, Mining Multimedia Data, Text Mining, Web Data Mining, Spatial Data Mining, Temporal Data Mining, Data Visualization.	13

Text Books:

- Jiawei Han, MichelineKamber (2013),” Data Mining Concepts& Techniques”, 3rdEdition, Morgan Kauffmann.

Reference Books:

- M. H. Dunham (2003), “Data Mining: Introductory and Advanced Topics”, 1st Edition, Pearson Education
- Sam Anahory, Dennis Murray (2009), “Data Warehousing in the Real World: A Practical Guide for Building Decision Support Systems”, 4th Edition, Pearson Education.
- Pang-Ning Tan, Michael Steinbach, Vipin Kumar (2006), “Introduction to Data Mining”, 2ndEdition, Addison-Wesley.
- Aggarwal, (2015), “Data Mining: The Textbook”, Springer.

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

CO1: Understand and apply the concept of data warehouse and mining in real-life applications.

CO2: Apply the principle algorithms used in modern machine learning.

CO3: Apply the information theory and probability theory to get the basic theoretical results in Data Mining.

CO4: Apply Data mining algorithms to real datasets, evaluate their performance and appreciate the practical issues involved.

CO5: Perform clustering using various clustering methods on data set

BCSE0181: DATA MINING & WAREHOUSING LAB

Credits: 01 **L–T–P: 0–0–2**

Objective: The Objective of this course is to implement and run the programme based on the basic concepts of Data Warehouse and Data Mining techniques

Module No.	Content	Lab Hours
I/II	• Demonstration of pre-processing on different dataset • Demonstration of Association rule process on different dataset using apriori algorithm • Demonstration of classification rule process on different dataset using FP Tree algorithm • Demonstration of classification rule process on different dataset using id3 algorithm • Demonstration of classification rule process on different dataset using naïve bayes algorithm • Demonstration of clustering rule process on different dataset using simple k-means • Demonstration of clustering rule process on different dataset using simple k-mediods • Demonstration of clustering rule process on different dataset using simple k-mode. • Demonstration of clustering rule process on different dataset using DBSCAN. • Demonstration of clustering rule process on different dataset using simple Hieratical based algorithm. • Implementation of K-NN Algorithms on different data sets. • Implementation of Sequential pattern SPADE algorithm on sequence data set. • Implementation of Sequential pattern GSP algorithm on sequence data set. • Implementation of SVM on a two dimensional data set. • Demonstration of Decision Tree on Weka Tool.	24

Text Books:

- Toby Segaran (2008), Programming Collective Intelligence Building Smart Web 2.0 Applications, O'Reilly Media.

Reference Books:

- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten (2009). The WEKA Data Mining Software: An Update. SIGKDD Explorations, Volume 11, Issue 1.
- https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf

Outcome:

CO1: After completion of this course students will be able to:
CO2: Study pre-processing of data sets.
CO3: Preprocess the data sets
CO4: Identify and apply appropriate classification and clustering techniques for a problem
Analyze various processes of data mining.

BCSE0153: BUSINESS INTELLIGENCE

Objective: *The objective of this course is for the students to achieve a profound understanding of Business Intelligence (BI) systems in terms of its tools, current practices and impacts.*

Credits: 03 **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**Introduction to Business Intelligence:** BI Definitions & Concepts, BI Framework, Data Warehousing Concepts and Its Role in BI; BI Infrastructure Components – BI Process, BI Technology, BI Roles & Responsibilities, Business Applications of BI, BI Best Practices. Effective and timely decisions – Data, information and knowledge – Role of mathematical models – Business intelligence architectures: Cycle of a business intelligence analysis – Enabling factors in business intelligence projects – Development of a business intelligence system – Ethics and business intelligence.**Basics of Data Integration:** Concepts of Data Integration, Needs and Advantages of using Data Integration, Introduction to Common Data Integration Approaches; Meta Data - Types and Sources, Introduction to Data Quality	20
II	Data Profiling Concepts and Applications, Introduction to ETL using Kettle. Marketing models – Logistic and Production models – Case studies. **Efficiency Measures and Metrics:** Efficiency measures – The CCR model: Definition of target objectives- Peer groups – Identification of good operating practices; cross efficiency analysis – virtual inputs and outputs – Other models. Introduction to Business Metrics and KPIS, Creating Cubes using Microsoft Excel. **Basics of Enterprise Reporting:** A Typical Enterprise, Malcolm Bridge - Quality Performance Framework, Balanced Scorecard, Enterprise Dashboard, Balanced Scorecard vs. Enterprise Dashboard, Enterprise Reporting using MS Access / MS Excel, Best Practices in the Design of Enterprise Dashboards.	20

Text Books:

- RN Prasad and Seema Acharya (2018), “Fundamentals of Business Analytics”, Wiley India, 2nd Edition.

Reference Books:

- U Dinesh Kumar (2017), “Business Analytics: The Science of Data - Driven Decision Making”, Wiley India, 1st Edition.
- David Loshin (2012), “Business Intelligence”, 2nd Edition, Elsevier Science & Technology.
- Mike Biere (2010), “Business Intelligence for the Enterprise”, Pearson.
- IBM (2004), “An Introduction to Building Data Warehouse”, Prentice Hall of India.
- Larissa Terpeluk Moss &ShakuAtre (2003), “Business Intelligence Roadmap”, Pearson.

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

CO1: Identify the major frameworks of computerized decision support: decision support systems (DSS), data analytics and business intelligence (BI).

CO2: Explain the foundations, definitions, and capabilities of DSS, data analytics and BI.

CO3: Design tested and effective advanced analytics models and simulations for decision making.

CO4: To study the methodology of engineering legacy databases for business intelligence to derive business rules for decision support systems.

CO5: Articulate assumptions, analyses, and interpretations of data in an oral formatApply big data technologies in business intelligence using geospatial data, location-based analytics, social networking, Web 2.0, reality mining, and cloud computing.

BCSE0182: BUSINESS INTELLIGENCE 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: 0–0–2**

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: At the end of the course, student is able to:

CO1: Use R-Studio, read R documentation, and write R scripts.
CO2: Import, export and manipulate data.
CO3: Analyse the data using data analytics latest tools based on HDFS like Pig, Hive
CO4: Implement the aggregation projection on data set using Cassandra, MongoDB.

Big Data Projects

Credits: 0 **L–T–P: 0–0–0**

Project No. 1.House Price Prediction

Objective: Floor area, River view many factor affect the house price. Predict price of houses.

Tasks: 1) Load Data in to HDFS

2) Analyse using Mapreduce/PIG/R

3) Use Mahout or R for Data Processing and prediction

Data Set:

https://www.kaggle.com/shree1992/housedata

Project No. 2.Predict Customer Churn

Objective: Companies invest significantly to acquire new customers. So, after acquisition, companies would like these customers to stay long and be loyal. Companies also make investments to keep engaging with these customers continuously and ensure they are happy and satisfied with their offerings. Losing customers mean loss of investment and loss of possible future revenue. So, it is important for companies to infer early signs of a customer about to churn and engage or offer incentives to retain them. But as targeting each customer is not possible, companies can take advantage of analytics to predict if a customer high probability of churning. A possible intervention can be made to retain the customer.

Data Set:

https://www.kaggle.com/aadrika/customer-churn

Project No. 3.Predict Coronary Heart Disease

Objective: There are many factors which affect heart so take most prominent of them and predict whether the patient may have risk of heart disease

Data Set: https://www.kaggle.com/johnsmith88/heart-disease-dataset

Project No. 4.Twitter Trends Analysis

Objective: The community of users participating in social media tends to share about common interests at the same time, giving rise to what are known as social trends. A social trend reflects the voice of a large number of users which, for some reason, becomes popular in a specific moment. Through social trends, users therefore suggest that some occurrence of wide interest is taking place and subsequently triggering the trend. In this work, we explore the types of triggers that spark trends on the micro blogging site Twitter, and introduce a typology that includes the following four types: news, ongoing events, memes, and commemoratives. The user will be allowed to search for the latest trends by inputting a keyword into search field. Based on user provided keyword, the system will search for similar keywords in database and summarize the total count to provide the trending tweets on twitter. The trending tweets with hashtag (#) will be displayed first and then the rest words will be displayed. By clicking on every trending tweet, the user commented tweets will be displayed. User can view all the tweets from the searched keyword.

Project No. 5.Stock Price Prediction

Objective: Predict the price stocks based on current trending factors.

Dat Set: https://www.kaggle.com/aumashe/stock-ew

Project No. 6. Movielens: Top 20 best rated movies

Objective: Find out 20 worst rated movies. But only consider those movies which are rated by at least 100 users. Find out best 10 and worst 10 movies in each category - Categories are tags given in the movies.

Project No. 7. Sales Performance Prediction

Objective: predict sales performance based on factors such as marketing expenditure, advertisement and others.

Project No. 8. Recommending movies to users using Collaborative Filtering

Objective: Knowing "What customers are most likely to buy in future" is key to personalized marketing for most of the businesses. Understanding customers past purchase behavior or customer demographics could be key to make future buy predictions. But how to use the customer behavior data, depends on many different algorithms or techniques. Some algorithms may use demographic information to make this prediction. But most of the times, the organizations may not have these kind of information about customers at all. All that organization will have been what customers bought in past or if the liked it or not. Recommendation systems use techniques to leverage this information and make recommendation, which has been proved to be very successful. For examples, Amazon.com's most popular feature of "Customers who bought this also buys this?"

Project No. 9. Detect Emotion in Text

Objective: When you read, "Why don't you ever text me", does it conveys an angry emotion or sad emotion?
Understanding Emotions in Textual Conversations is a hard problem in absence of voice modulations and facial expressions.

Project No. 10. Identify your Digits

Objective: Study, analyze and recognize elements in the images. That’s exactly how your camera detects your face, using image recognition! It’s your turn to build and test that technique. It’s a digit recognition problem.

BCSE0191: BUSINESS INTELLIGENCE PROJECT

Credits: 02 **L–T–P: 0–0–0**

Project No.11. Air-Traffic arrival as well as departure Visualization Project in Business Intelligence

Objective: Create a visual panel (report) with multiple prompts and filters to show multiple stages of Aeroplane Movement

Project No.12. University Placement Activity Visualization Project

Objective: Create multiple reports to visualize all kinds of placement activity such as placed students with unplaced students, company wise placed students, visited companies, upcoming companies etc. with filters and prompts

Project No.13. Presentation of Wal-Mart sales (region and product wise)

Objective: Create Dashboard for displaying multiple products and their sales analysis using multiple report types such as gauge chart, crosstab, pie chart etc and also represent it by colours such as high selling product in green, low selling product in red in same way for different region as well.

Project No. 14. Country Crime Statistics Reporting Project in Business Intelligence

Objective: Display multiple crime reports with drill down feature like from country to state and state to district and district to city...with filtering capability for various reasons such as display of city or state which had highest number of crimes.

Project No.15. Stock Market Statistics reporting Project in Business Intelligence

Objective: Display trends of various companies' shares with drill down as well as drill up feature on date basis

Project No.16. Company's Revenue Report Project in Business Intelligence

Objective: compares how different order methods are performing for each product line. This report should display the revenue that individual order methods generate for each product line and the average revenue all order methods generate for each product line.

Project No. Sales Performance Reporting Project

Objective: The Vice President of Sales has requested a report that shows sales performance in each country for 2012. He wants to see the performance for representatives in Southern Europe so he can present an award to the top seller when he visits next month

Project No.18.Create a Report Focused on Top Performing Product Types and Product Lines

Objective: create a report that displays revenue by product line and product type. The report must show the product types that generated revenue greater than $100 million and product lines that generated revenue greater than $400 million with various charts and colouring scheme

Project No.19. Designing of Dashboard for various aspects

Objective: create a dashboard report that contains a gauge chart that compares the gross profit of each product line by region, a combination chart that shows revenue earned by each product line by retailer type and region on separate axis, and finally a crosstab report that shows the gross margin of each product line by year and region.

Project No.20. Sales Titled reporting (with multiple perspectives) Project

Objective: You have been asked for a report that displays the quantity of products sold for each order year. You also need to display all product lines in uppercase. The report should contain an optional prompt that lets users view data by sales region. Add a report title that indicates which sales region users select in the prompt. It should also indicate if they do not select a region as well. You will use a layout calculation to display the report title

Project No. 21. Employee Development Program Reporting Project

Objective: HR would like a report that outlines the courses each employee has completed. They want to see the number of course days that each employee has accumulated to date and the cost of those courses. They would also like to be able to drill-through to specific information on each employee for each year.

Project No.22.Production analysis and it Planning Reporting Project

Objective: prepare a report that shows the quantity of products sold in each month of 2012 for all product lines, to help estimate production requirements for next year. The report must be broken into separate sections for each product line so that products from each line can be analyzed separately. The report name and logo must appear at the top and bottom of each page of the report.

Project No.23. Bollywood Movies trends reporting Project

Objective: This is to create Bollywood Movies reports with filter and prompts to let user to find out list of movies based on different users' taste (e.g. emotional, scary ..Etc) and also show other report to show most/least user rated movie

Project No.24. Analysis of Profit reporting Project in Business Intelligence

Objective: create a list report where users can review the gross profit generated by retailer types for each region gained by XYZ (product wise) providing multiple filter and prompts and also name the report

Project No.25. Analysis on various income class of any country population

Objective: Create a dashboard where a manager can visualize various Income class of people based on region (country, state, district and so on...) providing multiple prompts and filter to drill down in reporting levels

BCSE0154 : INFORMATION RETRIEVAL SYSTEMS

Objective: *This course aims to give students an understanding of the fundamental techniques for hyper-media architectures, design and usability, document management and retrieval, meta data management, and searching the web.*

Credits: 03 **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**Introduction to IR:** IR Concepts, Boolean Retrievals- An Example Information Retrieval Problem, A First Take at Building an Inverted Index, Processing Boolean Queries. **The Term Vocabulary and Postings Lists:** Document Delineation and Character Sequence Decoding, Determining the Vocabulary of Terms. **Dictionaries and Tolerant Retrieval:** Search Structures for Dictionaries, Wildcard Queries, Spelling Correction, Phonetic Correction. **Index Construction:** Hardware Basics Blocked Sort-Based Indexing. Scoring, Term Weighting and the Vector Space Model: Parametric and Zone Indexes, Term Frequency and Weighting, The Vector Space Model for Scoring.	22
II	**Evaluation in Information Retrieval:** Information Retrieval System Evaluation, Standard Test Collections, Evaluation of Unranked Retrieval Sets, Evaluation of Ranked Retrieval Results. **XML Retrieval:** Basic XML Concepts, Challenges in XML Retrieval, A Vector Space Model for XML Retrieval, Evaluation of XML Retrieval, Text-Centric vs. Data-Centric XML Retrieval. **Web Search Basics:** Web Characteristics, Advertising as the Economic Model, The Search User Experience, Index Size and Estimation, Near-Duplicates and Shingling. **Web Crawling and Indexes:** Overview, Crawling, Distributing Indexes, Connectivity Servers. **Link Analysis:** The Web as a Graph, Page Rank, Hubs and Authorities.	20

Textbook:

- Christopher D. Manning, PrabhakarRaghavan and HinrichSchütze (2009), "Introduction to Information Retrieval", Cambridge University Press.

Reference Books:

- Ricardo Baeza-Yate, Berthier Ribeiro-Neto (2011),"Modern Information Retrieval",2nd Edition, Addison Wesley.
- SoumenChakrabarti (2002), "Mining the Web: discovering knowledge from hypertext data", 2nd Edition, Morgan Kaufmann.
- David A. Grossman, Ophir Frieder (2004), "Information Retrieval: Algorithms, and Heuristics",2nd Edition, Springer.

Outcome: After successful completion of this course, students should be able to:
CO1: Use different information retrieval techniques in various application areas
CO2: Analyze indexing and pre-processing of textual documents for IR system
CO3: Apply IR principles into Spelling Correction, Phonetic Correction
CO4: Analyze performance of retrieval systems
CO5: Apply IR techniques to XML Retrieval
CO6: Implement retrieval systems for web search tasks.
CO7: Demonstrate similarity computation using link analysis
CO8: Apply big data technologies in business intelligence using geospatial data, location-based analytics, social networking, Web 2.0, reality mining, and cloud computing.

BCSE0155: DISTRIBUTED AND PARALLEL DATABASE

Objective: *The course introduces the types of parallelism in database systems, architecture of parallel and distributed database systems; parallel query processing; data partitioning.*

Credits: 03 **L-T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**Introduction:** Introduction to Databases, Overview of Relational DBMS, Relational Database concepts, Normalization. **Distributed Database:** Introduction, Comparison of Distributed and Centralized Database Systems, Distributed Database Architecture, Distributed DataBase Design, Types of Data Fragmentations, Fragmentation and Allocation of Fragments, Distributed Catalog Management. **Transactions Management in Distributed Database:** Properties and Goals of Transaction Management, Distributed Transactions, Two Phase Commit Protocol, Recovery Mechanism in case of Transaction Failures, Log Based Recovery, Communication and Site Failures.	20
II	**Concurrency Control in Distributed Database:** Serializability, Locking and Timestamp Based Concurrency Control Approach in Distributed Databases. Optimistic Concurrency Control Approach, Introduction to Distributed Deadlocks, Local and Global Wait-For-Graphs, Deadlock Detection and Prevention of Deadlocks. **Parallel Database:** Database Server Approach, Parallel Architectures, Parallel DBMS Techniques- Data Placement, Query Parallelism, Parallel Database Processing, Parallel Query Optimization; Parallel Execution Problems- Initialization, Interferences and Convoy Effect, Load Balancing, Parallel Execution for Hierarchical Architecture – Basic Concept, Load Balancing Strategy, Performance Evaluation.	20

Text Books:

- M.Tamer Oezsu, Patrick Valduriez(2011). "Principles of Distributed DatabaseSystems",2ndEdition, Prentice Hall.

Reference Books:

- Elmasri, Navathe (2011), "FundamentalsofDatabaseSystems",5thEdition, Pearson.
- Hector Garcia-Molina,Jeffrey D. Ullman,Jennifer Widom (2008),"Database Systems: The CompleteBook",2ndEdition,Pearson.
- Philip A. Bernstein, VassosHadzilacos, Nathan Goodman (1987), "Concurrency Control and Recovery in Database Systems",Addison-Wesley.
- Jim Gray, Andreas Reuter (1993),"Transaction Processing: Concepts and Techniques", Morgan Kaufmann.

Outcome: At the end of the course, student will able to:
CO1: understand the concepts of distributed database and its structures.
CO2: perform queries for effective, noiseless data retrieval in distributed database environment.
CO3: Implement various concurrency control approaches to ensure recovery of Database.
CO4: Execute primitive operations concurrently.
CO5: Manage parallel computing environments, and understand their implications and cost benefits and services.
CO6: apply design techniques, middleware and development tools for tuning an application environment.

BSCE0156: NATURAL LANGUAGE PROCESSING

Objective: *The objective of this course is to introduce to the students the leading trends and system in natural language processing and make them understand the concepts of morphology, syntax, semantics and pragmatics of the language.*

Credits: 03 **L-T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	Introduction to Natural Language Understanding: The study of Language, Applications of NLP, Evaluating Language Understanding Systems, Different levels of Language Analysis, Representations and Understanding, Organization of Natural Language Understanding Systems, Linguistic Background: An outline of English syntax. Applications: Named entity recognition and relation extraction- IE using sequence labelling-Machine Translation (MT) - Basic issues in MT-Statistical translation-word alignment- phrase-based translation – Question Answering Grammars and Parsing: Grammars and sentence Structure, Top-Down and Bottom-Up Parsers, Transition Network Grammars, Top-Down Chart Parsing. Feature Systems and Augmented Grammars: Basic Feature system for English, Morphological Analysis and the Lexicon, Parsing with Features, Augmented Transition Networks.	20
II	Grammars for Natural Language: Auxiliary Verbs and Verb Phrases, Movement Phenomenon in Language, Handling questions in Context-Free Grammars. Human preferences in Parsing, Encoding uncertainty, Deterministic Parser. Ambiguity Resolution: Statistical Methods, Probabilistic Language Processing, Estimating Probabilities, Part of speech Tagging - Rule-Based Part of Speech Tagging - Markov Models - Hidden Markov Models, Obtaining Lexical Probabilities, Probabilistic Context-Free Grammars, Best First Parsing. Semantics and Logical Form, Word senses and Ambiguity, Encoding Ambiguity in Logical Form.	20

Text Book:

1. James Allen, Natural Language Understanding, 2/e, Pearson Education, 2003

Reference Books:

1. Akshar Bharti, Vineet Chaitanya and Rajeev Sangal, NLP: A Paninian Perspective, Prentice Hall, New Delhi
2. D. Jurafsky, J. H. Martin, Speech and Language Processing, Pearson Education, 2002
3. L.M. Ivansca, S. C. Shapiro, Natural Language Processing and Language Representation
4. T. Winograd, Language as a Cognitive Process, Addison-Wesley

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

CO1: Understand the core tasks in NLP and its applications in real problems

CO2: Understand the human languages and be familiar with the most mainstream descriptive and theoretical frameworks for handling their properties

CO3: Grasp basics of knowledge representation, inference, and relations to the artificial intelligence.

CO4: Understand the algorithmic description of the main language levels: morphology, syntax, semantics, and pragmatics.

CO5: Understood the various translation strategies and techniques of maintaining balance among the three basic criteria of translation: accuracy, clarity and naturalness.

CO6: Apply the Markov Models and it's variants to do the Part of Speech Tagging

BCSE0157: INTRODUCTION TO BIG DATA ANALYTICS

Objective:

Credits: 03 **L-T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**Big Data Technology Landscape**: 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, 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 **Introduction to Mongo DB**: RDBMS vs. MongoDB, JSON, Unique Key, Dynamic Queries, Sharding, Replication, MongoDB QL: Create, Drop Database and Collections, CRUD: Create, Insert, Find, Update, Delete, Map Reduce Programming, Aggregations	20
II	**Introduction to Cassandra DB**: Features of Cassandra, CQL Data Types, CQLSH: CRUD, Counter, TTL, List, Set, Map, Tracing, Import Export csv files **HADOOP Ecosystem and Flume:** Introduction to Hadoop Ecosystem, Sqoop, Zookeeper, Plug-in Components: Impala, Hue, Flume: Introduction, Application, Advantage, Features. **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.	20

Text Book:

- *Seema Acharya and SubhashiniChellappan, "Big Data and Analytics", 1st Edition, 2015, Wiley, India.*
- *Jure Leskovec, AnandRajaraman, Jeff Ullman, "Mining of Massive Datasets", 2nd Edition, 2014, Cambridge University Press.*

Reference Books:

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

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

CO 1. Understand the concept and challenges of big data

CO 2. Work with existing technology to collect, manage, store, query, and analyze the various form of big data;

CO 3. Perform job scheduling of various applications and resource management in the cluster using Hadoop and Yarn

CO 4. Do the data summarization, query, and analysis over the big data with the help of pig and hive?

CO 5. Prepare the regression model, cluster and decision tree over the real big data

CO 6. Gain hands-on experience in large-scale analytics tools to solve some open big data problems;

BCSE0158: BIG DATA AND 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: 3–0–0**

Module No.	Content	Teaching Hours
I	**Big Data technology Landscape**: 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, 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 Spark & functional programming, Spark Concepts, RDD Fault Tolerance **Introduction to HBase:** What is HBase? HBase Architecture, HBase Components, Data model, HBase Storage Hierarchy, Cross-Datacenter Replication, Auto Sharding **Introduction to Cassandra DB:** Features of Cassandra, CQL Data Types, CQLSH: CRUD, Counter, TTL, List, Set, Map, Tracing, Import Export csv files	19
II	**HADOOP Ecosystem and Flume:** Introduction to Hadoop Ecosystem, Sqoop, Zookeeper, Plug-in Components: Impala, Hue, Flume: Introduction, Application, Advantage, Features. **Introduction to Mongo DB**: RDBMS vs. MongoDB, JSON, Unique Key, Dynamic Queries, Sharding, Replication, MongoDB QL: Create, Drop Database and Collections, CRUD: Create, Insert, Find, Update, Delete, Map Reduce Programming, Aggregations **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. **Machine Learning:** Linear Regression, Logistic Regression, Association Rule	19

Text Book:

- 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, 2014, Manning Publications.

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

CO 1. Understand the concept and challenges of big data

CO 2. Work with existing technology to collect, manage, store, query, and analyze the various form of big data.

Co 3. Perform job scheduling of various applications and resource management in the cluster using Hadoop and yarn.

CO 4. Do the data summarization, query, and analysis over the big data with the help of pig and hive?

CO 5. Prepare the regression model, cluster and decision tree over the real big data

CO 6. Gain hands-on experience in large-scale analytics tools to solve some open big data problems.

BCSE0183: BIG DATA AND 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: 0–0–2**

Module No.	Content	Lab Hours
I	**Module 1: Introduction to R** • Introduction and installation of R and RStudio • Data types, vectors, multidimensional array. • Functions and their use • Visualization using ggplot2. • Word-Count program using Java	24
II	**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	
III	**Module 3: Hands-On MapReduce** • Hands-On PIG • Hands-On HIVE • Twitter Data Fetching using Flume	

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: At the end of the course, student is 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.

BCSE0192: BIG DATA ANALYTICS PROJECTS

Credits: 2 **L–T–P-J: 0–0–0--**

Project No. 1.House Price Prediction

Objective: Floor area, River view many factor affect the house price. Predict price of houses.

Project No. 2.Predict Customer Churn

Objective: Companies invest significantly to acquire new customers. So, after acquisition, companies would like these customers to stay long and be loyal. Companies also make investments to keep engaging with these customers continuously and ensure they are happy and satisfied with their offerings. Losing customers mean loss of investment and loss of possible future revenue. So, it is important for companies to infer early signs of a customer about to churn and engage or offer incentives to retain them. But as targeting each customer is not possible, companies can take advantage of analytics to predict if a customer high probability of churning. A possible intervention can be made to retain the customer.

Project No. 3.Predict Coronary Heart Disease

Objective: There are many factors which affect heart so take most prominent of them and predict whether the patient may have risk of heart disease

Project No. 4.Twitter Trends Analysis

Objective: The community of users participating in social media tends to share about common interests at the same time, giving rise to what are known as social trends. A social trend reflects the voice of a large number of users which, for some reason, becomes popular in a specific moment. Through social trends, users therefore suggest that some occurrence of wide interest is taking place and subsequently triggering the trend. In this work, we explore the types of triggers that spark trends on the micro blogging site Twitter, and introduce a typology that includes the following four types: news, ongoing events, memes, and commemoratives. The user will be allowed to search for the latest trends by inputting a keyword into search field. Based on user provided keyword, the system will search for similar keywords in database and summarize the total count to provide the trending tweets on twitter. The trending tweets with hashtag (#) will be displayed first and then the rest words will be displayed. By clicking on every trending tweet, the user commented tweets will be displayed. User can view all the tweets from the searched keyword.

Project No. 5.Stock Price Prediction

Objective: Predict the price stocks based on current trending factors.

Project No. 6.Movielens: Top 20 best rated movies

Objective: Find out 20 worst rated movies. But only consider those movies which are rated by at least 100 users. Find out best 10 and worst 10 movies in each category - Categories are tags given in the movies.

Project No. 7.Sales Performance Prediction

Objective: predict sales performance based on factors such as marketing expenditure, advertisement and others.

Project No. 8.Recommending movies to users using Collaborative Filtering

Objective: Knowing "What customers are most likely to buy in future" is key to personalized marketing for most of the businesses. Understanding customers past purchase behavior or customer demographics could be key to make future buy predictions. But how to use the customer behavior data, depends on many different algorithms or techniques. Some algorithms may use demographic information to make these predictions. But most of the times, the organizations may not have these kind of information about customers at all. All that organization will have been what customers bought in past or if the liked it or not. Recommendation systems use techniques to leverage this information and make recommendation, which has been proved to be very successful. For examples, Amazon.com's most popular feature of "Customers who bought this also buys this?"

Project No. 9.Detect Emotion in Text

Objective: When you read, "Why don't you ever text me", does it conveys an angry emotion or sad emotion?

Understanding Emotions in Textual Conversations is a hard problem in absence of voice modulations and facial expressions.

Project No. 10.Identify your Digits

Objective: Study, analyze and recognize elements in the images. That’s exactly how your camera detects your face, using image recognition! It’s your turn to build and test that technique. It’s a digit recognition problem.

SYLLABUS

OF

PROGRAM ELECTIVE

BOUQUET: HIGH PERFORMANCE COMPUTING

DEPARTMENT OF COMPUTER ENGINEERING & APPLICATIONS

Under

Choice Based Credit System (CBCS)

Program Elective

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE- REQUISITES
			L	T	P	J			
Bouquet: High Performance Computing									
Theory									
1.	BCSE0201	Advanced Computer Architecture	3	0	0	0	3	3	Computer Organization
2.	BCSE0202	Embedded System	3	0	0	0	3	3	Microprocessors
3.	BCSE0203	Internet of Things	3	0	0	0	3	3	Microprocessors
4.	BCSE0204	Internet of Things and Its Applications	3	0	0	0	3	3	
5.	BCSE0205	Distributed System	3	0	0	0	3	3	CN /OS
6.	BCSE0206	Parallel Algorithms	3	0	0	0	3	3	CO& Algorithms
7.	BCSE0207	Cloud Computing	3	0	0	0	3	3	
8.	BCSE0208	Cloud Computing and Virtualization	3	0	0	0	3	3	
Labs									
1.	BCSE0231	Embedded System Lab	0	0	2	0	1	2	
2.	BCSE0232	Internet of Things Lab	0	0	2	0	1	2	
3.	BCSE0233	Parallel Algorithms Lab	0	0	2	0	1	2	
4.	BCSE0234	Cloud Computing lab	0	0	2	0	1	2	
Projects									
1.	BCSE0241	Internet of Things Project	0	0	0	-	2	-	
2.	BCSE0242	Cloud Computing and Virtualization Project	0	0	0	-	2	-	

BCSE0201: ADVANCE COMPUTER ARCHITECTURE

Objective: *The objective is to describe the operation of modern and high performance computers and ability to undertake performance comparisons of modern and high performance computers.*

Credits: 03 **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**Review of Traditional Architectures:** Evolution of Computer Architecture, Review of Fundamentals of CPU, Memory & IO, Classes Of Computers & Trends In Architecture & Technology. **Introduction to High Performance Computing:** Need for Parallel Computing, Introduction to Parallel Computer Models, Parallel Architectural Classification Schemes, Speedup Performance Laws. **Program and Network Properties**: Condition of Parallelism, data and resource dependence, Grain Size and Latency, System interconnect architecture- Mesh, cube, butterfly, hypercube, PM2I, Bus system. **Pipelining Techniques:** Introduction to Pipelining, Instruction Pipeline, Arithmetic Pipeline, Hazards, Hazards Detections and Resolution, Instruction Prefetching, Branch Handling Techniques. Scheduling of Static & Dynamic Pipelines.	21
II	**Memory:** Memory Hierarchy Technology, Cache Memory Organization, Memory Replacement Policies, Cache Coherence, Inclusion and Locality. **Shared Memory Organization**: Memory Interleaving, Bandwidth and Fault Tolerance, Memory Allocation Schemes. **Parallel and Scalable Architectures:** Multiprocessor and Multicomputer, System Interconnects, Cache Coherence and Synchronization, Message Passing Mechanism. **Different Architectures:** Superscalar and Vector Processor, VLIW Architecture, Data flow Computer, Multicore Architecture	19

Text Books:

- Kai Hwang, NareshJotwani (2011), "Advanced Computer Architecture", 2ndEdition, Tata McGraw-Hill.

Reference Books:

- John L. Hennessy (2011), "Computer architecture: A Quantitative Approach", 5th Edition, Elsevier.
- AnanthGrama, George Karypis, Vipin Kumar, Anshul Gupta (2003), "Introduction to Parallel Computing", 2ndEdition, Addison Wesley.
- Bhujade (2011), "Parallel Computing", 2ndEdition, New Age International.

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

CO1: Students are able to understand different processor architectures and system-level design processes.

CO2: Understand the components and operation of a memory hierarchy and the range of performance issues influencing its design.

CO3: Understand the organization and operation of current generation parallel computer systems, including multiprocessor and Multicore systems.

CO4: Understand the principles of I/O in computer systems, including viable mechanisms for I/O and secondary storage organization.

BCSE0202: EMBEDDED SYSTEM

Objective: *To acquaint students with methods of executive device control and to give them opportunity to apply and test those methods in practice.*

Credits: 03 **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I.	**Introduction:** Introduction to Embedded Systems, Hardware Needs Challenges and Design Process of Embedded System, Processor Architectures, Memory Organization, Embedded Devices and Its Network. **Architecture of Embedded System**: CPUs, Bus Based Computer Systems, Programming Design and Analysis, Model of Program, Basic Compilation Techniques, Program Optimization and Performance. **OS for Embedded Systems:** Real Time Operating System, ISR in RTOS, Basic Design Using an RTOS, RTOS Task Scheduling Models, RTOS Programming.	20
II.	**Software Architectures:** Processes and Operating Systems, Multiple Task and Multiple Processes, Preemptive RTOS, Priority Based Scheduling, Inter-process Communication Mechanism, Multiprocessors. **Program Modeling Concepts:** Program Model, DFG Models, Modeling of Multiprocessor Systems, UML Modeling, Embedded Software Development Process and Tools. **Networks &System Design Techniques:** Networks for Embedded and its Design, Internet Enabled System, Introduction to Sensor Network, Design Methodologies, Requirement Analysis, System Analysis and Quality Assurance.	20

Text Books:

- Wolf, Wayne (2008), “Computers as Components - Principles of Embedded Computing System Design”, Elsevier.

Reference Books:

- Raj Kamal (2011), “Embedded Systems–Architecture, Programming & Design”, Tata McGraw Hill.
- David A. Simon (1999), “An Embedded Software Primer”, Pearson Education.
- DanielW.Lewis (2012), “Fundamentals of Embedded Software Where C and Assembly Meet”, 2nd Edition, Pearson College Division.
- James K. Peckol (2012), “Embedded Systems: A Contemporary Design Tool”, Wiley India.

Outcome: After completion of course, student will be able to:
CO1: Visualize and analyze the design and behavior of the hardware components.
CO2: Apply knowledge of embedded systems along with some specialization in any area of computer engineering.

BCSE0231: EMBEDDED SYSTEM LAB

Objective: *To enable the students to program, simulate and test the 8085, 8051, PIC 18 and ARM processor based circuits and their interfaces.*

Credits: 01 **L–T–P: 0–0–2**

Module No.	Content	Lab Hours
I/II/III	• ASCII to fixed-point output to LCD • Debugging, oscilloscope fundamentals, logic analyzer, dump profile • Alarm clock, LCD, edge-triggered input interrupts, and SysTick periodic interrupts • IEE802.11Wi-Fi communication, TCP, client-server, IoT • 12-bit DAC, SPI, Music player, audio amp • Introduction to PCB Layout, PCB Artist (paper design only) • Design and Layout of an Embedded System • Software Drivers for an Embedded System • Temperature measurement, ADC, LCD • DC motor control, timer interrupts, PWM output, input capture, integral control • Final Design and Evaluation of Embedded System	12*2=24

Text Books:

- Raj Kamal (2011), “Embedded Systems–Architecture, Programming & Design”, Tata McGraw Hill.

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

After the successful conclusion of this course, students should be able to design embedded systems including hardware/software. interfaces for devices like LCD displays, motors, keyboards, analog sensors and speakers.

Students will be able to make hardware/software. Interfaces for devices like LCD displays, motors, keyboards, analog sensors and speakers.

Head of the Department
Computer Engineering & Applications
Institute of Engineering & Technology
GLA University, Mathura

BCSE0203: INTERNET OF THINGS

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

Credits: 03 **L–T–P: 3–0–0**

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:

- Understand the concepts of Internet of Things
- Analyze basic protocols in IOT
- Design IoT applications in different domain and be able to analyze their performance
- Implement basic IoT applications on embedded platform

BCSE0204: INTERNET OF THINGS AND ITS APPLICATIONS

Objective: *This course introduces the various types of advance technology related to IoT, and their applications, also introduces the methods of interfacing sensors to electronic systems.*

Credits: 03 **Semester II** **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**Developing IoTs** Introduction to Python, Introduction to different IoT tools, developing applications through IoT tools, developing sensor based application through embedded system platform, Implementing IoT concepts with python **Challenges in IoT** Design challenges, Development challenges, Security challenges, Other challenges. **Advance Technology in IoT and its application** IoT with Big data and advanced Analytics-Big Data Metadata Management in Smart Grids: semantic inconsistencies role of metadata, Sensor deployment & Node discovery, Data aggregation & dissemination	20
II	**Cloud Computing:** AWS Cloud, Sensor-Cloud, Connected Vehicles. Introduction & Application to Industrial IoT. How to handle IoT devices using voice interface like Alexa/Goggle Home **Fog Computing**: A Platform for Internet of Things and Analytics, Definition-Characteristics-Application Scenarios -Issues -Fog Computing and Internet of Things-Pros and Cons-Myths of Fog Computing -Need and Reasons for Fog Computing Fog Computing and Edge Computing-IoT, FOG, Cloud-Benefits **Case study:** Agriculture, Healthcare, Activity Monitoring, Wearable Sensing, Smart Cities and Smart Homes. Security Of IoT devices	20

Text Books:

Perry Lea (2018), "Internet of Things for Architects: Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics, and security", Packt Publishing

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

- CO1: Understand the concepts of Internet of Things.
- CO2: Analyze basic protocols in IOT.
- CO3: Design IoT applications in different domain and be able to analyze their performance.
- CO4: Implement basic IoT applications on embedded platform.

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: 0–0–2**

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 (2018), "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)"

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

- CO1: Students will be able to identifying the technical problems and be proficient in the analysis, design, test, and implementation of instrumentation and control systems utilizing appropriate software and hardware tools and devices.
- CO2: Understand the functionality of system components/devices for the automation of processes.

BCSE0241: IOT BASED PROJECT

Objective: *Identify research opportunities in IoT technology, applications and services, focusing on the context.*

Credits: 02 **L–T–P: 0–0–0**

1. IoT Weather reporting system using Raspberry pi:

Objective: This project helps you in weather reporting. We used many sensors that can help in sensing all-weather monitoring parameters and RaspberryPi gathers all the information from them and uploads it to the Things peak cloud, which is a wonderful application platform for IoT, where you can see the report that sensors send you.

2. IoT Connected Healthcare Applications:

Objective: The aim of developing this project is to monitor the health condition of a person anywhere and send the information to a specialized doctor to check up. Using this frequency of visiting doctor decreases. We developed a project using Wearable sensors with solar harvesting and Bluetooth low energy transmission that creates a wireless body area network (WBAN). Using this project, you can detect the heartbeat, Blood pressure, hemoglobin content etc., All these reports can be used for analyzing a person's health.

3. IoT Based Intelligent Traffic Management System:

Objective: As the population increases, the effort on Traffic is also increasing. In many metropolitan cities, it is being difficult for traffic Management team to control it. So, IoT can help the traffic system to be automated and monitor the traffic status using the internet.

4. IoT Based Smart Parking System Using RFID:

Objective: With the increase in usage of vehicles, Parking space is very difficult to find. We need to use the human effort to check where to park our vehicle. To overcome this problem, we developed Smart Parking system that shows you space to park your vehicle.

5. Smart Irrigation System Using IoT:

Objective: We have developed a Smart Irrigation system using the concept of IoT. As we know, farming has different stages, to help these farmers in understanding the climatic conditions, moisture content in the soil and to make their decisions easy we have developed Smart Irrigation System using IoT.

6. IoT Based Baby Monitoring System Using Raspberry Pi:

Objective: This project is to develop a smart umbrella system using IOT, which can measure rainfall and it can predict the weather condition with the notion in your smart phone. It is about the real-time weather condition and it can be design using Raspberry Pi.

7. IoT Based Fire Detection System Using FPGA:

Objective: Building a cloud based Fire monitoring system is very important to reduce the cost of maintaining servers, to avoid data losses and to make the access easy with multiple internet connected devices (computer, tablet, mobile phone) at the same time anywhere in the world. Using IOT. Here, we are going to design a fire detection system and data to be upload it to a Thing Speak cloud using FPGA

8. Cloud based Smart Energy meter using FPGA:

Objective: Efficient use of energy becomes more crucial when increase in the cost of energy is observed. Since energy management is required to define the amount of consumed energy in a specific period, utilization of Energy Meters is essential. It is possible to measure the consumed energy by using a simple energy meter. But sometimes the limited functionality of these meters restricts their area of application; especially in inaccessible positions or in the situations where visibility of the meter is poor, it is not possible to use such an appliance.

9. Data logger of Brain parameters in Google sheets using Mind wave mobile with Raspberry Pi:

Objective: Brain-Computer Interface technology is innovated to the analysis of the brain to predict the human thoughts or to control applications using the brain. As the next stage of analysis based on Machine Learning to our Brain, we need data logs of our Brain parameters, so we need the logs of data, which should contain Brain values. This can be done by using Raspberry Pi to take values from the Brain and to uploading the Brain values in Raspberry Pi.

10. MQTT based Monitoring System for Coal Mine using Raspberry Pi:

Objective: Due to global warming and climate, changes there are challenging situations in coalmine. To reduce the cost as well as to improve the productivity along with product quality the automation in the field of coalmine is necessary, which will also reduce the mine worker's efforts. This project proposes a design of a IOT system with MQTT protocol, by the help of Raspberry pi controller which is able to monitor the temperature, humidity, gas and status of smoke in an underground mine.

11. Industry Monitoring with data logger in Google sheet using Raspberry Pi:

Objective: This Raspberry Pi industry monitoring system which having some sensors, taking sensor values and converting using Analog to Digital, then for data logging which will be helpful to review the industry parameter, instead of data log manually, every data values will be uploaded to Google sheets which already get authenticated and by generating JSON file for every individual.

12. IoT Based Transmission Line Monitoring System Using Raspberry Pi:

Objective: The aim of this project is to decrease power consumption for delay sensitive and delay insensitive traffic. A fuzzy rule table is constructed with prediction of average inter arrival time of frame based on outcome sleep time and power is evaluated. Proposed method offer effective power efficiency than previous inter arrival time method using Raspberry Pi.

BCSE0205: DISTRIBUTED SYSTEMS

Objective: *The objective is to study the fundamental principles of distributed system and protocols. This course addresses communication, process, naming, synchronization, replication, and fault tolerance in distributed system.*

Credits: 03 **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**Introduction:** Overview to Distributed Systems, Characteristics, Examples of Distributed Systems, Issues and Challenges, System Models. **Theoretical Foundation:** Limitation of DS, Introduction to Time and Global States, Clocks, Events and Process States, Synchronizing Physical Clocks, Logical Time and Logical Clocks, Global States, Introduction to Distributed Operating System – Layered Architecture. **Distributed Deadlock Detection:** Distributed Mutual Exclusion – Introduction, Classification, Algorithms and their performances, Deadlock Prevention, Avoidance, Detection and Resolution, Deadlock Detection (Centralized & Distributed) Algorithms and their performances. **Agreement Protocols:** Introduction, Classification, Solution to Byzantine Agreement Problem and Applications. **Distributed Objects:** Introduction to Interprocess Communications, Client Server Communication, Group Communication, Communication between Distributed Objects, Remote Procedure Call, Events and Notifications, Case Studies – Java RMI and CORBA.	20
II	**Distributed File System:** Introduction to DFS, File Service Architecture, Sun Network File System, Andrew File System, Name Services, DNS, Directory and Directory Services. **Fault Tolerance:** Issues in Fault Tolerance, Commit Protocols, Voting Protocols, Dynamic Voting Protocols. **Failure Recovery in Distributed Systems:** Concepts in Backward and Forward Recovery, Recovery in Concurrent Systems, Recovery in Distributed Database Systems. **Load Balancing:** Introduction, Issues, Components and load distributing algorithms – Sender – Initiated, Receiver – initiated, symmetrically – initiated and Adaptive algorithm. **Overview of Security Techniques:** Introduction to Cryptographic Algorithms, Digital Signatures, Replication, System Model and Group Communications, Active and Passive Transactions with Replicated Data.	20

Text Book:

- MukeshSinghal and Niranjan Shivaratri (2017), “Advanced Concepts in Operating Systems”, McGraw-Hill Series in Computer Science.

Reference Books:

- A.S. Tanenbaum and M. Van Steen (2004), “Distributed Systems”, Pearson Education.
- George Coulouris, Jean Dollimore and Tim Kindberg (2011), “Distributed Systems Concepts and Design”, 5th Edition, Pearson Education Asia.

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

CO1: Understand basic elements and concepts related to distributed system technologies; and core architectural aspects of distributed systems.

CO2: Identify the advantages and challenges in designing distributed algorithms for different primitives like mutual exclusion, deadlock detection, and agreement.

CO3: Understand principle behind IPC and use various interposes communication techniques, such as remote method invocation, remote events for building distributed systems.

CO4: Introduce the concepts of distributed file system with its architecture and components along with case studies.
CO5: Distinguish the main failure types in a Distributed System and specify algorithms for achieving fault tolerance and error recovery within such a system.
CO6: Understand how balancing of resources is done; issues, components and algorithms for load balancing in distributed environment.

Head of the Department
Computer Engineering & Applications
Institute of Engineering & Technology
GLA University, Mathura

BCSE0206: PARALLEL ALGORITHM

Objective: *This course aims to introduce the concept of designing algorithms suitable for implementation on parallel computers. The focus will be on the algorithmic side. In particular, the emphasis will be on studying different parallel techniques and using them to design scalable parallel algorithms for a variety of problems and architectures.*

Credits: 03 **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**Introduction:** Need for parallel computers, Models of computation, analyzing parallel algorithms, Expressing parallel algorithms **Algorithm Models:** Sequential model, need of alternative model, parallel computational models such as PRAM, LMCC, Hypercube, Cube Connected Cycle, Butterfly, Perfect Shuffle Computers, Tree model, Pyramid model, Fully Connected model, CRCW-CREW, EREW models **Performance Measures ofParallelAlgorithms:** Speed-up and efficiency of PA, Cost- optimality, an example to illustrate Cost-optimal algorithms- such as summation, Min/Max on various models.	20
II	**Decomposition& Mapping techniques:** Database query processing, 15-puzzle problem, Parallel discrete event simulation. **Sorting:** Hyper quick sort, Bitonic merge sort, Enumeration sort (sorting on the CRCW model, CREW model and ERE Wmodel) **Searching and selection:** Searching on a sorted sequence (EREW, CREW, CRCW), Sequential selection algorithm, Parallel selection algorithm (EREW parallel solution) **Graph algorithms:** Graph coloring, Minimal spanning tree, Shortest path algorithm	20

Text Books:

- M.J. Quinn (1987), "Designing Efficient Algorithms for Parallel Computer", McGraw-Hill, Inc, ISBN:0-07-051071-7

Reference Books:

- S.G. Akl (1989), "Design and Analysis of Parallel Algorithms". Prentice-Hall, Inc., ISBN:0-13-200056-3
- S.G. Akl (1985)," Parallel Sorting Algorithm" by Academic Press
- Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar (2003), "Introduction to Parallel Computing", Second Edition, Addison Wesley, ISBN: 0-201-64865.

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

CO1: Explain and derive the complexity of algorithms for basic and collective communication operations.

CO2: Apply different methods and performance measures to analyze algorithms with respect to cost and scalability, including using the so-called iso-efficiency function.

BCSE0233: PARALLEL ALGORITHM LAB

Objective: *The goal of the course is to study the mathematical models, methods and technologies of parallel programming for multiprocessor systems.*

Credits: 01 **L–T–P: 0–0–2**

Module No.	Content	Lab Hours
I/II/III	• Parallel Programming with MPI • Parallel Programming with OpenMP • Lab for estimating the parallel method efficiency with the use of the ParaLab system • Lab for developing the parallel algorithms and programs • Lab works for parallel solving partial differential equations • Lab works for studying the parallel method libraries • Lab works for parallel solving the problem of multidimensional multiextremal optimization	12*2=24

Text Books:

- M.J. Quinn (1987), “Designing Efficient Algorithms for Parallel Computer”, McGraw-Hill, Inc, ISBN:0-07-051071-7

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

CO1: Doing the work each student should carry out the complete cycle of the parallel software developmenfor solving the problems of sufficiently high level of complexity.

BCSE0207: CLOUD COMPUTING

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: 03 **Semester - VI** **L–T–P: 3–0–1**

Module No.	Content	Teaching Hours
I	**Overview of Cloud Computing -** Brief history and Evolution of Cloud Computing, Traditional vs. Cloud Computing, Importance of Cloud Computing, Benefits and Challenges of Cloud Computing, Cloud computing vs. Cluster computing vs. Grid computing, Role of Open Standards Cloud **Computing Architecture:** Cloud computing stack Comparison with traditional computing architecture (client/server), Services provided at various levels, How Cloud Computing Works, Role of Networks in Cloud computing, protocols used, Role of Web services Service Models (XaaS) Infrastructure as a Service(IaaS), Platform as a Service(PaaS), Software as a Service(SaaS) Deployment Models Public cloud, Private cloud, Hybrid cloud, Community cloud. **Infrastructure as a Service(IaaS):** Introduction to virtualization, Different approaches to virtualization, Hypervisors, Machine Image, Virtual Machine(VM) Resource Virtualization Server, Storage, Network Virtual Machine (resource) provisioning and manageability, storage as a service, Data storage in cloud computing (storage as a service) Case Study: Amazon EC2. **Platform as a Service(PaaS):** Introduction to PaaS What is PaaS, Service Oriented Architecture (SOA) Cloud Platform and Management Computation Storage, Case study: Microsoft Azure as PaaS, Introduction, Service Offered, Creation of DB instance.	20
II	**Software as a Service (SaaS):** Introduction to SaaS, Web services, Web 2.0, Web OS, Open SaaS, SaaS with SOA Overview of Multi-Cloud Management Systems - Explain concept of multicloud management, Challenges in managing heterogeneous clouds, benefits of multi-cloud management systems. **Overview of Cloud Security -** Security concerns in Traditional IT, Challenges in Cloud Computing in terms of Application, Server, and Network Security. Security Concepts in VM, Abuse and Nefarious Use of Cloud Computing, Insecure Interfaces and APIs (Malicious Insiders, Shared Technology Issues, Data Loss or Leakage, Account or Service Hijacking, Unknown Risk Profile), Attacks in Cloud Computing **Cloud Security:** Infrastructure Security, Network level security, Host level security, Application level security Data security and Storage Data privacy and security Issues, Jurisdictional issues raised by Data Location Identity & Access Management, Access Control, Trust, Reputation, Risk, Authentication in cloud computing, IAM User, Groups and their Roles. **Service Management in Cloud Computing:** Service Level Agreements(SLAs), Billing & Accounting, Comparing Scaling Hardware: Traditional vs. Cloud, Economics of scaling: Benefitting enormously Managing Data Looking at Data, Scalability & Cloud Services Database & Data Stores in Cloud Large Scale Data Processing.	20

Text Books:

- Raj Kumar Buyya, James Broberg, Andrezei M. Goscinski (2011), Cloud Computing: Principles and• paradigms.

Reference Book:

- Anthony T. Velte, Toby J. Velte, and Robert Elsenpeter (2010) Cloud Computing: A Practical Approach,
- McGraw Hill. Rittinghouse, John, W, Cloud computing: Implementation, management and security.
- Barrie Sosinsky (2011), Cloud Computing Bible, Wiley.
- Rhoton, John, Cloud Computing Architected: Solution Design Handbook.
- Krutz, Ronald L.; Vines, Russell Dean, Cloud Security, A comprehensive Guide to Secure Cloud Computing.

Outcome: After successful completion of this student will be able to:

CO1: Articulate the main concepts, key technologies, strengths and limitations of cloud• computing and the possible applications for the state of the art cloud computing.
CO2: Analyze and study various components of AWS such as EC2, S3 and IAM.
CO3: Identify the architecture and infrastructure of cloud computing including SaaS, PaaS and IaaS, public cloud, private cloud and hybrid cloud.
CO4: Will be able to understand and define the multi-tenant cloud architecture, its advantage and• requirements.
CO5: Will be able to explain the core issues of cloud computing such as security and service• management.

BCSE0208: CLOUD COMPUTING AND VIRTUALIZATION

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: 03 **Semester III** **L–T–P : 3–0–0**

Module No.	Content	Teaching Hours
I	**Overview of Cloud Computing -** Brief history and Evolution of Cloud Computing, Traditional vs. Cloud Computing, Importance of Cloud Computing, Benefits and Challenges of Cloud Computing, Cloud computing vs. Cluster computing and Grid computing **Cloud Computing Architecture:** Cloud computing stack Comparison with traditional computing architecture (client/server), Services provided at various levels, How Cloud Computing Works, Role of Networks in Cloud computing, Role of Web services, Service Models (XaaS), Infrastructure as a Service(IaaS) , Platform as a Service(PaaS), Software as a Service(SaaS), Deployment Models, Public Cloud, Private Cloud, Hybrid Cloud, Community Cloud. **Infrastructure as a Service (IaaS)**: Introduction to virtualization, Different approaches to virtualization, Hypervisors and its types, Virtual Machine(VM), Resource Virtualization, Server Virtualization, Storage Virtualization, Network Virtualization, Virtual Machine Resource Provisioning and Manageability, Data storage in cloud computing (storage as a service), VM migration techniques, - Case Study: Amazon EC2. **Platform as a Service(PaaS):** Introduction to PaaS, Service Oriented Architecture (SOA), Cloud Platform and Management Computation Storage. **Software as a Service (SaaS:)** Introduction to SaaS, Web services, Web 2.0, Web OS, Open SaaS, SaaS with SOA.	19
II	**Load balancing**: Types of Load Balancing Scenarios in Cloud Computing Environment, Static, Dynamic and Centralized algorithms, Introduction to Open Stack, Horizon Web Interface, Using Keystone Identity Service, Swift Object Storage Service, Glance Image Service, Cinder Block Storage Service, Neutron Networking Service, and Nova Compute and Controller. **Overview of Cloud Security -** Security concerns in Traditional IT, Challenges in Cloud Computing in terms of Application, Server, and Network Security. Security Concepts in VM, Abuse and Nefarious Use of Cloud Computing, Insecure Interfaces and APIs (Malicious Insiders, Shared Technology Issues, Data Loss or Leakage, Account or Service Hijacking, Unknown Risk Profile), Attacks in Cloud Computing. **Cloud Security**: Infrastructure Security, Network level security, Host level security, Application level security Data security and Storage Data privacy and security Issues, **Energy Efficiency in Clouds**: Data Center Power Consumption, Green Data Centers. **Overview of Multi-Cloud Management Systems -** Explain concept of multi-cloud management, Challenges in managing heterogeneous clouds, benefits of multi-cloud management systems.	20

Text Book:

- Raj Kumar Buyya, James Broberg, Andrezei M. Goscinski (2011), Cloud Computing: Principles and paradigms.

Reference Books:

- Anthony T. Velte, Toby J. Velte, and Robert Elsenpeter (2010) Cloud Computing: A Practical Approach, McGraw Hill.
- Rittinghouse, John, W, Cloud computing: Implementation, management and security.

- Barrie Sosinsky (2011), Cloud Computing Bible, Wiley.
- Rhoton, John, Cloud Computing Architected: Solution Design Handbook.

- Krutz, Ronald L.; Vines, Russell Dean, Cloud Security, A comprehensive Guide to Secure Cloud Computing.

Outcome: After completion of Lab, student will be able to:
CO1: Articulate the main concepts, key technologies, strengths and limitations of cloud computing and the possible applications for the state of the art cloud computing.

CO2: Analyze and study various components of AWS such as EC2, S3 and IAM.

CO3: Identify the architecture and infrastructure of cloud computing including SaaS, PaaS and IaaS, public cloud, private cloud and hybrid cloud.

C04: Will be able to understand the concept of load balancing and auto scaling in AWS.

CO5: Will be able to explain the core issues of cloud computing such as security and energy management.

CO6: Will be able to understand and define the multi-tenant cloud architecture, its advantage and requirements.

CO7: Will be able to critically analyze case studies to derive the best practices model to apply when developing and deploying cloud based applications

BCSE0234: CLOUD COMPUTING LAB

Objective: *This lab aims to understand the concept of cloud and virtualization by the help of VMware.*

Credits: 01 **L–T–P: 0–0–2**

Module No.	Content	Lab Hours
I/II/III	• 1. a) Introduction to Packet Tracer. • b) Network Topologies. (Including explanation of Simple PDU &amp; Complex PDU.) • 2. Connecting 3 netwoks using routers. Also, configure DHCP and DNS server. • 3. Configuration of different Application services (SMTP, FTP, HTTP, TFTP, DHCP & DNS) • 4. Configuration of Vlan and Inter- Vlan Routing. • 5. Configure GRE over IP tunnel (VPN). • 6. Static NAT configuration. • 7. Configure Wireless network. • 8. Configure different IoT devices. • 9. Management of cloud resources using Cloud Analyst. • 10. Simulation of large scale Cloud computing data centers with Cloud Analyst • 11. Study on VMware • a. Creating a VM • b. Networking on VM • c. Merging and splitting disk on VM • d. Cloning the guest OS • e. Deploying VM with template • f. Creating Snapshots • g. Managing Users, Groups, Permissions and Roles • 12. Crating a EC2 instance on AWS • 13. Configuration of db in AWS. • 14. Creation of S3 bucket with single IAM user in AWS. • 14. Creating a AWS infrastructure for an organization on cloudcraft	12*2=24

Reference Book:

- Raj Kumar Buyya, James Broberg, Andrezei, M. Goscinski (2011), Cloud Computing: Principles and paradigms.

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

CO1: Understanding about the virtualization by the help of VMware
CO2: Understanding of CISCO packet tracer to build a cloud network infrastructure.

BCSE0242: CLOUD COMPUTING PROJECT

Credits: 02 **L–T–P: 0–0–-**

1. Developing a warehouse application using force.com cloud platform:

Objective: Force.com platform is a cloud service fits in the Platform as a Service model. Developers can leverage the runtime time environment and develop scalable web applications using Force.com platform. Design a warehouse application using Multi-Tenant MVC architecture of Force.com. The application should consist of all modules required to automate the warehouse and also it be able to accessible via android mobile.

2. Performance benchmarking of Type I Hypervisors:

Objective: The virtualization is carried out by the software layer called as the Hypervisor or Virtual Machine Monitor (VMM). Hypervisor is widely used in cloud datacenters. Benchmark is the measurement of best practice performance. Benchmarking is very essential term for the discovery of the best performance given by the particular system. Benchmarking can provide you the external references and the best practices on which to base your evaluations and to design your system processes which can be very useful in finding the gaps in the system to achieve the desired performance. Your target hypervisors are Xen, VMware Esxi (Free edition) and Hyper V (Community Edition). Performance analysis of network, cuprum, I/O need to carried out.

3. AgriERP: Agriculture Enterprise resource planning on cloud:

Objective: Agriculture management is complex, requires intimacy with the environment supporting the farm and must meet the specific needs of the farm family it supports in the community where they live. So this project is based on carrying and management of farming activities. By this project you are intended to reduce management worries and regular information of paper written form. Also you provide provision to farmers which give information regarding market update, weather report, and equipment information. Your ultimate goal is to enhance use of farming resource by using modern technology. Application can be developed using Force.com cloud platform.

4. Performance benchmarking of Type II Hypervisors:

Objective: The virtualization is carried out by the software layer called as the Hypervisor or Virtual Machine Monitor (VMM). Hypervisor is widely used in cloud datacenters. Benchmark is the measurement of best practice performance. Benchmarking is very essential term for the discovery of the best performance given by the particular system. Benchmarking can provide you the external references and the best practices on which to base your evaluations and to design your system processes which can be very useful in finding the gaps in the system to achieve the desired performance. Your target hypervisors are Virtual Box, VMWare workstation as a hosted hypervisors and KVM. Performance analysis of network, cuprum, I/O need to carried out.

5. Configure a devstack environment in personal computer and create a image with LAMP stack and deploy web an application:

Objective: DevStack is an opinionated script to quickly create an OpenStack development environment. It can also be used to demonstrate starting/running OpenStack services and provide examples of using them from a command line. Your objective is to deploy private cloud using devstack and create a necessary network components and projects. Than configure an image with (LAMP), Linux-Apache-MySQL-PHP. Develop a web application using html, PHP and MySQL database and deploy it on devstack.

6. Develop a three-tier web application using Web 3.0 Technologies and deploy it on AWS EC2 service with auto scaling property:

Objective: Amazon web services is a Public cloud provider consists of many services like storage, database, compute and container services. It also consists of advanced services like AI, Image voice recognition and other allied services. The high end three tier applications are with huge complex structure like e-commerce sites, social media sites and other variety of application where quality of user experience is most important. These types of applications can leverage the clouds on the fly based on the resource demands. Your objective is to create a web application using latest web technologies. Deploy the application by provisioning the EC2 instances of type micro. Perform auto scaling by generating high http network traffic using http-perf.

7. Performance benchmarking of Type I Hypervisors and containers:

Objective: The virtualization is carried out by the software layer called as the Hypervisor or Virtual Machine Monitor (VMM). Hypervisor is widely used in cloud datacenters. Benchmark is the measurement of best practice performance. Benchmarking is very essential term for the discovery of the best performance given by the particular system. Benchmarking can provide you the external references and the best practices on which to base your evaluations and to design your system processes which can be very useful in finding the gaps in the system to achieve the desired performance. Your target hypervisors are Virtual Box, VMWare workstation as a hosted hypervisors and KVM. Performance analysis of network, CPU, RAM I/O need to carried out.

8. Design and deploying of web application in Docker container:

Objective: Docker is an open-source project that automates the deployment of applications inside software containers. Docker containers wrap up a piece of software in a complete file system that contains everything it needs to run: code, runtime, system tools, and system libraries – anything you can install on a server. Docker provides an additional layer of abstraction and automation of operating-system-level virtualization on Windows and Linux. Docker uses the resource isolation features of the Linux kernel such as cgroups and kernel namespaces, and a union-capable file system such as Overlays and others to allow independent "containers" to run within a single Linux instance, avoiding the overhead of starting and maintaining virtual machines. The objective of proposed study is to deploy the containers on Docker. In that one container is a web server which has a web application which may be of any language such as java, python etc. and another containers consists of database of application. We had to connect these two containers and access the application of first container in third container with database connectivity.

9. Minisat: Open source provisioning, managing and monitoring tool for virtual machines:

Objective: The web interface allows the user to interact with the underlying infrastructure to manage its provisions. Technologies, which can be used to do so, are HTML, CSS, and Django. Application Server, which is the master node, will be used to provision the requirements of a user by using libvirt API. It will be able to monitor the status of VMs. Worker Node is the physical machines on which the VM will be provisioned.

10. Develop a three-tier web application using Web 3.0 Technologies and deploy it on Google compute service with auto scaling property:

Objective: Google compute services is a Public cloud provider consists of many services like storage, database, compute and container services. It also consists of advanced services like AI, Image voice recognition and other allied services. The high end three tier applications are with huge complex structure like e-commerce sites, social media sites and other variety of application where quality of user experience is most important. These type of applications can leverage the clouds on the fly based on the resource demands. Your objective is to create a web application using latest web technologies. Deploy the application by provisioning the EC2 instances of type micro. Perform auto scaling by generating high http network traffic using http-perf.

11. Performance analysis of load balancing algorithms in cloud heterogeneous environment:

Objective: The pervasiveness and power of cloud computing alleviates some of the problem's application administrators face in their existing hardware and software environments. However, the rapid increase in scale, dynamicity, heterogeneity, and diversity of cloud resources necessitates having expert knowledge about the way resources are scheduled. The project gives an insight on key resource types of cloud, its resource orchestration along with the analysis of the performance of algorithms.

12. Cloud Log Forensics: Foundations, State of the Art, and Future Directions:

Objective: Cloud log forensics (CLF) mitigates the investigation process by identifying the malicious behavior of attackers through profound cloud log analysis. However, the accessibility attributes of cloud logs obstruct accomplishment of the goal to investigate cloud logs for various susceptibilities. Accessibility involves the issues of cloud log access, selection of proper cloud log file, cloud log data integrity, and trustworthiness of cloud logs. Therefore, forensic investigators of cloud log files are dependent on cloud service providers (CSPs) to get access of different cloud logs.

13. Adaptive Threshold-Based Approach for Energy-Efficient Consolidation of Virtual Machines in Cloud Data Centers:

Objective: The rapid growth in demand for computational power driven by modern service applications combined with the shift to the Cloud computing model have led to the establishment of large-scale virtualized data centers. Such data centers consume enormous amounts of electrical energy resulting in high operating costs and carbon dioxide emissions. Dynamic consolidation of virtual machines (VMs) and switching idle nodes off allow Cloud providers to optimize resource usage and reduce energy consumption. However, the obligation of providing high quality of service to customers leads to the necessity in dealing with the energy-performance trade-off. A novel technique for dynamic consolidation of VMs based on adaptive utilization thresholds, which ensures a high level of meeting the Service Level Agreements (SLA). We validate the high efficiency of the proposed technique across different kinds of workloads using workload traces from more than a thousand Planet Lab servers.

14. An SLA-based Admission Control for a Software-as-a-service Provider in Cloud Computing Environments:

Objective: Software as a Service (SaaS) provides access to applications to end users over the Internet without upfront investment in infrastructure and software. To serve their customers, SaaS providers utilize resources of internal data centres or rent resources from a public Infrastructure as a Service (IaaS) provider. In-house hosting can increase administration and maintenance costs whereas renting from an IaaS provider can affect the service quality due to its variable performance. To overcome these limitations, we propose innovative admission control and scheduling algorithms for SaaS providers to effectively utilize public Cloud resources to maximize profit by minimizing cost and improving customer satisfaction level.

15. The Aneka Platform and Qos-driven Resource Provisioning for Elastic Applications on Hybrid Cloud:

Objective: How Aneka, a platform for developing scalable applications on the Cloud, supports a vision of QOS by provisioning resources from different sources and supporting different application models. We highlight the key concepts and features of Aneka that support the integration between Desktop Grids and Clouds and present an experiment showing the performance of this integration.

16. Data Leakage Detection Using Cloud Computing:

Objective: This project presents a data leakage detection system using various allocation strategies and which assess the likelihood that the leaked data came from one or more agents. For secure transactions, allowing only authorized users to access sensitive data through access control policies shall prevent data leakage by sharing information only with trusted parties and also the data should be detected from leaking by means of adding fake record`s in the data set and which improves probability of identifying leakages in the system. Then, finally it is decided to implement this mechanism on a cloud server.

17. Suboptimal Mechanism for load balancing in cloud:

Objective: Load balancing is important and major issue for cloud networks due to its elastic nature, user's unpredictable behavior and difficulty faced to foresee the request issued to the server. In order to efficiently utilize cloud resources, a new methodology for dynamic resource allocation based on the suboptimal mechanism of aggregation i.e. GIT (greedy incremental tree) is needed. The proposed technique improves path sharing at time of migration during load balance process. A cloud workload network is constructed based on-Greedy heuristics having the characteristics to build incrementally diverse set, selecting one compute node at a time to maximize diversity measure. The virtual machine manager can rely on the rules and can make efficient decision for optimal provisioning of virtual machine based on the aggregated route.

18. Towards Many-Core Processor Simulation on Cloud Computing Platforms:
Objective: There is a need for a many-core simulator that can be simple to use and learn from for students, dynamic and capable of emulating large systems for researchers, and flexible with fast turnover for industry designers. At the same time, as many-core systems have been becoming popular and complex, and hence their design, the long-standing field of Cloud Computing has become more prevalent and feasible to use. Therefore, the projects deal with the designing of many core processors on simulator.

19. Consumer Acceptance of Cloud Computing Based Gaming:
Objective: There is a lack of understanding on how the users perceive cloud gaming. Investigating the factors that affect the users' acceptance of cloud gaming is crucial in deter-mining the future of this new platform. A lack of awareness regarding these factors may lead to the users‟ rejecting the new technology. This is a research-based project, conducted through an inductive approach, using a survey as a research method, where the primary data comes from a structured interview. A descriptive study is conducted in order to obtain the full set of user-related features of cloud gaming. After that, the technology acceptance model is utilized in order to find out the user perception of these features.

20. Application performance Isolation in Cloud:
Objective: Performance isolation is the desirable thing in virtual machine based infrastructure to meet Service Level Objectives. Many experiments in this area measure the performance of applications while running the applications in different domains, which gives an insight into the problem of isolation. We run different kind of benchmark s simultaneously in virtual environment to evaluate the isolation strategy provided by the hypervisor.

21. Optimistic Synchronization of Parallel Simulations in Cloud Computing Environments:
Objective: Cloud computing offers the potential to make parallel discrete event simulation capabilities more widely accessible to users who are not experts in this technology and do not have ready access to high performance computing equipment. Services hosted within the cloud and can potentially incur processing delays due to load sharing among other active services, and can cause optimistic simulation protocols to perform poorly. This project proposes a mechanism termed the Time Warp Straggler Message Identification Protocol (TW-SMIP) to address optimistic synchronization and performance issues associated with executing parallel discrete event simulation in cloud computing environments

SYLLABUS

OF

PROGRAM ELECTIVE

BOUQUET: DEVELOPMENT TOOLS AND TECHNOLOGIES

DEPARTMENT OF COMPUTER ENGINEERING & APPLICATIONS

Under

Choice Based Credit System (CBCS)

Program Elective

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE- REQUISITES
			L	T	P	J			
Bouquet: Bouquet Development Tools and Technologies									
Theory									
1.	BCSE0251	Full Stack Using Scripting Technologies	3	0	0	0	3	3	
2.	BCSE0252	Full Stack using Node JS	3	0	0	0	3	3	
3.	BCSE0253	Full Stack using C#.net	3	0	0	0	3	3	
4.	BCSE0254	PHP - Scripting Language	3	0	0	0	3	3	
5.	BCSE0255	Digital Marketing and Transformation	3	0	0	0	3	3	
Labs									
1.	BCSE0281	Full Stack Using Scripting Technologies Lab	0	0	2	0	1	2	
2.	BCSE0282	Full Stack using Node JS Lab	0	0	2	0	1	2	
3.	BCSE0283	Full Stack using C#.net Lab	0	0	2	0	1	2	
4.	BCSE0284	PHP - Scripting Language Lab	0	0	2	0	1	2	
Projects									
1.	BCSE0291	Full Stack Using Scripting Technologies Project	0	0	0	-	2	-	
2.	BCSE0292	Full Stack using Node JS Project	0	0	0	-	2	-	
3.	BCSE0293	Full Stack using C#.net	0	0	0	-	2	-	
4.	BCSE0294	PHP - Scripting Language Project	0	0	0	-	2	-	

BCSE0251: FULL STACK USING SCRIPTING TECHNOLOGIES

Objective: *The objective is to provide a comprehensive study of the frontend. It stresses the strengths of Web, which provide students with the means of writing efficient, maintainable, and portable code.*

Credits: 03 **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**Hyper Text Mark-up Language (HTML5):** Introduction HTML, HTML Basics, HTML Elements, HTML5 Semantic, HTML Attributes, HTML Headings, HTML Paragraph, HTML Styles, HTML Formatting, HTML Quotations, HTML Computer Code, HTML Comments & Colours, HTML CSS, Links and Images, HTML Lists, HTML Blocks, HTML Classes, HTML Layout, HTML Responsive, HTML iframes, HTML JavaScript, HTML Head, HTML Entities and URI Code, HTML Symbols and XHTML, HTML Charset and Forms **Cascading Style Sheets (CSS3):** Introduction CSS3, CSS3 Syntax, CSS3 How To, CSS3 Colours, CSS3 Backgrounds, CSS3 Boarders, CSS Padding, CSS Height/Width, CSS3 Gradients, CSS3 Shadows, CSS3 Text, CSS3 Fonts, CSS3 2D Transforms, CSS3 3D Transforms, CSS Links, CSS Lists, CSS Tables, CSS Box Model, CSS Outline, CSS Display, CSS Max-width, CSS Position, CSS Float, CSS Inline-block, CSS Align, CSS Combinators, CSS Pseudo-class, CSS Pseudo-element, CSS Navigation Bar, CSS Dropdowns, CSS Tooltips, CSS3 Images, CSS Attr Selectors, CSS Forms, CSS Counters, CSS3 , CSS3 Button, CSS3 Pagination, CSS3 Multiple Columns, CSS3 User Interface, CSS3 Box Sizing, CSS3 Filters, CSS3 Media Queries, CSS3 Responsive	35
II	**JavaScript:** JavaScript Scope, JavaScript Events, JavaScript Strings, JavaScript Numbers, JavaScript Math, JavaScript Arrays, JavaScript Boolean, JavaScript Comparisons, JavaScript Conditions, JavaScript Switch, JavaScript Loops, JavaScript Type Conversion, JavaScript RegExp, JavaScript Errors, JavaScript Debugging, JavaScript Hoisting, JavaScript Strict Mode, JavaScript Functions, JavaScript Objects, JavaScript Forms, JavaScript HTML DOM. **Bootstrap:** Introduction to Bootstrap, Bootstrap Basics, Bootstrap Grids, Bootstrap Themes, Bootstrap CSS, Bootstrap JS **jQuery**: Introduction to jQuery, jQuery Syntax, jQuery Selectors, jQuery Events, jQuery Effects, jQuery HTML, jQuery Traversing, jQuery AJAX	30

Text Books:

- HTML 5 Black Book (Covers CSS3, JavaScript, XML, XHTML, AJAX, PHP, jQuery) by DT Editorial Services, 2016
- Learning Web Development with Bootstrap and AngularJS by Stephen Radford,

Outcome: After completion of course, student will be able to:
CO1: Implement interactive web page(s) using HTML, CSS and JavaScript.
CO2: Design a responsive web site using HTML5 and CSS3.
CO3: Demonstrate Rich Web Applications.
CO4: Use newer HTML5 tags with associated CSS instructions to organize information and content.

BCSE0281: FULL STACK USING SCRIPTING TECHNOLOGIES LAB

Objective: *The objective is to provide a comprehensive study of the frontend. It stresses the strengths of Web, which provide students with the means of writing efficient, maintainable, and portable code.*

Credits: 01 **L–T–P: 0–0–2**

Module No.	Content	Lab Hours
I/II	• Implements Basic HTML Tags • Implementation of Table Tag. • Implementation Text Formation • Implementation of Links • Implementation of CSS (All 4 Types) • DHTML(Layer/DIV), • Creating Classes in CSS • Use of Selectors, • Implementation of Menu. • Implement of JavaScript Events. • Implement of JavaScript Control Statements. • Implement of JavaScript Function. • Implementation of Responsive Page. • Implementation of Bootstrap Grid. • Implementation of Bootstrap theme. • Implementation of Bootstrap CSS & JS. • Implementation and use JQuery Selectors. • Implementation and use JQuery Events. • Implementation and use JQuery Effects.	12*2=24

Reference Books:

- HTML 5 Black Book (Covers CSS3, JavaScript, XML, XHTML, AJAX, PHP, jQuery) by DT Editorial Services, 2016
- Learning Web Development with Bootstrap and AngularJS by Stephen Radford

Outcome: After studying the subject, the students will be able to:
CO1: Implement interactive web page(s) using HTML, CSS and JavaScript.
CO2: Design a responsive web site using HTML5 and CSS3.
CO3: Demonstrate Rich Web Applications.
CO4: Use newer HTML5 tags with associated CSS instructions to organize information and content.

BCSE0291: FULL STACK USING SCRIPTING TECHNOLOGIES PROJECT

Objective: *The objective is to provide a comprehensive study of the frontend. It stresses the strengths of Web, which provide students with the means of writing efficient, maintainable, and portable code.*

Credits: 02 **L–T–P-J: 0–0–0–0**

Module No.	Content	Lab Hours
I/II	Students Have to develop one Project of Interactive Web Templates. • Ecommerce Template. • E-Learning Template. • Uni. Template. • Photography shop Templates • Service provider Template. • Event planner Template • Model Portfolio Template • Resume Based template • Property Site Template • Educational Site Template • Industry approved and relevant projects	-

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: 3–0–0**

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: Do programming using Node.Js
CO2: Develop web application using NodeJs, ExpressJs, MongoDB and Angular
CO3: Deploy project.
CO4: Understand project management and code.
CO5: Develop RESTful and MVC based web application

Head of the Department
Computer Engineering & Applications
Institute of Engineering & Technology
GLA University, Mathura

BCSE0282: FULL STACK USING NODE JSLAB

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: 0–0–2**

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: Do programming usingNode.Js
CO2: Develop web application using NodeJs, ExpressJs,MongoDB and Angular
CO3: Deploy your project.
CO4: Understand project management and code.
CO5: Develop RESTful and MVC based web application

Head of the Department
Computer Engineering & Applications
Institute of Engineering & Technology
GLA University, Mathura

BCSE0292: FULL STACK USING NODE JS PROJECT

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: 02 **L–T–P-J: 0–0–0–0**

Module No.	Content	Lab Hours
I/II	Students Have to develop one Project of Interactive Web Templates. • Ecommerce Template. • E-Learning Template. • Uni. Template. • Photography shop Templates • Service provider Template. • Event planner Template • Model Portfolio Template • Resume Based template • Property Site Template • Educational Site Template • Industry approved and relevant projects	-

BCSE0253: FULL STACK USING C#.NET

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: 3–0–0**

Module No.	Content	Teaching Hours
I	**.NET Framework & C#:** Expressions & Primitive Types, Non-Primitive Types, Control Flow, Arrays & Lists, Date & Time, Text, File System, Debug Application, Classes, Association, between Classes, Inheritance - Second Pillar of OOP, Polymorphism: Third Pillar of OOP, Interfaces, C# Advanced Topics & ADO.net. **Collections and Generics:** Introducing Collections, Benefits of Collection Classes, Understanding and using commonly used collections, Generics, Advantages of Generics, Generic Collections **ASP.NET Core:** Introduction to ASP.NET Core, ASP.NET Core First Application, - Controllers & Action Methods, Views, Helpers, Model Binding, - Validations & Data Annotations, State management Techniques, Security, MVC and Entity Framework Core, ASP.NET Core - Web Caching, Routing, Module Development & Web API and JQuery Ajax, Creating a Layout Using Master Pages&, Data bound Controls.	20
II	Mic**rosoft SQL Server:** Introduction to Basic Database Concepts, Introduction to SQL, Introduction to SQL, Joins and Set Operations, Working with Stored Procedures and Functions, Transaction Control Language (TCL) & Backup/Restore. **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. **React.js:** Starting with React, React Components, React State and Props, React Event Handling, Routing in React flux, &. Styling React	20

Text Books:

- Learning Angular for .NET Developers by Rajesh Gunasundaram
- ASP.NET Core Application Development - Building an Application in Four Sprints by James Chambers, David Paquette, Simon Timms
- Querying Microsoft SQL Server 2012 Exam 70-461 Training Kit by Gan B.I
- Introduction to React by Cory Gackenheimer,

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

CO1: Do programming using C#.
CO2: Develop web application using Asp.Net Core
CO3: Deploy your project using Visual Studio.
CO4: Understand project management and code.
CO5: Develop RESTful and MVC based web application
CO6: Getting familiar with EF and M.S. Sql Server Angular & React Js.

BCSE0283: FULL STACK USING C#.NET 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: 0–0–2**

Module No.	Content	Lab Hours
I/II	• WAP to Implement Types in C# • WAP to Illustrate Control Flow Statements • Implements Array and lists • WAP to Read write in files. • Implements Inheritance • Implements Interface • Create a data access layer with Entity Framework Core 2 • Initialize data for testing with EF Core 2 • Build a website with ASP.NET Core 2 • Configure the HTTP pipeline and routing • Creating Tag Helpers • Create and use View Components • 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 • Implementing React Components, • Implementing React Event.	12*2=24

Reference Books:

- Learning Angular for .NET Developers by Rajesh Gunasundaram
- ASP.NET Core Application Development - Building an Application in Four Sprints by James Chambers, David Paquette, Simon Timms
- Querying Microsoft SQL Server 2012 Exam 70-461 Training Kit by Gan B.I
- Introduction to React by Cory Gackenheimer,

Outcome: After studying the subject, the students will be able to:
CO1: Do programming using C#. Develop web application using Asp.Net Core
CO2: Deploy your project using Visual Studio.
CO3: Understand project management and code.
CO4: Develop RESTful and MVC based web application
CO5: Getting familiar with EF and M.S. Sql Server Angular & React Js

BCSE0293: FULL STACK USING C#.NET PROJECT

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: 02 **L–T–P-J: 0–0–0–0**

Module No.	Content	Lab Hours
I/II	Students Have to develop one Project of Interactive Web Templates. • Ecommerce Template. • E-Learning Template. • Uni. Template. • Photography shop Templates • Service provider Template. • Event planner Template • Model Portfolio Template • Resume Based template • Property Site Template • Educational Site Template • Industry approved and relevant projects	-

BCSE0254: PHP - SCRIPTING LANGUAGE

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

Credits: 03 **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**PHP Basics:** Introduction to PHP, Basic Syntax of PHP, Embedding PHP in HTML, Comments, Variables, Constants, Managing Variables, Operators and Operator Precedence and String Manipulation functions. **Conditional Control Structures:** If statement, If- else statement, If- else if statement, Nested If, Switch statement. **Functions in PHP:** Functions, User-Defined function, Call by value and call by references, Understanding variable scope, Global Variables, Static Variables, Include and Require, Built-in functions in PHP. **Arrays:** Arrays and its types in PHP, Accessing Elements of an Array, Modifying Elements of an Array, Functions in array, Array Sorting, Multidimensional Array. **PHP File Handling:** Introduction, File Open, File Creation, writing to files, reading from File, searching a record from a file, Closing a File.	20
II	**Class and Object:** Introduction, Object, Class, Defining Class in PHP, Object in PHP, Usage of this variable, Constructor, Constructor with Parameters. **Exception Handling:** Introduction to Exception, Exception Handling mechanisms, Creating Custom Exceptions, Multiple Catch Blocks, Exception Propagation, Error Handling in PHP. **Form Handling and Session Management in PHP:** Accessing and displaying Form data from different Form components, Differences among $_GET, $_POST and $_REQUEST variables, Session management, Session operations, Session tracking mechanism, Clearing/Modifying data from session, destroying a session, Setting and Retrieving Cookies, uploading a file, displaying its details, restricting various details of a file during upload, checking for errors and reading Error code table. **Database Management:** Introduction to DBMS, SQL Basics, Database connectivity in PHP with MySQL, Executing Queries from frontend. **XML:** Introduction to XML, Parsing XML document using DOM parser, Various operations on XML document using PHP.	20

Text Books:

- PHP Reference Beginner to Intermediate PHP 5 By Mario Lurig
- PHP and MySQL by Mike McGrath,

Outcome: After completion of course, student will be able to:
CO1: Do programming using PHP.
CO2: Develop web application using PHP
CO3: Understand project management and code.
CO4: Develop RESTful and MVC based web application
CO5: Getting familiar with XML and My-Sql.

BCSE0284: PHP - SCRIPTING LANGUAGE LAB

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

Credits: 01 **L–T–P: 0–0–2**

Module No.	Content	Lab Hours
I/II	• Creating simple webpage using PHP • Use of conditional statements in PHP • Use of looping statements in PHP • Creating different types of arrays • Usage of array functions • Creating user defined functions • Creation of files • File manipulation using PHP • Creation of sessions • Creation of cookies • Creating simple table with constraints • Insertion, Updating and Deletion of rows in MYSQL tables • Searching of data by different criteria • Usage of aggregate functions • Working with string, numeric and date functions • Database connectivity in PHP with MySQL • Validating Input	12*2=24

Reference Books:

- PHP Reference Beginner to Intermediate PHP 5 By Mario Lurig
- PHP and MySQL by Mike McGrath,

Outcome: After studying the subject, the students will be able to:
CO1: Do programming use PHP.
CO2: Develop web application using PHP
CO3: Understand project management and code.
CO4: Develop RESTful and MVC based web application
CO5: Getting familiar with XML and My-Sql.

Head of the Department
Computer Engineering & Applications
Institute of Engineering & Technology
GLA University, Mathura

BCSE0294: PHP - SCRIPTING LANGUAGE PROJECT

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

Credits: 02 **L–T–P-J: 0–0–0–0**

Module No.	Content	Lab Hours
I/II	Students Have to develop one Project of Interactive Web Templates. • Ecommerce Template. • E-Learning Template. • Uni. Template. • Photography shop Templates • Service provider Template. • Event planner Template • Model Portfolio Template • Resume Based template • Property Site Template • Educational Site Template • Industry approved and relevant projects	-

BCSE0255: DIGITAL MARKETING AND TRANSFORMATION

Objective: *The objective of this course is to provide knowledge about the digital marketing and the industry trends correspondence to the concepts.*

Credits: 03 **L–T–P: 3–0–0**

Module No.	Content	Lab Hours
I	**Introductionto Digital Marketing** Modern Marketing- How Marketing Works, Fundamentals Channels- Awareness, Consideration & Decision Making, Building Integrated Marketing Plan, Lead Journey- From Prospect to Sales. **Website and Blogging** The Power of Storytelling, Know Your Customer - Market and Content Research, Construct Prospective Buyer Personas & Journeys, Establishing The Content Ideation & Creation Framework, Creative Design Principles, Linking Blogs To Social Network (Conversation Blogs), Measuring And Optimizing Blog Performance, Importance Of Responsive Design, Leverage Landing Pages And Forms To Accelerate Conversion. **Content Promotion** SEO as an Art and as a Science, Ranking Algorithms, Website Audit, Optimizing Digital Assets & Metadata, Decoding Common Paid Media Platforms, Influencer Marketing, Black Hat, White Hat and Grey Hat SEO **Email Marketing** Types of Email (Promo/Trans/NL), ESP Setup & On-boarding, Permission Marketing, Subscriber welcome plan and journey, List segmentation and Management, Personalization and Responsive design, Multivariate Testing, E-commerce Integration, Deliverability and System Reputation Management, System Integrations & Automations.	20
II	**Social Media Marketing** Social Ads Type and their Design Structure, targeting strategy and planning – Laser/Broad, Effective targeting and custom audience set-up, Campaign setup and reporting on various social platforms, Social Split Advertising, Content Calendar, Peremptory traits for Social Advertising PPE, WC and CTW campaigns. **Mobile Marketing** Mobile landscapes for Marketing and Monetization, Conventional Advertising, Millennial Mobile Advertising, Versatile Promotions, Alternative focusing and promotions on Mobile, Push App and Game based promotions, Location evolution with mobile **Marketing Analytics & ROI** Key marketing engagement & ROI metrics, Primer on data science and analytics concepts, Web Traffic nuances, Multi-channel Analytics, Decoding CLV and RFM, Deciphering conversion and goal metrics, implement conjoint analysis & decision tree tactics, Avoiding common analytical pitfalls.	20

Text Book:

- Puneet Singh Bhatia, Fundamentals of Digital Marketing First Edition, Publication Pearson.

Reference Books:

- Ian Dodson, The Art of Digital Marketing: The Definitive Guide to Creating Strategic, Targeted and Measurable Online Campaigns, Publication Wiley India Pvt Ltd.

- Philip Kotler, HermawanKartajaya, IwanSetiawan, Marketing 4.0: Moving from Traditional to Digital, , Publication Wiley India Pvt Ltd.
- Vandana Ahuja, Digital Marketing 1st Edition, Publication Oxford.
- Rohan Yamagishi, Digital Marketing in Asia: A Start-Up Guide for Search Engine Marketing in APAC, Publication R. R. Bowker

Outcome: After learning the course the students should be able to:
CO1: Learn how to use new media such as mobile, search and social networking.
CO2: Learn the measurement techniques used in evaluating digital marketing efforts.
CO3: Understand how and why to use digital marketing for multiple goals within a larger marketing and/or media strategy.
CO4: Understand the major digital marketing channels - online advertising: Digital display, video, mobile, search engine, and social media.
CO5: Learn to develop, evaluate, and execute a comprehensive digital marketing strategy and plan
CO6: Explore the latest digital ad technologies.

BCSO0002: INTRODUCTION TO OBJECT ORIENTED PROGRAMMING
(Open Elective)

Objective: *The objective of the course is to understand the fundamentals the fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.*

Credits: 03 **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**OO Fundamentals:** Need of OO approach, OO Concepts. Characteristics of Object oriented programming. **Basics of Java:** Features of Java, Byte Code and Java Virtual Machine, JDK, Data types, Operator, Control Statements. **Array and String:** Single and Multidimensional Array, String class, Operations on string, Command line argument, Use of Wrapper Class. **Classes, Objects and Methods:** Class, Object, Constructor, new operator, Constructor Overloading, Method Overloading, Recursion, Passing and Returning object form Method, this and static keyword (variable, method, class and package), Access control, modifiers, Nested class, Inner class, Abstract class, Java Standard Libraries. **Polymorphism:** Methodoverloading. **Inheritance and Interfaces:** Use of Inheritance, Inheriting Data members and Methods, constructor in inheritance, Types of Inheritance, super keyword, Final keyword, Creation and Implementation of an interface, Dynamic method dispatch, Comparison between Abstract Class and interface.	20
II	**Multithreaded Programming:** Use of Multithread programming, Thread State Diagram, Thread class methods, Runnable interface, Thread priority. **Exception Handling:** Exception and Error, built in Exception, Use of try, catch, throw, throws and finally, Custom exception. **GUI Programming:** Java Applet, Applet life cycle, Applet Vs Application, Graphics methods, Layout- Flow, Grid, Border, Introduction to AWT Programming, Introduction to Swing, AWT Vs Swing, GUI development in AWT, Swings, Event Delegation Model, Event Handling using Button. **JDBC:** Database Connectivity Model, Types and Roles of Drivers, Database Connectivity Statements, Communicating with Database.	21

Text Books:

- Herbert Schildt, “Java the Complete Reference”, TMH. 8th edition.

References:

- Kathy Sierra & Bert Bates, “Head First Java”, O’Reilly, 2nd Edition.
- Patrick Naughton, “Java Handbook”, Osborne McGraw-Hill.
- Khalid A Mughal, Rolf W Rasmussen, “A Programmer's Guide to Java SE 8 Oracle Certified Associate (OCA)”, Addison-Wesley Professional.

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

CO1: Understand differences between procedures oriented and object oriented approach.
CO2: Understand the relevance of Object Oriented Programming techniques.
CO3: Understand how to write, compile and execute a Java Program.
CO4: Understand the use of polymorphism and Inheritance.
CO5: Understand how to define user exceptions and its uses.
CO6: Understand what is a thread and Multithreading model.
CO7: Understand how to develop a GUI application.

BCSE0603: Information Security Audit & Monitoring

Credits: 02 **Semester - VI** **L–T–P: 2–0–1**

Module No.	Content	Teaching Hours
I	**Introduction to Information security.** Difference between Cyber Security and Information Security. Introduction to governance, risk, and compliance standards/framework **Information security for Business**. Business - Information security alignment(BISA), Requirement of BISA, Methodology of BISA, Industry best practices in BISA. Information Security as a business enabler, Organizational structure of information security professionals, Roles & responsibilities in the information security functions. Introduction to global standards. Information security framework **CASE STUDY**: Eramba - open Governance, Risk and Compliance (GRC) solution **ISO-27001 standard:** Introduction to ISO27001-2013, ISO27001-2013 Requirements, Implementation lifecycle of ISO 27001 standard, Control objectives and controls, Requirements of ISO standard & statement of applicability, Detailed description of Clause 6 and 8, Steps of risk assessment. ISMS framework	13
II	**PCI-DSS standard**: Implementation of PCI-DSS standard, Requirements of PCI-DSS standard, Feature of Eramba > controls catalogue Security controls report, Security services, Support contracts, Business continuity plan(BCP), Security Policies, Policy Exceptions. **CASE STUDY**: RBI Guidelines for Cyber Security Framework **Security regulatory requirements**: Implementation of security regulatory requirements, Security in BFSI, Security in telecom sector, Security in Healthcare sector, IT Act of India **Security Assurance & Audits:** Introduction to security assurance and audit, Auditing and security concepts, Audit methodology, Business Skills for Auditors, Creating audit checklists. **CASE STUDY :** Security Auditing Tools (Netwrix Auditor, Nessus, Acunetix, NetworkMiner, etc)	13

Text Books:

IT - Information Security, Audit and Monitoring - Student Guide V 1.0 (IBM ICE Publication)
Information Security Fundamentals, 2nd Edition by Thomas R. Peltier
Information Security: Principles and Practices, Second Edition by Mark S. Merkow, Jim Breithaupt

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 Riak. **Database Programming using Python**: Database connectivity, Retrieving Data from Database, Parameters Passing, Executemany Methods, Cursor Attributes, Invoke Stored Procedures, Invoke Stored Functions.	26

Text Books:

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

References Books:

- Date C J," *An Introduction to Database Systems*", 8th Edition, Addison Wesley.
- Korth, Silbertz and Sudarshan (1998), "*Database Concepts*", 5th Edition, TMH.
- 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:

- Master the basic concepts and appreciate the applications of database systems.
- Be familiar with the relational database theory, and be able to write relational algebra expressions for queries and design principles for the logical design of databases, including the E-R method and normalization approach.
- Be familiar with the basic issues of transaction processing and concurrency control.
- Be familiar with the various NoSQL database system.

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 (1998), *"Database Concepts"*, 5th Edition, TMH.
- Majumdar& Bhattacharya, *"Database Management System"*, TMH
- Sadalage, P. & Fowler (2012), "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Pearson Education

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

- Ability to create database tables
- Ability to formulate SQL queries based on the problems given
- Ability to apply PL/SQL.
- Ability to create NoSQL databases.
- Ability to connect database using Python program.

BCSE0507: CLOUD BACKUP & DISASTER RECOVERY

Objective: *This course aims to explain various ways to take cloud backups also how to prepare data for backup. This course also covers various scenarios to build a highly available system. Course also explains various reasons about cloud disaster and possible recovery from it.*

Credits: 03 **Semester VII** **L–T–P: 3–0–0**

Module No.	Content	Teaching Hours
I	**Fundamentals**: Objective, Disk Storage, Characteristics and types of Disk Drives: Access Centric & Capacity Centric Drives, Disk Systems: Specifications of LTO- 6, WORM, Automated Tape Library. **Backup**: Recovery Objectives: RPO, RTO, Types of Backup: Full, Incremental, Differential, and Progressive Incremental, Architecture of Backup: Network based backup, D2D2T backup, Network free backup, Server free or Server less backup, NDMP backup. **Data Preparation for Backup**: Data Compression, Data Deduplication, Virtual Tape Library, Archive, Data Protection for a practical world, 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.	19
II	**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. **Availability:** Availability Terminologies, Networking/ Communication Terminologies, Location Designations, Disaster Recovery Planning, Phases of Planning, DR Technology Tree, Virtualization. **Replication:** Local Replication, Remote Replication, Synchronous Replication, Asynchronous Replication, Replication Tools, Software based Tools, Deployment **Topologies:** Two Site Replication, Multi- Site Replication, DR Drill and the DR, Operating System Performance Tuning, Hypervisor Performance Tuning.	20

Reference Books:

- IBM Book (2016), Cloud Backup & DR.
- Robin M Hostings (2017), Planning Cloud-Based Disaster Recovery for Digital Assets: The Innovative Librarian's Guide
- Bryan Strawser (2016), Rethinking Disaster Recovery: The Impact of Cloud Computing (Bryghtpath LLC White Papers Book 2)

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

- Explain the concepts of Cloud backup and how to prepare data for backup.
- Describe how to make highly available system.
- Explain about different reasons for cloud disaster and how to recover from it.

INTRODUCTION TO MACHINE LEARNING (BCSE-0701)

Credits: 03 **L–T–P: 3–0–0**

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. **Statistical Learning**-Meaning, learning based on probability distribution (Normal distribution) **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 Book:

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

Reference Books:

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

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

- CO1: Understand the basic concepts of machine learning including bias-variance tradeoff.
- CO2: Explain the concepts of regression and re-sampling methods.
- CO3: Conceptualize supervise and re-enforcement learning for classification.
- CO4: Understand the ensemble methods for improving classification.
- CO5: Understand relationship of modelling and machine learning.
- CO6: Develop the projects based on machine learning.