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.	BECG0001/ BECG0001	Electrical Engineering / Electronics Engineering	3	1	0	4	4
5.	BCSC0001	C Programming	4	1	0	5	5
PRACTICALS							
6.	BCHS0801/ BPHS0801	Engineering Chemistry/ 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	C 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/ BECG0001	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 & 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.	BCSC00010	Discrete Mathematics	3	1	0	0	4	4	
12.	BCSC0011	Theory of Automata & Formal Language	3	1	0	0	4	4	
13.	BCSC0012	Design & Analysis of Algorithms	3	0	0	0	3	3	Programming , Data Structures
14.	BCSC0013	Compiler Design	3	1	0	0	4	4	Theory of Automata & Formal Language
PRACTICALS									
15.	BCSG0800	Python Programming Lab	0	0	2	0	1	2	
16,	BCSC0800	Computer Programming Lab	0	0	2	0	1	2	
17.	BCSC0801	Object Oriented Programming Lab	0	0	2	0	1	2	Programming Lab
18.	BCSC0802	Database Management System Lab	0	0	2	0	1	2	
19.	BCSC0803	Operating Systems Lab	0	0	2	0	1	2	
20.	BCSC0804	Computer Organization Lab	0	0	2	0	1		
21.	BCSC0805	Data Structures & Algorithms Lab	0	0	2	0	1	2	Programming Lab
22.	BCSC0806	Microprocessors Lab	0	0	2	0	1	2	
23.	BCSC0807	Competitive Programming Lab	0	0	2	0	1	2	Programming , Data Structures
Total			**44**	**7**	**18**	**0**	**60**	**67**	

Program Elective

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE- REQUISITES
			L	T	P	J			
Bouquet: 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	Cybersecurity 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 & 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	4	1	1	Computer Networks

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
PRACTICALS									
6.	BCSE0081	Software Testing Lab	0	0	2	0	1	2	
PROJECTS									
7.	BCSE0091	Software Testing Project	0	0	0	8	2	-	

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	Digital Image Processing
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	Introduction to Machine Learning	3	0	0	0	3	3	Mathematics, Programming
6.	BCSE0106	Computer Vision	3	0	0	0	3	3	Digital Image Processing
PRACTICALS									
7.	BCSE0131	Digital Image Processing Lab	0	0	2	0	1	2	Programming
8.	BCSE0132	Soft Computing Lab	0	0	2	0	1	2	Programming
9.	BCSE0133	Machine Learning Project	0	0	2	0	1	2	
PROJECTS									
10.	BCSE0141	Machine Learning Project	0	0	0	8	2	-	

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	Advance Database Management System	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	Big Data Analytics	3	0	0	0	3	3	DBMS
6.	BCSE0156	Distributed and Parallel Databases	3	0	0	0	3	3	DBMS
7.	BCSE0157	Natural Language Processing	3	0	0	0	3	3	TAFL/Compiler Design
8.	BCSE0158	Data Science	3	0	0	0	3	3	
PRACTICALS									
8.	BCSE0181	Data Mining and Warehousing Lab	0	0	2	0	1	2	
9.	BCSE0182	Business Intelligence Lab	0	0	2	0	1	2	
10.	BCSE0182	Big Data Analytics Lab	0	0	2	0	1	2	
PROJECTS									
11.	BCSE0191	Data Mining and Warehousing Project	0	0	0	8	2	-	
12.	BCSE0192	Business Intelligence Project	0	0	0	8	2	-	

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	4	0	0	0	4	4	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	Distributed System	4	0	0	0	4	4	CN /OS
5.	BCSE0205	Parallel Algorithms	3	0	0	0	3	3	CO & Algorithms
6.	BCSE0206	Cloud Computing and Storage	3	0	0	0	3	3	Distributed System
PRACTICALS									
7.	BCSE0231	Embedded System Lab	0	0	2	0	1	2	
8.	BCSE0232	Internet of Things Lab	0	0	2	0	1	2	
9.	BCSE0233	Parallel Algorithms Lab	0	0	2	0	1	2	
10.	BCSE0234	Cloud Computing lab	0	0	2	0	1	2	
PROJECTS									
11.	BCSE0241	Internet of Things Project	0	0	0	8	2	-	
12.	BCSE0242	Cloud Computing and Storage Project	0	0	0	8	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	Mobile App Development for Android	3	0	0	0	3	3	
2.	BCSE0252	Mobile App Development for IOS	3	0	0	0	3	3	
3.	BCSE0253	Advanced Java Programming	3	0	0	0	3	3	
4.	BCSE0254	.Net Framework using C#	3	0	0	0	3	3	
5.	BCSE0255	Web Technology	3	0	0	0	3	3	
PRACTICALS									
6.	BCSE0281	Mobile App Development for Android Lab	0	0	2	0	1	2	
7.	BCSE0282	Mobile App Development for IOS Lab	0	0	2	0	1	2	
8.	BCSE0283	Advanced Java Programming Lab	0	0	2	0	1	2	
9.	BCSE0284	.Net Framework using C# Lab	0	0	2	0	1	2	
10.	BCSE0285	Web Technology Lab	0	0	2	0	1	2	

Projects

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

Mandatory Non Graded Course

S. NO.	CODE	SUBJECT	TEACHING SCHEME				CREDITS	CONTACTS HR/WK	PRE- REQUISITES
			L	T	P	J			
THEORY									
1.	BCSM0001	Introduction to Cyber Security	2	0	0	0	0	2	
2.	BCHM0101	Disaster Management	2	0	0	0	0	2	
3.	MBAM0001	Basic Course in Entrepreneurship	2	0	0	0	0	2	
4.	MBAM0002	Leadership And Organizational Behaviour	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 Purpose – I	2	0	0	0	2	2	
4.	BELH0004	English for Professional Purpose – 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	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	4	4	
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	BCSG0800	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	Introduction To Object Oriented Programming	3	0	0	0	3	3	
2.	BCSO0002	Data Structures And Applications	3	0	0	0	3	3	
3.	BCSO0003	Essentials Of Information Technology	3	0	0	0	3	3	
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	Introduction To Object Oriented Programming Lab	0	0	2	0	1	2	
8.	BCSO0071	Data Structures And Applications Lab	0	0	2	0	1	2	
9.	BCSO0072	Essentials Of Information Technology Lab	0	0	2	0	1	2	
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.	14
II	**Tuple:** Introduction, accessing tuples, Operations, Working, Functions and Methods. **Dictionaries:** Introduction, accessing values in dictionaries, Working with dictionaries, Properties, Functions. **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;	14
III	**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.	14

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:

- Understand to solve problems with smaller Lines of Code using Python as compared to other programming languages
- Use OO concepts while programming in Python
- Use in-built packages defined in Python
- 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.	17
II	**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. **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	17
III	**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.	18

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:

- Familiar with the concepts related to structured programming constructs
- Design an algorithmic solution for a given problem and as well C program for a given algorithm.
- Demonstrate their knowledge of, and ability to apply, programming fundamentals in different programming languages.
- 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.
- To apply the concepts and design principles relating to: data structures, computer architecture and organization, programming languages, operating systems, and networks.

BCSC0002: OBJECT ORIENTED PROGRAMMING

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

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

Module No.	Content	Teaching Hours
I	Object Oriented Programming features, Introduction to Java & Python for OOPS, Characteristics of Java & Python for OOPS **Java View:** Concepts of Objects, Methods & Classes, Constructors, Polymorphism, Inheritance, Abstract class, final, Packages and Interfaces. Exception Handling. Python View: Importance of self, __init__() method, Instance Methods, Class Methods and Static Methods, Using default parameters in Methods.	14
II	**Object Design Implementation:** Multithreaded Architecture and Thread Control Methods, String handling, Wrapper Class(java Only). Generic Class and Generic Methods. Regular Expression processing. **Collection Framework (Java Only):** Collections Overview, Collection Interface, List Interface, Set Interface. Collection classes, ArrayList, Linked List, Java Security API, Introduction to Java Web Services.	14
III	**JDBC:** Types of Drivers, Connectivity Model, JDBC/ODBC Bridge, Communicating with Database. Database Programming(Python): Database Connectivity, Retrieving Data from Database, Parameters Passing, Executemany Method, Cursor Attributes, Invoke Stored Procedures, Invoke Stored Functions. GUI Programming: GUI Programming Toolkit, Overview of Tkinter, Visual Tkinter IDE CGI Web Programming: CGI and Apache Server Configurations, CGI Module and Debugging, CGI Cookies	14

Text Books:

- Michael R Blaha, James R. Rumbaugh, "*Object Oriented Modeling and Design with UML*", 2nd Edition, Pearson Education.
- Naughton, Schildt, "*The Complete Reference JAVA2*", 9th Edition, Oracle Press.

Reference Books:

- Cay Horstmann, "*Big Java*", 5th Edition, Wiley India Edition.
- Loy & Wood, "*Java Swing*", O'Reilly.
- Bhave&Patekar, "*Programming with Java*", Pearson Education

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

- Learn Features of object-oriented programming.
- Learn Designing of GUIs with different layouts and Graphical programming
- Design object oriented solutions for small systems involving database and event handling concepts.

BCSC0003: DATABASE MANAGEMENT SYSTEM

Objective: *The objective of the course is to enable students to understand and use a relational database 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), Overall Database Structure, Database Users, Database Development Life Cycle (DDLC). **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, Relationship of Higher Degree. **Relational Data Model and Language:** Relational Data Model Concepts, Integrity Constraints, Entity Integrity, Referential Integrity, Keys Constraints, Domain Constraints, Relational Algebra, Relational Calculus, Tuple and Domain Calculus.	13
II	**Data Base 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, Canonical Cover, Redundant Cover, Synthesis the Set of Relation, MVD and JDs, Inclusion Dependence. **File Organization**: Indexing, structure of index files and types, Dense and sparse indexing, Introduction and properties of B trees and B+ trees, Finding order and capacity of B+ trees.	14
III	**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. **Distributed Database:** Introduction of Distributed Database, Data Fragmentation and Replication. **Concurrency Control Techniques:** Concurrency Control, Locking Techniques for Concurrency Control, 2PL, Time Stamping Protocols for Concurrency Control, Validation Based Protocol, Multiple Granularity, Multi Version Schemes.	13

Text Books:

- Elmasri and Navathe (2010), "*Fudamentals of Database Systems*", 6th Edition, Addision Wesley.

References Books:

- Date C J," *An Introduction to Database Systems*", 8th Edition, Addision 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:

- 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 logical design of databases, including the E-R method and normalization approach.
- Be familiar with the basic issues of transaction processing and concurrency control.

BCSC0004: OPERATING SYSTEMS

Objective: *The objective of the course is to provide basic knowledge of computer operating system structures and functioning including CPU scheduling, memory management, concurrent processes, deadlocks, security, and integrity.*

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.	14
II	**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. **Deadlock:** System model, Deadlock characterization, Prevention, Avoidance and detection, Recovery from deadlock, Combined Approach.	13
III	**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.	13

Text Books:

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

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:

- Identify the services provided by operating systems.
- Understand the internal structure of an operating system and be able to write programs using system calls.
- Understand and solve problems involving process control, mutual exclusion, deadlock and synchronization.

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	**Introduction:** Basic organization of the computer and block level description of the functional units, Number representation; fixed and floating-point number representation, IEEE standard for floating point representation., Instruction set, Instruction cycles. Register, bus and memory transfer. **Central Processing Unit**: Addition and subtraction of signed numbers, carry look ahead adders. Multiplication: Signed operand multiplication, Booths algorithm and array multiplier. Processor organization, general registers organization, stack organization and addressing modes. Introduction to Combinational Circuit, Multiplexer, demultiplexer, Decoder, Encoder. Introduction to Sequential Circuit, Flip-Flops, Synchronous and Asynchronous Counters	13
II	**Multiprogramming and Multiprocessing**; Introduction to pipelined operation. **Control Unit:** Instruction types, formats, micro-operations, execution of a complete instruction. Hardwired and micro programmed control: micro programmed sequencing, Microinstruction with next address field, pre-fetching microinstructions, concept of horizontal and vertical microprogramming.	14
III	**Memory:** Basic concept and hierarchy, RAM memories, ROM memories. Cache memories: concept and design issues, performance, address mapping and replacement. Virtual memory: concept and implementation. **Input/Output**: Peripheral devices, I/O interface, I/O ports, Interrupts: interrupt hardware, types of interrupts and exceptions. Buses, bus architecture, types of buses and bus arbitration. Modes of Data Transfer: Programmed I/O, interrupt initiated I/O and Direct Memory Access., I/O channels and processors. Standard communication interfaces.	13

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:

- Understand the organization of the modern computer system hardware.
- Analyze the performance of component, able to calculate the effective address of different operands, arithmetic operations of positive and negative numbers.
- Understand the Basic hardware architectural issues that affect the nature and performance of software.

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.	13
II	**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. **Search Trees:** Binary Search Trees (BST), Insertion and Deletion in BST, AVL Trees, Introduction to M-Way Search Trees, B Trees.	14
III	**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.	13

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:

- Understand the concept of Dynamic memory management, data types, different data structure and asymptotic notations.
- Student will be able to choose appropriate data structure as applied to specified problem.
- Able to compare, implement and know when to apply sorting algorithm including bubble sort, selection sort, heap sort.

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.	13
II	**Programming Technique with Additional Instruction**: Looping, Counting, Indexing, Additional Data Transfer and 16-Bit Arithmetic Instruction, Counters and Time Delays, Stack and Subroutine. **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.	14
III	**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.	13

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:

- Understand the processor organization, behavior of processing elements in different addressing modes.
- Analyze the data and instructions flow inside the internal components like computing elements, memory and input output devices

BCSC0008: COMPUTER NETWORKS

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

Credits:04 **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	13
II	**Data Link Layer –**Error detection and correction, Flow control (sliding window protocol) **Network Layer:** Network Layer –IP addressing, subnet, CIDR, VLSM, Internetworking, Address mapping, routing. Connecting devices.	14
III	**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.	13

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:

- Understand the terminology and concepts of the OSI reference model and the TCP/IP reference model.
- Apply the concepts of protocols, network interfaces, and design/performance issues in local area networks and wide area networks.

BCSC0009: SOFTWARE ENGINEERING

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

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

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.	13
II	**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. **OO Analysis and OO Design:** OO Concepts, Introduction to UML Design Patterns, Activity Diagram, Sequence Diagram, State Chart Diagram. **Coding**: Coding Process, Verification – Code Inspections, Software Metrics.	13
III	**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.	14

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:

- The ability to apply software engineering theory, principles, tools and processes, as well as the theory and principles of computer science and mathematics, to the development and maintenance of complex software systems.
- The ability to design and experiment with software prototypes and to select and use software metrics.
- Effective communications skills through oral and written reports and software documentation evaluated by both peers and faculty.
- The ability to elicit, analyze and specify software requirements through a productive working relationship with project stakeholders.

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.	13
II	**Probability Theory**: Introduction to Probability Theory, Conditional Probability, Total Probability, Bayes' Theorem. **Propositional Logic** - Logical Connectives, Truth Tables, Normal Forms (Conjunctive and Disjunctive), Validity; **Predicate Logic** - Quantifiers, Inference Theory, Methods of Proof: Direct, Indirect, Mathematical Induction.	14
III	**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.	13

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:

- Understand the notion of mathematical thinking, mathematical proofs, and algorithmic thinking, and be able to apply them in problem solving.
- Understand the basics of discrete probability and number theory, and be able to apply the methods from these subjects in problem solving.
- Use effectively algebraic techniques to analyze basic discrete structures and algorithms.

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 **L-T-P-J:3-1-0-0**

Module No.	Content	Teaching Hours
I	**Introduction:** Alphabets, Strings and Languages; Automata and Grammars, Deterministic Finite Automata (DFA), Nondeterministic Finite Automata (NFA), Equivalence of NFA and DFA, Minimization of Finite Automata, Myhill-Nerode Theorem; FA with Output - Moore and Mealy machine, Applications and Limitations of FA. **Regular expression (RE):** Regular Expression to FA, DFA to Regular Expression, Arden Theorem, Non Regular Languages, Pumping Lemma for Regular Languages, Applications of Pumping Lemma, Closure Properties of Regular Languages.	13
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. **Push Down Automata (PDA):** Introduction, Language of PDA, Acceptance by Final State, Acceptance by Empty Stack, Deterministic PDA, Equivalence of PDA and CFG.	14
III	**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	13

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:

- Construct FA and minimize automata;
- Construct an automaton for a given regular expression;
- Construct a pushdown automaton for a given context-free language;
- Construct a Turing machine deciding a given problem,
- Prove whether a language is or is not a regular or context-free by using the Pumping Lemma;
- Prove that a given context-free grammar generates a given context-free language;
  Prove un-decidability of a given problem by reducing from a known un-decidable problem.

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.	13
II	Divide and Conquer with examples such as Sorting, Matrix Multiplication, Convex hull and Searching. 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.	14
III	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.	13

Text Books:

- Thomas H. Coremen, Charles E. Leiserson and Ronald L. Rivest(2008), "*Introduction to Algorithms, Third edition*", Prentice Hall of India.
- Aaron M. Tanenbaum, Yedidyah Langsam and Moshe J. Augenstein (2009), "Data Structures Using C and C++", 2nd Edition, PHI.

Reference Books:

- Gilles Brassard Paul Bratley (1996)," *Fundamentals of Algorithms*", Prentice Hall.
- Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran (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:

- Learn good principles of algorithm design;
- Apply the algorithms and design techniques to solve problems;
- Analyze the complexities of various problems in different domains and estimate their worst-case, average-case and best case behavior.
- Discuss various searching, sorting and graph traversal algorithms.
- Understand NP completeness and identify different NP.
- Know how to design algorithms using the divide-and-conquer. Dynamic programming, greedy approach strategy and recite algorithms that employ these strategies.

BCSC0013: COMPILER DESIGN

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

Credits: 4 **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.	14
II	**Advance Parser:** Constructing Canonical LR parsing tables, Constructing LALR parsing tables, using ambiguous grammars, an automatic parser generator, implementation of LR parsing tables. **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.	13
III	**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 Generation: Design Issues, Target Language. Addresses in the Target Code, Basic Blocks and Flow Graphs, Optimization of Basic Blocks, Code Generator. Code **optimization:** Machine-Independent Optimizations, Loop optimization, DAG representation of basic blocks, Value numbers and algebraic laws, Global Data-Flow analysis.	13

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:*

- Apply the knowledge of l ex tool & yacc tool to develop a scanner & parser.
- Deal with different translators.
- Develop program to solve complex problems in compiler.
- Learn the new code optimization techniques to improve the performance of a program in terms of speed & space.

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 and III	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:

- solve problems with smaller Lines of Code using Python
- use OO concepts while programming in Python
- use in-built packages defined in Python
- use 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 and III	• 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.	17
	• 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	17
	• 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.	18

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:

- Write, compile and debug programs in C language.
- Use different data types in a computer program.
- Design programs involving decision structures, loops and functions.
- Understand the concepts of functions, recursion, pointers and file handling.
- Write, compile and debug programs in C language.
- Use different data types in a computer program.
- 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 and III	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:

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

BCSC0802: DATABASE MANAGEMENT SYSTEM LAB

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

Credits:1 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours
I, II and III	• 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 concept of grouping of Data. • To implement 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 cursor. • To implement the concept of Procedure function and Triggers.	24

References Books:

- Date C J," *An Introduction to Database Systems*", 8th Edition, Addision 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:

- Ability to crate database tables
- Ability to formulate SQL queries based on the problems given
- Ability to apply PL/SQL.

BCSC0803: OPERATING SYSTEMS LAB

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

Credits:1 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours
I, II and III	• 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:

- Various operations on UNIX operating systems.
- Understand the working of systems calls.
- 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 and III	• 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:

- Design any sequential circuit for logical operations.
- 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 and III	• 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:

- Iden*tify, implement and use the appropriate data structures for a given problem*
- *Apply algorithmic skills for computing and engineering practice.*
- *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 and III	• 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:

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

BCSC0807: COMPETITIVE PROGRAMING 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 and III	1. Program for Recursive Binary & Linear Search. 2. Program for sorting(Heap Sort, Merge Sort, Selection Sort, Quick Sort) 3.Program to understand Recursion. 4. Program to understand Backtracking 5. Program to understand Matrix Multiplication. 6. Program to understand Convex hull and Searching. 7. Program to understand 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. 8. Program to understand Dynamic Programing with examples such as Longest Increasing Subsequence. Finding best path in maze. Coin Change Problem. Knapsack. Warshal's and Floyd's algorithms 9. Program to understand Divide and Conquer Etc.	24

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

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

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