Credit Structure for B.Tech. (Information Technology)

(For newly admitted students from session 2020-21)

Credit Courses											
	Category Semesters	I	II	III	IV	V	VI	VII	VIII	Total	Min. Req.
Undergraduate Core Courses (158 min. credits)	Basic Sciences & Maths (BSM)	9	14	9	4	-	-	-	-	**36**	**36**
	Engineering Fundamentals (EF)	11	7	6	2	-	-	-	-	**26**	**24**
	Department Core (DC)	-	-	10	15	19	24	10	4	**82**	**78**
	Management (M)	-	-	-	3	3	-	-	-	**6**	**6**
	Humanities & Social Science Core (HSSC)	4	-	-	-	-	-	-	-	**4**	**4**
	Project (P)	-	-	-	-	-	-	5	5	**10**	**10**
Undergraduate Programme Electives (22 min. credits)	Programme Electives (PE)	-	-	-	-	-	-	8	8	**16**	**16**
	Open Electives (OE)	-	-	-	-	-	-	-	4	**4**	**3**
	Humanities & Social Science Electives (HSSE)	-	3	-	-	-	-	-	-	**3**	**3**
Min. Credits Required (158+22=180)	**Total**	**24**	**24**	**25**	**24**	**22**	**24**	**23**	**21**	**187**	**180**

Audit Courses		
	Total	Min. Req.
(Min. 3 Credits audit subjects from other departments will be offered during Semester I-V)	21	15
Seminar	3	3
Industrial/Practical Training (IPT)	1	1

Course Structure of BTech (Information Technology)

Credit Value

1 Period Lecture=1 Credit **1 Period Tutorial=1 Credit** **2 Period Practical=1 Credit**

Freshman Year, Semester-I

S.N.	Category	Paper Code	Subject	L	T	P	Credit
1.	BSM	BAS-01	Engineering Mathematics-I	3	1	0	4
2.	BSM	BAS-02	Engineering Physics-I	3	1	2	5
3.	EF	BIT-01	Fundamentals of Information Technology	3	1	-	4
4.	EF	BEE-01	Principles of Electrical Engineering	3	1	2	5
5.	HSSC	BAS-03	Professional Communication	3	1	0	4
6.	EF	BIT-02	Software Tools-I	0	0	4	2
7.	AC		Audit Course				-
			Total	**15**	**5**	**8**	**24**

Freshman Year, Semester-II

S.N.	Category	Paper Code	Subject	L	T	P	Credit
1.	BSM	BAS-07	Engineering Mathematics-II	3	1	0	4
2.	BSM	BAS-08	Engineering Physics-II	3	1	2	5
3.	BSM	BAS-24	Applied Computational Methods	3	1	2	5
4.	EF	BIT-03	Programming Fundamentals	3	1	2	5
5.	HSSE	BAS-**	Humanities & Social Science Electives	2	1	0	3
6.	EF	BCE-10	Engineering Graphics	0	0	4	2
7.	AC		Audit Course				-
			Total	**14**	**5**	**10**	**24**

Sophomore Year, Semester-III

S.N.	Category	Paper Code	Subject	L	T	P	Credit
1.	BSM	BAS-01	Discrete Mathematics	3	1	0	4
2.	BSM	BAS-14	Graph Theory	3	1	2	5
3.	EF	BIT-11	Switching Theory & Logic Design	3	1	-	4
4.	DC	BIT-12	Data Structures	3	1	2	5
5.	DC	BIT-13	Object Oriented Programming	3	1	2	5
6.	EF	BIT-14	Software Tools-II	0	0	4	2
7.	AC		Audit Course				-
			Total	**15**	**5**	**10**	**25**

Sophomore Year, Semester-IV

S.N.	Category	Paper Code	Subject	L	T	P	Credit
1.	BSM	BAS-26	Optimization Techniques	3	1	0	4
2.	M	MBA-113	Management Information System	2	1	-	3
3.	DC	BIT-15	Design & Analysis of Algorithm	3	1	2	5
4.	DC	BIT-16	Computer Organization & Architecture	3	1	2	5
5.	DC	BIT-17	Database Management System	3	1	2	5
6.	EF	BIT-18	Software Tools-III	0	0	4	2
7.	AC		Audit Course				-
			Total	**14**	**5**	**10**	**24**

Junior Year, Semester-V

S.N.	Category	Paper Code	Subject	L	T	P	Credit
1.	M	MBA-02	Engineering & Managerial Economics	2	1	0	3
2.	DC	BIT-26	Operating System	3	1	2	5
3.	DC	BIT-27	Computer Networks	3	1	2	5
4.	DC	BIT-28	Software Engineering	3	1	2	5
5.	DC	BIT-29	Automata Theory	3	1	-	4
6.	AC		Audit Course				-
			Total	**14**	**5**	**06**	**22**

Junior Year, Semester-VI

S.N.	Category	Paper Code	Subject	L	T	P	Credit
1.	DC	BIT-31	Data Mining & Ware Housing	3	1	0	4
2.	DC	BIT-32	Artificial Intelligence	3	1	2	5
3.	DC	BIT-33	Machine Learning	3	1	2	5
4.	DC	BIT-34	Wireless Sensor Network & IoT	3	1	2	5
5.	DC	BIT-35	Network Security & Cryptography	3	1	2	5
6.	AC	BIT-30	Seminar	-	-	6	-
			Total	**15**	**5**	**08**	**24**

Senior Year, Semester-VII

S.N.	Category	Paper Code	Subject	L	T	P	Credit
1.	DC	BIT-41	Graphics & Visual Computing	3	1	2	5
2.	DC	BIT-42	Mobile Computing	3	1	2	5
3.	PE-1	BIT-*	Programme Elective-1	3	1	0	4
4.	PE-2	BIT-*	Programme Elective-2	3	1	0	4
5.	P	BIT-40	Project Part-1	0	0	10	5
6.	AC	BIT-45	Industrial/Practical Training	0	0	2	-
			Total	**12**	**4**	**14**	**23**

Senior Year, Semester-VIII

S.N.	Category	Paper Code	Subject	L	T	P	Credit
1.	DC	BIT-43	Distributed System	3	1	0	4
2.	PE-3	BIT-*	Programme Elective-3	3	1	0	4
3.	PE-4	BIT-*	Programme Elective-4	3	1	0	4
4.	OE	BOE-*	Open Elective Offered by other dept.	3	1	0	4
5.	P	BIT-50	Project Part-2	0	0	10	5
			Total	**12**	**4**	**10**	**21**

Engineering Fundamentals & Departmental Core (Information Technology)

Sr. No.	Paper Code	Subject	Prerequisite	L	T	P	Credit
1.	BIT-01	Fundamentals of Information Technology	-	3	1	0	4
2.	BIT-02	Software Tools-I	-	0	0	4	2
3.	BIT-03	Programming Fundamentals	-	3	1	2	5
4.	BIT-11	Switching Theory & Logic Design	-	3	1	0	4
5.	BIT-12	Data Structures	-	3	1	2	5
6.	BIT-13	Object Oriented Programming	-	3	1	2	5
7.	BIT-14	Software Tools-II	-	0	0	4	2
8.	BIT-15	Design & Analysis of Algorithm	-	3	1	2	5
9.	BIT-16	Computer Organization & Architecture	-	3	1	2	5
10.	BIT-17	Database Management System	-	3	1	2	5
11.	BIT-18	Software Tools-III	-	0	0	4	2
12.	BIT-26	Operating System	-	3	1	2	5
13.	BIT-27	Computer Networks	-	3	1	2	5
14.	BIT-28	Software Engineering	-	3	1	2	5
15.	BIT-29	Automata Theory	-	3	1	0	4
16.	BIT-31	Data Mining & Ware Housing	-	3	1	0	4
17.	BIT-32	Artificial Intelligence	-	3	1	2	5
18.	BIT-33	Machine Learning	-	3	1	2	5
19.	BIT-34	Wireless Sensor Network & IoT	-	3	1	2	5
20.	BIT-35	Network Security & Cryptography	-	3	1	2	5
21.	BIT-30	Seminar	-	0	0	6	0
22.	BIT-41	Graphics & Visual Computing	-	3	1	2	5
23.	BIT-42	Mobile Computing	-	3	1	2	5
24.	BIT-40	Project Part-1	-	0	0	10	5
25.	BIT-45	Industrial/Practical Training	-	0	0	2	1
26.	BIT-43	Distributed System	-	3	1	0	4
27.	BIT-50	Project Part-2	Project-1	0	0	10	5

Programme Electives (Information Technology)

Sr. No.	Paper Code	Subject	Prerequisite	L	T	P	Credit
		PE-1 & PE-2					
1.	BIT-51	.Net Technology	-	3	1	0	4
2.	BIT-52	Advanced JAVA	-	3	1	0	4
3.	BIT-53	Real Time System	-	3	1	0	4
4.	BIT-54	Artificial Intelligence Search Methods for problem Solving	-	3	1	0	4
5.	BIT-55	Aspect Oriented Programming	-	3	1	0	4
6.	BIT-56	Big Data Computing	-	3	1	0	4

7.	BIT-57	Blockchain Architecture Design and Use Cases	-	3	1	0	4
8.	BIT-58	Cloud Computing and Distributed Systems	-	3	1	0	4
9.	BIT-59	Compiler Design	-	3	1	0	4
10.	BIT-60	Computer Vision: Foundations and Applications	-	3	1	0	4
11.	BIT-61	Functional Programming	-	3	1	0	4
12.	BIT-62	Data Science for Engineers	-	3	1	0	4
13.	BIT-63	Database Administration with ORACLE	-	3	1	0	4
14.	BIT-64	Deep Learning	-	3	1	0	4
		PE-3 & PE-4					
15.	BIT-65	Android Programming	-	3	1	0	4
16.	BIT-66	Embedded System	-	3	1	0	4
17.	BIT-67	Hardware Modelling using Verilog	-	3	1	0	4
18.	BIT-68	Hardware Security	-	3	1	0	4
19.	BIT-69	High Performance Computing	-	3	1	0	4
20.	BIT-70	Introduction to Parallel Programming in Open MP	-	3	1	0	4
21.	BIT-71	Linux Administration & Networking	-	3	1	0	4
22.	BIT-72	Digital Signal Processing	-	3	1	0	4
23.	BIT-73	Multi-Core Computer Architecture – Storage and Interconnects	-	3	1	0	4
24.	BIT-74	Network Programming	-	3	1	0	4
25.	BIT-75	Parallel Algorithms	-	3	1	0	4
26.	BIT-76	Scalable Data Science	-	3	1	0	4
27.	BIT-77	Software Design, Construction & Quality Management	-	3	1	0	4
28.	BIT-78	Software Verification & Validation	-	3	1	0	4

Open Electives for other department

Sr. No.	Paper Code	Subject	Prerequisite	L	T	P	Credit
1.	BOE-25	Linux & Shell Programming	-	3	1	0	4
2.	BOE-26	Web Technology	-	3	1	0	4
3.	BOE-27	Digital Forensic & Cyber Laws	-	3	1	0	4
4.	BOE-28	Network Security	-	3	1	0	4

Audit Courses for BTech (IT)

S.N.	**Category**	**Paper Code**	**Subject**	**L**	**T**	**P**	**Credit**
1.	AC	BAS-05	Environment & Ecology	2	1	0	-
2.	AC	BEC-01	Fundamentals of Electronics Engineering	2	1	0	-
3.	AC	BCS-13	Internet & Java Programming	3	1	2	-
4.	AC	BCS-53	LAMP Technology	3	1	0	-
5.	AC	BCS-73	Neural Network & Fuzzy Systems	3	1	0	-
6.	AC	BEE-15	Introduction to Microprocessors	3	1	2	-
7.	AC	MAS-109	Foreign Language- French	2	1	0	-
8.	AC	MAS-109	Foreign Language- German	2	1	0	-
9.	AC	MAS-109	Foreign Language- Spanish	2	1	0	-

Humanities & Social Science Electives (HSSE)

S.N.	Category	Paper Code	Subject	L	T	P	Credit
1.	AC	BAS-10	Technical Writing	2	1	0	3
2.	AC	BAS-11	Human Values & Professional Ethics	2	1	0	3
3.	AC	BAS-12	Industrial Psychology	2	1	0	3
4.	AC	BCS-13	Industrial Sociology	2	1	0	3

Computer Fundamental (CF)courses for BBA

S.N.	Category	Paper Code	Subject	L	T	P	Credit
1.	CF	BIT-81	Fundamentals of Computer Applications	2	0	0	2
2.	CF	BIT-82	IT Tools for Business	2	0	2	3

Syllabus (B.Tech.-I)

BIT-01 **Fundamentals of Information Technology**

Course category : Engineering Fundamentals (EF)
Pre-requisite Subject : NIL
Contact hours/week : Lecture: 3, Tutorial: 1 , Practical: 0
Number of Credits : 4
Course Assessment methods : Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination

Course Outcomes : The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course

1. understand the basics of computers Hardware/Software
2. understand the importance of data compression and the algorithms for lossy and lossless data compression
3. understand the concept of operating system and fundamentals of computer networking

UNIT-I **9**

Introduction to Computer Hardware/Software: Processor, Motherboard, I/O Devices, peripherals, Memory Types & Hierarchy: Cache, Primary & Secondary memories with examples, Concept of Computer Languages: Low-Level, Assembly and High-Level, System Software: Assembler, Compiler, Interpreter, Loader/Linker

UNIT-II **9**

Data & Information, Digital representation of Information, Number Systems & Comparisons: Binary, Octal, Decimal, Hexadecimal, Text Representation: ASCII, EBCDIC, Unicode, Multimedia Data, Data Compression Types and Techniques: Lossy / Lossless, Huffman, Shannon-Fano, Dictionary Based Compression techniques

UNIT-III **9**

Operating System: Concept, Functions, Types, Single-user/Multi-user operating system, Architectural differences, Shell fundamentals, Exemplary commands: Internal & External, Basics of Primary and Secondary Memory Management

UNIT-IV **9**

Network Basics: Concept, Types, Transmission modes, Topologies, OSI & TCP/IP Models: Functions of different Layers, concept of MAC, IP (Private/Public) and TCP addresses, Basic Introduction to CSMA/CD, IP & TCP/UDP and HTTP Protocols, Current Internet Applications

Text Books & References

1. Mark Nelson and Jean-Loup Gailly"The Data Compression Book", M&T Books, A Division of MIS: Press,Inc.
2. K Sayood, "Introduction to Data Compression" 3/e, Elsevier 2006
3. Forouzan, Data Communication and Networking, TMH
4. Silberschatz, A., Galvin, P. and Gagne, G., Applied Operating Systems Concepts, John Wiley& Sons Inc.

BIT-02 **Software Tools-I**

Course category : Engineering Fundamentals (EF)

Pre-requisite Subject : NIL

Contact hours/week : Lecture: 0, Tutorial: 0 , Practical: 4

Number of Credits :2

Course Assessment methods : Continuous assessment through Viva-voce, Practical work/Record, attendance and Major Practical Examination

Course Outcomes : The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Understanding of Booting Process and installation of Operating system
2. Usage of Operating system commands
3. Understanding of Shell and its usage as a programming language
4. Understanding of Computer Networking concepts

Experiments

1. Understanding CMOS settings of operating system
2. Installation of Linux operating system using virtualization technique
3. Understanding and practice of various Linux commands
4. Creation/usage of various types of files supported by Linux
5. Practice of Computer networking commands
6. Programs using shell programming

BIT-03 **Programming Fundamentals**

Course category : Engineering Fundamentals (EF)

Pre-requisite Subject : NIL

Contact hours/week : Lecture: 3, Tutorial: 1, Practical: 2

Number of Credits : 5

Course Assessment methods : Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination

Course Outcomes : The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Describing the basics of terminologies used in computer programming.
2. Practicing C language programming by writing, compiling and debugging the code.
3. Designing programs involving simple statements, conditional statements, iterative statements, array, strings, functions, recursion and structure.
4. Discussing the dynamic memory allocations and use of the pointers.
5. Applying basic operations on files through programs.
6. Studying and implementing the codes using macros, preprocessor directives and command line arguments

UNIT-I **9**

Basics of Computers and Programming: Functional diagram of computer; Language Processors; Approaches to problem solving, Concept of algorithm and flow charts. **Simple Statements:** Datatypes; Tokens and its types; Variable declaration and initialization; User defined type declaration: typedef, enum; Comments; Format specifiers; Standard I/O: taking input and displaying output; **Operators:** types, precedence and associativity; Expressions; Type conversion, C short-hands.

UNIT-II **9**

Conditional Statements: Simple if, if-else, nested if-else, else-if ladder, switch statements, nested switch, advantages of switch over nested if, restrictions on switch values. **Iterative Statements:** Concepts of entry and exit controlled loops; Uses of for, while and do while loops; Nested Loops; Printing various patterns using nested loops; Using break, continue and goto statements.

UNIT-III **9**

Arrays: Single-dimensional, multi-dimensional array and their applications; declaration and manipulation of arrays; strings and string handling functions. **Pointers:** Pointer and address arithmetic; dereferencing; pointers and arrays;

dynamic memory allocation and de-allocation. **Functions:** Function prototype; Arguments and its types: actual, formal and default arguments; Scope of a variable; Argument passing methods; Passing pointer as the function argument; Recursion: types, advantages and disadvantages; Storage class specifies; Character test functions.

UNIT-IV **9**

Structure: Declaring and defining structures; Array within structure; Array of structure; Defining and using some data structures: Stack, Queue, and Linked lists. **File Handling:** Types of files; Text files and different operations on text files, opening a file, closing a file;Data structure of a file; EOF; I/O operations on files; Random access to the files. **Standard C Preprocessors & C Library:** Pre-processor, Directives, Macro, Macro substitution; Conditional Compilation; Command Line Arguments; Standard C Library.

Text Books & References

1. Brian W. Kernighan and Dennis M. Ritchie, "The C programming language", Pearson
2. E. Balagurusamy, "Programming in ANSI C", McGraw Hill Education
3. Yashavant Kanetkar, "Let Us C", bpb publication
4. Jeri R. Hanly, Elliot B. Koffman, "Problem Solving and Program Design in C", Pearson
5. Herbert Schildt,"C: The Complete Reference",McGraw Hill Education

EXPERIMENTS

Implementing programs in following categories using programming language 'C':

1. Programs of simple statements, conditional statements and iterative statements with their applications.
2. Programs of single and multi dimensional arrays and their applications.
3. Programs of strings and their applications
4. Programs of pointer and their applications
5. Programs of function and their applications
6. Programs of structure and their applications
7. Codes of file handling and management
8. Codes with Preprocessor, Macro, Conditional Compilation and Command Line Arguments

BIT-11 **SWITCHING THEORY & LOGIC DESIGN** **4 Credits (3-1-0)**

Course Objectives:

The course objectives of the course are:

1. To introduce the concept of digital and binary systems
2. To be able for designing and analysing combinational logic circuits
3. To be able for designing and analysing sequential logic circuits
4. To understand basic software tools for the design and implementation of digital circuits and systems
5. To reinforce theory and techniques taught in the classroom through experiments and projects in the laboratory.

Learning Outcomes:

On completion of this course, students will be able to:

1. Design a finite state machine and sequential logic design.
2. Synthesize a logic design from a natural language description of a problem.
3. Realize a complete arithmetic and logic unit.
4. Generate a realization of combinational logic in a programmable gate array.
5. Simulate a complete design to evaluate functional correctness and timing.

UNIT- I: **9**

Binary Codes - Weighted and Non-Weighted - Binary Arithmetic Conversion Algorithms - Error Detecting and Error Correcting Codes - Canonical and Standard Boolean Expressions - Truth Tables.

UNIT- II: **9**

K-Map Reduction - Don't Care Conditions - Adders / Subtractors- Carry Look-Ahead Adder - Code Conversion Algorithms - Design of Code Converters - Equivalence Functions.

Binary/Decimal Parallel Adder/Subtractor for Signed Numbers - Magnitude Comparator - Decoders / Encoders - Multiplexers / Demultiplexers- Boolean Function Implementation using Multiplexers.

UNIT- III: **9**

Sequential Logic - Basic Latch - Flip-Flops (SR, D, JK, T and Master-Slave) - Triggering of Flip-Flops - Counters - Design Procedure - Ripple Counters - BCD and Binary - Synchronous Counters.

UNIT- IV: **9**

Registers - Shift Registers - Registers with Parallel Load - Memory Unit - Examples of RAM, ROM, PROM, EPROM - Reduction of State and Flow Tables - Race-Free State Assignment - Hazards.

Text Books & References:

1. Morris Mano, Digital Design, Prentice Hall of India
2. W. H. Gothmann, Digital Electronics -An Introduction to Theory and Practice, Prentice Hall of India

---

BIT-12 **DATA STRUCTURES** **5 Credits (3-1-2)**

Course Objectives:

The course objectives of the course are:

6. To understand the various techniques of searching and sorting
7. To design and implement arrays, stacks, queues, and linked lists
8. To design and implement the non-linear data structures such as trees and graphs

Learning Outcomes:

On completion of this course, students will be able to:

1. Write the algorithms and understand their complexities
2. Learn various linear data structures such as stack and queue
3. Learn various non-linear data structures such as such as tree and graph
4. Know applications of linear and non-linear data structures
5. Implement the different data structures statically (using array)
6. Implement the different data structures dynamically (using pointer or linked list)
7. Understand various searching and sorting techniques
8. Understand different hashing techniques

UNIT- I : Introduction **9**

Basics: Data and Information, Need of data structure, Algorithms and their complexities, Time complexity, Space Complexity, Time Space Trade-off, Big Oh notation. **Array:** Definition, Different representations – row major, column major, address calculation, Basic operations on matrix, Sparse matrix and its types, Basic array operations (Creation, Insertion into and Deletion from an array), Applications of array, Array representation of polynomials. **Linked List:** Definition and its types, Singly linked list, Circular linked list, Doubly linked list, Basic linked list operations (Creation, Insertion into and Deletion from a linked list), Applications of linked list, Linked list representation of polynomial.

UNIT II : Linear Data Structures **9**

Stack: Definition and its implementations (static using array and dynamic using linked list), PUSH and POP operations, Applications of stack, Infix, Prefix and Postfix Expressions and their inter-conversions, Evaluation

of Postfix expressions using stack, Recursion, Types of recursion, difference between recursion and iteration, Tower of Hanoi Problem. **Queue**: Definition and its types, Circular queue, Double ended queue, Implementation of queue (static using array and dynamic using linked list), Basic operation (Creation, Insertion into and Deletion from a queue), Applications of queue.

UNIT III : Nonlinear Data Structures **9**

Trees: Definition, Basic terminologies, difference between tree and forest, tree representation, Types of tree, Implementation of tree (static using array and dynamic using linked list), Basic operations (Creation, Insertion into and Deletion from a tree), Traversal techniques (Inorder, Preorder and Postorder), Applications of tree, Binary trees, B Tree, B+ Tree, Binary Search Tree, Height balanced binary tree – AVL Tree, Threaded binary tree. **Graph**: Definitions, Graph representations – adjacency matrix, adjacency list, Basic operations (Creation, Insertion into and Deletion from a graph), Traversal techniques (Depth first search - DFS, Breadth first search - BFS), Applications of graph. Minimum spanning tree – Prim's algorithm, Kruskal algorithm.

UNIT IV : Searching, Sorting and Hashing **9**

Searching Algorithms: Sequential or Linear search, Binary search. **Sorting Algorithms:** Bubble sort, Insertion sort, Merge sort, Quick sort, Heap sort (concept of max heap, and Min Heap), Radix sort. **Hashing Techniques:** Definition, Difference between Searching and Hashing, Hash functions, Collision.

List of Experiments:

Write C/C++ programs to illustrate the concept of the following:

1. Arrays
2. Linked List
3. Stack
4. Queue
5. Tree
6. Graph
7. Searching
8. Sorting
9. Hashing
10. Applications of various data structures

Text Books & References:

1. "Fundamentals of Data Structures of C", Ellis Horowitz, Sartaj Sahni, Susan Anderson-freed.
2. "Data Structures", S. Lipschutz.
3. "Introduction to Algorithms", T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein.
4. "Data Structures and Program Design in C", Robert L. Kruse, Bruce P. Leung.
5. "Data Structures in C", Aaron M. Tenenbaum.

BIT-13 OBJECT ORIENTED PROGRAMMING 5 Credits (3-1-2)

Course Objectives:

The course objectives of the course are:

1. To give introductory as well as advanced knowledge of object oriented programming
2. To provide various syntax and object oriented programming concepts
3. To improve problem solving skills with Python

Learning Outcomes:

Upon completion of this course, students will be able to:

1. Write basic and advance object oriented programs
2. Create and use classes and objects
3. Write code for Constructors and Destructors
4. Write code for Inheritance
5. Write code for Polymorphism
6. Write code for Operator Overloading
7. Write code for Exception handling
8. Write code for file handling and various file operations

UNIT I **9**

Tokens, Keywords, Identifiers and Constants, Basic Data Types, User-Defined and Derived Data Types, Type Casting, Implicit Conversion, Operators and Expressions, Operator Precedence, Simple statements, Conditional statements, Iterative statements, Array, Function, Pointer, Structure

UNIT II **9**

Basic Concepts of Object Oriented Programming, Object Oriented Programming Paradigm, Benefits of OOP, Object Oriented Languages, Class and Objects, Scope Resolution Operator, Access specifiers, Data members, Accessing class members, Data hiding, Member function, Inline function, Friend function, Passing objects as arguments, Returning objects from functions

UNIT III **9**

Constructors and its types, Destructor, Constructor overloading, Order of construction and destruction, Inheritance, Single, Multilevel, Multiple, Hierarchical, Hybrid Inheritance, Base class, Derived class, Virtual function, Polymorphism, Operator Overloading, Overloading Unary Operators, Overloading Binary Operators

UNIT IV **9**

Exception Handling, Throwing and Catching Mechanism, Templates, File handling, Types of files, End of File, Basic file operations: creating, opening, closing, reading, writing and appending a file, copying a file to another, Object oriented system development

List of Experiments:

Write programs to illustrate the following concepts:

1. Operators and expressions
2. Simple statements, Conditional statements and Iterative statements
3. Arrays
4. Functions
5. Pointers
6. Structures
7. Objects and Classes
8. Inline Function, Friend function and Virtual Functions
9. Scope Resolution Operator
10. Constructors and Destructors
11. Inheritance
12. Polymorphism
13. Operator Overloading
14. Exception Handling
15. File operations

Text Books & References:

1. P. Deitel and H. Deitel, “C++ How to Program”, Pearson.
2. E. Balagurusamy, “Object Oriented Programming with C++”, TMH Publication.

3. Yashavant Kanetkar, “Let us C++”, BPB Publications
4. Robert Lafore, “Object Oriented Programming in Turbo C++”, Galgotia Publication.
5. B. Trivedi, "Programming with ANSI C++", Oxford University Press.
6. Ira Pohl, "Object Oriented Programming using C++", Pearson Education, Second Edition Reprint
7. B. Stroustrup, "The C++ Programming language", Pearson Education.
8. Timthy Budd, “An Introduction to Object Oriented Programming with C++,” Addition-Wesley.
9. Kip R. Irvine, “C++ and Object-Oriented Programming,” Prentice Hall.

BIT-14 **Software Tools-II** **2 Credits (0-0-4)**

Course Objectives:

The objectives of this course are:

1. To give overview of advanced C programming (command line arguments & file handling)
2. To give knowledge about practical usage of LINUX system calls
3. To develop capability of system call programming for practical understanding of operating system’s internal functioning

Learning Outcomes:

The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Understanding of process handling in Operating system
2. Usage of kernel system calls
3. Understanding of multiprogramming function of operating system using system calls

Experiments

1. Programs related to advanced C programming (command line arguments & file handling)
2. Understanding and usage of various LINUX kernel system calls
3. System call programming (programs based on LINUX kernel system calls using C language)