University of Engineering and Management

Institute of Engineering & Management, Salt Lake Campus

Institute of Engineering & Management, New Town Campus

University of Engineering & Management, Jaipur

3rd Semester Syllabus for BCA Admission Batch 2023

Index:

Syllabus Structure:

BCA 2nd Year Course Structure: 2023 – Odd Semester

BCA 2023 Course structure									
Semester 3									
Sl	Type of Course	Subject Code	Subject name	L	T	P	S	Total Contact Hours	Credit Points
THEORY1									
1	Computer Science and Application	BCACC301	Data Structures with C	3	1	0	0	4	4
2	Computer Science and Application	BCACC302	Object Oriented Programming With C++	3	1	0	0	4	4
3	Multidisciplinary	BCAMD301	Computer Networking	3	1	0	0	4	3
4	Ability Enhancement Course	BCAAE301	Python Programming	3	1	0	0	4	3
5	Value Added Course	BCAESP301	General Studies & Current Affairs-III	2	0	0	0	2	2
PRACTICAL									
6	Computer Science and Application	BCACC391	Data Structures Laboratory with C	0	0	4	0	4	2
7	Computer Science and Application	BCACC392	Programming Laboratory with C++	0	0	4	0	4	2
SESSIONAL									
8	Skill Enhancement Course	BCASDP381	Competitive Aptitude Training-III	0	0	0	2	2	1
9	Minor Course	BCAMS281	Business Ethics and Corporate Governance	0	0	0	4	2	4
MOOCS/MAR/IFC									
10	Value Added Course	MOOCs	Massive Open Online Course	0	0	0	0	0	0
11	Value Added Course	IFC	Industry and Foreign Certification	0	0	0	0	0	0
12	Value Added Course	MAR	Mandatory Additional Requirements(MAR)	0	0	0	0	0	0
	Total							**30**	25

University of Engineering and Management

Institute of Engineering & Management, Salt Lake Campus Institute of Engineering & Management, New Town Campus University of Engineering & Management, Jaipur

Syllabus for BCA Admission Batch 2023

Subject Name: **Data Structure with C** Credit: 4 Lecture Hours: 40

Subject Code: **BCACC301**

Pre-requisite: Fundamental computer knowledge

Relevant Links:

Study Material Coursera LinkedIn Learning NPTEL

COURSE OBJECTIVES:

1. Allow to assess how the choice of data structures and algorithm design methods impacts the performance of programs
2. To choose the appropriate data structure and algorithm design method for a specified application.
3. To solve problems using data structures such as linear lists, stacks, queues, binary trees and binary search trees writing programs for these solutions.
4. To efficiently implement the different data structures and solutions for specific problems.

COURSE OUTCOMES:

CO1: Analyze algorithms for time and computational complexity.
CO2: Select appropriate search techniques (Linear Search, Binary Search, Hashing) based on the problem's nature.
CO3: Implement and apply Stacks, Queues, and Linked Lists to various problems.
CO4: Use non-linear data structures for search, insertion, and retrieval; evaluate time complexity of balanced and unbalanced trees, and apply these to relevant problems.

Module No	Topic	Chapter Name	Sub Topic	Mapping with Industry and International Academia	Lecture Hours	Corresponding Lab Assignment
1	Introduction	An Introduction to Data Structures . Chapter 1, Data Stuctures Through C (A practical Approach) G.S.BALUJA	Overview of fundamental concepts	***International Academia:*** https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/resources/lecture-1-algorithms-and-computation/ ***IndustryMapping:****C Lion, Eclipse(IDE),Visual Studio, CLion, Clang*	6	**1.**Write a c program to print even numbers using a loop. **2.** Write a c program to print the sum of n natural numbers using a loop. **3.** Create a student database using structure. **4.** Write a C program that allows the user to input details of multiple students (name, roll number, and marks). Implement a function to search for a student by roll number and display their details if found. **5.** Write a C program to generate the Fibonacci series using recursion. Ask the user to enter the number of terms in the series and then display the Fibonacci series up to that number of terms.
	Data Structure	An Introduction to Data Structures Data Stuctures Through C (A practical Approach) G.S.BALUJA . Chapter 1,(1.10)	Definition of data structure, Data structure operation			
	Algorithms	Data Stuctures Through C (A practical Approach) G.S.BALUJA An Introduction to Data Structures . Chapter 1(1.6-Algorithms)	Complexity of Algorithms, Time Space trade off, Asymptotic Notations for Complexity of Algorithms, Subalgorithms, Role of variables in data structures and algorithms, Importance and types of data in computing			

2	Arrays	Data Stuctures Through C (A practical Approach) G.S.BALUJAArrays Chapter 3	Definition, Linear arrays, arrays as ADT, Representation of Linear Arrays in Memory, Traversing Linear arrays, Inserting and deleting, Multi-dimensional arrays, Matrices, Sparse Matrices	***International Academia:*** https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/resources/lecture-2-data-structures-and-dynamic-arrays/ ***IndustryMapping:****C Lion, Eclipse(IDE),Visual Studio, CLion, Clang*	14	**1.** Write a C program to input elements into an array of integers and calculate their sum and average. Take the input from user to enter the size of the array and its elements. Display the sum and average of the elements. **2.** Develop a C program to reverse the elements of an array of integers. Ask the user to input the size of the array and its elements. Reverse the elements of the array in-place and then display the reversed array. **3.** Create a C program to sort an array of integers in ascending order. Allow the user to input the size of the array and its elements. Implement any sorting algorithm (e.g., bubble sort, selection sort) to sort the elements, and then display the sorted array. **4.** Write a C program to check whether the given expression has balanced parentheses using a stack. The program should ask the user to input an expression containing only parentheses (such as "(({}))" or "{[()]}" etc.). Implement a stack data structure
	Linked Lists	Data Stuctures Through C (A practical Approach) G.S.BALUJALinked List Chapter 7	Linked Lists, Representation in memory, Traversing, Searching, Memory allocation (Garbage collection, overflow, underflow), Insertion, Deletion, Circular linked lists, Doubly linked lists, Header linked lists			

	Stacks	Data Stuctures Through C (A practical Approach) G.S.BALUJAStacks Chapter 5	Definition, Array representation, Linked representation, Polish notation, Evaluation of Postfix Expressions, Transforming Infix to Postfix Expressions			and use it to verify if the parentheses are balanced or not. **5.** Develop a C program to evaluate a postfix expression using a stack. Take input from user a postfix expression (e.g., "53+2*") and implement a stack-based algorithm to evaluate the expression and display the result. 6. Write a C program to create a linked list of integers. Implement functions to insert elements at the beginning, end, and at a specified position in the linked list. Also, include a function to display the elements of the linked list. **7.** Develop a C program to reverse a linked list. Implement a function to reverse the elements of a given linked list in-place and then display the reversed linked list. **8.** Develop a C program to reverse a linked list. Implement a function to reverse the elements of a given linked list in-place and then display the reversed linked list.
	Queues	Data Stuctures Through C (A practical Approach) G.S.BALUJAQueue s	Definition, Array representation, Linked representation, Circular queues,			**1.** Write a C program to implement a queue data structure. Implement functions to enqueue (insert) elements into the queue, dequeue (remove) elements from

		Chapter 6	Priority Queue, D-Queue			the queue, and display the elements of the queue. **2**. Develop a C program to implement a circular queue data structure. Implement functions to enqueue, dequeue, and display elements in a circular queue. **3.** Design a C program to implement a queue using a linked list. Implement functions to enqueue, dequeue, and display elements in the linked list-based queue. **4.** Create a C program to implement a priority queue using arrays. Implement functions to enqueue elements with priority, dequeue elements based on their priority, and display the elements of the priority queue. **5.** Write a C program to simulate the functioning of a ticket counter using a queue. Implement functions to simulate customers arriving at the ticket counter and getting served. Display the sequence of customers served.
3	Trees	Trees Chapter 8	Introduction and Definition of Trees, Tree Terminology	***International Academia:*** 1. https://ocw.mit.edu/courses/6-006-introduction-to-	10	**1.** Write a C program to create a binary search tree (BST) of integers. Implement functions to insert elements into the BST, perform inorder, preorder, and
	Binary Tree	Trees Chapter 8	Representing Binary Trees in			

			Memory, Traversing Binary Tree: Preorder traversal, In-order traversal, Post-order traversal, Traversal algorithms using stacks	algorithms-spring-2020/resources/lecture-6-binary-trees-part-1/ 2. https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/resources/lecture-7-binary-trees-part-2-avl/ ***IndustryMapping:****C Lion, Eclipse(IDE),Visual Studio, CLion, Clang*		postorder traversals, and display the elements accordingly. **2.** Develop a C program to calculate the height of a binary tree. Implement a function to calculate the height (depth) of the binary tree and display the result. **3.** Create a C program to find the mirror image of a binary tree. Implement a function to convert the given binary tree into its mirror image and display the mirror image tree. 4. Design a C program to perform operations on a binary search tree (BST). Implement functions to search for an element in the BST, delete an element from the BST, and display the resulting BST after each operation. **5.** Write a C program to perform level order traversal of a binary tree. Implement a function to traverse the binary tree level by level and display the elements in each level.
	Binary Search Trees	Trees Chapter 8	Searching in Binary Search Trees, Inserting in Binary Search Trees, Deleting in a Binary Search Tree			
	Advanced Tree Structures	Data Stuctures Through C (A practical Approach) G.S.BALUJATrees Chapter 8	AVL trees and implementations, M-trees, B-Trees (definition only)			
4	Searching	Searching and Hashing Chapter 8	Sequential Search, Binary Search, Indexed Search	***International Academia:*** 1. https://ocw.mit.edu/courses/6-006-introduction-to-	10	**1.** Write a C program to perform linear search on an array of integers. Implement a function to search for a given element in the array using linear search. Ask the user to input the size of the array
	Sorting	Sorting Chapter 10	Introduction and Notation,			

			Insertion Sort, Selection Sort, Shell Sort, Divide And Conquer, Merge sort for Linked List, Quick sort for Contiguous List.	algorithms-spring-2020/resources/lecture-3-sets-and-sorting/ 2. https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/resources/lecture-4-hashing/ 3. https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/resources/lecture-5-linear-sorting/		and its elements, as well as the element to be searched. Display the index of the element if found, otherwise indicate that the element is not present in the array. **2.** Develop a C program to perform binary search on a sorted array of integers. Implement a function to search for a given element in the sorted array using binary search. Ask the user to input the size of the array and its elements in ascending order. Then, ask for the element to be searched and use the binary search algorithm to find its index. Display the index of the element if found, otherwise indicate that the element is not present in the array. **3.** Write a C program to sort an array of integers in ascending order using the bubble sort algorithm. Implement a function to perform bubble sort on the array. Ask the user to input the size of the array and its elements. After sorting, display the sorted array. **4.** Develop a C program to sort an array of integers in ascending order using the merge sort algorithm. Implement a function
	Hashing	Data Stuctures Through C (A practical Approach) G.S.BALUJASearching and Hashing Chapter 8	Sparse Tables, Choosing a Hash function, Collision Resolution with Open Addressing, Collision Resolution by Chaining.			

						to perform merge sort on the array. Ask the user to input the size of the array and its elements. After sorting, display the sorted array. **5.** Write a C program to implement a hash table data structure. Implement functions to insert key-value pairs into the hash table, retrieve the value associated with a given key, and handle collisions using chaining. Ask the user to input keys and values, and demonstrate the functionality of the hash table

TEXT BOOK:

1. **Seymour Lipschutz, "Data Structures with C", Schaum's out Lines, Tata Mc Graw Hill, 2011.**
2. **Robert Kruse, C.L.Tondo, Bruce Leung,Shashi Mogalla,"Data Structures and Program Design using C", Pearson Education, 2009.**

REFERENCE BOOKS:

1. **Mark Allen Weiss," Data Structures and Algorithm Analysis in C", Second Edition, Pearson Education, 2013**
2. **Reema Thareja, "Data Structures Using C", Oxford Universities Press 2014, 2nd Edition.**
3. **Alfred V Aho, John E Hopcroft and Jeffery D Ullman, " Data Structures and Algorithms", Pearson Education.**
4. **Samiran Chattopadhyay, Debabrata Ghosh Dastidar and Matagini Chattopadhyay, " Data Structures through C Language", BPB Publication.**

List of equipment/apparatus for laboratory experiments:

Sl. NO	Requirements
1.	Computer with moderate configuration
2.	A programming language compiler

University of Engineering and Management

Institute of Engineering & Management, Salt Lake Campus Institute of Engineering & Management, New Town Campus University of Engineering & Management, Jaipur

Syllabus for BCA Admission Batch 2023

Subject Name: **Object Oriented Programming with C++** **Credit: 4** **Lecture Hours: 40**
Subject Code: **BCACC302**

Pre-requisite: Basics knowledge of programming language, Logic building skills.

Relevant Links:

Study Materials Coursera LinkedIn Learning Infosys Springboard NPTEL

COURSE OBJECTIVES:

1. To introduce students to a powerful programming language
2. To understand the basic structure of object oriented program
3. To gain knowledge of various programming errors
4. To enable the students to make flowchart and design an algorithm for a given problem
5. To enable the students to develop logics and programs

COURSE OUTCOMES:

CO 1: Students will be able to learn different programming techniques using object-oriented technology with C++.
CO 2 Students will be able to learn how to solve real life problems by implementing data, security, reuse of code, polymorphism etc.
CO 3: Students will be able to learn how to solve real life problems by using pointer, virtual function and FILE handling.
CO 4: Students will be able to learn how to solve real life problems by implementing exception handling and generic programming.

Module number	Topic	Chapter Name	Sub-Topics	Mapping with Industry and International Academia	Lecture Hours	Corresponding Lab Assignment
1	Introduction of Object Oriented Programming, Introduction of C++ Tokens and Keywords, Function	E. Balagurusamy, Object Oriented Programming with C++, Tata Mc Graw Hill, Fourth Edition (1,2,3,4)	A look of object oriented programming, Object oriented programming paradigm, Basics of object oriented programming, Benefits of object oriented programming, Object oriented languages, Applications of object oriented programming, Application of C++, A simple C++ program, More C++ statements, An example with Class, Structure of C++ program, Creating a source file, Compiling and linking Tokens, Keywords, Identifiers and constants, Basic data types, User defined data types, Derived data types, Symbolic constants, Type compatibility, Declaration of variables, Dynamic initialization of variables, Reference variables, Operators in C++,Scope resolution operator, Member dereferencing operators, Memory management operators, Manipulators, Type cast operator, Expressions and their types, Special assignments expressions, Implicit conversions, Operator overloading, Operator precedence, Control structures Introduction, The main function, Function prototyping, Call by reference, Return by reference, Inline functions, Default arguments, Const. arguments, Function overloading, Friend and virtual functions,	***International Academia:*** https://ocw.mit.edu/courses/6-096-introduction-to-c-january-iap-2011/ ***Industry Mapping:*** *CLion, Eclipse (IDE), Sublime Text, Atom (Text Editors), Visual Studio Debugger (Debuggers)*	8	1. Console based C++ program to display, insert, search and sort elements of an array, prime no, factorial, Fibonacci series, palindrome etc.

			Math library function			
2	Introduction to class, Constructor and Destructor, Operator Overloading, Inheritance	E. Balagurusamy, Object Oriented Programming with C++, Tata Mc Graw Hill, Fourth Edition (5,6,7,8)	Introduction, Specifying a class, Defining a member functions, A C++ program with class, Making an outside function inline, Nesting a member functions, Private member function, Arrays within a class, Memory allocation for objects, Static data members, Static member functions, Arrays of objects, Objects as function arguments, Friendly function, Returning objects, Const member functions, Pointers to member, Local classes. Constructors, Parameterized Constructors, Multiple Constructors in a class, Constructors with default arguments, Dynamic initialization of objects, Copy Constructors, Dynamic Constructors, Constructing two dimensional array, Const objects, Destructors, Defining operator overloading, Overloading unary operators, Overloading binary operators, Overloading unary operators using friends, Manipulation of string using operators, Rules for Overloading operators, Type conversions Defining derived classes, Single inheritance, Making private member inheritable, Multilevel inheritance, Multiple inheritance, Hierarchical inheritance, Hybrid inheritance, Virtual base classes, Abstract classes, Constructors in derived classes, Member classes: nesting of classes	***International Academia:*** https://ocw.mit.edu/courses/6-096-introduction-to-c-january-iap-2011/ ***International Academia:*** https://ocw.mit.edu/courses/6-s096-introduction-to-c-and-c-january-iap-2013/pages/lectures-and-assignments/c-inheritance/ ***Industry Mapping:*** *CLion, Eclipse (IDE), Sublime Text, Atom (Text Editors), Visual Studio Debugger (Debuggers)*	12	1. Program based on object, class, and encapsulation property 2. Program based on array of object 3. Program based on various (default, parameterize, copy) constructor 4. Constructor overloading program 5. Destructors program 6. Program based on various operator overloading 7. Program based on function overloading 8. Program based on friend function 9. Program based on various type of inheritance

3	Pointer and virtual function C++ Streams FILE handling in C++	E. Balagurusamy, Object Oriented Programming with C++, Tata Mc Graw Hill, Fourth Edition (9,10,11)	Introduction, Pointers, Pointers to objects, this pointer, Pointers to derived classes, Virtual functions, Pure Virtual functions C++ streams, C++ stream classes, Unformatted I/O operators, Formatted I/O operations, Managing output with manipulator Classes for File stream operations, Opening and closing a File, Detecting end-of-file, More about Open();File modes, File pointers and their manipulations, Sequential input and output operations, Updating a File: Random access, Error handling during File operations, Command-line arguments.	***International Academia:*** https://ocw.mit.edu/courses/6-096-introduction-to-c-january-iap-2011/ ***Industry Mapping:*** *CLion, Eclipse (IDE), Sublime Text, Atom (Text Editors), Visual Studio Debugger (Debuggers)*	10	1. Program based on pointer concept 2. Virtual and Pure virtual function programs 3. File and streams based programs
4	Template, Exception Handling, Standard Template Library	E. Balagurusamy, Object Oriented Programming with C++, Tata Mc Graw Hill, Fourth Edition (12,13)	Introduction to generic programming, Class templates, Class templates with multiple parameters, Function templates, Function templates with multiple parameters, Overloading of template functions, Member function templates, Non-type template arguments Introduction of exception, Basics of exception handling, Exception handling mechanism, Throwing mechanism, Catching mechanism, Rethrowing an exception, Specifying exceptions Components of STL, Containers, Algorithms, Iterators, Application of container classes, Function objects	***International Academia:*** https://ocw.mit.edu/courses/6-096-introduction-to-c-january-iap-2011/ ***Industry Mapping:*** *CLion, Eclipse (IDE), Sublime Text, Atom (Text Editors), Visual Studio Debugger (Debuggers)*	10	1. Function and class template based program 2. Program based on various type of exceptions.

Learning Resources:

TEXT BOOK:

1. **E. Balagurusamy, Object Oriented Programming with C++, Tata Mc Graw Hill, Fourth Edition, Download eBook**

REFERENCE BOOKS:

1. **Herbert Schildt, C++: The Complete Reference, Tata Mc Graw Hill, Fourth Edition,** Download eBook

List of equipment/apparatus for laboratory experiments:			
SL. No.			
1	Computer with moderate configuration		
2	g++ compiler and other software's as required		
3	C++ compiler (online)	Online compiler-1	Online compiler-2

Syllabus for BCA Admission Batch 2023

Subject Name: **Computer Networking** Credit: 3 Lecture Hours: 40
Subject Code: **BCAMD301**

Pre-requisite: Knowledge of basic data communication & network security.

Relevant Links:

Study Materials Coursera LinkedIn Learning Infosys Springboard

COURSEOBJECTIVES:

1. To deliver comprehensive view of Computer Network.
2. To enable the students to understand the Network Architecture, Network type and topologies
3. To understand the design issues and working of each layer of OSI model.
4. To familiarize with the benefits and issues regarding Network Security.

COURSEOUTCOMES:

CO1: Identify the different components in a Communication System and their respective roles.
CO2 Describe the technical issues related to the Networks.
CO3: Defining the standard model and protocols of networking
CO4: Understand the basics of data communication, networking, internet and their importance.

Module number	Topic	Chapter Name	Sub-topics	Mapping with Industry and International Academia	Lecture Hours
1	**Introduction**	**Data Communications and Networking, B. A. Forouzan,TMH(Chapter 1,Chapter 2,Chapter 3**	Introduction to communication systems, Data, signal and Transmission: Analog and Digital, Transmission modes, components, Transmission Impairments, Performance criteria of a communication system. Goals of computer Network, Networks: Classification, Components and Topology, categories of network [LAN, MAN,WAN];Internet: brief history, internet today; Protocols and standards; OSI and TCP/IP model.	***International Academia:*** (https://ocw.mit.edu/courses/6-829-computer-networks-fall-2002/resources/l2internetworking/ /) ***Industry Mapping:*** Wireshark, Network Simulator 3	5
2	**Physical Layer & Data link layer**	**Data Communications and Networking, B. A. Forouzan,TMH(Chapter 3,Chapter 4,Chapter 5,Chapter 6)**	**Physical Layer:** Overview of data[analog & digital], signal[analog & digital], transmission [analog & digital] & transmission media [guided & unguided]; Circuit switching: time division & space division switch, TDM bus; Telephone Network **Data link layer:** Types of errors, framing [character and bit stuffing], error detection & correction methods; Flow control; Protocols: Stop & wait ARQ **Medium access sub layer:** Point to point protocol, FDDI, token bus, token ring; Reservation, polling, concentration; Multiple access protocols: ALOHA, CSMA,FDMA, TDMA,	***International Standards:*** (https://ocw.mit.edu/courses/6-003-signals-and-systems-fall-2011/) (https://ocw.mit.edu/courses/6-829-computer-networks-fall-2002/resources/l0singlelink/) ***IndustryMapping:***Wireshark, Network Simulator 3	10

			CDMA; Ethernet		
3	**Network layer & Transport layer:**	**Data Communications and Networking, B. A. Forouzan,TMH (Chapter 17,Chapter 18,Chapter 22,Chapter24)**	**Network layer:** Internetworking & devices: Repeaters, Hubs, Bridges, Switches, Router, Gateway; Addressing : Internet address, Classful address, Routing: techniques, static vs. dynamic routing , Protocols: IP, IPV6. **Transport layer:** Process to process delivery; UDP; TCP; Congestion control algorithm: Leaky bucket algorithm, Token bucket algorithm, Quality of services [Qos]	***International Standards:*** (https://ocw.mit.edu/courses/6-829-computer-networks-fall-2002/resources/l2internetworking/ /) ***Industry Mapping :***Wireshark, Network Simulator 3	10
4	**Application Layer**	**Data Communications and Networking, B. A. Forouzan,TMH (Chapter 26)**	**Application Layer** DNS, SMTP, FTP, HTTP & WWW	***International Standards: (*** https://ocw.mit.edu/courses/6-829-computer-networks-fall-2002/pages/lecture-notes/) ***Industry Mapping: Industry Mapping:*** Wireshark, Network Simulator 3	10
5	**Network Security**	**Data Communications and Networking, B. A. Forouzan,TMH (Chapter 32)**	Security: Cryptography [Public, Private Key based], Digital Signature, Firewalls [technology & applications]	***International Standards: (***https://ocw.mit.edu/courses/6-829-computer-networks-fall-2002/resources/l7security/) ***Industry Mapping:*** Wireshark, Network Simulator 3	5

TEXTBOOK:

1. **B. A. Forouzan, Data Communications and Networking, Tata Mc Graw Hill.**
2. **A. S.Tanenbaum, Computer Networks, Pearson Education/PHI**

REFERENCEBOOKS:

1. **W. Stallings , Data and Computer Communications, PHI/Pearson Education**

Subject Name: **Python Programming** Credit: 3 Lecture Hours: 40
Subject Code: **BCAAE301**
Pre-requisite: Basics of programming language, Logic building skills.

Relevant Links:

Study material Linkedin Learning Infosys Springboard Coursera

COURSE OBJECTIVES:

Throughout the course, students will be expected to demonstrate their understanding of Python Programming by being able to do each of the following:

1. To apply fundamental concepts of Python Programming for problem solving
2. To understand the fundamentals different data types, operators and conditional operators in python.
3. To understand and apply and work with string, list, tuple and dictionary.
4. To apply logical reasoning to design programs using object oriented programing and use python modules and packages.

COURSE OUTCOMES:

CO1: Understand the importance and basic concepts of Python Programming and be able to apply them in problem solving
CO2: To understand basic concepts of flow control statements and concept of iterators and able to apply flow control statements to solve problems
CO3: To get familiarize and understand basic set and dictionary operations and be able to apply the concept for solving real word problems
CO4: Understand some basic properties of object oriented programming in python, and be able to analyze practical examples.

Module number	Topic	Chapter Name	Sub-Topics	Mapping with Industry and International Academia	Lecture Hours
1	Introduction to Python Programming	Python Programming, S.SRIDHAR, J INDUMATI VM HARIHARAN Chapter No : 3	Introduction to Python Programming: Python variables, expressions, statements Variables, Keywords, Operators & operands, Expressions, Statements, Order of operations, String operations, Comments, Keyboard input, Example programs Functions Type conversion function, Math functions, Composition of functions, Defining own function, parameters, arguments, Importing functions, Example programs	International Standards: https://ocw.mit.edu/courses/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/resources/lecture-1-what-is-computation/ Industry Mapping: TensorFlow PyTorch: scikit-learn: NumPy and) ,Pandas, **Generative Adversarial Networks (GAN)**	10
2	Control flow, Iterators & Functions	Python Programming, S.SRIDHAR, J INDUMATI VM HARIHARAN Chapter No: 4,5,6	**Control flow, Iterators**: Conditions and iterations Modulus operator, Boolean expression, Logical operators, if, if- else, if-elif-else, Nested conditions, Example programs Iteration while, for, break, continue, Nested loop, Example programs **Function** :Function	International Standards: 1.https://ocw.mit.edu/courses/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/resources/lecture-2-branching-and-iteration/	10

			Definition, Call, Signature, Function Python Recursive Function	2.https://ocw.mit.edu/courses/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/resources/lecture-4-decomposition-abstraction-and-functions/ Industry Mapping TensorFlow PyTorch: scikit-learn: NumPy and ,Pandas Jupyter Notebooks , Prompt Engineering	
3	Strings, List, Dictionaries, Tuples	Python Programming, S.SRIDHAR, J INDUMATI VM HARIHARAN Chapter No: 7,8,9	**Strings**: Accessing values in string, Updating strings, Slicing strings, String methods – upper(), find(), lower(), capitalize(), count(), join(), len(), isalnum(), isalpha(), isdigit(), islower(), isnumeric(), isspace(), isupper() max(), min(), replace(), split(), Example programs **List**: Introduction, Traversal, Operations, Slice, Methods, Delete element, Difference between lists and strings, Example program **Dictionaries** : Introduction, Brief idea of dictionaries & lists	International Standard: 1. https://ocw.mit.edu/courses/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/resources/lecture-3-string-manipulation-guess-and-check-approximations-bisection/ 2. https://ocw.mit.edu/courses/6-0001-introduction-to-computer-science-and-programming-in-python-fall-	10

			Tuples: Introduction, Brief idea of lists & tuples, Brief idea of dictionaries & tuples	2016/resources/lecture-5-tuples-lists-aliasing-mutability-and-cloning/ Industry mapping: Django and Flask. FastAPI, Selenium Ansible, Docker,Fabric Kubernetes, NLP	
4	Modules, Packages, Object Oriented Programming	Python Programming, S.SRIDHAR, J INDUMATI VM HARIHARAN Chapter No 12, 15	Python Modules, Python Packages, Creating class, Instance objects, Accessing attributes, Built in class attributes, destroying objects, Inheritance, Abstraction, Polymorphism Example program	International Standard: 1. https://ocw.mit.edu/courses/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/resources/lecture-8-object-oriented-programming/ Industrymapping: Apache Spark: PySpark. Apache Airflow: Apache Kafka	10

Learning Resources:

TEXTBOOK:

1. **S.SRIDHAR, J INDUMATI VM HARIHARAN, Python Programming, PEARSON. Download eBook**

REFERENCEBOOKS:

1. **Veena Ghuriani, Akansha Gautam, Vaishali Chawla, Python Programming, A comprehensive Approach, PHI/Pearson Education, Fourth Edition,** Download eBook
2. **Kenneth A Lambert, Fundamentals of Python, First Programs, 2nd Edition**