

**Open Elective Courses
Syllabus
Semester V**

Program:	B. Tech. (Computer)			Semester: V			
Course:	Data Structures Using Python (OEC-2)			Code: BCE5601			
Teaching Scheme				Evaluation Scheme			
Lecture	Practical	Tutorial	Credit	IE	MTE	ETE	Total
03	-	-	03	20	30	50	100
Prior knowledge of Python Programming is essential.							
Course Objectives: <ol style="list-style-type: none"> To understand Python Specific Data Structures. To illustrate and demonstrate Stacks, Queues. To understand how searching and sorting is performed in Python. To understand how linear and non-linear data structures work. To learn the fundamentals of writing Python scripts. To learn the operations on tree and graph data structure. 							
Course Outcomes: <p>After learning the course, students will be able to:</p> <ol style="list-style-type: none"> Differentiate the type of data structure. Create, run and manipulate Python Programs using core data structures like Lists. Comprehend the searching & sorting algorithms. Apply suitable data structures to solve the programming problems. Use effective and efficient data structures in solving various Computer Engineering domain problems. Comprehend nonlinear data structures such as tree and graph. 							
Detailed Syllabus							
Unit	Description						Duration (H)
I	Introduction to Data Structures Introduction to Python programming, Data Structures – Definition, Linear Data Structures, on-Linear Data Structures, Python Specific Data Structures - List, Tuples, Set, Dictionaries, Comprehensions and its Types, Strings, slicing. Arrays - Overview, Types of Arrays, Operations on Arrays, Arrays vs. List.						06
II	Searching and Sorting Techniques Searching - Linear Search and Binary Search Sorting - Bubble Sort, Selection Sort, Insertion Sort, Merge Sort and Quick Sort.						06
III	Linked List Linked Lists – Introduction, Implementation of Singly Linked Lists, Doubly Linked Lists, Circular Linked Lists						06
IV	Stack & Queue Stacks - Overview of Stack, Implementation of Stack, Applications of Stack, Queues- Overview of Queue, Implementation of Queue, Applications of Queues, Priority Queues.						06

V	Tree Trees - Overview of Trees, Tree Terminology, Binary Trees - Introduction, Implementation. Tree Traversals, Binary Search Trees - Introduction	06
VI	Graph Introduction, directed vs. Undirected Graphs, Weighted vs. Unweighted Graphs, Representations - Adjacency Matrix, Adjacency list, Graph Traversals - Breadth First Search, Depth First Search.	06
	Total	36
Text Books:		
<ol style="list-style-type: none"> 1. Data structures and algorithms in python by Michael T. Goodrich, ISBN-13: 978-1118290279, ISBN-10: 1118290275, Publisher: Wiley; 1st edition (March 18, 2013). 2. Problem Solving with Algorithms and Data Structures Using Python by Bradley N Miller and David L. Ranum. ISBN-13: 978-1590282571, ISBN-10: 1590282574, Publisher: Franklin, Beedle & Associates; 2nd edition (August 22, 2011). 		
Reference Books:		
<ol style="list-style-type: none"> 1. Hands-On Data Structures and Algorithms with Python: Write complex and powerful code using the latest features of Python 3.7, 2nd Edition by Dr. Basant Agarwal, Benjamin Baka. ISBN: 9781788991933, 2018. 2. Core Python Programming -R. Nageswara Rao, ISBN-10: 9789351199427, ISBN-13: 978-9351199427, Willy; 1st edition (January 1, 2016). 		

Program:	B. Tech. (Computer)			Semester: V			
Course:	Programming with C++ (OEC-2)			Code: BCE5602			
Teaching Scheme				Evaluation Scheme			
Lecture	Practical	Tutorial	Credit	IE	MTE	ETE	Total
03	-	-	03	20	30	50	100
<p>Prior knowledge of Python Programming is essential.</p>							
<p>Course Objectives:</p> <ol style="list-style-type: none"> 1. To explore the principles of Object-Oriented Programming (OOP). 2. To use the concept of inheritance and polymorphism. 3. To understand the use of exception handling in C++ programs. 4. To provide a foundation for advanced programming using File handling and STL. 5. To provide lifelong learning attitude towards problem solving. 							
<p>Course Outcomes:</p> <p>After learning the course, students will be able to:</p> <ol style="list-style-type: none"> 1. Compare the strengths of object-oriented programming with respect to procedural programming. 2. Demonstrate working with primitive data types. 3. Understand and demonstrate dynamic memory management techniques. 4. Analyze and apply the concept of function overloading & operator overloading for real time problem solving. 5. Classify inheritance with the understanding of early and late binding, usage of exception handling, generic programming. 6. Demonstrate the use of various advanced object-oriented concepts with the help of programs. 							
Detailed Syllabus							
Unit	Description						Duration (H)
I	<p>Introduction of OOPs Concepts</p> <p>Overview of procedural programming and object-oriented programming, Syntax of variables declaration, Classes and objects, Member functions, memory management. Case Study: Write a program in c++ to create an employee class with appropriate features.</p>						06
II	<p>Inheritance</p> <p>Introduction, benefits, Access specifiers, Types of inheritance - single, multiple, multilevel, hybrid and hierarchical. Case Study: Write a program in c++ to derive class bicycle from class vehicle with appropriate syntax.</p>						06
III	<p>Polymorphism</p> <p>Introduction, Types of polymorphism: function and operator, Virtual functions, Pure virtual functions, Virtual base class, Overloading and overriding. Case study: Write a program in c++ to overload '+' and '-' operator.</p>						06
IV	<p>Exception Handling</p> <p>Introduction to exception, Benefits of exception handling, try, throw and catch blocks, pre-defined exceptions in c++, Re-throw. Case Study: Write a program in c++ to create a class student with name, age, roll no and telephone number as parameters. Program should throw an exception if telephone_number >10.</p>						06

V	<p>File Handling</p> <p>Classes for file stream operation, Opening and closing a file, File pointers and their manipulation, File operations on binary files – variables, class objects, sequential file organization, Direct access files. Case Study: Write a program in c++ to create a database for airline reservation system using file handling.</p>	06
VI	<p>Templates</p> <p>Introduction, Function templates, Class template with multiple parameters. Introduction to STL: Introduction of STL components, Sequential container, Algorithms, Iterators. Case Study: Write a program in c++ to create vector template using STL container.</p>	06
	Total	36
<p>Text Books:</p> <ol style="list-style-type: none"> 1. E. Balagurusamy, “Object -Oriented Programming with C++”, McGraw Hill Education, Eighth Edition, Sept. 2020, ISBN-13: 978-9389949186. 2. Ivor Horton, Peter Van Weert, “Beginning C++20”, Novice Professional, Sixth Edition, 2020, ISBN-13: 978-1484258835 (ISBN-10: 1484258835) 3. Robert Lafore, “OOP in C++”, Pearson Publishing, 4th Edition, 2001, ISBN:0672323087 (ISBN 13: 9780672323089). 		
<p>Reference Books:</p> <ol style="list-style-type: none"> 1. Bjarne Stroustrup, The C++ Programming language, Third edition, 2008, Pearson Education. ISBN 9780201889543. 2. Deitel, C++ How to Program, 4 th Edition, Pearson Education,2002, ISBN:81-297-0276-2. 3. Herbert Schildt, C++ the complete reference, Eighth Edition, McGraw Hill Professional, 2011, ISBN:978-00-72226805. 		
<p>MOOC Courses:</p> <ol style="list-style-type: none"> 1. An Introduction to Programming Through C++, NPTEL, 12 weeks 		

Vision and Mission of Computer Department

Department Vision

To be a Premier Hub in Computer Engineering in Education and Research.

Department Mission

To build technologically competent and ethically strong individuals for serving the needs of industry and society by providing state-of-the-art resources, opportunities for Learning and Research in Computer Engineering.