

Course Code CSE3011	Python Programming	Course Type LP	Credits 3
Pre-requisite knowledge is required on Object-oriented programming paradigm and data structures.			
Course Objectives			
<p>This course will introduce the Python Programming language, its functionality, code constructs, and its applications. This course is devised for following objectives,</p> <ul style="list-style-type: none"> • To study object oriented paradigm in Python. • To develop their skill set using Python. • To familiarize with the functionalities and applications of Python. 			
Course Outcomes			
<p>Students will be able to solve problems, explore real-world software development challenges, and create practical and contemporary applications. At the completion of this course, students should be able to do the following:</p> <ul style="list-style-type: none"> • Understand and use the Object Oriented paradigm in Python • Use the IO model in Python to read and write disk files. • Write Python programs using collections, regular expression, classifying and categorizing text. 			
<p>Student Outcomes (SO): b, c, i, k, l</p> <p>b. An ability to analyze a problem, identify and define the computing requirements appropriate to its solution.</p> <p>c. An ability to design, implement and evaluate a system / computer-based system, process, component or program to meet desired needs</p> <p>i. Design and conduct experiment as well as analyze and interpret data.</p> <p>k. An ability to use current techniques, skills and tools necessary for computing engineering practice.</p> <p>l. An ability to apply mathematical foundations, algorithmic principles and computer science theory in the modeling and design of computer-based systems (CS)</p>			
Unit No	Unit Content	No. of hours	SOs
1	A Brief History of Python, Different Versions, Python 2 vs Python 3, Installing Python, Environment Variables, Executing Python from the Command Line, Editing Python Files, Basic Python Syntax, String Values, String Operators, Numeric Data Types Conversions, Simple Input and Output, Language components - Control Flow structures and Syntax - Relational Operators - Logical Operators - Bit Wise Operators, Python for Windows	6	b,c,i
2	Conditions, boolean logic, logical operators, ranges, Control statements: if-else, loops (for, while), Flow control, Functions, Scoping, Exceptions, Input and output, Modules, Collections, Lists, Tuples, Sets, Dictionaries, Modules, Standard Modules, Regular Expressions, Quantifiers, Basic String Operations	6	c,i
3	Principles of Object Orientation, Classes in Python, Creating Classes, Instance Methods, Access Specification, data modeling, persistent storage of objects, inheritance, polymorphism, operator overloading, abstract classes, exception handling, try block	6	c,i
4	File Handling, Writing Data to a File, Reading Data From a File - Additional File Methods: Using Pipes as Data Streams, Handling	6	c,i

	IO Exceptions, Working with Directories, Metadata, File Organization, Database Programming - Generic Database Connectivity using ODBC, Postgres connection in Python, MySQL connection in Python.		
5	Graphical user interfaces, event-driven programming paradigm, tkinter module, creating simple GUI, buttons, labels, entry fields, dialogs, widget attributes - sizes, fonts, colors layouts, nested frames, Multithreading, Networks, and Client/Server Programming, introduction to HTML, interacting with remote HTML server, running html-based queries, downloading pages; CGI programming, programming a simple CGI form.	6	c,i,k
6	Guest Lecture on Contemporary Topics	02	
	Total Hours:	32	
Mode of Teaching and Learning: Flipped Class Room, Activity Based Teaching/Learning, Digital/Computer based models, wherever possible to augment lecture for practice/tutorial and minimum 2 hours lectures by industry experts on contemporary topics			
Mode of Evaluation and assessment: The assessment and evaluation components may consist of unannounced open book examinations, quizzes, student’s portfolio generation and assessment, and any other innovative assessment practices followed by faculty, in addition to the Continuous Assessment Tests and Final Examinations.			
Text Books:			
1.	Mark Summerfield, “Programming in Python 3”, 2nd Edition, Pearson Education, 2011.		
2.	Harvey M. Deitel, “Python – How to program” , Prentice Hall, 2002		
Reference Books:			
1.	Michael Dawson, “Python Programming for the Absolute Beginner”, Third Edition, Cengage Learning, 2010.		
2.	Mark Lutz, “Learning Python”, 4th Edition, O’Reilly, 2000.		
List of Suggested Experiments:			
1	Write python program to print list of numbers using range and for loop.		
2	Write python program to print first n prime numbers.		
3	Write python program to multiply matrices.		
4	Write python program to take command line arguments (word count).		
5	Write python program in which a function is defined and calling that function prints ‘Hello World’.		
6	Write python program to let user enter some data in string and then verify data and print welcome to user.		
7	Write python program to store strings in list and then print them.		
8	Write python program to find the most frequent words in a text read from a file.		
9	Write python program in which a class is defined, then create object of that class and call simple ‘print function’ defined in class.		
10	Write python program in which an function (with single string parameter) is defined and calling that function prints the string parameters given to function.		
11	Simulate elliptical orbits in Pygame.		
12	Simulate bouncing ball using Pygame.		

Recommendation by the Board of Studies on	June 25, 2018
Approval by Academic council on	July 18, 2018
Compiled by	Dr S Raju and Dr R Ganesan