Semester: I / II						
INTRODUCTION TO PYTHON PROGRAMMING **Category: Programming Language Course** **(Common to all Programs)** **(Theory & Practice)**						
Course Code	**:**	**AI115AI / AI125AI**		**CIE**	**:**	100 Marks
Credits: L:T:P	**:**	2:0:1		**SEE**	**:**	100 Marks
Total Hours	**:**	28L+28P		**SEE Duration**	**:**	3 Hours

Unit-I	**05 Hrs**
Getting Started: Introducing Python, Setting Up Python in windows, Setting Up Python in other Operating Systems, introducing IDLE. **Types, Variable, and Simple I/O:** Using Quotes with Strings, Concatenating and Repeating Strings, Working with Numbers, Understanding the Variable, Getting User Input, Converting Values.	
Unit – II	**05 Hrs**
Branching, While Loops, and Program Planning: Using the If statement, Using the else Clause, Using the elif clause, creating while Loops, Avoiding Infinite Loops, Creating Intentional infinite Loops, Using Compound Conditions.	
Unit –III	**06 Hrs**
For Loops, Strings, and Tuples: Using for Loops, counting with the For Loops, Using Sequence Operators and Functions with Strings, Indexing Strings, Slicing the Strings, Creating the Tuple, Using Tuple. **Lists and Dictionaries:** Using Lists, Using List Methods, understanding when to use the tuple instead of Lists.	
Unit –IV	**06 Hrs**
Functions: Creating Functions, Using Parameters and Return Values, Using Keyword Arguments and Default Parameters Values, Using Global Variables and Constants. **Files and Exceptions:** Reading from Text Files, Writing to Text Files, Handling Exceptions.	
Unit –V	**06 Hrs**
Software Objects: Defining a Class, Defining Method, Instantiating an Object, invoking a Methods, Using Constructor, Using Class Attributes and Static Methods, Understanding Object Encapsulation. **Object-Oriented Programming:** Using Inheritance to Create New Classes, creating a Base Class, inheriting from a Base Class, extending a Derived Class, Using the Derived Class, extending a Class through Inheritance, Understanding Polymorphism.	

Course Outcomes: After completing the course, the students will be able to	
CO1	Apply fundamental knowledge of Python programming to solve the engineering problems.
CO2	Identify the problems in various application domains and solve them using different concepts of Python Programming.
CO3	Design a solution using Python programming with societal, environmental, and other concerns by engaging in lifelong learning for emerging technology.
CO4	Demonstrate the use of modern tools by exhibiting teamwork and effective communication skills.

Reference Books	
1	Michael Dawson, Python programming for the absolute beginner, 3rd Edition, CENGAGE, ISBN-13:978-93-86668-00-4, ISBN-10: 93-86668-00-9, 2010.
2	John V. Guttag. Introduction to Computation and Programming using Python, The MIT Press, Cambridge, Massachusetts, London, ISBN: 978-0-262-51963-2, 2013
3	Mark Summerfield, Programming in Python 3: A Complete Introduction to the Python Language, 2nd Edition, ISBN-13: 978-0-321-68056-3, ISBN-10: 0-321-68056-1.
4	Paul Gries, Jennifer Campbell, Jason Montojo, Practical Programming: An Introduction to Computer Science Using Python 3.6, 3rd Edition, The Pragmatic Bookshelf, ISBN-13: 978-1-6805026-8-8, 2017.
5	Mark Lutz, Learning Python, 5th Edition, 2013, Oreilly Media, ISBN: 978-1-449-35573-9.
6	Burkhard A. Meier, Python GUI Programming Cookbook, Packt Publishing, 2015, ISBN 978-1-78528-375-8.

Laboratory Experiments	
PART-A	
1	Introductory Lab-Installation and Working with the Sample Programs.
2	Write a program to find the largest prime factor of a given integer.
3	Write a program to find the height of the ball thrown by a basketball player.
4	Write a program to find the Golden ratio.
5	Read a paragraph from the user and count the number of words, and frequency of Words appearing, and search for the specific word.
6	Consider a sequence of numbers with some missing values. Write a python program for inserting the missing values, and remove some of the values from the sequence. Also, add a few more values to the existing sequence.
7	Create an Employee 'Employee' Database using dictionaries and perform the insert, search and display operations.
8	Implement Set and Tuple Operations.
9	Create a text file called my_file.txt with some content, capitalize the first letter of every word, and print the content of the file in reverse order.
PROGRAMMING ASSIGNMENT	
Design and develop a python GUI application connected to interested Sustainable Development Goals (SDG).	

RUBRIC FOR THE CONTINUOUS INTERNAL EVALUATION (THEORY WITH LAB)		
#	**COMPONENTS**	**MARKS**
1	**QUIZZES:** Quizzes will be conducted in online/offline mode. **TWO QUIZZES** will be conducted & Each Quiz will be evaluated for 10 Marks. **THE AVERAGE OF TWO QUIZZES WILL BE THE FINAL QUIZ MARKS.**	**10**
2	**TESTS:** Students will be evaluated in test, descriptive questions with different complexity levels (Revised Bloom's Taxonomy Levels: Remembering, Understanding, Applying, Analyzing, Evaluating, and Creating). **TWO tests will be conducted**. Each test will be evaluated for **50 Marks**. **FINAL TEST MARKS WILL BE REDUCED TO 30 MARKS.**	**30**
3	**EXPERIENTIAL LEARNING:** Students will be evaluated for their creativity and practical implementation of the problem. Case study based teaching learning (10), Program specific requirements (10), Video based seminar/presentation/demonstration (10**) ADDING UPTO 30 MARKS**.	**30**
4	**LAB:** Conduction of laboratory exercises, lab report, observation and analysis **(30 Marks)**, lab test **(10 Marks)** and Innovative Experiment/ Concept Design and Implementation (**10 Marks)** adding up to 50 Marks. **THE FINAL MARKS WILL BE REDUCED TO 30 MARKS.**	**30**
	MAXIMUM MARKS FOR THE CIE THEORY	**100**

RUBRIC FOR SEMESTER END EXAMINATION (THEORY)		
Q. NO.	**CONTENTS**	**MARKS**
	PART A	
1	Objective type questions covering entire syllabus	10
	PART B (Maximum of TWO Sub-divisions only)	
2	Unit 1 : (Compulsory)	14
3 & 4	Unit 2 : Question 3 or 4	14
5 & 6	Unit 3 : Question 5 or 6	14
7 & 8	Unit 4 : Question 7 or 8	14
9 & 10	Unit 5 : Question 9 or 10	14
11	Lab Component (Compulsory)	20
	MAXIMUM MARKS FOR THE SEE THEORY	**100**