

COURSE CODE	COURSE TITLE	L	T	P	C
1153CS301	PYTHON PROGRAMMING LAB	0	0	4	2

Course Category: Allied Elective

A. Preamble:

This course focuses on developing the python programming to do a variety of programming tasks where the students are encouraged to develop application using pygame. At the end of the course the student will be developing adequate skills in programming and will be known to understand the implementation of various applications using python.

B. Pre-requisite:

Sl. No	Course Code	Course Name
1	1150CS201	Problem Solving using C

C. Related Courses:

Sl. No	Course Code	Course Name
		NIL

D. Course Outcomes

At the end of the course, the students are able to:

CO Nos.	Course Outcomes	Level of learning domain (Based on revised Bloom's taxonomy)
CO1	Write, Test and Debug Python Programs	K3, S3
CO2	Implement Conditionals and Loops for Python Programs	K3, S3
CO3	Use functions and represent Compound data using Lists, Tuples and Dictionaries	K3, S3
CO4	Read and write data from & to files in Python and develop Application using Pygame	K3, S3

K3-Apply, S3-Processes

E. Correlation of COs with POs :

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	M	H			H				L				L	M	
CO2	M	H			H				L					M	
CO3	M	H			H				L						M
CO4	H	H	H		H				H		H			H	H

F. Course Content

LIST OF EXPERIMENTS

1. Running instructions in Interactive interpreter and a Python Script.
2. Write a program to purposefully raise Indentation Error and correct it.
3. Scientific problem solving using decision making and looping.
4. Simple programming for one dimensional and two dimensional arrays.
5. Python Programming to explore string functions.
6. Utilizing 'Functions' in Python
 - Find mean, median, mode for the given set of numbers in a list.
 - Write a function dups to find all duplicates in the list.
 - Write a function unique to find all the unique elements of a list.
 - Write function to compute gcd, lcm of two numbers
7. Demonstrate the use of Lists, Dictionaries.
8. Write a program to implement Turtle.
9. Building a Classical Data Structure using Python Programming.
10. Implement Searching Operations: Linear and Binary Search.
11. To sort the 'n' numbers using: Selection and Insertion Sort.
12. Find the most frequent words in a text read from a file.
13. Demonstrate Exceptions in Python.
14. Simulate elliptical orbits and bouncing ball using Pygame

F. Learning Resources:

i. Reference Books:

1. Allen B. Downey , “ Think Python: How to Think Like a Computer Scientist”, Second Edition, Updated for Python 3, Shroff/O'Reilly Publishers, 2016.
2. Shroff “Learning Python: Powerful Object-Oriented Programming; Fifth edition, 2013.
3. David M. Baezly “Python Essential Reference”. Addison-Wesley Professional; Fourth edition, 2009.
4. David M. Baezly “Python Cookbook” O'Reilly Media; Third edition (June 1, 2013) by.
5. <http://www.edx.org>