

<b>Subject Code : 1CS2010306</b>	<b>Subject Title: PROGRAMMING WITH PYTHON</b>
<b>Pre-requisite :</b>	Knowledge of some programming language like C, Java

**Course Objective:**

The objectives of the course are to:

- To develop proficiency in creating based applications using the Python Programming Language.
- To be able to understand the various data structures available in Python programming language and apply them in solving computational problems.
- To be able to do testing and debugging of code written in Python.
- To be able to draw various kinds of plots using PyLab.
- To be able to use generators for generating series like Fibonacci.

Teaching Scheme (Hours per week)				Evaluation Scheme (Marks)				
Lecture	Tutorial	Practical	Credit	Theory		Practical		Total
				University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	
3	-	3	6	60	40	30	20	150

Subject Contents			
Sr. No	Topic	Total Hours	Weight (%)
1	<b>Introduction to Python:</b> The basic elements of Python, Branching programs, Strings and Input, Iteration <b>Functions, Scoping and Abstraction:</b> Functions and Scoping, Specifications, Recursion, Global variables, Modules, Files	07	20
2	<b>Testing and Debugging:</b> Testing, Debugging <b>Structured Types, Mutability and Higher-order Functions:</b> Tuples, Lists and Mutability, Functions as Objects, Strings, Tuples and Lists, Dictionaries	07	20
3	<b>Exceptions and assertions:</b> Handling exceptions, Exceptions as a control flow mechanism, Assertions <b>Classes and Object-oriented Programming:</b> Abstract Data Types and Classes, Inheritance, Encapsulation and information hiding	07	20
4	<b>Some Simple Algorithms and Data Structures:</b> Search Algorithms, Sorting Algorithms, Hash tables <b>Plotting and more about Classes:</b> Plotting using PyLab, Plotting mortgages and extended examples	10	25
5	<b>Dynamic Programming:</b> Fibonacci sequence revisited, Dynamic programming and the 0/1 Knapsack algorithm, Dynamic programming and divide and conquer	05	15

**Course Outcome:**

At the end of this course, the student would be able

- To create robust applications using the Python programming language
- To test and debug applications written using the Python programming language
- To create applications for solving computational problems using the Python Programming Language.

**List of References:**

1. John V Guttag. "Introduction to Computation and Programming Using Python", Prentice Hall of India
2. Allen Downey, Jeffrey Elkner and Chris Meyers "How to think like a Computer Scientist, Learning with Python", Green Tea Press
3. Swaroop C H. "A Byte of Python", <http://www.swaroopch.com/notes/python>
4. "Python Programming", [http://en.wikibooks.org/wiki/Python\\_Programming](http://en.wikibooks.org/wiki/Python_Programming)
5. "The Python Tutorial", <http://docs.python.org/release/3.0.1/tutorial/>
6. "Learn Python the Hard way", <http://learnpythonthehardway.org/>

**Suggestions for Lab Sessions :**

Labs may be done using version 3.0 of the Python Programming Language. The main text book has sufficient problems for lab exercises.

**Practical List:**

1	Write a Python Program to Convert Celsius to Fahrenheit and vice a versa.
2	Write a program in python to swap two variables without using temporary variable.
3	Write a Python Program to Convert Decimal to Binary, Octal and Hexadecimal
4	Write a program to make a simple calculator (using functions).
5	Write a program in python to find out maximum and minimum number out of three user entered number.
6	Write a program which will allow user to enter 10 numbers and display largest odd number from them. It will display appropriate message in case if no odd number is found.
7	Write a Python program to check if the number provided by the user is an Armstrong number or not.
8	Write a Python program to display all the prime numbers in user entered range.
9	Write a Python program to check if the number provided by the user is a palindrome or not.
10	Write a Python program to perform following operation on given string input: a) Count Number of Vowel in given string b) Count Length of string (donot use len() ) c) Reverse string d) Find and replace operation e) check whether string entered is a palindrome or not
11	Define a procedure histogram() that takes a list of integers and prints a histogram to the screen. For example, histogram([4, 9, 7]) should print the following: **** ***** *****
12	Write a program in python to implement Fibonacci series up to user entered number. (Use recursive Function)
13	Write a program in python to implement Factorial series up to user entered number. (Use recursive Function)
14	Write a program in python to implement simple interest and compound interest values on chart using PyLab. Show the difference between both. (Note: Use of object oriented paradigm is compulsory.)
15	Write a program in Python to implement read lines, write line using file handling mechanisms.
16	Write a program in python to implement Salary printing file read operation. (File format: EmployeeNo, name,

	deptno, basic, DA, HRA, Conveyance) should perform below operations. a) Print Salary Slip for given Employee Number b) Print Employee List for Given Department Number
<b>17</b>	1) Write a program in python to implement Railway Reservation System using file handling technique. System should perform below operations. a. Reserve a ticket for a passenger. b. List information all reservations done for today's trains. (Note: Use of object oriented paradigm is compulsory.)
<b>18</b>	Write a program in python to implement Library Management System using file handling technique. System should perform below operations. a. Issue a book for student. b. List information today's issued books. (Note: Use of object oriented paradigm is compulsory.)
<b>19</b>	Write a program in python to implement Bank System using Class and Methods and perform below Operations. (Note: Use of object oriented paradigm is compulsory.) a) Add Bank account b) Deposit of money c) withdrawal operation d) Money transfer e) Show Balance
<b>20</b>	Write a program in python to implement Stack using Class and Methods and perform below operations. (Note: Use of object oriented paradigm is compulsory.) a) Create Stack b) Pop c) Push d) Merge two stack e) List element
<b>21</b>	Write a program in python to implement Queue using Class and Methods and perform below operations. (Note: Use of object oriented paradigm is compulsory.) a) Create Queue b) Add an element c) Remove an element d) Merge two Queues e) List elements
<b>22</b>	Write a program in python to implement Queue using Class and Methods and perform below operations. (Note: Use of object oriented paradigm is compulsory.) a) Create Queue b) Add an element c) Remove an element d) Merge two Queues e) List elements
<b>23</b>	Write a program in python to implement Merge Sort. (Note: Use of object oriented paradigm is compulsory)
<b>24</b>	Write a program in python to implement Bubble Sort. (Note: Use of object oriented paradigm is compulsory)
<b>25</b>	Write a program in python to implement 0/1 Knapsack algorithm. (Note: Use of object oriented paradigm is compulsory.)
<b>26</b>	Write a program in python to implement Fibonacci Series and Factorial using memo. (i.e. FastFibo & FastFact) (Note: Use of object oriented paradigm is compulsory.)