Department of Electronic Sciences

BSc. (Hons.) Electronic Sciences
Category-I

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Programming Fundamentals using Python ELDSC-1	**4**	**3**	**0**	**1**	**Course Admission Eligibility**	**Nil**

Learning Objectives

The Learning Objectives of this course are as follows:
This course introduces the student to the fundamental understanding of the Python programming language. The main objective is to help students learn to use the Python programming language to solve problems of interest to them. It introduces the core programming basics including data types, operators, input/output, control structures, iterative and recursive constructs, compound data types, and program design with functions. The course also discusses the fundamental principles of Object-Oriented Programming (OOP), as well as comprehensive data and information processing technique.

Learning outcomes

The Learning Outcomes of this course are as follows:

CO1 Read, write and debug python programs to solve computational problems.

CO2 Select and use a suitable programming construct and data objects like lists, sets, tuples and dictionaries for solving a given problem.

CO3 Be proficient in the handling of strings and functions

CO4 Use Python libraries

CO5 Articulate OOP concepts such as encapsulation, inheritance and polymorphism and use them in applications

SYLLABUS OF DSC-1

UNIT – I Starting with Python (12 Hours)

Introduction to Python: Python Interpreter-IDLE (script and interactive mode), Python shell, using Python as calculator, concept of data types; variables, Identifiers and keywords, Literals, Strings, Operators (Arithmetic operator, Relational operator, Logical operator, Boolean operator, Assignment operators, Membership operators(in and not in), Identity operators, Bit wise operator, Increment or Decrement operator), comments in the program, understanding error messages.
Creation of a Python Program: Input and Output Statements, Control statements -Branching (if-else, if-elif-else), indentation in python, iteration (using for, while), Conditional Statement, exit function, Difference between break, continue and pass, Nested conditionals

UNIT – II Strings and Lists (12 Hours)
Data objects in Python: Mutable and immutable
Strings- Creating and Storing Strings, Accessing Characters in String by Indexing (positive and negative), String Operations: concatenation, replication (*), membership, comparison, Slicing, string built-in functions, String method
Lists- Creating Lists, Accessing list elements, traversing a list , Aliasing a list, comparing list , list Operations:- concatenation, replication(*), membership, slicing, Indexing, nested list, list built-in functions List methods , del statement.
Sets: Creating sets, Sets built-in functions, Set Methods

UNIT – III Tuples and Dictionaries (12 Hours)
Tuples: Creating Tuples, Tuple operations: slicing, concatenation, replication, membership, comparing and deletion, tuple built-in functions
Dictionaries: Dictionary in python (key : value pairs), creating a dictionary, element accessing and traversing a dictionary, appending values, updating values, removing items from dictionary, membership, dictionary built-in functions, dictionary methods , clear statement
Object Oriented Programming: Introduction to Classes, Objects and Methods, Encapsulation, Inheritance, Polymorphism, Abstraction

UNIT – IV Functions and Modules (12 Hours)
Functions: Built in function (math, statistics), User defined functions: Defining Functions, arguments: positional, default, keyword, variable length arguments, scope of variables, parameter passing (string list, dictionary, tuples, sets), return statement, recursion, importing (using import) user defined function (path).
Modules in python: use of keyword from, namespacing, module aliasing, introduction to python packages (matplotlib, pandas, numpy, scikitlearn, nltk, openCV) and libraries and their applications

Practical component (if any) – Programming Fundamentals using Python Lab (30 Hours)

Learning outcomes
CO1 Develop algorithms and write programs in Python language for arithmetic and logical operations, conditional branching.
CO2 Write programs in Python language using construct and data objects like strings, lists, sets, tuples, dictionaries, Python libraries and use concept of OOP.
CO3 Prepare the technical report on the experiments carried.

1. Write a python menu driven program to calculate area of circle, rectangle, square using if-elif-else.
2. Write a python program to print Fibonacci series up to a certain limit (use 'while').
3. Write a python program to print the Pascal triangle.
4. Write a python program to find HCF (GCD) of two numbers.
5. Write a python program to find LCM of two numbers.
6. Write Python programs to illustrate the various functions of the "Math" module, "Statistics" module in Python.
7. Write a Python program to count number of vowels using sets in given string
8. Write a Python program to Remove all duplicates from a given string in Python
9. Write a Python program to count positive and negative numbers in a list
10. Write a Python program to find sum of elements in list
11. Write a python program to read a list of 'n' integers (positive and negative) and create two new lists one having all positive numbers and the other having all negative numbers from the given list. Print all three lists.
12. Write a python program to create a list of tuples from given list having number and its cube in each tuple
13. Create a Python program to create a dictionary which has record of a student information: Admission number, Roll Number, Name and Marks. Display information on the basis of Admission number
14. Write a python program which contains user defined functions as a 'module' to calculate area, perimeter or surface area , volume for various shapes like square, cube, circle, cylinder. The user defined functions should accept the values for calculation as parameters and calculated values should be returned. Import the module and use appropriate functions.
15. Create a menu driven Python program using user defined functions to implement a calculator to perform :

(a) Basic arithmetic operations
(b) log10(x), sin(x) , cos(x)

Note: Students shall sincerely work towards completing all the above listed practicals for this course. In any circumstance, the completed number of practicals shall not be less than twelve.

Essential/recommended readings

1. Allen B. Downey, "Think Python: How to Think Like a Computer Scientist", 2nd edition, Updated for Python 3, Shroff/O'Reilly Publishers, 2016 (http://greenteapress.com/wp/thinkpython/)
2. Guido van Rossum and Fred L. Drake Jr, —An Introduction to Python – Revised and updated for Python 3.2, Network Theory Ltd., 2011.
3. John V Guttag, —Introduction to Computation and Programming Using Python", Revised and expanded Edition, MIT Press , 2013
4. Robert Sedgewick, Kevin Wayne, Robert Dondero, —Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson India Education Services Pvt. Ltd., 2016.

5. Timothy A. Budd, —Exploring Python‖, Mc-Graw Hill Education (India) Private Ltd.,, 2015.

Suggestive readings

1. Kenneth A. Lambert, —Fundamentals of Python: First Programs‖, CENGAGE Learning, 2012.
2. Charles Dierbach, —Introduction to Computer Science using Python: A Computational Problem-Solving Focus, Wiley India Edition, 2013.
3. Paul Gries, Jennifer Campbell and Jason Montojo, —Practical Programming: An Introduction to Computer Science using Python 3‖, Second edition, Pragmatic Programmers, LLC, 2013.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DISCIPLINE SPECIFIC CORE COURSE – 2 (DSC-2): Circuit Theory &

Credit distribution, Eligibility and Prerequisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Circuit Theory & Network Analysis ELDSC-2	4	3	0	1	Course Admission Eligibility	Nil

Learning Objectives

The Learning Objectives of this course are as follows:

- To study the basic circuit concepts in a systematic manner suitable for analysis and design.
- To study the steady state analysis of AC Circuits.
- To study and analyse electric circuits using network theorems.
- To study and design passive filters using R, L and C

Learning outcomes

The Learning Outcomes of this course are as follows:

CO1 Study basic circuit concepts in a systematic manner suitable for analysis and design.
CO2 Determine AC steady state response.
CO3 Analyse the electric circuits using network theorems.
CO4 Determine frequency response of filters