

Programming using Python

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 with Python	2	0	0	2	12 th Pass	NIL

Learning Objectives

The Learning Objectives of this course are as follows:

- To provide exposure to basic problem-solving techniques with computers
- To develop logical thinking abilities and to propose novel solutions for real world problems through programming language constructs.
- To deepen the empirical knowledge on applying programming on business domains.

Learning outcomes

The Learning Outcomes of this course are as follows:

- After studying this course, students will be able to interpret the basic representation of the data structures and sequential programming
- After studying this course, students will be able to gain knowledge of, and ability to use control framework terminologies.
- After studying this course, students will be able to work out using the core data structures as lists, dictionaries, tuples, and sets.
- After studying this course, students will be able to choose appropriate programming paradigms, interrupt and handle data using files to propose solutions through reusable modules.
- After studying this course, students will be able to propose possible error-handling constructs for unanticipated states/inputs .
- After studying this course, students will be able to implements exemplary applications on real-world problems.

SYLLABUS

Unit-1: Introduction (3 weeks)

Relationship between computers and programs, Basic principles of computers, File systems, Using the Python interpreter, Introduction to binary computation, Input / Output

Unit-2: Data types and control structures (4 weeks)

Operators (unary, arithmetic, etc.), Data types, variables, expressions, and statements, Assignment statements, Strings and string operations, Control Structures: loops and decision

Unit-3: Modularization and Classes (4 weeks)

Standard modules, Packages, Defining Classes, Defining functions, Functions and arguments(signature)

Unit-4: Data structures and Object-oriented design (4 weeks)

Data Structures (array, List, Dictionary), Error processing, Exception Raising and Handling Programming types, Object Oriented Programming, Object Oriented Design, Inheritance and Polymorphism

Practical Exercises (15 weeks)

- Running instructions in Interactive interpreter and a Python Script
- Write a program to purposefully raise Indentation Error and Correct it
- Write a program to compute distance between two points taking input from the user. (Pythagorean Theorem)
- Write a program add.py that takes 2 numbers as command line arguments and prints its sum.
- Write a Program for checking whether the given number is an even number or not.
- Using a for loop, write a program that prints out the decimal equivalents of 1/2, 1/3, 1/4, 1/10
- Write a program using a for loop that loops over a sequence. What is the sequence?
- Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.
- Find the sum of all the primes below two million. Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...
- By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.
- Write a program to count the numbers of characters in the string and store them in a dictionary data structure.
- Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure.
- Write a program combining lists that combines these lists into a dictionary.
- Write a program to count the frequency of characters in a given file. Can you use

character frequency to tell whether the given file is a Python program file, C program file or a text file?

- Write a program to print each line of a file in reverse order.
- Write a program to compute the number of characters, words and lines in a file.
- Write a function ball collide that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding. Hint: Represent a ball on a plane as a tuple of (x, y, r), r being the radius. If (distance between two balls centers) \leq (sum of their radii) then (they are colliding)
- Find mean, median, mode for the given set of numbers in a list.
- Write a function nearly equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.
- Write a function dups to find all duplicates in the list.

Essential/recommended readings

- "Starting Out with Python plus My Programming Lab with Pearson eText --Access Card Package (3rd Edition) Tony Gaddis ISBN-13: 978-0133862256".
- Python Crash Course: A Hands-On, Project-Based Introduction to Programming (2nd Edition).
- Head-First Python: A Brain-Friendly Guide (2nd Edition) by Paul Barry.
- Learn Python the Hard Way: 3rd Edition by Zed A. Shaw.
- Python Programming: An Introduction to Computer Science (3rd Edition) by John M. Zelle.
- Python Cookbook: Recipes for Mastering Python 3 (3rd Edition) by Brian Jones and David Beazley.

Examination scheme and mode:

Total Marks: 100

Internal Assessment: 25 marks

Practical Exam (Internal): 25 marks

End Semester University Exam: 50 marks

The Internal Assessment for the course may include Class participation, Assignments, Class tests, Projects, Field Work, Presentations, amongst others as decided by the faculty.

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

