



# Indraprastha College for Women

## University of Delhi

Course Name:	SEC-II (BA/BCom/BSc)
Paper Title:	Programming using Python
Unique Paper Code:	2342011101
Semester:	II
Faculty(s):	Dr. Pardeep Kumar
Year:	2024

<b>Work Plan</b>			
Unit No.	Learning Objective	Lecture No.	Topics to be Covered
I	<b>Introduction</b>	1	Relationship between computers and programs
		2	Basic principles of computers
		3	File systems
		4	Download and Install Python(PR. Ex.)
		5	The Python interpreter
		6	Introduce Python interface (PR. Ex.)
		7	Running instructions in an Interactive interpreter and a Python Script (PR. Ex.)
		8	Introduction to binary computation
		9	Introduction to binary computation
		10	Write a program to purposefully raise an Indentation Error and Correct it (Pr. Ex.)
		11	Write a program to compute the distance between two points taking input from the user. (Pythagorean Theorem) (Pr. Ex)

		12	Write a program add.py that takes 2 numbers as command line arguments and prints its sum.(Pr. Ex.)
		13	Comment in Programme(Pr. Ex.)
		14	Input / Output
		15	Programme that specifies Input and Output
Unit-II	Data types and control structures	16	Operators (unary, arithmetic, etc.)
		17	Data types with examples
		18	Variables and expressions
		19	Statements, Assignment Statements
		20	Strings and string operations
		21	Write a Program for checking whether the given number is an even number or not (Pr. Ex.)
		22	Control Structures: loops and decision
		23	If with For, If with Do
		24	If with While
		25	23 and 24 (Pr. Ex.)
		26	If with two arguments
		27	Using a for loop, write a program that prints out the decimal equivalents of 1/2, 1/3, 1/4, 1/10 (Pr. Ex.)
		28	Write a program using a for loop that loops over a sequence. What is the sequence? (Pr. Ex.)
		29	Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero. (Pr. Ex.)
		30	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, .....(Pr. Ex.)

Unit-III	Modularization and Classes	31	By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms. (Pr. Ex.)
		32	Introduction of Modules
		33	Introduction of Packages
		34	32 and 33 (Pr. Ex.)
		35	Explain Classes
		36	35 (Pr. Ex.)
		37	Defining functions, Functions and arguments
		38	37 (Pr. Ex.)
		39	Combination of Function and Modules
		40	Combination of Function and Packages
		41	Combination of Function, Modules and Packages.
		42	Write a program to count the numbers of characters in the string and store them in a dictionary data structure. (Pr. Ex.)
		43	Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure. (Pr. Ex.)
		44	Write a program combining lists that combines these lists into a dictionary (Pr. Ex.)
		45	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? (Pr. Ex.)
Unit-IV		46	Types of Array
		47	Error processing and detection
		48	Handling Programme file type
		49	Object Oriented Programming
		50	Object Oriented Design,

Data structures and Object-oriented design	51	Write a program to print each line of a file in reverse order (Pr. Ex.)
	52	Write a program to compute the number of characters, words and lines in a file. (Pr. Ex.)
	53	Write a function ball collide that takes two balls as parameters and computes if they are colliding.(Pr. Ex.)
	54	Inherited structured Programme
	55	Polymorphic programming
	56	Boolean Programming
	57	Find mean, median, mode for the given set of numbers in a list.(Pr. Ex.)
	58	Express 57 by Graph. (Pr. Ex.)
	59	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. (Pr. Ex.)
	60	Write a function dups to find all duplicates in the list. (Pr. Ex.)

Syllabus		
Unit	Contents	Contact Hours
I	Relationship between computers and programs, Basic principles of computers, File systems, Using the Python interpreter, Introduction to binary computation, Input / Output	8
II	Operators (unary, arithmetic, etc.), Data types, variables, expressions, and statements, Assignment statements, Strings and string operations, Control Structures: loops and decision	8
III	Standard modules, Packages, Defining Classes, Defining functions, Functions and arguments (signature)	8

IV	Data Structures (array, List, Dictionary), Error processing, Exception Raising and Handling Programming types, Object Oriented Programming, Object Oriented Design, Inheritance and Polymorphism	8
V	<ul style="list-style-type: none"> <li>● Running instructions in Interactive interpreter and a Python Script</li> <li>● Write a program to purposefully raise Indentation Error and Correct it</li> <li>● Write a program to compute distance between two points taking input from the user. (Pythagorean Theorem) <ul style="list-style-type: none"> <li>● Write a program add.py that takes 2 numbers as command line arguments and prints its sum.</li> </ul> </li> <li>● Write a Program for checking whether the given number is an even number or not.</li> <li>● Using a for loop, write a program that prints out the decimal equivalents of 1/2, 1/3, 1/4, 1/10</li> <li>● Write a program using a for loop that loops over a sequence. What is the sequence?</li> <li>● Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.</li> <li>● 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, .....</li> <li>● By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.</li> <li>● Write a program to count the numbers of characters in the string and store them in a dictionary data structure.</li> <li>● Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure.</li> <li>● Write a program combining lists that combines these lists into a dictionary.</li> <li>● 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?</li> <li>● Write a program to print each line of a file in reverse order.</li> <li>● Write a program to compute the number of characters, words and lines in a file.</li> <li>● 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) &lt;= (sum of their radii) then (they are colliding) .</li> <li>● Find mean, median, mode for the given set of numbers in a list.</li> </ul>	28

	<ul style="list-style-type: none"> <li>• 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.</li> <li>• Write a function dups to find all duplicates in the list.</li> </ul>	
	<b>Total</b>	<b>60</b>
<b>Text Books/Suggested Readings:</b>		
S. No.	Name of Authors/Books/Publishers	Year of Publication/ Repr int
1.	"Starting Out with Python plus My Programming Lab with Pearson eText --Access Card Package (3rd Edition) Tony Gaddis ISBN-13: 978-0133862256".	2021
2.	Python Crash Course: A Hands-On, Project-Based Introduction to Programming (2nd Edition).	2019
3.	Python Programming: An Introduction to Computer Science (3rd Edition) by John M. Zelle	2013
4.	Python Cookbook: Recipes for Mastering Python 3 (3rd Edition) by Brian Jones and David Beazley.	2013

Paper Components			
Credits	Lecture (L)	Tutorial (T)	Practical (P)
2	0	0	2 (for 4 hrs.)
Assessment Scheme			
S.No.	Component	Marking Scheme	Total Marks
1	Internal Assessment <ul style="list-style-type: none"> <li>• Assignment/Quiz/Project/Presentation</li> <li>• Class Test</li> <li>• Attendance</li> </ul>	20 20 0	40
2.	Continuous Assessment ( <b>Tutorial</b> ) <ul style="list-style-type: none"> <li>• Activity 1</li> <li>• Activity 2</li> </ul>	NA NA NA	NA

	<ul style="list-style-type: none"> <li>• Attendance</li> </ul>	NA	
3.	Practical <ul style="list-style-type: none"> <li>• Continuous Assessment</li> <li>• End Term Written/Practical Exam</li> <li>• Viva</li> </ul>	10	40
		20	
		10	
4.	End Semester Examination		NA