Department of Artificial Intelligence and Machine Learning

Academic Year	:	2026-2027	Course Name	:	Python Programming
Semester	:	I	Course Code	:	1BPLC105B
Scheme	:	2025	L: T: P: S	:	3:0:2:0 (4 credits)
Total Contact hours	:	40 + 24 (Practical)	CIE Marks	:	50
CoursePlan Author	:	Mrs. Lavanya	SEE Marks	:	50
Date	:	17/08/2026	Total Marks	:	100

Course Prerequisites:

-Basic knowledge of using a computer
-Basic mathematics (addition, subtraction, multiplication, division)
-Simple logical thinking and problem-solving skills
-No prior programming or coding experience required

Learning Objectives:

-Apply variables, data types, expressions, and statements to solve computational problems

-Implement iterative constructs (`for`, `while`) and control statements to handle repetition logic

-Apply string operations, list and tuples to manipulate data efficiently

-Understand and apply dictionaries, numpy and file operations to manage data in Python programs

-Apply OOP concepts such as encapsulation, inheritance, and polymorphism to develop Python programs

Course Outcomes:

CO		*At the end of the course, student should be able to . . .*	**Blooms'Level**
CO1	:	Develop scripts using primitive language constructs of python	L3
CO2	:	Identify the methods to manipulate primitive python data structures	L2
CO3	:	Make use of Python standard libraries for programming..	L3
CO4	:	Build scripts for performing file operations.	L3
CO5	:	Illustrate the concepts of Object-Oriented Programming as used in Python.	L2

Blooms' Taxonomy:

L1	L2	L3	L4	L5	L6
Remembering	Understanding	Applying	Analyzing	Evaluating	Creating

Program Outcomes:

PO1	:	Engineering knowledge	**PO7**	:	Ethics
PO2	:	Problem analysis	**PO8**	:	Individual and Collaborative Team Work
PO3	:	Design/development of solutions	**PO9**	:	Communication
PO4	:	Conduct investigations of complex problems	**PO10**	:	Project Management and Finance
PO5	:	Modern tool usage	**PO11**	:	Life-Long Learning
PO6	:	The engineer and the world			

Program Specific Outcomes of AIML Department:

PSO1:	Adapt, Contribute Innovate ideas in the field of Artificial Intelligence and Machine Learning.
PSO2:	Enrich the abilities to qualify for Employment, Higher studies and Research in various domains of Artificial Intelligence and Machine Learning with ethical values.
PSO3:	Acquire practical proficiency with niche technologies to become Entrepreneur in Artificial Intelligence and Machine Learning domain.

CO-PO-PSO Mapping:

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	-	2	-	-	-	-	-	2	2	2	1
CO2	3	3	2	-	2	-	-	-	-	-	2	3	2	1
CO3	2	2	3	-	3	-	-	-	-	-	3	3	3	2
CO4	2	2	3	2	3	-	-	-	-	-	2	2	3	2
CO5	3	2	3	-	3	-	-	2	1	2	3	3	3	3

CO-PO-PSO Justification:

	PO/PSO	Correlation Weightage	Justification
CO1	PO1	3	Applies fundamental programming concepts such as variables, control structures, and operators, thereby strengthening core engineering knowledge.
	PO2	2	Enables students to analyze problems logically and break them down into step-by-step algorithmic solutions.
	PO3	1	Supports the design and development of basic software solutions using appropriate programming constructs.
	PO5	2	Promotes the use of Python, a widely used modern programming tool, for developing computational solutions.
	PO11	2	Builds a strong programming foundation that motivates learners to continuously update their skills and adapt to new technologies
	PSO1	2	Establishes the basic programming foundation required to understand and implement Artificial Intelligence and Machine Learning concepts
	PSO2	2	Enhances problem-solving and programming skills essential for employment, advanced studies, and research in AI and ML domains
	PSO3	1	Provides essential coding skills that support the development of basic applications, forming a foundation for entrepreneurial ventures.
	PO1	3	Strengthens engineering fundamentals by applying core concepts of data representation and manipulation using Python data structures such as lists, tuples, sets, and dictionaries.
	PO2	3	Enables learners to analyse problems effectively by selecting and applying appropriate data structures to organize, process, and interpret data.

CO2	PO3	2	Supports the development of efficient and optimized solutions by enabling the use of suitable data structures for algorithm implementation.
	PO5	2	Promotes effective use of Python as a modern programming tool for managing and manipulating structured data.
	PO11	2	Builds a strong foundation in data handling, motivating learners to continuously adapt and learn advanced data-centric technologies
	PSO1	3	Provides the fundamental data manipulation skills required to implement algorithms and models in Artificial Intelligence and Machine Learning.
	PSO2	2	Enhances analytical and programming skills essential for employment, higher education, and research in AI and ML domains with ethical data usage.
	PSO3	1	Supports the development of data-driven applications, forming a technical foundation for entrepreneurial initiatives using niche AI and ML technologies.
CO3	PO1	2	Enhances engineering knowledge by applying built-in and standard Python libraries to implement efficient and reliable computational solutions.
	PO2	2	Enables effective problem analysis by selecting appropriate libraries and functions to simplify complex computational tasks.
	PO3	3	Supports the development of optimized and scalable solutions by leveraging reusable and well-tested library components.
	PO5	3	Strongly supports the use of modern programming tools by effectively utilizing Python's rich ecosystem of standard libraries.
	PO11	3	Encourages continuous learning by motivating students to explore and adapt to new libraries and evolving technologies.

	PSO1	3	Provides essential exposure to libraries that form the foundation for Artificial Intelligence and Machine Learning development.
	PSO2	3	Enhances professional competence by building skills in library-based programming required for employment, research, and advanced studies.
	PSO3	2	Supports entrepreneurial initiatives by enabling rapid development of practical, technology-driven applications using Python libraries.
CO4	PO1	2	Strengthens engineering knowledge by applying programming concepts for data storage, retrieval, and management using file systems
	PO2	2	Enables learners to analyze data-related problems and determine suitable file-handling techniques for efficient solutions.
	PO3	3	Supports the design and development of practical software solutions that require persistent data storage and retrieval.
	PO4	2	Facilitates investigation of complex problems by enabling data collection, processing, and analysis through file-based operations.
	PO5	3	Promotes the effective use of Python as a modern tool for handling real-world data using file input/output mechanisms.
	PO11	2	Builds foundational skills that motivate learners to explore advanced data management, databases, and storage technologies.
	PSO1	2	Supports AI and ML applications by enabling data collection, preprocessing, and storage essential for model development.
	PSO2	3	Enhances practical programming skills required for employment, research, and advanced studies involving data-intensive applications.
	PSO3	2	Enables development of data-driven applications, supporting entrepreneurial ventures through effective data management solutions.

CO5	PO1	3	Strengthens engineering knowledge by applying object-oriented concepts such as classes, objects, inheritance, polymorphism, and encapsulation in Python.
	PO2	2	Enables learners to analyze complex problems by decomposing them into interacting objects and classes.
	PO3	3	Strongly supports the design and development of modular, reusable, and scalable software solutions using object-oriented principles.
	PO5	3	Promotes effective use of Python as a modern programming tool for developing structured and industry-oriented software applications
	PO8	2	Supports teamwork by enabling modular development, where team members can work independently on different classes and modules.
	PO9	1	Improves communication through clear program structure, meaningful class design, and documentation of object-oriented code.
	PO10	2	Aids in project planning and management by supporting structured development, code reuse, and cost-effective software solutions.
	PO11	3	Prepares learners for continuous learning by providing a foundation for advanced programming paradigms and emerging technologies.
	PSO1	3	Provides essential object-oriented design skills required for implementing scalable and maintainable AI and ML systems.
	PSO2	3	Enhances professional competence and research readiness by developing industry-relevant software design skills with ethical practices.
	PSO3	3	Strongly supports entrepreneurial initiatives by enabling development of robust, scalable, and market-ready applications using object-oriented design.

Course Content (Syllabus)

Module 1	
The way of the program: The Python programming language, what is a program? What is debugging? Syntax errors, Runtime errors, Semantic errors, Experimental debugging. **Variables, Expressions and Statements:** Values and data types, Variables, Variable names and keywords, Statements, Evaluating expressions, Operators and operands, Type converter functions, Order of operations, Operations on strings, Input, Composition, The modulus operator. **Iteration**: Assignment, Updating variables, the for loop, the while statement, The Collatz 3n + 1 sequence, tables, two-dimensional tables, break statement, continue statement, paired data, Nested Loops for Nested Data. **Functions**: Functions with arguments and return values. Chapters: 1.1-1.7, 2.1-2.12, 3.3, 4.4, 4.5	08
Module 2	
Strings: Working with strings as single things, working with the parts of a string, Length, Traversal and the for loop, Slices, String comparison, Strings are immutable, the in and not in operators, A find function, Looping and counting, Optional parameters, The built-in find method, The split method, Cleaning up your strings, The string format method. **Tuples:** Tuples are used for grouping data, Tuple assignment, Tuples as return values, Composability of Data Structures. **Lists**: List values, accessing elements, List length, List membership, List operations, List slices, Lists are mutable, List deletion, Objects and references, Aliasing, cloning lists, Lists and for loops, List parameters, List methods, Pure functions and modifiers, Functions that produce lists, Strings and lists, list and range, Nested lists, Matrices. **Chapter**: 5.1, 5.2, 5.3	08
Module 3	
Dictionaries: Dictionary operations, dictionary methods, aliasing and copying. **Numpy:** About, Shape, Slicing, masking, Broadcasting, dtype. **Files**: About files, writing our first file, reading a file line-at-a-time, turning a file into a list of lines, Reading the whole file at once, working with binary files, Directories, fetching something from the Web.	08

Chapter: 5.4, 6.1-6.5, 7.1-7.8	
Module 4	
Modules: Random numbers, the time module, the math module, creating your own modules, Namespaces, Scope and lookup rules, Attributes and the dot Operator, Three import statement variants. **Mutable versus immutable and aliasing** **Object oriented programming**: Classes and Objects – The Basics, Attributes, Adding methods to our class, Instances as arguments and parameters, Converting an instance to a string, Instances as return values. **Chapter**: 8.1-8.8, 9.1, 11.1	**08**
Module 5	
Object oriented programming: Objects are mutable, Sameness, Copying. Inheritance: Pure functions, Modifiers, Generalization, Operator Overloading, Polymorphism. Exceptions: Catching Exceptions, Raising your own exceptions. **Chapter:**11.2.2-11.2.4,11.3.2-11.3.9,12.1,12.2	**08**

Delivery Methods:

***C&T:** Chalk & Talk ***PPT** ***CBT:** Case-Based Teaching

***Dem:** Demonstration ***PBL:** Problem Based Learning

Class No	Topic	Date	RBT	CO	Mode
Module-1					
1.	Course website and Introduction about the subject	17/08/26	L1	CO1	PPT
2.	The way of the program: The Python programming language, what is a program? What is debugging? Syntax errors, Runtime errors, Semantic errors, Experimental debugging.	18/08/26	L1	CO1	C&T
3.	Variables, Expressions and Statements: Values and data types, Variables, Variable names and keywords, Statements, Evaluating expressions	19/08/26	L1	CO1	C&T
4.	Operators and operands, Type converter functions, Order of operations	20/08/26	L2	CO1	C&T
5.	Operations on strings, Input, Composition, The modulus operator.	24/08/26	L2	CO1	C&T
6.	Iteration: Assignment, Updating variables, the for	25/08/26	L2	CO1	C&T

Class No	Topic	Date	RBT	CO	Mode
	loop, the while statement, The Collatz 3n + 1 sequence, tables				
7.	Two-dimensional tables, break statement, continue statement, paired data, Nested Loops for Nested Data.	27/08/26	L2	CO1	C&T
8.	Functions: Functions with arguments and return values.	29/08/26	L2	CO1	C&T
9.	Revision of Module1	31/08/26			
10.	Module1 test	1/09/26			
Module-2					
11.	Strings: Working with strings as single things, working with the parts of a string, Length, Traversal and the for loop	2/09/26	L3	CO2	C&T
12.	Slices, String comparison, Strings are immutable, the in and not in operators	3/09/26	L3	CO2	C&T
13.	A find function, Looping and counting, Optional parameters, The built-in find method, The split method, Cleaning up your strings, The string format method.	7/09/26	L3	CO2	PPT, C&T
14.	Tuples: Tuples are used for grouping data, Tuple assignment, Tuples as return values, Composability of Data Structures.	8/09/26	L3	CO2	C&T
15.	Lists: List values, accessing elements, List length, List membership, List operations	9/09/26	L3	CO2	C&T
16.	List slices, Lists are mutable, List deletion, Objects and references	10/09/26	L3	CO2	C&T
17.	Aliasing, cloning lists, Lists and for loops, List parameters, List methods, Pure functions and modifiers	15/09/26	L3	CO2	C&T
18.	Functions that produce lists, Strings and lists, list and range, Nested lists, Matrices.	16/09/26	L3	CO2	C&T
19.	Revision of Module2	17/09/26			
20.	Module2 test	21/09/26			
Module-3					
21.	Dictionaries: Dictionary operations, dictionary methods	22/09/26	L2	CO3	PPT, C&T
22.	Aliasing and copying	23/09/26	L3	CO3	Demo with hands on

Class No	Topic	Date	RBT	CO	Mode
23.	Numpy: About, Shape, Slicing, masking, Broadcasting, dtype.	24/09/26	L3	CO3	PPT, Demo with hands on
24.	Files: About files, writing our first file, reading a file line-at-a-time	28/09/26	L3	CO3	PPT, Demo with hands on
25.	Turning a file into a list of lines	29/09/26	L3	CO3	PPT, Demo with hands on
26.	Reading the whole file at once	30/09/26	L3	CO3	Demo with hands on
27.	working with binary files	1/10/26	L3	CO3	Demo with hands on
28.	Directories, fetching something from the Web.	8/10/26	L3	CO3	Demo with hands on
29.	Revision of Module3	12/10/26			
30.	Module3 Test	13/10/26			
Module-4					
31.	Modules: Random numbers, the time module, the math module	14/10/26	L2	CO4	PPT/PBL
32.	creating your own modules, Namespaces	15/10/26	L3	CO4	PPT
33.	Scope and lookup rules, Attributes and the dot Operator	22/10/26	L3	CO4	PPT
34.	Three import statement variants	26/10/26	L3	CO4	PPT/ C&T
35.	Mutable versus immutable and aliasing	27/10/26	L3	CO4	PPT/Demo with hands on
36.	Object oriented programming: Classes and Objects — The Basics, Attributes,	28/10/26	L3	CO4	PPT/Demo with hands on

Class No	Topic	Date	RBT	CO	Mode
37.	Adding methods to our class, Instances as arguments and parameters	29/10/26	L3	CO4	PPT/Demo with hands on
38.	Converting an instance to a string, Instances as return values	2/11/26	L3	CO4	PPT/Demo with hands on
39.	Revision of Module4	3/11/26			
40.	Module4 Test	4/11/26			
Module-5					
41.	Object oriented programming: Objects are mutable	5/11/26	L2	CO5	PPT/ C&T
42.	Sameness, Copying.	9/11/26	L3	CO5	PPT/ C&T
43.	Inheritance: Pure functions	16/11/26	L3	CO5	PPT/ C&T
44.	Modifiers, Generalization	17/11/26	L3	CO5	PPT/ C&T
45.	Operator Overloading	18/11/26	L3	CO5	PPT/ C&T
46.	Polymorphism	19/11/26	L3	CO5	PPT/ C&T
47.	Exceptions: Catching Exceptions	23/11/26	L3	CO5	PPT/ C&T
48.	Raising your own exceptions.	24/11/26	L3	CO5	PPT/ C&T
49.	Revision of Module5	25/11/26			
50.	Module5 Test	1/12/26			

PRACTICAL COMPONENTS OF IPCC

Lab	PART – A: FIXED SET OF EXPERIMENTS	Date	RBT	CO	Mode
1.	Introduction Python Programming environment and Installation	C1 -20/8 C2 -27/8	L2		Hands on
2.	Develop a python program to read 2 numbers from the keyboard and perform the basic arithmetic operations based on the choice. (1-Add, 2-Subtract, 3-Multiply, 4-Divide). b. b. Develop a program to read the name and year of birth of a person. Display whether the person is a senior citizen or not.	C1 - C2 -	L3	CO1	Hands on
3.	a. Develop a program to generate Fibonacci sequence of length (N). Read N from the console. b. Write a python program to create a list and perform the following operations a. Inserting an element b. Removing an element c. Appending an element d. Displaying the length of the list e. Popping an element f. Clearing the list	C1 - C2 -	L3	CO2	Hands on
4.	a.Read N numbers from the console and create a list. Develop a program to print mean, variance and standard deviation with suitable messages. b. Read a multi-digit number (as chars) from the console. Develop a program to print the frequency of each digit with a suitable message.	C1 - C2 -	L3	CO2	Hands on
5.	Develop a program to print 10 most frequently appearing words in a text file. [Hint: Use a dictionary with distinct words and their frequency of occurrences. Sort the dictionary in the reverse order of frequency and display the dictionary slice of the first 10 items	C1 - C2 -	L3	CO2	Hands on
6.	Develop a program to read 6 subject marks from the keyboard for a student. Generate a	C1 -	L3	CO2, CO3	Hands on

	report that displays the marks from the highest to the lowest score attained by the student. [Read the marks into a 1-Dimesional array and sort using the Bubble Sort technique].	C2 -			
7.	Develop a program to sort the contents of a text file and write the sorted contents into a separate text file. [Hint: Use string methods strip(), len(), list methods sort(), append(), and file methods open(), readlines(), and write()].	C1 - C2 -	L3	CO3	Hands on
8.	Define a function that takes TWO objects representing complex numbers and returns a new complex number with the sum of two complex numbers. Define a suitable class 'Complex' to represent the complex number. Develop a program to read N (N >=2) complex numbers and to compute the addition of N complex numbers.	C1 - C2 -	L3	CO2, CO3, CO5	Hands on
9.	Define a function that takes TWO objects representing complex numbers and returns a new complex number with the sum of two complex numbers. Define a suitable class 'Complex' to represent the complex number. Develop a program to read N (N >=2) complex numbers and to compute the addition of N complex numbers.	C1 - C2 -	L3	CO2, CO3, CO4	Hands on
10.	Text Analysis Tool: Build a tool that analyses a paragraph: frequency of each word, longest word, number of sentences, etc.	C1 - C2 -	L3	CO2, CO3	Hands on
11.	Develop Data Summary Generator: Read a CSV file (like COVID data or weather stats), convert to dictionary form, and allow the user to run summary queries: max, min, average by column.	C1 - C2 -	L3	CO2	Hands on
12.	Develop Student Grade Tracker: Accept multiple students' names and marks. Store	C1 -	L3	CO2	Hands on

	them in a list of tuples or dictionaries. Display summary reports (average, topper, etc.).	C2 -			
13.	Develop a program to display contents of a folder recursively (Directory) having sub-folders and files	C1 - C2 -	L3	CO3	Hands on
14	Lab IA1 (1-6 programs)	C1 - C2 -			
15	Lab IA 2 (7-12 programs)	C1 - C2 -			

Self-study hours:

For 4 credits: Self-study hours = 120 hours – Course delivery planned hours = 45 hours(TH) +26(Pr)

Sl No	Self-study	Hours
1	E-Learning	**40**
2	Assessment Conduction	**9**
	TOTAL	**49**

Textbook and Reference books:

T1	Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers- How to think like a computer scientist: learning with python 3. Green Tea Press, Wellesley, Massachusetts,2020 https://media.readthedocs.org/pdf/howtothink/latest/howtothink.pdf
	Reference books / Manuals: 1. Al Sweigart," Automate the Boring Stuff with Python, 2nd Edition: Practical Programming for Total Beginners",2nd Edition, No Starch Press, 2022. (Available under CC-BY-NC-SA license at https://automatetheboringstuff.com/) 2. Kyla McMullen, Elizabeth Matthews and June Jamrich Parsons, Programming with Python, Cengage, 2023.

Web links and Video Lectures (e-Resources):

1.	Web links and Video Lectures (e-Resources): https://www.learnbyexample.org/python/
2.	https://www.learnpython.org/
3.	https://pythontutor.com/visualize.html#mode=edit
4	https://sites.google.com/skit.org.in/lavanya-k/course-material/python-programming

Assessment Schedule:

S.N.	Assessment Type	Content	CO	Duration	Marks	Date
1.	IA-1 (Theory)	M1,M2,M3	CO1,CO2,CO3 (1st half)	90 min	50(scaled down to 15)	
2.	IA-2 (Theory)	M3,M4,M5	CO3(2nd half), CO4,CO5	90 min		
3.	Assignment	M1 to M5	CO1 to CO5		10	
4.	**Part A (IA+ Assignment)**				**25**	
5.	Practical Component: Continuous Assignment	M1 to M5(Program 1 to 12)	CO1 to CO5	2 hours /week	15	
6.	Practical Component: Lab test	All Programs from syllabus	CO1 to CO5	2(2*1) hours	10	
7.	**Part B(CA+Lab Test)**				**25**	
8.	**TOTAL CIE (Part A+Part B)**				**50**	
9.	Semester End Examination	M1 to M5	CO1 to CO5	3hr	100(scaled down to 50)	

***RB** – Text Book/**R**eference **B**ook, ***L** – **L**ecture, **V**- Videos or any other mode, ***RBT** – **R**evised **B**looms' **T**axonomy, **L: T: P: C** – Theory/**L**ecture: **T**utorial: **P**ractical/Drawing: **C**redits, **SEE**: **S**emester **E**nd **E**xamination, **CIE**: **C**ontinuous **I**nternal **E**valuation, **Seminar**: Group of 6-8 students, **Module** 1,2,3,4 & 5,*

*The weightage of Continuous Internal Evaluation **(CIE) is 50%** and for Semester End Exam **(SEE) is 50%.** The **minimum passing mark for the CIE is 40%** of the maximum marks (20 marks out of 50) and for the **SEE minimum passing mark is 35%** of the maximum marks (18 out of 50 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures a **minimum of 40%(40 marks out of 100)** in the **sum total** of the **CIE** (Continuous Internal Evaluation) and **SEE** (Semester End Examination) taken together.*

Faculty In charge **Course Coordinator** **HoD**