<table>
<tr><td><b>Course Title</b></td><td colspan="3"><b>Mastering Python: From Basics to Advanced</b></td><td colspan="6"><b>CREDITS</b></td><td><b>3</b></td></tr>
<tr><td rowspan="2"><b>Course Code</b></td><td rowspan="2"></td><td rowspan="2"><b>Course Category</b></td><td rowspan="2"><b>Core</b></td><td><b>L</b></td><td><b>T</b></td><td><b>P</b></td><td><b>S</b></td><td><b>C</b></td><td><b>CH</b></td><td><b>1-1- 2-0-3</b></td></tr>
<tr><td><b>1</b></td><td><b>1</b></td><td><b>2</b></td><td><b>0</b></td><td><b>3</b></td><td><b>60</b></td><td></td></tr>
<tr><td><b>Version</b></td><td><b>1.0</b></td><td><b>Approval Details</b></td><td><b>ACM</b></td><td colspan="7"><b>LEARNING LEVEL – BTL4</b></td></tr>
<tr><td><b>Course Coordinator</b></td><td colspan="10">Dr. Lenin. J</td></tr>
<tr><td><b>Course Description</b></td><td colspan="10">This course covers the fundamental concepts of Python programming, including data types, control structures, functions, modules, and object-oriented programming. It also includes practical applications of Python in various fields such as web development, data analysis, and machine learning. The course is designed to provide hands-on experience through practical exercises and projects.</td></tr>
<tr><td><b>Course Objectives</b></td><td colspan="10">1. Understand the basics of Python programming language.
2. Develop problem-solving skills using Python.
3. Learn to write efficient and readable code.
4. Gain practical experience through hands-on projects.
5. Explore the applications of Python in various domains.</td></tr>
</table>

Course Outcome	**CO Statement**	**Bloom's Taxonomy Level**
1	Apply basic constructs in Python to solve programming problems.	BTL - 3
2	Analyze the application of control structures and loops in Python programs to manage program complexity.	BTL – 4
3	Use functions and modules to write reusable code.	BTL – 3
4	Apply Python programming concepts to solve a variety of programming problems.	BTL – 3
5	Develop real-world applications using Python.	BTL – 6

Course Syllabus

Module	**Name**	**Contact Hours:**

Module -I Introduction to Python	History and features of Python - Installing Python and setting up the environment - Basic syntax - variables and data types - Input and output operations - Conditional statements (if, if-else, nested if) - Loops (for, while, nested loops) – Break - continue and pass statements.	**12 Hrx.**
Module -II Functions and Modules	Defining and calling functions - Function arguments and return values - Lambda functions - Importing modules and using standard libraries - Creating and using custom modules	**12 Hrx.**
Module -III Data Structures	Lists – tuples - sets and dictionaries – List comprehension - Set comprehensions - Dictionary comprehensions - Working with strings	**12 Hrx.**
Module -IV OOPs and File, Exception Handling	Classes and objects - Attributes and methods - Inheritance and polymorphism - Encapsulation and data hiding - Reading and writing files - Working with CSV and JSON files - Handling exceptions using try-except blocks - Custom exceptions	**12 Hrx.**
Module -V Advanced Topics	Regular expressions - Decorators and generators - Introduction to web development with Flask/Django - Basics of data analysis with Pandas - Introduction to machine learning with Scikit-learn	**12 Hrx.**

Practical Syllabus

Experiment No.	**Name**	**Tool/ Library Used**
1.	Writing and running basic Python programs Using the Python interpreter and IDEs	**Python**
2.	Variables, data types, and basic operators Simple input and output operations	**Python**
3.	Implementing conditional statements Writing loops (for and while)	**Python**

4.	Defining and calling functions Using function arguments and return values	**Python**
5.	Importing and using standard libraries Creating custom modules	**Python**
6.	Creating and manipulating lists List comprehensions	**Python**
7.	Working with tuples and sets Operations and methods	**Python**
8.	Creating and manipulating dictionaries Dictionary comprehensions	**Python**
9.	String operations and methods Regular expressions for pattern matching	**Python**
10.	Reading from and writing to files Working with CSV and JSON files	**Python**
11.	Using try-except blocks Creating custom exceptions	**Python**
12.	Defining classes and creating objects Working with attributes and methods	**Python**
13.	Implementing inheritance in classes Overriding methods and using polymorphism	**Python**

Self-Study topics for Advanced Learners

Advanced Python libraries (e.g., NumPy, Matplotlib).Web scraping with BeautifulSoup. Automating tasks with Python scripts.

Textbooks

1.	Mark Lutz, "Learning Python," O'Reilly Media, 5th Edition.
2.	Zed A. Shaw, "Learn Python the Hard Way," Addison-Wesley.

Reference books

1.	Al Sweigart, "Automate the Boring Stuff with Python," No Starch Press.
2.	Eric Matthes, "Python Crash Course," No Starch Press.

Coursera Credit Course

1.	https://www.coursera.org/specializations/python

Online Resources

1	https://www.w3schools.com/python/

CO PO Mapping

1: Weakly related, 2: Moderately related and 3: Strongly related

Course Outcome	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10
CO1	2	2	1	1	NA	2	2	NA	NA	2
CO2	2	2	1	1	NA	2	2	NA	NA	2
CO3	2	2	1	1	NA	2	2	NA	NA	2
CO4	2	2	1	1	NA	2	2	NA	NA	2
CO5	2	2	1	1	NA	2	2	NA	NA	2

CO PSO Mapping

Course Outcome	Program Specific Outcome							
	MCA General		MCA Generative AI		MCA Game Development		MCA Data Science	
	PSO 1	PSO 2	PSO 1	PSO 2	PSO1	PSO 2	PSO1	PSO2
CO1	NA	1	NA	3	NA	1	3	3
CO2	NA	1	NA	2	NA	1	3	3
CO3	NA	1	NA	3	NA	1	3	3
CO4	NA	1	NA	2	NA	1	3	3
CO5	NA	1	NA	2	NA	1	3	3

Assessment Pattern

Components	Internal Assessment	Semester End Examination (SEE)
Marks	50	50
Total Marks	100	

Assessment Plan

Assessment Component	Details of Assessment	Weightage
Continuous assessment – 1	Assignment	10
Continuous assessment – 2	Practical Examination	15
Continuous assessment – 3	Quiz	10
Continuous assessment – 4	MSE	10
Continuous assessment – 5	Attendance	5
Continuous assessment – 6	Semester End Examination	50