

**MANIPAL UNIVERSITY JAIPUR**  
Department of Computer Science & Engineering  
Course

**Hand-out A. Basic Details:**

Programme Name:	<b>B. Tech</b>
Course Name:	<b>Object Oriented Programming USING PYTHON</b>
<b>Course Code:</b>	<b>CSE2122</b>
LTPC ( <i>Lecture Tutorial Practical Credits</i> ):	<b>3104</b>
Session:	<b>July 2025</b>
Class:	<b>III</b>
Course Coordinator:	<b>Dr. Sunita Singhal &amp; Dr. Deepti Sharma</b>
Course Instructor(s):	<b>Dr. Akshay Jadhav, Dr. Deepti Sharma, Dr. Tapan</b>
Additional Practitioner(s) – if any ( <i>Industry Fellow/ Visiting Faculty/ Adjunct Faculty, etc.</i> ):	

**B. Introduction:** The main objective of this course is to familiarize students with object-oriented programming with python.

**C. Course Outcomes:**

<i>CO Statement</i>	<i>CO</i>	<i>Bloom's Level</i>	<i>Target Attainment %</i>	<i>Target Attainment level</i>
Demonstrate the basic programming skills of Python programming.	CSE2122.1	2: Understand	$\geq 80$	3
Apply the concept of data structures and reusability of programs.	CSE2122.2	3: Apply	$\geq 70\% < 80\%$	2
Develop the concept of object-oriented programming using python.	CSE2122.3	3: Apply	$\geq 70\% < 80\%$	2

Apply packages and exception handling in Python to solve a variety of real-world programming challenges	CSE2122.4	3: Apply	$\geq 60\% < 70\%$	2
Analyse and apply file handling and python libraries to address data analysis challenges.	CS2122.5	4: Analysis	$\geq 60\% < 70\%$	2

#### D. Program Outcomes and Program Specific Outcomes

[PO.1]	<b>Engineering knowledge:</b> Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems
[PO.2]	<b>Problem analysis:</b> Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
[PO.3]	<b>Design/development of solutions:</b> Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
[PO.4]	<b>Conduct investigations of complex problems:</b> Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions
[PO.5]	<b>Modern tool usage:</b> Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
[PO.6]	<b>The engineer and society:</b> Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice
[PO.7]	<b>Environment and sustainability:</b> Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development
[PO.8]	<b>Ethics:</b> Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practices
[PO.9]	<b>Individual and teamwork:</b> Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
[PO.10]	<b>Communication:</b> Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions
[PO.11]	<b>Project management and finance:</b> Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

[PO.12]	<b>Life-long learning:</b> Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
---------	--

**Program Specific Outcomes:**

[PSO.1]	Analysis to design, develop and implement efficient software for a given real life problem.
[PSO.2]	Develop through practice apply knowledge of AI, Machine Learning and Data Mining in analyzing big data for extracting useful information from it and for performing predictive analysis.
[PSO.3]	Implement to design, manage, and secure wired/ wireless computer networks for transfer and sharing of information.

**E. Assessment Plan:** Clearly write the criteria, its description, and associated marks for assessment of student achievements. In addition, the attendance requirements, assignments need to be mentioned in this section.

Criteria	Description	Maximum Marks
Internal Assessment (Formative and Summative)	Sessional Exam	30
	Continuous Evaluation Lab (15 Marks) (Record (5) + Execution (5) + Viva (5)) Test cases-based coding assignment <b>three</b> (15 Marks)	30
End Term Exam (Summative)	End Term Exam (Closed Book)	40
	Total	100

**Syllabus: Introduction:** Programming a computer, Programming languages; **Python basics:** Getting started with Python, Essentials of a Python program; Integers, Floating-point numbers, Strings; **Variables and scope:** Variables, Modifying values, Type conversion; **Selection control statements:** Selection: if statement, Boolean values, operators, and expressions; **Collections:** Lists, Tuples, Sets, Ranges, Dictionaries, Conversion, Sequences; **Loop control statements:** while, for statements, Nested loops, Iterables, iterators and generators, Comprehensions, The break and continue statements

**Functions:** Input parameters, Return values, Default parameters, \*args and \*\*kwargs, Decorators, Lambdas. Generator functions and yield; **Data Structure in Python:** Array, Linked List, Stack, Queue, Tree, Searching and Sorting; **Object-Oriented programming:** OOP's Concepts, **Classes, and Objects:** Defining and using a class, Instance attributes, Class attributes, Class decorators,

inspecting an object, Constructor, Abstraction, Composition. **Inheritance: Types of Inheritance. overriding magic methods; I/O and Errors Handling:** Errors, exceptions, handling exceptions, debugging programs, Logging, Testing; **Packaging:** Modules, Packages, Documentation, **File Handling:** Introduction, Access Methods, Read and write operation, Working with directories. **Python Libraries: Pandas, Matplotlib, NUMPY**

#### C. Text Books:

1. D. Phillips, Python 3 Object-Oriented Programming Build robust and maintainable software with object-oriented design patterns in Python 3.8, (3e), Packt Publishing, January 2018
2. W. J. Chun, Core Python Applications Programming, (3e), Prentice Hall Publishers, 2012
3. J. Grus, Data Science from Scratch: First Principles with Python, (1e), O'Reilly Media, 2015.
4. Python Data Science Essentials: A practitioner's guide covering essential data science principles, tools, and techniques, Third Edition By Alberto Boschetti, Luca Massaron.
5. Data Structures and Algorithms in Python, An Indian Adaptation Paperback – 1 July 2021 by Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, Wiley Publication

#### D. Reference Books:

1. A. Martelli, Python in a Nutshell, (3e), O'Reilly Media, Inc, 2017.
2. J. Georzen, T. Bower, B. Rhodes, Foundations of Python Network Programming, (3e), APress, 2014.
3. D. M. Beazley, Python Essential Reference, (4e), Pearson Addison-Wesley Professional, 2009.
4. M. Lutz, Programming Python, (4e), O'Reilly Media, 2010

#### E. Web Reference and Coursera Courses:

Python for Everybody Specialization – offered by University of Michigan

<https://www.coursera.org/programs/manipal-university-jaipur-ura2a/specializations/python>

#### F. Lecture Plan:

Lecture No.	Content Breakdown	Session Outcome (SOs)	Corresponding CO	Mode of delivery	Mode of assessing CO
1	Introduction to programming, types of programming languages, installing Python	Understand the role of programming languages and environment setup for Python	CSE2122.1	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / Mid-term Exam
2	Writing basic Python programs, syntax rules, printing, indentation	Write and run simple Python scripts	CSE2122.1	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / Mid-term Exam
3	Integer, float, string data types, operations, type conversions	Perform arithmetic and string operations in Python	CSE2122.1	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / Mid-term Exam

4	Lab 1: Basic Python Programs	Understand and implement basic Python syntax, data types, and simple input/output operations	CSE2122.1	Lab	Viva, Record, Execution
5	String indexing, slicing, string methods, escape sequences	Manipulate strings using slicing and built-in methods	CSE2122.1	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / Mid-term Exam
6	Variables, scope (local/global), constants, dynamic typing	Explain variable declarations and scope in Python	CSE2122.1	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / Mid-term Exam
7	Boolean expressions, if, elif, else control structures	Implement conditional logic in programs	CSE2122.1	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / Mid-term Exam
8	Lab 2: Control Flow	Apply conditional statements to control the flow of a program.	CSE2122.1	Lab	Viva, Record, Execution
9	Logical operators, nested conditions, conditional expressions	Apply complex decision-making using logical constructs	CSE2122.1	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / Mid-term Exam
10	Lists: declaration, indexing, slicing, common list methods	Manipulate list structures using indexing and methods	CSE2122.1	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / Mid-term Exam
11	Tuples and sets: characteristics, usage, set operations	Use tuples and sets in appropriate data contexts	CSE2122.1	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / Mid-term Exam
12	Lab 3: Loops	Use loops to perform iterative tasks and apply loop control statements	CSE2122.1	Lab	Viva, Record, Execution
13	Dictionaries: creation, key-value pairs, methods	Create and access data using Python dictionaries	CSE2122.1	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / Mid-term Exam
14	while and for loops, loop control variables	Implement iteration using basic loop structures	CSE2122.1	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / Mid-term Exam
15	Nested loops, break, continue, loop with else	Control loop execution using nested and control statements	CSE2122.1	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / Mid-term Exam

16	Lab 4: List	Perform various operations on lists such as indexing, slicing, and appending	CS22122.1	Lab	Viva, Record, Execution
17	Comprehensions (list, dict, set), use of iterables	Write concise code using comprehensions and iterables	CSE2122.2	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / Mid-term Exam
18	Function definition, parameters, return values	Create reusable functions with parameters and returns	CSE2122.1	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / Mid-term Exam
19	Default parameters, *args, **kwargs, lambdas	Handle flexible arguments and use lambda functions	CSE2122.1	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / Mid-term Exam
20	Lab 5: Function	Define and invoke functions using parameters and return values.	CSE2122.1	Lab	Viva, Record, Execution
21	Generator functions, yield, decorator syntax	Use decorators and generators for functional abstraction	CSE2122.3	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / Mid-term Exam
	Midterm Examination				
22	Arrays, Linked List basics (using lists or classes)	Represent arrays and linked lists using Python data types	CSE2122.2	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / ETE Exam
23	Stack and Queue implementation using list or deque	Implement stack and queue using standard libraries	CSE2122.2	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / ETE Exam
24	Lab 6: Basic Data Structure	Use basic data structures like tuples, sets, and dictionaries for data management	CSE2122.2	Lab	Viva, Record, Execution
25	Binary tree basics, node structure using classes	Construct and traverse simple binary trees	CSE2122.2	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / ETE Exam
26	Searching: linear, binary; Sorting: bubble, selection, insertion	Apply basic search and sort algorithms on data	CSE2122.2	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / ETE Exam
27	OOP Concepts: class, object, encapsulation	Understand and define Python classes and objects	CSE2122.3	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / ETE Exam

28	Lab 7: Class and Methods	Create classes and define methods to implement object-oriented programming	CSE2122.3	Lab	Viva, Record, Execution
29	Constructors, instance vs class variables	Use constructors and differentiate class/instance attributes	CSE2122.3	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / ETE Exam
30	Abstraction and composition	Apply abstraction and composition in object design	CSE2122.3	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / ETE Exam
31	Inheritance: single, multiple, multilevel	Implement inheritance and understand class hierarchy	CSE2122.3	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / ETE Exam
32	Lab 8: Constructor	Understand and use constructors for class initialization	CSE2122.3	Lab	Viva, Record, Execution
33	Method overriding	Customize class behavior using magic methods	CSE2122.3	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / ETE Exam
34	Exceptions: try, except, else, finally blocks	Handle runtime errors gracefully using exception handling	CSE2122.4	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / ETE Exam
35	Raising exceptions, debugging techniques	Raise exceptions and debug using Python tools	CSE2122.4	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / ETE Exam
36	Lab 9: Inheritance	Demonstrate inheritance and method overriding.	CSE2122.3	Lab	Viva, Record, Execution
37	Unit testing with unittest, using logging module	Write and test code using Python's testing and logging tools	CSE2122.4	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / ETE Exam
38	File handling: open(), read/write, file modes	Perform file I/O operations in different modes	CSE2122.5	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / ETE Exam
39	File operations: seek, tell, working with directories	Manipulate files and directories using built-in modules	CSE2122.5	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / ETE Exam
40	Lab 10: Exception Handling	Implement exception handling to write robust and error-resilient code	CSE2122.4	Lab	Viva, Record, Execution

41	Creating modules, importing modules	Organize code using Python modules	CSE2122.5	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / ETE Exam
42	Packages, documentation, __init__.py, help system	Structure code with packages and document them	CSE2122.5	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / ETE Exam
43	NumPy: arrays, basic operations, array indexing	Use NumPy for numerical operations and data handling	CSE2122.5	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / ETE Exam
44	Lab 11: Polymorphism	Understand and apply the concept of polymorphism	CSE2122.5	Lab	Viva, Record, Execution
45	Pandas: Series, DataFrame, data import/export	Handle tabular data using Pandas	CSE2122.5	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / Mid-term Exam
46	Pandas: Series, DataFrame, data import/export	Demonstrate knowledge through a mini project or recap	CSE2122.5	PPT, Lecture and Notes	Class Quiz/ Sessional Exam I / Mid-term Exam
47	Lab 12: Package and File	Create and use packages and modules; perform file operations such as read, write, and append	CSE2122.5	Lab	Viva, Record, Execution



**Pedagogical Approaches:**

- I. Lecture based teaching-learning
- II. Individual learning/ self-study
- III. Inquiry based learning
- VI. Learning through problem-solving
- VII. Flipped Classroom

**H. Learner centric activities:**

- Peer-assisted learning or buddy systems
- Step-by-step problem-solving exercises

**High-difficulty level:**

- Assignments involving real-life application or case studies
- Encouragement to explore MOOCs or online certifications related to the course

**I. Outside Class Engagement Activities:**

- Participating in academic clubs, technical competitions, or seminars
- Attending faculty-guided doubt-clearing or reinforcement sessions
- Watching recommended video lectures, tutorials, or educational webinars

**J. Course Articulation Matrix:**

CO	STATEMENT	CORRELATION WITH PROGRAM OUTCOMES												PSO1	PSO2
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		
CSE2122.1	Demonstrate the basic programming skills of Python programming.	3	2	1		2								1	1
CSE2122.2	Apply the concept of data structures and reusability of programs.	3	3	2						1			1	2	2
CSE2122.3	Illustrate and develop Develop the concept of object-oriented programming using python.	3	2	2	1					1	1			1	2
CSE2122.4	Apply packages and exception handling in	2	2		2	2							1	2	1

	Python to solve a variety of real-world programming challenges.														
<b>CSE2122.5</b>	Analyse and apply file handling and python libraries to address data analysis challenges.	<b>3</b>	<b>3</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>1</b>			<b>2</b>	<b>1</b>		<b>2</b>	<b>3</b>	<b>3</b>