



बिरसा मुंडा ट्रायबल युनिवर्सिटी Birsa Munda Tribal University

राजपिपला, जि० नर्मदा Rajpipla, Dist. Narmda

Established by Tribal Development Department, Govt. of Gujarat

School of Computer Science

M. Sc. (Information Technology) Programme

Subject Code & Name: MITME105-2 - Programming in Python

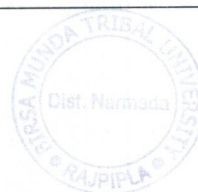
Teaching And Evaluation Scheme:

Teaching Scheme				Examination Scheme			
Credits				Component Weightage			
				CCE		SEE	
L	T	P	Total	TH	PWE	TH	PWE
2	-	2	4	25%	25%	25%	25%

Programme Name	M. Sc. (Information Technology)
Semester	1
Course Code	MITME105-2
Course Title	Programming in Python
Course Content Type (Th. / Pr.)	Th. + Pr.
Course Credit	2+2
Sessions + Lab Per Week	2+4
Total Teaching/ Lab Hours	30+60
* 2 Lab = 1 session	

Learning Objectives

Upon successful completion of this course, students will be able to:



- Understand the fundamental concepts of Python programming including syntax, variables, data types, operators, and control structures.
- Apply problem-solving techniques using conditional, iterative, and functional programming constructs.
- Develop modular and reusable programs using functions, recursion, and advanced Python features.
- Utilize Python collections and NumPy libraries for efficient data manipulation and numerical computation.
- Analyse real-world datasets using statistical functions and basic visualization techniques.
- Design and implement structured, readable, and maintainable Python applications.
- Evaluate alternative programming approaches and select appropriate data structures and algorithms for solving computational problems.
- Create practical Python-based solutions for scientific, business, and data-oriented applications.

Learning Outcomes

After successful completion of the course, students will be able to:

- Explain the fundamental concepts of Python programming including syntax, variables, data types, input-output operations, type conversion, and string manipulation techniques.
- Develop Python programs using arithmetic, logical, relational, and bitwise operators along with conditional and iterative control structures to solve computational problems.
- Utilize Python collections (Lists, Tuples, Sets, Dictionaries) and NumPy library functions for efficient data storage, manipulation, and statistical analysis.
- Analyse and implement modular programming concepts using user-defined functions, recursion, lambda expressions, decorators, and higher-order functions.



- Evaluate alternative programming approaches and select appropriate Python constructs, libraries, and data structures for solving real-world problems effectively.
- Design and develop structured, efficient, reusable, and maintainable Python applications by integrating multiple programming concepts and libraries.

CLO – PLO Mapping Matrix

CLO/ PLO	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	3	2	1	1	–	–	–	1	2
CLO2	3	3	3	2	–	1	–	1	2
CLO3	3	3	3	3	1	2	–	1	2
CLO4	3	3	3	3	1	2	–	2	3
CLO5	3	3	3	3	2	3	–	2	3
CLO6	3	3	3	3	2	3	1	2	3

Detailed Contents

Unit	Topic/Sub Topic	Teaching Hours
I Python Fundamentals	Python Fundamentals: • Concepts of Interpreter based programming language • Structure of Python Programming language. • Python code Indentation and execution • Python Variables (Naming of variables and Dynamic declaration of variables) • Comments in Python • Input, Output and Import Functions • Assigning values to multiple variables • Global variables	7



	Python Datatypes: • Text (str) • Numeric Type(int, float, complex) • Boolean (bool) • Setting Datatypes • Type conversion (int, float, complex), casting (int, float, str) Python Strings: • Multiline string • String as character array • triple quotes • Slicing string • negative indexing • string length, concatenation • String Methods(centre, count, join, len, max, min, replace, lower, upper, replace, split)	
II Operators, Conditional and Iterative Statements	Operators: • Arithmetic Operators • Comparison Operators • Assignment Operators • Bitwise Operators • Logical Operators • Membership Operators • Identity Operators • Operator Precedence Python conditional statements: • if statement, if..elif statement, if..elif...else statements, nested if	8

	Python Iterative statements: • while loop • nested while loop • break , continue statements. • for loop, range, break, continue, pass and Else with for loop, • nested for loop	
III Python Collections and Array Handling	Python Collections : • Lists(Creating list, indexing, accessing list members, range in list, list methods(append, clear, copy, count, index, insert, pop, remove, reverse, sort) • Tuples (Declaring tuple, indexing tuple, changing tuple values, adding and removing data from tuple, Use of tuple() method to create tuple, count() and index() methods) • Sets(declaring set, access set data, set methods (add, clear, copy, discard, pop, remove, union, update)). • Dictionary (Creating Dictionary, Adding, Accessing and Removing element ,Dictionary methods : get(), pop(), popitem(), clear(), copy()) Array handling using Numpy : • Overview of numpy • Numpy methods (Mean, Median, Mode, Standard Deviation and Variance) • Implementation of Numpy methods on numeric dataset created using list. • Introduction to data visualization.	8
IV	<u>Functions</u> • <u>Pass in Functions</u>	7

Python Functions	• <u>Global and Local Variables</u> • <u>Recursion</u> • <u>*args and **kwargs in Function</u> • <u>First Class Function</u> • <u>Lambda Function</u> • <u>Map, Reduce and Filter Function</u> • <u>Inner Function</u> • <u>Decorators</u> User defined function • Defining function, Function with Parameters • Parameter with default value, Function with return value	
Reference Books		
1. PYTHON PROGRAMMING, 3E by <u>Reema Thareja</u> , Publisher: Oxford University Press , ISBN-13 -978-9367257654 2. Python: The Complete Reference by <u>Martin C. Brown (Author)</u> , Publisher McGraw Hill Education, ISBN-13 : 978-9387572942 3. Ultimate Python Programming: Learn Python with 650+ programs, 900+ practice questions, and 5 projects by <u>Deepali Srivastava (Author)</u>, Publisher : BPB Publications , ISBN-13 -978-9355516558 4. Core Python Programming : Covers fundamentals to advanced topics like OOPS, Exceptions, Data structures, Files, Threads, Net, 3ed by <u>R. Nageswara Rao (Author)</u> , Publisher Dreamtech Press , ISBN-13 - 978-9390457151 5. Learning with Python by <u>Allen Downey (Author)</u>, <u>Jeffrey Elkner (Author)</u>, & 1 More, Publisher Dreamtech Press, ISBN-13- 978-9351198147		
Web References:		
1. https://docs.python.org/3/tutorial/index.html 2. https://www.geeksforgeeks.org/python/python-programming-language-tutorial/		

3. <https://www.w3schools.com/python/>
4. <https://www.tutorialspoint.com/python/index.htm>

L:: Lecture, **T::** Tutorial, **P::** Practical

CCE:: Continuous & Comprehensive Evaluation

(CCE theory includes Mid Semester Examination, Assignment, MCQ Quizzes, Seminar, Reflective Notes, Class Participation, Case Analysis and Presentation, Slip Tests (Announced/ Surprised), Attendance etc. or any combination of these)

PWE: Practical Work Examination

(PWE includes Laboratory Practical Work, Project Work, Viva simulation exercise work etc.)

SEE:: Semester End Evaluation

Sample Practical Exercises:

UNIT-I:

1. Write a Python program to display "Hello World".
2. Write a Python program to print your personal information (Name, Course, College, Mobile No.).
3. Write a program to declare and display variables of different data types.
4. Write a program to accept user input and display it.
5. Write a program to assign multiple values to multiple variables and to assign one value to multiple variables.
6. Demonstrate dynamic variable declaration in Python.
7. Write a program to calculate the sum of two numbers entered by the user.
8. Write a program to swap two variables without using a third variable.
9. Write a program to identify the data type of different variables using type().
10. Write a program demonstrating integer, float, and complex numbers.
11. Write a program to perform arithmetic operations on complex numbers.
12. Write a program to demonstrate Boolean values and logical operations.
13. Write a program to create and display a set datatype.



14. Write a program to demonstrate type casting using `int()`, `float()`, and `str()`.
15. Write a program to access individual characters of a string using indexing.
16. Write a program to perform string slicing operations.
17. Write a program to demonstrate negative indexing in strings.
18. Write a program to find the length of a string using appropriate method.
19. Write a program to concatenate two strings.
20. Write a program demonstrating string methods: `upper()`, `lower()`, `replace()`, `split()`, `count()`, and `join()`.

UNIT – II:

1. Perform arithmetic operations on two numbers.
2. Find the largest of two numbers.
3. Find the largest of three numbers.
4. Check whether a number is positive, negative, or zero.
5. Check whether a number is even or odd.
6. Check voting eligibility.
7. Prepare Marksheet of 12th Standard.
8. Implement a simple calculator using conditional statements.
9. Check whether a year is a leap year.
10. Find the smallest of three numbers.
11. Generate numbers from 1 to 10 using loop.
12. Find the sum of first N natural numbers using while loop.
13. Generate multiplication tables using loop.
14. Generate numbers from 1 to 20 using for.
15. Generate multiplication tables using for.
16. Demonstrate `range()` function.
17. Demonstrate pass statement.
18. Demonstrate else with for loop.
19. Demonstrate nested for loops by printing patterns.



20. Generate Fibonacci series using loops.

UNIT – III:

1. Create and display a list.
2. Access list elements using indexing.
3. Perform various list slicing operations.
4. Perform Add, Insert, Remove, Delete, Sort and Reverse elements in a list using appropriate end().
5. Count occurrences of an element in the list.
6. Perform Copy operation on a list.
7. Create and display tuples.
8. Access tuple elements.
9. Use count() and index() on tuples.
10. Convert a list into a tuple.
11. Create a set and display elements.
12. Add and Remove elements to a set.
13. Perform union of two sets.
14. Demonstrate update() method.
15. Create dictionary and Add and Access a dictionary elements.
16. Remove elements using pop().
17. Copy dictionaries.
18. Create a NumPy array from a list.
19. Calculate Mean, Median, Mode, Standard Deviation and Variance using NumPy.
20. Create a simple data visualization (Bar Chart/Line Chart) using Python.

UNIT – IV:

1. Define and call a simple function.
2. Create a function with parameters.



3. Create a function with return value.
4. Demonstrate local variables.
5. Demonstrate global variables.
6. Create a function with default arguments.
7. Create a function using keyword arguments.
8. Demonstrate *args and **kwargs with proper example.
9. Write a recursive function to calculate factorial.
10. Write a recursive function for Fibonacci series.
11. Create a function to check prime numbers.
12. Create a function to calculate simple interest.
13. Create a function to find maximum of three numbers.
14. Create a lambda function for addition.
15. Create a lambda function for square of a number.
16. Use map() to square list elements.
17. Use map() to convert strings to uppercase.
18. Use filter() to find even numbers.
19. Use filter() to find prime numbers.
20. Use reduce() to calculate product of list elements.