

‘ओ’ लेवल के संशोधित पाठ्यक्रम (M3-R5.1) पर आधारित

प्रोग्रामिंग एंड प्रोब्लम सोल्विंग थ्रू पायथन



हिन्दी माध्यम

लेखक

Manish Mathur

[B.Sc.(Maths), M.Sc.(IT), MBA]

Director: Softech Informatic Systems, Ajmer

Mohit Mathur

[B.Sc., SCCP (Sun, USA), M.Sc.(IT), MCA]

Chief Executive: Softech Informatic Systems, Ajmer

आस्था पब्लिकेशंस

17A, जेम्स बगीची, विद्याधर नगर, जयपुर (राजस्थान)

Published by:

AASTHA PUBLICATIONS

17-A, Jems Bagichi Vidhyadhar Nagar Jaipur-302023

Phone Lines: 98-8787-1616, 722-0000-720

E-mail: aasthapublications@Yahoo.com

Website: aasthapublications.com

© All Right Reversed

Copyright©2025 by the Aastha Publications No part of this publication may be reproduced, stored in a retrieval system, used in a spreadsheet, or transmitted in any form or by any means electronic, mechanical, photocopying or otherwise-without prior permission in written permission of the publisher.

Second Edition: 2025

ISBN: 978-81-983945-0-7

SKU ID : 200PTP135H352L1

MRP: ₹ 200/- (Two Hundred Only)

For any clarification regarding this book, the student may please write to Aastha Publications giving the above reference number for this including chapter number and page number. While every possible care has been taken in typesetting and printing this book, publisher welcomes suggestions from students for improvement in future editions.

Registered Office: B-45, Ganesh Colony, Pilani-333031 (Rajasthan)

PREFACE

प्रोग्रामिंग को हमेशा एक कला के रूप में माना जाता है और प्रोग्रामर हमेशा प्रसंस्करण और आवश्यक कार्यक्षमता के साथ अपने प्रोग्राम को आसान बनाने के प्रयास में कोड के संस्करणों के साथ संघर्ष करते हैं। इन सभी आवश्यकताओं के लिए पायथन सही सूट करता है। यह बहुत कॉम्पैक्ट, हल्के वजन, फ्री और ओपन सोर्स भाषा है। इस पुस्तक को तैयार करने के हमारे प्रयास में हमने आपको पायथन में एक संरचित प्रोग्राम लिखने के लिए सभी आवश्यक टिप्स और ट्रिक्स को समझाने की पूरी कोशिश की है।

इस पुस्तक को 12 अध्यायों में विभाजित किया गया है जहां प्रत्येक अध्याय के अंत में ओ—लेवल परीक्षा पैटर्न के आधार पर प्रश्न (बहु विकल्पीय प्रश्न) दिए गए हैं। इसके अलावा पुस्तक के अंत में एक Sample Paper भी दिया गया है जो कि नवीनतम पाठ्यक्रमानुसार है। इस पुस्तक के सभी पाठकों से अनुरोध है कि वे उन अभ्यासों को हल करें। हमें यकीन है कि आप इस पुस्तक का पूरा लाभ उठाएंगे और अपनी परीक्षा में सर्वोच्च अंक प्राप्त करके अपने माता—पिता और शिक्षकों को गर्व महसूस कराएंगे।

हमने इस पुस्तक को बिना त्रुटियों और प्रिंटिंग या प्रूफ रीडिंग में अन्य गलतियों के बिना, पूरा करने की पूरी कोशिश की है, लेकिन फिर भी कुछ त्रुटियां संभव हो सकती हैं और इस संबंध में आपके सहयोग की आवश्यकता होगी। हम पाठकों से अनुरोध करते हैं कि वे पुस्तक की गुणवत्ता में सुधार करने के लिए अपने सुझाव अवश्य दें। अपने सुझाव दी गई ईमेल पर भेजें।

लेखक 

Outline of Module

Objectives

The objectives of this module are to make the learners understand the programming language concepts like Data Types, Loops, Functions; Python Lists, Strings, Tuples, Dictionaries, Elementary Data Handling using Pandas, NumPy etc.

After completion of this course, the learner is expected to analyze the real life problem and write a program in Python to solve the problem. The main emphasis of the module will be on writing algorithm to solve problems and implement in Python. After completion of the module, the learner will be able to

- Draw flow charts for solving different problems
- Develop efficient algorithms for solving a problem
- Use the various constructs of Python viz. conditional, iteration
- Program making judicious use of Lists, Strings, Tuples, Dictionaries wherever required
- Manage data using NumPy
- Handle files and create Modules in Python

Duration

120 Hours - (Theory: 48hrs + Practical: 72 hrs)

Sr.	Module Unit or Chapter name	Duration		Marks (Max.)
		Theory	Practical	
1.	Introduction to Programming	2	3	20
2.	Algorithm and Flowcharts to Solve Problems	6	9	
3.	Introduction to Python	2	3	30
4.	Operators, Expressions and Python Statements	10	15	
5.	Sequence Data Type	6	9	
6.	Functions	10	15	40
7.	File Processing	6	9	
8.	Modules	2	3	
9.	NumPy Basics	4	6	10
Total		48	72	100

SYLLABUS

1. Introduction to Programming

The basic Model of computation, algorithms, flowcharts, Programming Languages, compilation, testing & debugging and documentation.

2. Algorithms and Flowcharts to Solve Problems

Flow Chart Symbols, Basic algorithms/flowcharts for sequential processing, decision based processing and iterative processing. Some examples like: Exchanging values of two variables, summation of a set of numbers, Decimal Base to Binary Base conversion, Reversing digits of an integer, GCD (Greatest Common Divisor) of two numbers, Test whether a number is prime, factorial computation, Fibonacci sequence, Evaluate 'sin x' as sum of a series, Reverse order of elements of an array, Find largest number in array, Print elements of upper triangular matrix.

3. Introduction to Python

Python Introduction, Technical Strength of Python, Introduction to Python Interpreter and program execution, Using Comments, Literals, Constants, Python's Built-in Data types, Numbers (Integers, Floats, Complex Numbers, Real, Sets), Strings (Slicing, Indexing, Concatenation, other operations on Strings), Accepting input from Console, printing statements, Simple 'Python' programs.

4. Operators, Expressions and Python Statements

Assignment statement, expressions, Arithmetic, Relational, Logical, Bitwise operators and their precedence, Conditional statements: if, if-else, if-elif-else; simple programs, Notion of iterative computation and control flow –range function, While Statement, For loop, break statement, Continue Statement, Pass statement, else, assert.

5. Sequence Data Types

Lists, tuples and dictionary, (Slicing, Indexing, Concatenation, other operations on Sequence datatype), concept of mutability, Examples to include finding the maximum, minimum, mean; linear search on list/tuple of numbers, and counting the frequency of elements in a list using a dictionary.

6. Functions

Top-down approach of problem solving, Modular programming and functions, Function parameters, Local variables, the Return statement, DocStrings, global statement, Default argument values, keyword arguments, VarArgs parameters. Library function-input(), eval(), print(), String Functions: count(), find(), rfind(), capitalize(), title(), lower(), upper(), swapcase(), islower(), isupper(), istitle(), replace(), strip(), lstrip(),rstrip(), split(), partition(), join(), isspace(), isalpha(), isdigit(), startswith(), endswith(), encode(), decode(), String: Slicing, Membership, Pattern Matching, Numeric Functions: eval(), max(), min(), pow(), round(), int(), random(), ceil(), floor(), sqrt(), Date & Time Functions, Recursion.

7. File Processing

Concept of Files, File opening in various modes and closing of a file, Reading from a file, Writing onto a file, File functions-open(), close(), read(), readline(), readlines(), write(), writelines(), tell(), seek(),

8. Scope and Modules

Scope of objects and Names, LEGB Rule Module Basics, Module Files as Namespaces, Import Model, Reloading Modules.

9. NumPy Basics

Introduction to NumPy, ndarray, datatypes, array attributes, array creation routines, Array From Existing Data, Array From Numerical Ranges, Indexing & Slicing.

1. Introduction to Programming.....	1-20
2. Problem Solving Through Flowchart.....	21-36
3. Python Fundamentals.....	37-57
4. Control Statement.....	58-92
5. Sequence Data Type : LIST.....	93-122
6. Sequence Data Type : TUPLE.....	123-143
7. Sequence Data Type : Dictionary	144-165
8. Library Functions.....	166-205
9. User Defined Functions.....	206-227
10. File Processing.....	228-259
11. Scope and Modules.....	260-277
12. NumPy Basics:Array.....	278-306
13. Project.....	307-330
14. Sample Paper.....	331-340

DETAILED SYLLABUS

1. Introduction to Programming

- 1.1 Basic Model of Computation
 - 1.1.1 Recording
 - 1.1.2 Classification
 - 1.1.3 Sorting
 - 1.1.4 Calculation
 - 1.1.5 Summarization
 - 1.1.6 Reporting
- 1.2 Programming Languages
 - 1.2.1 Machine Language
 - 1.2.2 Assembly Language
 - 1.2.3 High Level Language
 - 1.2.4 Fourth Generation Language
 - 1.2.5 Fifth Generation Language
- 1.3 Steps of Programming
 - 1.3.1 Program Analysis
 - 1.3.2 Program Designing
 - 1.3.3 Program Coding
 - 1.3.4 Program Debugging
 - 1.3.5 Program Documentation
 - 1.3.6 Program Maintenance
- 1.4 Compilation
- 1.5 Testing and Debugging
- 1.6 Documentation
- 1.7 Algorithm

2. Problem Solving Through Flowchart

- 2.1 Introduction
- 2.2 Standard Boxes or Diagram
 - 2.2.1 Terminal, Input-Output Box
 - 2.2.2 Process Box, Decision Box
- 2.3 Examples

3. Python Fundamentals

- 3.1 Introduction
- 3.2 Working with Python
 - 3.2.1 Interactive Mode of Python
 - 3.2.2 Program mode
- 3.3 Creating a Module/Program file
- 3.4 Executing Python Program
- 3.5 Python Tokens
 - 3.5.1 Keyword
 - 3.5.2 Identifier
 - 3.5.3 Literal
 - 3.5.4 Operator
 - 3.5.5 Punctuation
- 3.6 print() statement
- 3.7 input ()

4. Control Statement

- 4.1 Introduction
- 4.2 Conditional Statements
 - 4.2.1 'if' statement
 - 4.2.2 if-else statement
 - 4.2.3 if-elif statement
 - 4.2.4 Nested if Statement
 - 4.2.5 Complex Conditions
 - 4.2.6 Stored Conditions
- 4.3 Looping Statement
 - 4.3.1 while() loop
 - 4.3.2 for() loop
 - 4.3.3 Jump statements
 - 4.3.4 Loop-else Statement
 - 4.3.5 Nested Loops

4.4 'pass' statement

4.5 Assert Statement

5. Sequence Data Type: LIST

5.1 Introduction

5.2 Creating a List

5.2.1 Creating List by assigning values

5.2.2 Creating List through other sequences

5.2.3 Creating List dynamically via keyboard

5.3 Accessing the List

5.3.1 Accessing whole List

5.3.2 Accessing individual element

5.3.3 Traversing the List

5.4 List Operations

5.4.1 Combining the Lists

5.4.2 Replicating the List

5.4.3 Slicing the List

5.4.4 Adding elements to List

5.4.5 Updating the Element

5.4.6 Deleting Elements

5.4.7 Copying the List

5.5 Functions and Methods of List

5.5.1 len()

5.5.2 append()

5.5.3 extend()

5.5.4 insert()

5.5.5 pop()

5.5.6 remove()

5.5.7 reverse()

5.5.8 sort()

6. Sequence Data Type:TUPLE

6.1 Introduction

6.2 Creating a Tuple

6.2.1 Creating Tuple by assigning values

6.2.2 Creating Tuples from Existing Sequence

6.2.3 Creating Tuple dynamically via keyboard

6.3 Accessing the Tuple

6.3.1 Accessing whole Tuple

6.3.2 Accessing individual element

6.3.3 Traversing the Tuple

6.4 Tuple Operations

6.4.1 Combining the Tuples

6.4.2 Replicating the Tuple

6.4.3 Slicing the Tuple

6.4.4 Unpacking the Tuple

6.5 Functions and Methods of Tuple

6.5.1 len()

6.5.2 max()

6.5.3 min()

6.5.4 count()

6.5.5 index()

7. Sequence Data Type: Dictionary

7.1 Introduction

7.2 Creating a Dictionary

7.2.1 Creating Dictionary using Assignment method

7.2.2 Creating Dictionary dynamically via keyboard

7.3 Accessing the Dictionary

7.3.1 Accessing whole Dictionary

7.3.2 Accessing individual element

7.3.3 Traversing the Dictionary

7.4 Dictionary Operations

7.4.1 Adding elements to Dictionary

7.4.2 Creating Dictionary using keys and values separately

7.4.3 Creating Dictionary using key: value pair as List

7.4.4 Updating Dictionary element

7.4.5 Deleting Dictionary element

7.4.6 Checking existing of key

7.5 Nested Dictionary

7.6 Dictionary Functions and Methods

7.6.1 len()

7.6.2 clear()

7.6.3 get()

7.6.4 has_key()

7.6.5 items ()

7.6.6 keys()

7.6.7 values()

7.6.8 update()

8.3.8 pow()

8.3.9 sin()

8.3.10 cos()

8.3.11 tan()

8.3.12 degrees()

8.3.13 radians()

8.3.14 trunc()

8.4 Random module

8.4.1 random()

8.4.2 randint()

8.5 String Manipulations

8.5.1 String Operators

8.5.2 String Slice

8.5.3 String Functions

8.5.4 String Methods

8.6 Date and Time Functions

8.6.1 'time' Module

8.6.2 'calendar' Module

8. Library Functions

8.1 Introduction

8.2 Importing Module

8.2.1 To import entire module

8.2.2 To import entire module

8.2.3 To import entire module

8.3 Math module

8.3.1 abs()

8.3.2 sqrt()

8.3.3 exp()

8.3.4 ceil()

8.3.5 floor()

8.3.6 log()

8.3.7 log10()

9. User Defined Functions

9.1 Introduction

9.2 Defining a function

9.3 Invoking a function

9.4 Function Arguments

9.4.1 Positional Arguments

9.4.2 Default Arguments

9.4.3 Named Arguments

9.5 Type of Functions

9.5.1 Function with Input and Output

9.5.2 Function with Input and without Output

9.5.3 Function without Input and with Output

9.5.4 Function without Input-Output

9.6 Scope of Variables

9.7 RECURSION

10. File Processing

10.1 Introduction

10.2 Text File Processing

10.2.1 open()

10.2.2 close()

10.2.3 write()

10.2.4 readline()

10.2.5 More commands to write and read the file

10.2.6 Random Reading in file

10.3 Binary File Processing

10.3.1 'pickle' module

10.3.2 open()

10.3.3 close()

10.3.4 dump()

10.3.5 load()

10.3.6 Checking End of File

10.3.7 Deletion of records

10.4 Object Oriented File Processing

11. Scope and Module

11.1 Scope Rule

11.1.1 Global Scope

11.1.2 Local Scope

11.1.3 Accessibility of Global and Local variable

11.1.4 LEGB Rule

11.1.5 Local and Global variable with same name

11.2 Module

11.2.1 Creating Module

11.2.2 Importing Module

11.2.3 Calling Module Objects

11.2.4 Module commands

12. File Processing

12.1 Introduction

12.2 Creating NumPy Array

12.2.1 Creating array using 'array()' function

12.2.2 Creating array using 'arange()' function

12.2.3 Creating array using linspace() function

12.3 Dimensions of Array

12.3.1 Zero-Dimensional Array

12.3.2 Single-dimensional Array

12.3.3 Multi-dimensional Array

12.4 Shape of Array

12.5 Array Indexes

12.6 Array Slicing

12.7 Array Functions

12.7.1 reshape()

12.7.2 may_share_memory()

12.7.3 ones()

12.7.4 zeros()

12.7.5 ones_like()

12.7.6 zeros_like()

12.7.7 empty()

12.7.8 copy()

12.8 Identity Array

12.8.1 identity()

12.8.2 eye()