M3-R5

PROGRAMMING AND PROBLEM SOLVING THROUGH PYTHON LANGUAGE

OVERVIEW OF MODULE

Introduction to Module

Python is easy to use, powerful and versatile programming language, making it a great choice for developers. Python is used widely in different areas likes building Raspberry Pi applications, writing script program for desktop applications, configuring servers, developing machine learning& data analytics applications and developing web applications.

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
- Write programs making judicious use of Lists, Strings, Tuples, Dictionaries wherever required
- Manage data using NumPy
- Handle files and create Modules in Python

Duration

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

Outline of Module

Module Unit	Duration (Theory) in Hours	Duration (Practical) in Hours	Learning Objectives
1. Introduction to Programming	2	3	After completion of this unit of module, Learner will be able to • Understand the concept of Programming. • Understand evolution of Programming.
2. Algorithm and Flowcharts to solve problems	6	9	After completion of this unit of module, Learner will be able to • Understand the concepts and purposes of algorithm and flowchart. • Use algorithm and flowchart to solve problem independent of language. • Gain knowledge of different constructs of algorithm and flowchart.
3. Introduction to Python	2	3	After completion of this unit of module, candidate will be able to • Understand features of Python that make it one the most popular languages in the industry. • Understand structure of Python problem. • Understand the areas where Python is used.
4. Operators, Expressions and Python Statements	10	15	After completion of this unit of module, Learner will be able to • Use the basic operators and expressions available in Python in developing program. • Understand and use various Python statements like conditional constructs, looping constructs in writing Python program.

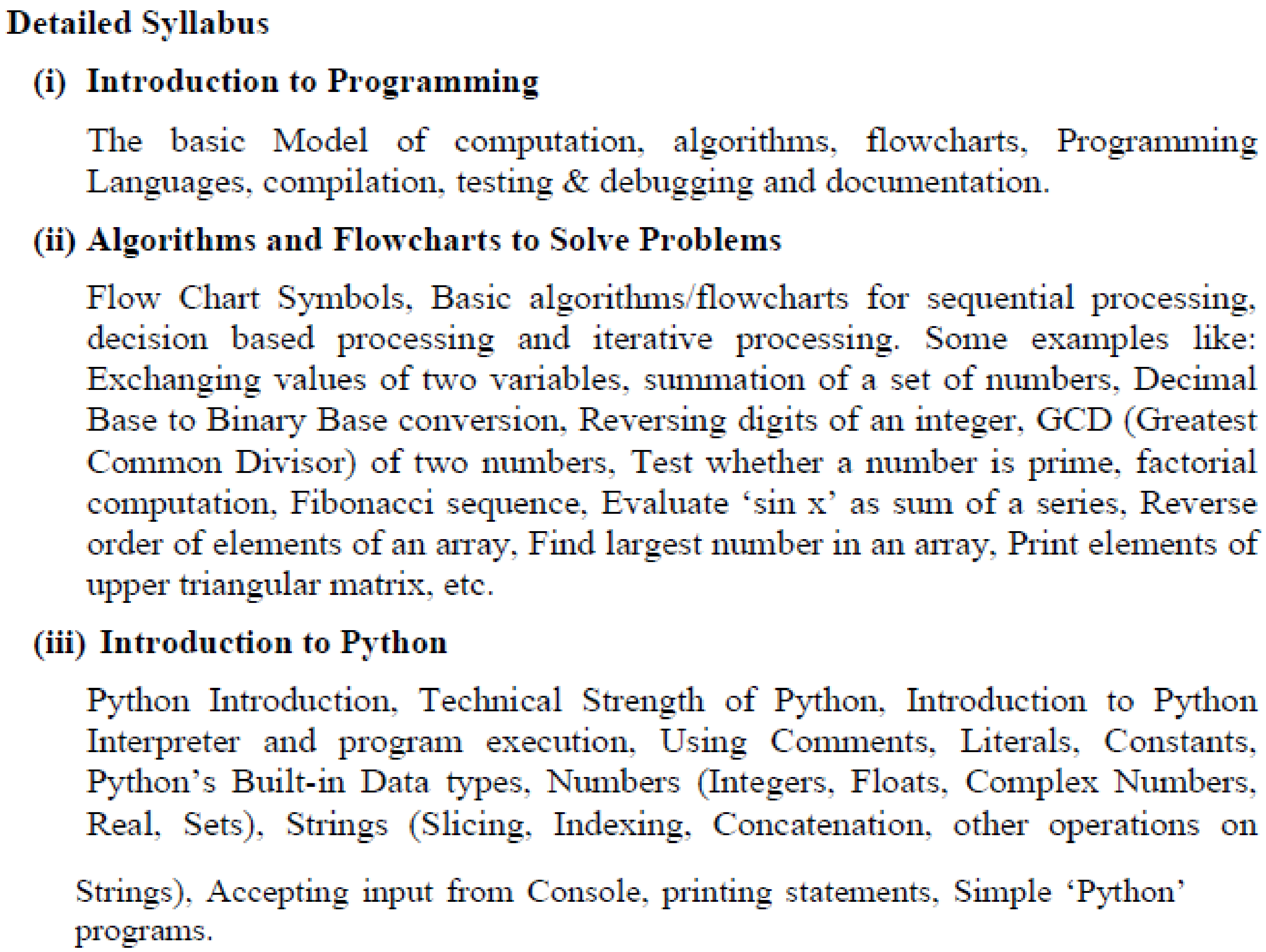
5. Sequence data types	6	9	After completion of this unit of module, Learner will be able to • Work with various built-in Sequence datatypes and their use • Understand the concept of mutable and immutable objects
6. Functions	10	15	After completion of this unit of module, Learner will be able to • Apply the in-built functions available in Python in solving different problems. • Work with modular approach using user defined functions.
7. File Processing	6	9	After completion of this unit of module, Learner will be able to work with files and reading /writing onto files.
8. Modules	2	3	After completion of this unit of module, Learner will be able to • Understand the concept of modules and importing, loading and reloading of modules in programs.
9. NumPy Basics	4	6	After completion of this unit of module, Learner will be able to • Work on NumPy array manipulation to access data and subarrays and to split, reshape, join arrays etc
Total	48	72	

Marks Distribution

Module Unit	**Written Marks (Max.)**
1. Introduction to Programming, Algorithm and Flowcharts to solve problems	20
2. Introduction to Python, Operators, Expressions and Python Statements, Sequence data types	30
3. Functions, File Processing, Modules	40
4. NumPy Basics	10
5. Total	100

M3-R5: PROGRAMMING AND PROBLEM SOLVING THROUGH PYTHON LANGUAGE

SYLLABUS



Detailed Syllabus

(i) Introduction to Programming

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

(ii) 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 an array, Print elements of upper triangular matrix, etc.

(iii) 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.

(iv) 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.

(v) 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.

(vi) 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(), aplit(), partition(), join(), isspace(), isalpha(), isdigit(), isalnum(), 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.

(vii) 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(), Command Line arguments.

(viii) Scope and Modules

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

(ix) NumPy Basics

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