



Faculty of: **Computer Science**
 Course: **Master of Computer Applications**
 Semester: **II**
 Subject Code: **5CS02CPP1**
 Subject Name: **Python Programming**

Sr. No	Subject Code	Subject Name	Teaching hours/ Week			Credit hours	Credit Points	Evaluation Scheme/ Semester								Total
			Th	Tu	Pr			Theory				Practical				
								Internal Assessment		End Semester Exams		Internal Assessment		End Semester Exams		
								Marks	Duration	Marks	Duration	Marks	Duration	Marks	Duration	
4	5CS02CPP1	Python Programming	4	--	2	6	5	15	1	70	2½	15	1	--	--	100

Objectives:

- To able to develop, automate, and test applications and systems using one of the open source programming language.

Pre-requisites:

- Students should have prior programming experience and be familiar with basic concepts such as variables/scopes, flow-control, and functions.
- Prior exposure to object-oriented programming concepts is not required, but definitely beneficial.

Course outline:-

Sr. No.	Course Contents	Number of Hours
1	Installing Python, Simple program using Python, Expressions and Values, Variables and Computer Memory, error detection, Multiple line statements, Designing and using functions, functions provided by Python, Tracing function calls in memory model, omitting return statement. Working with Text: Creating Strings of Characters, Using Special Characters in Strings, Creating a Multiline String, Printing Information, Getting Information from the Keyboard.	08
2	A Boolean Type , Choosing Statements to Execute, Nested If Statements , Remembering the Results of a Boolean Expression Evaluation , A Modular Approach to Program Organization, Importing Modules , Defining Your Own Modules, Testing Code Semi automatically Grouping Functions Using Methods: Modules, Classes, and Methods , Calling Methods the Object-Oriented Way, Exploring String Methods, Underscores.	10
3	Storing and Accessing Data in Lists, modifying Lists, Operations on Lists, Slicing Lists, Aliasing, List Methods, Working with a List of Lists. Repeating Code Using Loops: Processing Items in a List, Processing Characters in Strings, Looping Over a Range of Numbers, Processing Lists Using Indices, Nesting Loops in Loops, Looping Until a Condition Is Reached, Repetition Based on User Input, Controlling Loops Using Break and Continue Reading and Writing.	10

4	Kinds of files, Opening a File, Techniques for Reading Files, Files over the Internet, Writing Files, and Writing Algorithms That Use the File-Reading Techniques, Multiline Records. Storing Data Using Other Collection Types: Storing Data Using Sets, Storing Data Using Tuples, Storing Data Using Dictionaries, Inverting a Dictionary, Using the In Operator on Tuples, Sets, and Dictionaries, Comparing Collections.	10
5	Understanding a Problem Domain , Function “Isinstance,” Class Object, and Class Book , Writing a Method in Class Book, Plugging into Python Syntax: More Special Methods ,Creating Graphical User interface: Building a Basic GUI, Models, Views, and Controllers, Customizing the Visual Style Introducing few more Widgets, Object-Oriented GUIs, Keeping the Concepts from Being a GUI Mess.	10
Total Hours		48

PRACTICAL LIST: Perform practical based on above contents.

Reference Books:

1. Practical Programming: An introduction to Computer Science Using Python, second edition, Paul Gries, Jennifer Campbell, Jason Montojo, The Pragmatic Bookshelf.
2. Learning with Python: How to Think Like a Computer Scientist Paperback – Allen Downey , Jeffrey Elkner, 2015
1. Introduction to Python for Computational Science and Engineering (A beginner's guide), Hans Fangohr.
2. Exploring Python, Timothy A. Budd, Mc Graw Hill Education
3. Python for Informatics: Exploring Information, Charles Severance.
4. Learning Python, Fourth Edition, Mark Lutz, O'Reilly publication

NPTEL Resources:

1. The Joy of Computing using Python, IIT Ropar Prof. Sudarshan Iyengar
<https://nptel.ac.in/courses/106106182>