**BLUE PRINT FORMAT**

**III Semester**

**PYTHON PROGRAMMING**

**SUBJECT : PYTHON PROGRAMMING**

Maximum Marks : 70

Duration : 3 hours

Course Code : PP203.3C

Course Title : **PYTHON PROGRAMMING**

Section A : 5 Questions of 8, each carrying 6 marks (6 x 5 = 30)

Section B : 4 Questions of 6, each carrying 10 marks (4 x 10 = 40)

**Distribution of Marks**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Unit** | **Hours Allotted in the Syllabus** |  | **No. of Questions & Marks Distribution** | | | |
| **Section A** Conceptual/Objective Questions  **(No.of questions including Choice in Section A)** | **(Total Marks including Choice in Section A)** | **Section B** Analytical Questions  **(No.of questions including Choice in Section B)** | **(Total Marks including Choice in Section B)** | **Total Marks** |
| I | 13 | 2 | 12 | 1 | 10 | 22 |
| II | 13 | 2 | 12 | 2 | 20 | 32 |
| III | 13 | 2 | 12 | 2 | 20 | 32 |
| IV | 13 | 2 | 12 | 1 | 10 | 22 |
| **Total** | **52** | **8** | **48** | **6** | **60** | **108** |

**MCA**

**III SEMESTER**

**PYTHON PROGRAMMING**

**Total Teaching Hours: 52 Hours/Week: 4**

**Max Marks: 100 Credits: 4**

**Course Description**

Python is high-level object-oriented programming language. It is vastly used for Artificial Intelligence (AI), Machine Learning, and Deep Learning projects. This course covers programming paradigms brought in by Python with a focus on Regular Expressions, List and Dictionaries. The course explores the various modules and libraries to cover the complete landscape of Python programming.

**Course Objectives:**

1. This course introduces core programming basics—including data types, control structures, algorithm development, and program design with functions
2. The course discusses the fundamental principles of Object-Oriented Programming, as well as in-depth data and information processing techniques.
3. Students will solve problems, explore real-world software development challenges, and create practical and contemporary applications.

**Learning Outcomes:**

1. Discover Python as a useful scripting language for developers.
2. Apply lists, tuples, and dictionaries in Python programs.
3. Identify Python object types indexing and slicing to access data in Python programs.
4. Define the structure and components of a Python program.
5. Design and package Python modules for reusability.

**UNIT I [13 Hours]**

**Introduction to Python Programming Language**: Introduction to Python Language, Strengths and Weaknesses, IDLE, Dynamic Types, Naming Conventions, String Values, String Operations, String Slices, String Operators, **Numeric Data Types**, Conversions, **Built in Functions**. **Data Collections and Language Component:** Introduction, Control Flow and Syntax, Indenting, The if Statement, Relational Operators, Logical, Operators, True or False, Bit Wise Operators, The while Loop, break and continue, for Loop, Lists, Tuples, Sets, Dictionaries, Sorting Dictionaries, Copying Collections

**UNIT II [13 Hours]**

**Strings and text files**; manipulating files and directories, os and sys modules; **text files**: reading/writing text and numbers from/to a file; creating and reading a formatted file (csv or tab-separated). **String manipulations**: subscript operator, indexing, slicing a string; strings and **number system**: converting strings to numbers and vice versa. Binary, octal, hexadecimal numbers

**Lists, tuples, and dictionaries**; basic list operators, replacing, inserting, removing an element; searching and sorting lists; dictionary literals, adding and removing keys, accessing and replacing values; traversing dictionaries. **Design with functions**: hiding redundancy, complexity; arguments and return values; formal vs actual arguments, named arguments. Program structure and design. **Recursive functions**.

**UNIT III [13 Hours]**

**Simple Graphics and Image Processing**: “turtle” module; simple 2d drawing - colors, shapes; digital images, image file formats, image processing - Simple image manipulations with 'image' module (convert to bw, greyscale, blur, etc). **Classes and OOP**: classes, objects, attributes and methods; defining classes; design with classes, data modeling; persistent storage of objects. OOP, continued: inheritance, polymorphism, operator overloading; abstract classes; exception handling, try block

**UNIT – IV: [13 Hours]**

Graphical user interfaces; event-driven programming paradigm; tkinter module, creating simple GUI; buttons, labels, entry fields, dialogs; widget attributes - sizes, fonts, colors layouts, nested frames **Multithreading, Networks, and Client/Server Programming**; **introduction to HTML**, interacting with remote HTML server, running html-based queries, downloading pages; **CGI programming**, programming a simple CGI form. **Searching, Sorting, and Complexity Analysis**

**Text Books:**

1. Kenneth A. Lambert, The Fundamentals of Python: First Programs, 2011, Cengage Learning, ISBN: 978-1111822705.
2. Python Data Science Handbook: Essential Tools for Working with Data, [Jake VanderPlas](https://www.amazon.in/Jake-VanderPlas/e/B01N5KCUPA?ref=sr_ntt_srch_lnk_1&qid=1640335621&sr=1-1), ISBN : 1491912057
3. Python Cookbook: Recipes for Mastering Python 3, OREILLEY [David Beazley](https://www.amazon.in/s/ref=dp_byline_sr_ebooks_1?ie=UTF8&field-author=David+Beazley&text=David+Beazley&sort=relevancerank&search-alias=digital-text), ISBN: 978-1449340377
4. Steven Bird, Ewan Klein and Edward Loper, ―Natural Language Processing with Python, First Edition, OReilly Media, 2009.

**Reference Books:**

1. Python for Data Analysis, 2e: Data Wrangling with Pandas, Numpy, and Ipython, Wes Mckinney, OREILLY, ISBN: 978-1491957660
2. Python Programming: An Introduction to Computer Science, John M Zelle, ISBN: 978-1590282755