| Course Type | Course Code | Name of the Course | L | T | P | Credits |
|---|---|---|---|---|---|---|
| DC | NHSC508 | Python Programming Essentials | 3 | 1 | 0 | 4 |

| **Course Objective** |
|---|
| To provide social science and humanities undergraduates with a comprehensive understanding of Python programming fundamentals enabling them to apply programming concepts to their respective fields of study |
| **Learning Outcomes** |
| After successful completion of the course, the student should be able to:<br>1. Develop foundational programming skills in Python.<br>2. Gain proficiency in problem-solving and algorithmic thinking.<br>3. Apply Python programming concepts to analyze data and solve real-world problems. |

| Unit No. | Topics to be Covered | Lecture + Tutorial Hours | Learning Outcome |
|---|---|---|---|
| 1 | Introduction to Python Basics<br>Installation, Coding environments, Text Editors | 2L | Understand the foundational elements of Python programming, including installation, coding environments, variables, strings, numbers, and commenting practices |
| 2 | Programming Basics:<br><br>Variables: In-built, types, operations<br>Expressions: assignment, comparison | 2L + 1T | Understand the concept of computer program, its syntax and utility of programs in research. |
| 3 | Data Types: types, operations, working with stings<br><br>Data Structures: Lists, Dictionaries, Tuples and their methods and applications | 7L + 2T | . Understanding and using basic data types and their use in writing programs.<br>Understand various in built data structures and their methods. |
| 4 | Control Statements<br>Condition Statements: syntax and use of IF, ELSE, ELIF, AND, OR<br>Loops: WHILE loop, user input, FOR loop<br>Files – opening, reading, handling exceptions | 9L + 2T | Learn control statements in Python, covering conditional statements , loop structures, and file operations including opening, reading, and exception handling for robust file processing |
| 5 | Functions: definition, purpose, arguments, return values, error-handling, flexible arguments (*args, **kwargs), lambda<br>Scripts: splitting code | 5L + 2T | Learn functions in Python, covering their definition, purpose, handling of arguments and return values, error management, and their integration within scripts for modular and efficient code organization |
| 6 | Object Oriented Programming: Classes – creation, instances<br>Inheritance – child class, overriding methods, importing classes | 6L + 3T | Grasp object-oriented programming principles in Python, covering class creation, instantiation, inheritance with child classes, method overriding, and importing classes for modular code organization |
| 7 | Working with Data<br>String manipulation, regular expressions – searching pattern matching<br>Preview of Data Science tools: NumPy, Pandas, Matplotlib | 8L + 3T | Gain proficiency in working with text data in Python, encompassing string manipulation techniques and regular expressions for efficient pattern matching and searching |
| 8 | Version Control: git, making project, initializing, making and abandoning changes | 3L + 1T | Learn version control using Git, including project creation, initialization, making changes, and effectively managing them through commits, as |

| | | | |
|---|---|---|---|
| | | | well as strategies for reverting or abandoning changes when necessary |
| | **Total** | **42L+14T** | |

**Text Books:**

1. Matthes, E (2023). Python Crash Course, 3rd Ed. No Starch Printing.
2. VanderPlas, J. (2016). A Whirlwind Tour of Python. O'Reilly Media, Inc

**Reference Books:**

1. Montfort, N. (2021). Exploratory Programming for the Arts and Humanities. 2nd Ed. MIT
2. Wang, H. (2023). Introduction to Computer Programming with Python. Remix