ADVANCED COMPUTER PROGRAMMING

ENCT 153

Lecture : 3 **Year : I**
Tutorial : 1 **Part : II**
Practical : 3

Course Objectives:
This course gives overview about the programming paradigms and focuses on multi-paradigm programming model Python. This course introduces with the semantics of Python programming language and helps learners to master the skills of basic Python programming and object-oriented programming. It also highlights why Python is a useful scripting language for all developers.

1 Programming Paradigms (2 hours)

1.1 Introduction
1.2 Different programming paradigms
1.3 Advantages, disadvantages of different paradigms and examples

2 Introduction to Python Programming (3 hours)

2.1 Need of Python
2.2 History
2.3 Features and limitations of Python
2.4 Python with respect to other languages (C, C++, JAVA, JavaScript)
2.5 Top Python implementations

3 Basic Programming Concept in Python (5 hours)

3.1 Keywords
3.2 Basic data types
3.3 Variables and inputs
3.4 Logic and comparison operations
3.5 Conditional statement
3.6 Loop
3.7 Functions
3.8 Recursion function call

4 Advanced Data Types and Operation in Python (8 hours)

4.1 Mutable and immutable data types
4.2 List and tuple data types
4.3 Dictionary data types
4.4 Sequence data types

- 4.5 Two-dimensional lists
- 4.6 Set data types
- 4.7 Lambda
- 4.8 Operation of mutable and immutable data types

5 Object Oriented Programming (12 hours)

- 5.1 Concepts of object-oriented programming
- 5.2 Classes and objects
  - 5.2.1 Attributes and methods
  - 5.2.2 __init__() and __str__() methods
  - 5.2.3 Delete properties and objects
  - 5.2.4 Iterator in a class
- 5.3 Aggregation and composition
- 5.4 Inheritance
  - 5.4.1 Parent and child classes
  - 5.4.2 __init__() in child class
  - 5.4.3 The super() function
  - 5.4.4 Member overriding
  - 5.4.5 Forms of Inheritance (Single, hierarchical, multiple, multilevel)
- 5.5 Polymorphism and dynamic binding
  - 5.5.1 Abstract class and concrete class
  - 5.5.2 Abstract methods and abstract attributes
- 5.6 Operator overloading in Python
  - 5.6.1 Arithmetic operators
  - 5.6.2 Bitwise and shift operators
  - 5.6.3 Comparison operators
  - 5.6.4 Assignment operators
  - 5.6.5 Unary operators

6 Exceptions and File Handling in Python (5 hours)

- 6.1 Types of errors
- 6.2 Types of exceptions
- 6.3 Catching and handling exceptions
- 6.4 User-defined exceptions
- 6.5 Debugging programs with the assert statement
- 6.6 Logging the exceptions
- 6.7 Introduction to file handling
- 6.8 Opening and closing a file
- 6.9 Working with text and binary files
- 6.10 Random file access

7 Python Libraries and Maths (10 hours)

- 7.1 Modules, packages and libraries

7.2 The standard library and library functions
7.3 Adding more Python libraries
7.4 Python frameworks
7.5 Introduction to the numPy library
7.6 Creating, indexing and slicing numPy arrays
7.7 Copying and editing numPy arrays
7.8 Stacking and restructuring numPy arrays
7.9 Arithmetic operations with numPy arrays
7.10 Operations with numPy arrays of different shapes
7.11 Concatenation, reversion and persistence of numPy arrays
7.12 Applications of numPy random number generation
7.13 Applications of numPy statistics
7.14 Applications of numPy linear algebra

Tutorial **(15 hours)**

After completing each chapter some problems are solved and students are asked to solve programming problems with the teacher's assistance**.**

Assignment

Appropriate assignment problems are given to students after the completion of each chapter.

Practical **(45 Hours)**

There will be ten laboratories exercise covering all the topics. At the end of the course students must submit a programming project report.

Final Exam

The questions will cover all the chapters in the syllabus. The evaluation scheme will be as indicated in the table below:

Chapter	Hours	Mark distribution*
1	2	4
2	3	4
3	5	6
4	8	10
5	12	18
6	5	6
7	10	12
Total	**45**	**60**

* There may be minor deviation in marks distribution.

References

1. Gries, P., Campbell, J., Montojo, J. (2017). Practical programming: An introduction to computer science using Python 3.6. Pragmatic Bookshelf.
2. Savaliya, M.T., Maurya, R.K., Magar, G.M. (2010). Programming through Python. Sybgen Learning India.
3. Lanaro, G., Nguyen, Q., Kasampalis, S. (2019). Advanced Python programming: Build high performance, concurrent, and multi-threaded apps with Python using proven design patterns. Packt Publishing.
4. Summerfield, M. (2018). Programming in Python 3. Pearson Education.
5. Brown, M.C. (2018). Python: The complete reference. McGraw Hill.
6. Hetland, M.L. (2017). Beginning Python: From novice to professional. Apress.