# **Year II Semester II**

**IRE 211: IoT Architecture and Technologies+ Sessional Missing**

1. Description of Course PROG 301

**Section A: General Information**

1.1 Course Title Advanced Programming with Python and Scripting

1.2 Type of Course Common

1.3 Credit Hour 3.00

1.4 Pre-requisite Course(s) PROG 101: Structured Programming Language, PROG 111: Object Oriented Programming, CSE 113: Data Structure and Algorithms.

**Section B: Course Details**

1.5 Course Content

**Course Content**

1. Basic Python
   * Python data model and data structures
     + Dynamic typing and python basic types and conversion.
     + Data structures: lists, string, tuple, sets, dictionaries, collections.
     + Unicode and byte.
     + Mutable vs immutable types.
   * Python’s Structure
     + Module, package and scope.
     + Control flow, loop, context manager
     + Object reference and recycling.
     + Typing as type basic hinting.
   * Functional Programming
     + Function and lambdas.
     + Comprehensions.
     + Closure and decorators.
     + Generators and coroutine.
     + Map, filter, functools and itertools.
   * Object Oriented Programming
     + Class and object.
     + Interface, protocols and ABCs.
     + Inheritance and advanced typing.
     + Magic functions and operator overloading.
     + Data class and pydantic.
2. System Programming
   * Handling command line arguments with argparse.
   * Into to logging, pipes and signal.
   * File and directory management.
   * Intro os and sys module.
   * Networking
     + Sockets and addresses.
     + TCP, UDP.
     + Intro to Paho: MQTT and SSL.
   * Threading.
     + The threading module
     + The queue module.
     + GIL in python.
   * Multiprocessing.
     + The multiprocessing module.
     + Process pool.
     + Shared memory.
   * Asynchronous programming.
     + The asyncio module
     + Event loop
     + Coroutine and task.
3. Data Processing, Visualization and Manipulation
   * Intro to NumPy
     + Data Type
     + Arrays
     + Universal Functions
     + Aggregation
     + Broadcasting
     + Indexing
   * Intro to Pandas
     + Series, dataframe.
     + Indexing (simple, hierarchical), selection.
     + Join, merge, concatenation, append.
     + Aggregation and grouping.
     + Pivot table.
   * Data visualization using Matplotlib.
     + Line plot, scatter plot, density, and contour plot
     + Subplot, legend, test, and annotation.
     + 3D plotting.
   * ML with Scikit-learn, TensorFlow.
     + Data representation in scikit-learn.
     + Hyperparameters and model validation.
     + TensorFlow basics
     + Keras APIs and tensorboard visualization
4. Image Processing with Python
   * Intro to scikit-image and OpenCV.
5. Web Technology
   * Web scraping with beautiful soup.
   * Intro to request and urllib module.
   * Working with json.
   * Web development with FastAPI/Django.
   * ODM and ORM for database query.

1.6 Course Objectives

* Design and implement efficient programs by taking the advantage of Python build-in capabilities and in-depth knowledge.
* Understand functional programming, dynamic object-oriented programming concepts in python.
* Implement different protocols of networking, data analysis, data manipulation, data visualization, parallel programing, system programming concepts and web programming.

1.7 Knowledge required

* Basics of programming: Structured Programming, Object oriented programming etc.

1.8 Course Outcomes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CO No. | CO Statement | Corresponding PO(s)\* | Domains and Taxonomy Level(s)\*\* | Delivery Method(s) and Activity(-ies) | Assessment Tool(s) |
| 1 | **Remember** the syntax, keywords and terminologies related to the Python Programming language. | e.g., PO(a) | C1 | Lectures, Homework. | Written exams, assignment |
| 2 | **Gain** knowledge over efficient programs using Python’s build-in and library capabilities. |  | C2 | Lectures, Homework, Lab. | Written exams, Assignment,  Lab assessment |
| 3 | **Understand** different programming concepts in Python. |  | C3 | Lectures, Homework | Written exams, Assignment,  Lab assessment |
| 4 | **Analyze** the data driven architectures, networking, and system protocols in Python |  | C4 | Lectures, Lab. | Written exams, Assignment,  Lab assessment |
| 5 | **Apply** the gained knowledges to create real world application. |  | C5 | Lectures, Lab. | Written exams,  Project evaluation |

1.9 Mapping of Knowledge Profile, Complex Engineering Problem Solving and Complex Engineering Activities

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| COs | K1 | K2 | K3 | K4 | K5 | K6 | K7 | K8 | P1 | P2 | P3 | P4 | P5 | P6 | P7 | A1 | A2 | A3 | A4 | A5 |
| CO-1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| CO-2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| CO… |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1.10 Lecture Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Lec# | Lecture Topics | Teaching-learning Strategy(-ies) | Assessment Strategy(-ies) | Corresponding CO(s) |
| 1 | * Python data model and data structures * Dynamic typing and python basic types and conversion. * Data structures: Lists, String, Tuple, Sets, Dictionaries, Collections. * Unicode and Byte. * Mutable vs Immutable types. | Lectures, Lab | Class Participation/ Presentation, Class  Performance, Homework/  Assignment, Class  Test, Mid Term and Semester Final  Examination  (Written) | C1 |
| 2 | * Python Structure * Module, package, and scope. * Control flow, loop, context manager * Object reference and recycling. * Typing as type basic hinting. | C1, C2 |
| 3 | * Functional Programming * Function and lambdas. * Comprehensions. * Closure and decorators. * Generators and coroutine. * Map, filter, functools and itertools. | C1, C2, C3 |
| 4 | * Object Oriented Programming * Class and Object. * Interface, Protocols and ABCs. * Inheritance and advanced typing. * Magic functions and operator overloading. * Data class and pydantic. | C1, C2, C3 |
| 5 | * Handling command line arguments with argparse. * Into to logging, pipes and signal. * File and directory management. * Intro os and sys module.   + Networking   + Sockets and addresses. * TCP, UDP. * Intro to Paho: MQTT and SSL. | C1, C2, C3 |
| 6 | * Threading.   + The threading module   + The queue module.   + GIL in python. * Multiprocessing.   + The multiprocessing module.   + Process pool.   + Shared memory. | C2, C3, C4 |
| 7 | * Asynchronous programming.   + The asyncio module   + Event loop   + Coroutine and task. | C2, C3, C4 |
| 8 | * Intro to NumPy * Data Type * Arrays * Universal Functions * Aggregation * Broadcasting * Indexing | C2, C3, C4 |
| 9 | * Intro to Pandas * Series, dataframe. * Indexing (simple, hierarchical), selection. * Join, merge, concat, append. * Aggregation and grouping. * Pivot table | C2, C3, C4 |
| 10 | * Data visualization using Matplotlib. * Line plot, scatter plot, density and contour plot * Subplot, legend, test and annotation. * 3D plotting. | C2, C3, C4 |
| 11 | * ML with Scikit-learn, TensorFlow. * Data representation in scikit-learn. * Hyperparameters and model validation. * TensorFlow basics * Keras APIs and Tensorboard visualization | C2, C3, C4 |
| 12 | * Web Scraping with beautiful soup. * Intro to request and urllib module. * Working with json. | C2, C3 |
| 13 | * Web Development with FastAPI/Django. | Class Participation/ Class Performance,  Semester Final  Examination  (Written), Project | C2, C3, C4, C5 |
| 14 | * ODM and ORM for Database query. |

1.11 Assessment Strategy

* Class Attendance: Class attendance will be recorded in every class.
* Continuous Assessment: Continuous assessment any of the activities such as class participation, quizzes, assignment, presentation, etc. The scheme of the continuous assessment for the course will be declared on the first day of classes.
* Mid-term Examination: A comprehensive mid-term examination will be held at the mid of the term (according to academic calendar) following the guideline of Academic Council.
* Final Examination: A comprehensive final examination will be held at the end of the term (according to academic calendar) following the guideline of Academic Council.

1.12 Distribution of Marks

Class Attendance 10%

Continuous Assessment 30%

Mid-term Examination 24%

Final Examination 36%

Total 100%

* 1. Textbook

1.13.1 **Introducing Python**: Modern Computing in Simple Packages, Bill Lubanovic, 2nd Edition, O’Reilly Media, Inc.

1.13.2 **Fluent Python:** Clear, Concise, and Effective Programming, Luciano Ramalho, 2nd Edition, O’Reilly Media, Inc.

* 1. Reference Book

1.14.1 **Python Data Science Handbook:** Essential Tools for Working with Data, Jake Vanderplas, 2nd Edition, O’Reilly Media, Inc.

1.14.2 **Learning Python,** Mark Lutz, 5th Edition, O’Reilly Media, Inc.

1.14.3 **Programming Python**, Mark Lutz, 4th Edition, O’Reilly Media, Inc.

**PROG 301 Advanced Programming with Python and Scripting Sessional as per theory**