
ADVANCED PYTHON PROGRAMMING

Subject Code: STPY202

Total Hours: 30

Credits: 3

Course Learning Objectives (CLO)

The objective of this course is to make students to

- Understand how to handle different forms of data in python and to visualize it.
- Write code to scrape data from the web and parse HTML and JSON content.
- Understand how REST APIs work and how they can be implemented in Python.
- Implement GUIs in Python.
- Understand how to develop robust software by following testing and debugging practice.

Unit1: Handling Data in Python

[06 hours]

Numerical Computing, Using Pandas for data manipulation, Using Matplotlib and Seaborn, Plotly for Visualisation, Handling common data types: Images, Sound, Pdfs, Spreadsheets and Text

Unit2: Web Scraping and HTML Parsing

[04 hours]

Analyzing a web page, Sequential and Threaded Crawlers, Concurrent Downloading, Caching, Solving CAPTCHA, Requests, HTML Parsing, BeautifulSoup, Scrapy for Web Scraping, Interacting with forms, RSS feeds and third-party APIs

Unit3: Rest API, JSON, and Flask

[08 hours]

What is an API, HTTP, REST Principles, JSON parsing, , Security and Authentication, SQL Databases, SQLAlchemy, Installing Flask, Deploying Flask apps, Dockerizing Flask apps, asynchronous jobs using queues, Celery or Redis

Unit4: GUI Programming and Streamlit

[08 hours]

Tkinter. Labels, message widget, buttons, canvas, entry, canvas, slider, layout management, menus, events and binds

Streamlit: What is Streamlit, Installation, Displaying Text, Displaying Pandas data frames and Json, streamlit widgets, working with files and text input, streamlit forms and apps, Publishing on Streamlit Cloud, Tkinter, PyQt, PySide, QT designer, Dash.

Unit5: Testing and Debugging

[04 hours]

Difference between Testing and Debugging, doctest, unittest and pytest

SKILL BASED EXERCISE (SBE):

Note: - These Projects/activities are only indicative; the faculty member can innovate

Case studies: - These are the case studies that are not published. The teacher will be provided with Notebooks-

- Predicting the Health Insurance Premium
- Identifying questions in video transcripts

Assignments: -

- Data Processing and visualization
- Web scraping Accessing data from 3rd Party APIs, API for simple CRUD operations,
- GUI Development, Steamlit app development for visualizing a dataset

Lab Exercises: -

1. Using any open-source structured data, use Seaborn to visualize and analyze the data statistically
2. Read and Display images and do basic operations on the images like changing the size, Accessing and Modifying pixel values. Accessing Image Properties. Image ROI. Splitting and Merging Image Channels. Making Borders for Images, Read Videos and Access frames of the video
3. Read and play audio files filter, resample and build spectrograms

4. Read and Write data from excel files
5. Write a crawler to crawl a web site. Using an HTML Parser, extract the main text of pages and create word clouds for each page.
6. Create a rest API, dockerize and deploy the API, use Postman to call the API
7. Create a simple GUI application that interacts with the API from the previous exercise and displays the output to the user
8. Create a streamlined app for visualizing data used in the first exercise
9. Add test cases and error handling to the API created in exercise 6

Mini projects:

1. Build a web scraper for a specific website and use the data collected to create a data visualization or predictive model.
2. Develop a Rest API for a small business, such as a bookstore or a restaurant, to manage inventory, orders, and customer data.
3. Create a graphical user interface for a scientific calculator or a simple game using Tkinter.
4. Build a Streamlit application to analyze and visualize data from a popular dataset, such as the Iris or the Boston Housing dataset.
5. Create a data dashboard using Dash or Plotly to visualize and analyze real-time data from a public API's.

The project list is designed to allow learners to apply the skills and techniques covered in the course and showcase their abilities to potential employers or clients. It should involve a significant amount of programming, data analysis, and problem-solving, and demonstrate the learner's ability to work independently and creatively to solve real-world problems.

TEXTBOOKS:

1. Brian Heinold, “A Practical Introduction to Python Programming”, 2022

REFERENCES:

1. Jamie Buelta. Python Automation Cookbook: 75 Python automation ideas for web scraping, data wrangling, and processing Excel, reports, emails, and more
2. Richard Lawson, Web Scraping with Python
3. <https://python-course.eu/>

Course Outcomes:

At the end of the course, students will be able to

Course Outcomes	Description	Bloom’s Taxonomy Level
CO1	Use data from third party APIs, RSS feeds and web sites parsing the data.	Applying (3)
CO2	Demonstrate the parsed data through visualization.	Applying (3)
CO3	Demonstrate security and authentication using built APIs.	Applying (3)
CO4	Present the simulation results with help of GUI.	Applying (3)
CO5	Demonstrate the robustness of deployed optimized code.	Applying (3)
CO6	Demonstrate different the types of testing and debugging techniques	Applying (3)