"Advanced Python Programming for Everybody"

Five reasons you should learn Python programming language now:

1. Python is easy to learn.
2. Python is a flexible language and platform.
3. Python has one of the most mature package libraries around - more than 1,000+ available today.
4. Python is the most popular programming language in Data Science projects.
5. Python is cross-platform and Open Source technology (FREE!)

Instructor: Ernest Bonat, Ph.D.
Senior Software Engineer
Senior Data Scientist
ebonat@15itresources.com
Cell: 503.730.4556

Objective: Learn and improve your Python programming skills with real practical examples.

Prerequisites: Previous Python programming experiences, Probability and Statistics (undergraduate level).

Requirements: A laptop with the latest Python Anaconda distribution package and any popular Python IDE programs (PyCharm, Spyder, Eclipse, NetBeans, Python Tools for Visual Studio, Visual Studio Code, etc.) installed.

Duration: Eight weeks module class, meeting on Wednesdays, 6:00 – 9:00 pm.

Cost: Free.

Module #	Date	Location
1	04/18/2018	CR-HF3-AUD-186
2	04/25/2018	CR-HF3-AUD-186
3	05/02/2018	CR-HF3-AUD-186
4	05/09/2018	CR-HF3-AUD-186
5	05/16/2018	CR-HF3-AUD-186
Not Available	05/23/2018	Not Available
6	05/30/2018	CR-HF3-AUD-186
Not Available	06/06/2018	Not Available
7	06/13/2018	CR-HF3-AUD-186
Not Available	06/20/2018	Not Available
8	06/27/2018	CR-HF3-AUD-186

Weekly Modules:

1. Object-Oriented Programming (functions, input/output arguments, return values, lambda functions, optional arguments, nested functions, recursive functions, multiple arguments, classes, static methods, constructor, basic inheritance, polymorphism, encapsulation, operator overloading, get/set properties, class composition, singleton class, metaclasses, real practical examples, week assignment).

2. Multithreading and Asynchronous Programming (threads, communicating between threads, advanced use of multithreads, stoppable thread with a while loop, concurrency and parallelism, asyncio asynchronous programming, coroutine, tasks, event loop, order of execution, future states, real practical examples, week assignment).

3. Database Access and Data Manipulation (Python drivers for MySQL, SQL Server and MongoDB databases, database connection, SQL data retrieval (Select), CRUD operations and transactions (Insert, Update, Delete, Commit, and Rollback), error handling, calling stored procedures with and without input parameters, SQLAlchemy - SQL toolkit and object relational mapper, real practical examples, week assignment).

4. Extract-Transform-Load (ETL) System Design and Development (system design and implementation (extract, transform and load tiers), pandas library for fast data structure and manipulation, data parse and conversion for CSV, TXT, XML, JSON files, data helper class, real practical examples, week assignment).

5. Design and Developer Python GUI Desktop Applications (GUI desktop application design and development, GUI guides lines, PyQT5 library and GUI components, GUI EXE application deployment using PyInstaller, real practical examples, week assignment).

6. Unit Testing Implementation (unittest - unit testing framework, test discovery, organizing test code, skipping tests and expected failures, classes and functions test cases, loading and running tests, real practical examples, week assignment).

7. Python Data Ecosystem for Data Science Projects (NumPy, pandas SciPy, Statsmodels, Matplotlib, Seaborn, scikit-learn Machine Learning, Cross-Validation, Ensemble Methods (Boosting, Bagging, Boostrap), OOP applied to Data Science projects, real practical examples, week assignment).

8. How to speed up a Python Program? (NumPy, PyPy, compile with Cython, caching computation, compile with Numba, GPU accelerated computing with PyCUDA and Numba libraries combination, real practical examples, week assignment).

Join us today and learn real Python programming!