To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

# High-Performance Python with C and Cython

## Class Duration

- 35 hours of live online training
- Delivered over 5 days

## Target Audience

- All students should have significant experience with the Python and C languages
- The class has no review of the Python or C programming languages
- If you need to learn C programming, please consider the C Programming for Python Developers course

## Description

The High-Performance Python with C training course is for students who have completed a C and Python training class or are self-taught programmers with many years of continuous Python and C experience. The course focuses on supporting legacy C code for Python Extensions and writing new highly performant code with Cython. The course includes content on using NumPy arrays with Python, C, and Cython. Finally, the class will discuss using the Zig language as an alternative to C for coding extensions.

## Learning Objectives

- Explore how and why Python is slow
- Compare the performance of various Python programming techniques
- Leverage the C programming language to write fast code called from Python code
- Learn how to debug, detect memory leaks, and performance profile C code
- Go deep on PyObject and apply reference counting techniques
- Incorporate Numpy into C Extensions
- Build and Package C Extensions created with C
- Build and Package C Extensions created with Cython

## Training Materials

All students receive comprehensive courseware covering all topics in the course. Courseware is distributed via GitHub in the form of documentation and extensive code samples. Students practice the topics covered through challenging hands-on lab exercises.

## Software Requirements

Students will need a free, personal GitHub account to access the courseware. Students will need permission to install Python, C Compiler, and Visual Studio Code on their computers. Also, students will need permission to install Python Packages and Visual Studio Extensions. If students are unable to configure a local environment, a cloud-based environment can be provided.

## Training Topics

### Getting Started
- Debugging Python and C code with VS Code
- Profiling Python Applications
- Speed Gains of Cython
- Speed Gains of C Extensions
- CPU-bound vs. IO-bound
- How to Choose the Best Option

### Debugging
- Compile the C Extension with the Debug Options
- GDB in the Terminal
- GDB, Microsoft C/C++ Extension, and Visual Studio Code
- GDB/Python Integration for Viewing PyObjects
- Debugging Tactics

### Profiling
- Reference Counting and Memory Leaks
- Build a Debug Version of Python
- Detect Memory Leaks with Valgrind
- Perform Sample Profiling with py-spy

### Interfacing Python with C
- CTypes FFI

- C Extensions
- Cython

## CTypes FFI
- Compile C code to a Shared Object
- Load Shared Object into Python
- Call C Functions from Python
- Python Structure Class
- Python Binary Formatted Strings

## Building C Extensions
- Learn to use the Python C API documentation
- Basic Structure of a C Extension
- Compiling C Extensions
- Import C Extensions in Python
- Using PyObject
- Managing Memory and Reference Counting
- Challenges of Multithreaded Programming and the GIL

## Distributing C Extensions
- Packaging C Extensions for PyPi
- Packaging C Extensions for Conda

## NumPy and C Extensions
- Learn to use the Numpy C API documentation
- Using NumPy Arrays with C Code
- Passing NumPy Arrays to C Extension
- Creating NumPy Arrays in a C Extension
- Using the NumPy C API

## Cython Extensions
- What is Cython?
- Installing Cython
- Compile Python Code

## Cython Static Typing
- Python Code Decorations with Cython
- Cython and Python Type Hints

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

## Enhance Python to Cython
- Create Python Extension with Cython
- Cython Implementation and Declaration Files
- Cython's Three Kind of Functions
- Optimizing Loops
- Cython, Numpy, and Types Memory Views
- Parallel Programming with Cython
- Challenges of Multithreaded Programming and the GIL
- Wrap C Libraries in Cython

## Wrapping C with Cython
- Declare and Wrap External C Code in Cython
- Understand Wrapping
- C structs, unions, enums, and typedefs
- C functions
- Error Handling
- Callbacks

## Distribute Cython Extensions
- Packaging Cython Extension with PyPi
- Package a Cython Extension with Conda

## Conclusion
- Summary of Key Concepts
- Q&A
- Further Resources and Next Steps