http://www.oopsla.org/2006/2006/index.php?option=com_content&task=view&id=82&Itemid=315

OOPSLA 2006

ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications
October 22-26, Portland, Oregon, USA

T48: Totally Awesome Computing: Python as a General-Purpose Programming Language

T48: Totally Awesome Computing: Python as a General-Purpose Programming Language

Thursday, Oct 26, from 08:30 to 12:00

Python is well-known as a scripting language and is often used in server-side web programming. What many people don't know is that it is a full-powered object-oriented programming language, with some functional programming support thrown in for good measure. Python is a portable, modular, very-high-level language. Programs in Python are typically 3-10 times smaller in code size than their Java or C++ equivalents, and take a fraction of the time to develop. The Python library is among the most robust of any available--whatever you need, chances are there's a Python library for you. (As they say, the Python release comes with "batteries included".) Another lesser-known fact is that Python evolved from a language designed to teach programming to children, hence its easy learning curve and simple syntax. Python interfaces nicely with C++ and Java, so you can call upon their facilities in the rare case that the standard Python distribution doesn't already solve your problem. This is a soup-to-nuts tutorial for attendees already familiar with most any modern programming language.

Introductory: Attendees should have experience with one or more modern programming languages and familiarity with the principles of object-oriented programming.

Goals: This tutorial convincingly illustrates the awesome power of Python, which will likely cause participants to prefer to use Python for many of their routine programming tasks. Participants will learn and come to appreciate the "Pythonic" way of doing things (using list comprehensions, generators, anonymous functions, etc.). They are also likely to expand their knowledge of the principles of programming languages, since advanced language features (like novel, SmallTalk-like applications of polymorphism) are so easy to teach in Python.

Format: The presentation will be a mix of lecture, discussion, and hands-on exercises, proceeding through the following topics:

1. A First Look
2. Types and Operations
3. Text Processing
4. Lists and Dictionaries
5. Tuples and Files
6. Statements and Control Structures
7. Functions and Functional Programming
8. Modules and Packages
9. Object-oriented Programming

Chuck Allison, Utah Valley State College: *Chuck Allison has over 20 years of industrial software development experience, and is a professor of Computer Science at Utah Valley State College. He was a contributing member of the C++ Standards Committee throughout the 1990s and designed std::bitset. He was an editor of one sort or another for the C/C++ Users Journal from 1992-*

While Space Available

- Lightning Talks

Search

please enter searchwords

Search

- Conference Schedule
- Week at a Glance
- Receptions and Special Events
- The OOPSLA Wiki

OOPSLA Venues

- Awards
- Birds of a Feather (BoF)
- Demonstrations
- DesignFest
- Doctoral Symposium
- Dynamic Languages Symposium
- Essays
- Invited Speakers
- Lightning Talks
- Onward!
- Panels
- Posters
- Practitioner Reports
- Receptions and Special Events
- Research Papers
- Student Research Competition
- Tutorials
- Workshops

2003 and is the founding editor of The C++ Source. Chuck has been a regular lecturer at Software Development Conference since 1993, is author of C & C++ Code Capsules: A Practitioner's Guide, and co-author with Bruce Eckel of Thinking in C++, Volume 2: Practical Programming.

Related Onward! Papers

- A Commensalistic Software System
- Collaborative Diffusion: Programming Antiobjects

Related Panels

- The Convergence of XP and SCRUM

Related Practitioner Reports

- Evolving an Embedded Domain-Specific Language in Java

Related Research Papers

- Design Fragments Make Using Frameworks Easier
- JTL - The Java Tools Language

Related Tutorials

- T08: Ajax: Introduction and Architecture
- T18: Making the Most of Eclipse
- T22: Generate the Repetitive, Boring Code: How to Write Code Generators
- T38: Introduction to the Eclipse Modeling Framework
- T45: Storytelling with FIT
- T46: Software Architecture: Principles, Strategies, Qualities
- T47: Programming Mobile Devices: An Introduction
- T49: Model-Driven Development of Distributed Systems
- T50: Enterprise JavaBeans and the Java Persistence API
- T53: An Introduction to C++ Library Functionality in TR1 and Boost

Related Workshops

- W01: Library-Centric Software Design
- W02: Eclipse Technology eXchange
- W03: Building Software for Sensor Networks
- W07: Fifth "Killer Examples" for Design Patterns Workshop
- W09: Creating an Informative Workspace
- W12: NetBeans plug-in/RCP application development workshop
- W16: Ultra Large Scale Systems

program **registration** **submissions** **committee** **lodging** **portland**