ISSCENS

Intensive Summer School for Computing in Environmental Sciences

Sponsored by
The University of Virginia Alliance for Computational Science and Engineering (UVACSE)
and
The National Aeronautics and Space Administration (NASA)

A Program for Computational Education and Internship Training for Students in Environmental Sciences

Program Structure

WEEK 1: will begin Wednesday and will teach basic programming and software engineering with an emphasis on the particular interests of environmental science students.

WEEK 2: will introduce advanced visualization and compiled languages.

WEEK 3: will cover material for high-performance computing, including serial optimization, high-throughput computing, MPI, and OpenMP.

FOLLOWING 8 WEEKS: Once attendees have completed this Summer School they should be well prepared for internships with modeling groups at any NASA laboratory.

Those selected will begin 8-week internships at various NASA research centers nationwide immediately following the Summer School.

Program Logistics

2.5-Week Summer School:

- Dates of Summer School: **May 29-June 14, 2013**
- 20 students will be admitted
- Summer School will take place at The University of Virginia
- Housing will be provided
- Breakfast and lunch will be provided on weekdays, and a stipend for dinners and meals on weekends will be paid

8-Week NASA Internships:

- Dates of Internships: **June 17-August 8, 2013**
- 10 students will be selected from the Summer School participant pool
- Internships will take place at various NASA centers
- Interns will receive a stipend of $600 per week

Application Process

Submission of online application, unofficial undergraduate transcript, 2 letters of reference and a letter of intent.

Application Deadline

March 15, 2013

To learn more about the program, visit: http://www.uvacse.virginia.edu/isscens

ISSCENS Program Goals

- To teach basic software engineering concepts to environmental science students
- To introduce students to high-performance computing
- To enable students to work with advanced models used within environmental sciences
- To encourage durable and valuable connections among mentors and students

TENTATIVE SCHEDULE

	WEEK ONE	WEEK TWO	WEEK THREE
DAY 1 Monday	**Week 1 begins Wednesday. The first week will focus on Python.**	**Lecture**: Sorting and searching. Plotting using Matplotlib. More advanced visualization using ParaView. Introduction to Unix.	**Lecture**: Serial optimization for both Python and Fortran.
		Lab: Searching data. Line plots, contour plots, and surface plots. Volumetric rendering examples with ParaView.	**Lab**: Optimization of an intentionally badly-written program.
DAY 2 Tuesday		**Lecture**: Classes and object-oriented programming. Constructors and methods. Data hiding. Inheritance. Software design principles. More advanced Unix.	**Lecture**: High-throughput computing
		Lab: Writing a class containing topographic data. Application of the class to a 2D grid. Interaction of the class with NumPy arrays and Matplotlib.	**Lab**: Writing a bash script to submit multiple jobs, each rendering several frames of a ”movie,” to a queuing system.
DAY 3 Wednesday	**Lecture**: Introduction to programming. Fundamental data types including integers, floating-point numbers, and characters/strings. Pitfalls of working with floating-point numbers. Type conversions. Input and output to the console. Conditionals and looping.	**Lecture**: Introduction to Fortran 2003. Use of a compiler. Differences between compiled languages and interpreted languages. Data types and typing. Arrays and array operations. Input and output. Conditionals and loops.	**Lecture**: Beginning MPI. Collective communications.
	Lab: Programming simple expressions. Writing correct conditional expressions. Programming loops (while, for, and repeat).	**Lab**: Repeat the Day 1/Week 1 exercises with a compiler and different syntax. Reading and writing data as arrays.	**Lab**: Monte Carlo simulations.
DAY4 Thursday	**Lecture**: Functions, variable scope. Modules and namespaces. File input and output.	**Lecture**: Functions and subroutines. Modules. Defined types. Procedure and operator overloading. Make and make files. Simple bash scripting.	**Lecture**: More advance MPI. Point-to-point communications.
	Lab: Reading data into a program. Writing functions. Grouping functions into a module.	**Lab**: Writing simple functions and subroutines. Designing a programmer defined type within a module and developing related procedures. Writing a simple bash script to compile and run a program.	**Lab**: Solving Laplace’s equation
DAY 5 Friday	**Lecture**: Lists and list operators. Tuples and dictionaries. Introduction to NumPy. Arrays and array operators. Differences between arrays and lists.	**Lecture**: Inheritance and polymorphism. Introduction to Co-Array Fortran. Introduction to using a queuing system.	**Lecture**: OpenMP.
	Lab: Working with NumPy arrays. Array operations on data. Mathematical operations. Development of a short module to analyze climate data.	**Lab**: Conversion of the defined type from the previous day into a class. Submitting jobs to a queue.	**Lab**: Another way to solve Laplace’s equation.

To learn more about the program, visit:
http://www.uvacse.virginia.edu/isscens