An Introduction to Computation as Research Tool
CS 5014

Instructors

Ed Hall (edhall@virginia.edu)
Katherine Holcomb (kholcomb@virginia.edu)
Jacalyn Huband (jhuband@virginia.edu)

Small Hall, Room 112
UVACSE
Office Hours: upon request

Time/Location

Time: MWF 1:00 – 1:50pm
Location: MEC 213 / 214 / 215

Course Goals and Outcomes

Course Goals: Many graduate students come to UVA with little or no programming experience. At the same time their research often involves extensive use of computers, particularly for modeling and simulation. Without formal training they eventually learn to program, but often in an ad hoc, inefficient way. The result is a combination of reduced productivity and potentially lower quality research results. The goals of this 3-credit course are to enable students in the various disciplines to both understand the nature and limitations of computation and to become productive members of their research group's computational science projects.

The first 10 weeks of the course will provide a foundation in the core abstractions in computational thinking using a programming language chosen by students (from a finite set) to use for the course. The basic programming abstractions will be framed in pseudo-code with weekly, language-specific lecture and lab sessions to demonstrate how these abstractions are implemented in the particular programming language.

In the last 4 weeks of the course, the class will divide into separate sections to cover domain-specific applications of the programming concepts acquired in the first part of the course. The types of domain-specific sections offered will depend on departmental participation in the content development, and determine which programming languages are taught in the first half of the course.

The anticipated language choices are Matlab, Mathematica, Python, and R. Each section of the course will feature only two of the language choices. Students registered for one section may be moved to another section, depending on their choice of programming language.

The topics to be covered in the first 10 weeks are listed as follows; however, the order in which they are presented may vary by instructor.

Week	Topic
1	Course overview and Introduction to Computation, Variable assignment and input/output Choices for languages and last four weeks possible topics (Language choice decided by next Monday) Unix intro, logging into cluster frontend
2	Logical expressions, conditional statements.
3	Looping constructs, algorithm complexity.
4	Arrays/Lists and File I/O.
5	Functions, Variable scope, abstraction, and encapsulation,

	modules/packages.
6	More about functions
7	Multi-dimensional Arrays, Programmer-defined types and structures
8	Classes or similar (as appropriate to language)
9	Code Validation and debugging
10	Code Profiling and Timing; Using a managed computing resource
1-10	Unix operating system and shell scripting
11-14	Domain-specific applications
15	Last class, course wrap-up.

Applications or tracks within specific domains will be taught in the last 4 weeks. Actual tracks will be determined based on availability of instructors and student interest. Examples of previously taught tracks include:

Atomistic Simulation
Bioinformatics
Machine Learning
Natural Language Processing
Parallel Computing
Partial Differential Equations
Pre-processing of Statistical Data
Signal Filtering/Optimization
Software Design
Text Analysis

Course Outcomes: Following the course, the students will be able to:

- Think computationally, and be able to write/debug programs in a programming language appropriate to their field using a variety of programming abstractions.
- Use the Unix operating system and perform basic shell scripting.
- Run programs on a high-performance computing platform (e.g. the local Linux clusters)
- Perform computational research tasks relevant to their field.

Text

None

Additional Reading Materials

Programming language specific reference books will be suggested, and online resources will be made available.

Course Prerequisites

Graduate status or permission of the instructor. Not open for computer science graduate students.

Policies and Procedures

The grade will be based on homeworks from the first 8 weeks (~70%) and homeworks (or projects) from the domain-specific sections (~30%).