

The IDL Way: An Introduction to Computer Programming for Astronomers

Student Facilitator: James McBride
Sponsoring Professor: James Graham
Class Website: http://ugastro.berkeley.edu/~jmcbride/idl_class/index.html
Lectures: Monday 6:00 - 7:30 PM
Classroom: 544 Campbell Hall
Office Hours: TBA

Course Description :

“The IDL Way” is aimed primarily at astronomy majors, as the course focuses on learning to use IDL, the programming language used in each of the undergraduate astronomy laboratory courses. The course assumes no knowledge of computer programming, starting with an introduction to basic programming terms and techniques, before getting in to performing more complicated science related tasks. Primary instruction occurs in weekly lectures, with additional one on one instruction on the undergraduate astronomy lab computers during the rest of the week.

Course Objectives :

By the end of the semester, students will be able to write basic programs in IDL and will be comfortable working on a Unix/Linux operating system. Such skills will greatly ease the transition in to Astronomy 120, 121, or 122, which all use IDL for data acquisition, reduction, and analysis. The skills learned in this course will also be useful in research.

Required Texts :

No texts will be necessary for the course.

Additional Resources :

- <http://dfanning.com/documents/tips.com> - A good general reference site, with in depth explanations of common problems, such as what a “floating underflow” error means.
- http://idlastro.gfc.nasa.gov/idl_html_help/idl_alph.html - A list of standard IDL routines with information on calling sequence as well as examples.
- <http://idlastro.gsfc.gov> - A collection of astronomy related procedures.
- <http://www.idlwave.org> - A resource with with information on using Emacs to write IDL code.

Grading : 2 units, P/NP

Homework :

There will be weekly or biweekly assignments due by midnight on Mondays. Programs will not be submitted, but saved in to an ay98 directory on the ugaastro server, where they will be viewed and graded for achievement of initial objectives and completeness.

Participation and Attendance	20%
Homework	60%
Final Project	20%

Schedule :

This is a rough guide of what will be covered during the semester, though it is subject to change according to class progress and the availability of guest speakers from the Astronomy department.

Week 1: Getting to Know IDL and Unix

- Working from the command line
- Variables and data types
- Basic math and logical operators
- Customizing your IDL session using .idlenv
- Loops and conditionals
- Writing a main-level IDL program

Week 2: Digging in to IDL

- Functions and procedures
- Compiling programs
- Positional parameters
- Keyword parameters
- Scope of procedures and functions
- Intro to vectors

Week 3: IDL as a Vector Based Language (part one)

- Vectors (1-D arrays)
- Accessing array elements
- Loops or arrays?
- Generating arrays
- The endless utility of where()
- Other array operators

Week 4: IDL as a Vector Based Language (part two)

- More on vectors
- Wrappers
- Cool 1-D tricks
- histogram(), I knew ye not!

Week 5: Fun with Strings

- String variables
- String operations
- Formatting
- Printing to the screen
- ASCII I/O
- Interacting with Unix from IDL

Week 6: FITS Files

- Dealing with Astronomy's ubiquitous storage format
- 2D arrays
- Matrix operations
- Fanning a vector
- Array concatenation
- Avoiding the first position asterisk

Week 7: Data Structures

- Intro to data structures
- Using MPFIT

Week 8: Pseudo Object Oriented Programming (POOP)

- Efficient use of data structures
- Managing large projects

Week 9: Approaching Large Programming Tasks

- Debugging techniques
- General coding philosophy

Week 10: Implementing POOP (part one)

- The advantage of modular code
- Fractional shifting
- The power of "save"

Week 11: Implementing POOP (part two)

- Inheritance
- Code recycling

Week 12: Keeping up to Date With CVS

- Keeping track of file changes
- Collaboration with others

Week 13: Using Pointers

- Creating pointers
- Pointer referencing and dereferencing

Week 14: Designing Effective Figures

- Plotting and displaying
- Controlling output

Week 15: Course Wrap Up

- Review of most important topics covered
- What to take from the course
- Tips for taking lab