Computing for Imaging Science (SIMG-726)

Introduction
- Course Information
  - Course Title
  - Course Objective
  - Prerequisites/Expectations
  - Course Grading
  - Course Text
  - Computing Resources

Course Information
- Course Title
  - Computing for Imaging Science
- Course Number
  - SIMG 726
- Meeting Times
  - Monday and Wednesdays 11:00-1:00 PM
- Meeting Locations
  - Mondays (Classroom), Wednesdays (Lab)

Contact Information
- Instructor
  - Rolando Raqueño
  - E-mail (rolo@cis.rit.edu)
  - Phone (475-6907)
  - Office (76-3108)
- Office Hours
  - Monday and Wednesdays (2:30-3:30 PM)
  - Other times by appointment

Course Objective
- Basic proficiency in UNIX operating system for problem solving in imaging
- Learning elementary programming constructs in IDL and refinement of programming skills
- Useful data manipulation techniques
- Establish software development, validation, and documentation practices

UNIX Proficiency
- Basic UNIX Environment Proficiency
  - X-Windowing Environment (networked)
  - The universally available editor 'vi'
  - Navigation and File Manipulation
  - Basic Computing Data Type and Hardware differences
Why UNIX?

- Why the UNIX Environment?
  - Very mature technology
  - Maintains a large base of high quality "copy-left" programs
  - LINUX is "free" and can run on a 386 PC
  - Remote computing is a natural part of UNIX
  - Very close control of the computer
  - Minimalist approach to computing

Shortcuts and Timesavers

- Learn shortcuts and time savers when solving programming related problems (when not to do traditional program)
  - a.k.a "Stupid UNIX Tricks"
  - Shell Scripts
  - AWK & SED
  - Makefiles
- Application in an imaging context

Compiling Traditional Programs in UNIX

- Legacy FORTRAN and C code requires some knowledge of how to compile these programs
- Tools to aid in compiling (makefiles)
- Common pitfalls
- Setup tips for input and output data files

Imaging Concepts

- Application in an imaging context
  - Bi-level
  - Monochrome
  - Color
  - Multispectral
  - Hyperspectral

IDL Environment

- IDL Basics, Syntax, and Semantics
  - Your responsibility get started
- IDL Graphical User Interface (GUI) development
- How IDL interacts with the UNIX environment.

This Course is/is not

- This course is not...
  - an algorithm course in image processing or computational theory
  - a programming course in C/C++
- This course will...
  - give you a means to prototype and test image processing algorithms
  - provide a foundation for prototyping algorithms that can be translated into traditional programming languages
Will/will not make you...
- This course is not...
  - Designed to make you a software engineer
- This course, however, will...
  - Give you the skills to communicate with software engineers about your algorithm
  - Provide them necessary concepts, specifications and test cases, i.e.,

  **The working prototype**

Documentation, Documentation
- This course will...
  - Teach you about documentation in the form of source code control.

  **RCS or SCCS**
- Inherently lacking in other operating systems (OS)

Prerequisites/Expectations
- Experience in a high level programming language (e.g. C, FORTRAN, BASIC, PASCAL, ADA, etc.) in the last 20+ years.
- Ability to make lots of mistakes (and remember them)
- Ability to deal with frustration, and know when to quit
- Good Typing Skills a plus
- Good Technical Writing Skills

Key Milestones in the Progression of Programming Experience
- Programming in traditional languages
  - Beginning (Monolithic programs)
    - single letter variables
    - Loops and maybe an occasional GOTO
  - Intermediate (Use of subroutines and functions)
    - Favor Local variables rather than Global Variables
    - Understand parameter passing schemes
  - Advanced (Structures and abstract data types)
    - More meaningful variable names

Milestones in the Progression of Programming Experience
- Programming in traditional languages
  - Learning that hand-optimizing code at the development is simply not worth the time
  - Learning to separate the algorithm from the bells and whistles (i.e., input and output code)
  - Creating a library of "tools"
  - Use of revision control utilities along with README's and formal documentation.

Milestones in the Progression of Programming Experience
- Programming in non-traditional languages
  - Thinking about the problems in terms of matrix and vector operations
  - Avoiding loops and conditionals
  - Thinking about the "Non-expert" by designing the Graphical User Interface (GUI)
  - Realizing that there is no single software tool that solves everything.
Course Grading
• Final IDL GUI Project 35%
  –Presentation and Report
• Mid-term Exam 35%
• Programming Assignment(s) 20%
  –Will be questions in Mid-term
• Quizzes (every Monday) 10%

Course Text
• Fanning’s Book - IDL Programming (Required)
• Research Systems Incorporated (RSI)
  –Online Manuals
    • Using IDL
    • Building IDL Applications
    • Reference Manuals
• UNIX in a Nutshell (O’Reilly & Associates)
• The Cuckoo’s Egg (Cliff Stoll)

Other References
• Text
  –Learning the UNIX Operating System
  –UNIX Power Tools
• On-line
  –News groups
    • comp.lang.idl-pvwave
  –Web sites
    • www.rsinc.com

Other References
• Notes
  –These Powerpoint slides will be available on-line for your reference.
  –These Notes will be available the evening before class.
  –If not, last years notes will be available
  –Send me mail and I will reply with the location of the site.

Computing Resources
• RIT’s ISC
  –Provides general computing resources to the RIT community (e.g. internet connectivity, LIMITED dial in line services)
• CIS computing facilities
  – Imaging specific capabilities (IDL, etc.)
  – Undergraduate and Graduate DIP Lab
  – The “DEAL” with RSI

Syllabus until Holiday Break
• Class #0 - Today
  –This Lecture
• Assignment #1
  –Get your account setup
  –Send e-mail to rolo@cis.rit.edu with personal info (Name, Address, Phone Number, Undergraduate/Graduate Majors, Computer Experience)
  –Go over the on-line tutorial vitutor (CIS)
  • -rvrpci/pub/vitutor/vitutor
**Syllabus**

- **Class #1 - December 1**
  - Tour of UNIX, X-Windows
  - Introduction vi
  - Getting started and Tour of IDL
    - Working with Data
    - Plotting
    - Hardcopy output from IDL

- **Project Assignment #1 (Due December 15)**
  - Implementing Statistics Functions in IDL

- **Class #2 - December 6**

- **Class #3 - December 8**
  - The PBMPLUS Utilities
  - IDL and PBMPLUS
  - Basic IDL Image Manipulations

- **Class #4 - December 13**
  - Reading/Writing Data in IDL
  - Image Formats
  - Final Project Examples

- **Quiz #2**
  - Topics
    - man, ls, cp, mv, ped

- **Class #5 - December 15**
  - IDL debugging & GUI Introduction

- **Break Assignments**
  - Read the Cuckoo's Egg (Quiz Material)
  - Mockup Drawing of GUI Final Project

- **Due Today**
  - Project Assignment #1
    - Abstract and high level overview of final project.

**Summary**

- Course Information and Objectives
- Prerequisites/Expectations and Experience
- Course Grading
- Course Text and Resources
- Syllabus until Break
Keep in mind

- There is more clout to say that you can build a car rather than just saying that you know how to drive one.
- Our goal is to get you to understand how a car works as well as drive it.