

ROCHESTER INSTITUTE OF TECHNOLOGY  
Rochester, New York

COLLEGE of SCIENCE  
Department of IMAGING SCIENCE

REVISED COURSE: 1051-211

- 1.0 TITLE: COMPUTING FOR IMAGING SCIENCE  
DATE: 27 February 2003  
CREDIT HOURS: 4  
PREREQUISITE(S): 1016-204 (Algebra & Trigonometry)  
COREQUISITE(S): none  
COURSE PROPOSED BY: Carl Salvaggio

2.0 COURSE INFORMATION:

	Contact Hours	Maximum Students / Section
Classroom	4	12
Lab	n/a	n/a
Studio	n/a	n/a
Other	n/a	n/a

QUARTER(S) OFFERED:

Fall     Winter     Spring     Summer

STUDENTS REQUIRED TO TAKE THIS COURSE:

Imaging Science, 2nd year

STUDENTS WHO MIGHT ELECT TO TAKE THE COURSE:

Environmental Science, Applied Mathematics, Physics, Computer Engineering

3.0 GOALS OF THE COURSE:

This course is intended to provide the student with the necessary problem solving and computer programming skills to be able to utilize the IDL programming language to arrive at solutions to imaging-related problems that they will encounter during their academic career at the Center and during their professional career in the field of imaging science.

4.0 COURSE DESCRIPTION:

**1051-211**

**Computing for Imaging Science**

This course will introduce the student to the IDL environment as a data visualization tool and a programming language. The student will learn the various capabilities of the package and how they can rapidly prototype solutions to various science and engineering problems. As these solutions are developed, fundamental concepts of programming and data structures will be

introduced. Programming assignments will include fundamental imaging related problems and will work with scalar, vector and array processes. This course will emphasize the need for concrete problem definition, problem decomposition into smaller sub-problems, implementation/testing, and presentation/documentation of the algorithms and results. (1016-204) Class 4, Credit 4 (F)

## 5.0 POSSIBLE RESOURCES:

- 5.1 Gumley, Liam E., *Practical IDL Programming*, Morgan Kaufmann Publishers, New York, ISBN: 1-55860-700-5
- 5.2 Student version of IDL for Windows

## 6.0 TOPICS:

- 6.1 Overview of the UNIX operating system
  - 6.1.1 Structure of the UNIX operating system
  - 6.1.2 Editing text files under the UNIX operating system
  - 6.1.3 Fundamental commands; ls, man, rm, cp, mv, alias, xhost, setenv
  - 6.1.4 Concepts of piping and redirection
- 6.2 Scientific problem solving
  - 6.2.1 Input
  - 6.2.2 Processing
  - 6.2.3 Output
- 6.3 Computer science concepts
  - 6.3.1 Alternative numbering systems; binary, octal, hexadecimal
- 6.4 Programming concepts
  - 6.4.1 Variables
  - 6.4.2 Data types
  - 6.4.3 Data conversion functions
  - 6.4.4 Arrays
    - 6.4.4.1 Index generation functions
    - 6.4.4.2 Array creation functions
    - 6.4.4.3 Array addressing
    - 6.4.4.4 Determining dimensionality; N\_ELEMENTS() and SIZE()
    - 6.4.4.5 Arithmetic operators with arrays and vectors
    - 6.4.4.6 Array functions (descriptive statistics)
      - 6.4.4.6.1 MIN() / MAX()
      - 6.4.4.6.2 MOMENT()
      - 6.4.4.6.3 MEAN() / STDEV() / VARIANCE()
    - 6.4.4.7 Finding elements within arrays; WHERE()
    - 6.4.4.8 Reordering array elements
      - 6.4.4.8.1 REVERSE()
      - 6.4.4.8.2 ROTATE()
      - 6.4.4.8.3 TRANSPOSE()
      - 6.4.4.8.4 SHIFT()
      - 6.4.4.8.5 SORT()
      - 6.4.4.8.6 UNIQ()

- 6.4.4.9 Resizing arrays; CONGRID()
- 6.4.4.10 Matrix mathematical operators
- 6.3.5 Program structure
  - 6.3.5.1 Input/output statements
  - 6.3.5.2 Procedures and functions
  - 6.3.5.3 Argument passing
    - 6.3.5.3.1 Parameters
    - 6.3.5.3.2 Keywords
    - 6.3.5.3.3 Boolean keywords
    - 6.3.5.3.4 COMMON blocks
  - 6.3.5.4 RETURN procedure
- 6.3.5 Data structures
- 6.3.6 String manipulation
  - 6.3.6.1 Differences between strings and arrays
  - 6.3.6.2 String manipulation routines
- 6.3.7 IF () THEN ELSE () statements
- 6.3.8 CASE / SWITCH control statements
- 6.3.9 FOR / WHILE / REPEAT looping constructs
- 6.4 GUI/event-driven programming
  - 6.4.1 Difference between procedural and event-driven programming
  - 6.4.2 Concept of widgets
  - 6.4.3 GUI construction
    - 6.4.3.1 Bases
    - 6.4.3.2 Labels
    - 6.4.3.3 Textboxes
    - 6.4.3.4 Buttons
      - 6.4.3.4.1 Action
      - 6.4.3.4.2 Exclusive / radio buttons
      - 6.4.3.4.3 Non-exclusive / checkboxes
    - 6.4.3.5 Widget realization
    - 6.4.3.6 WIDGET\_CONTROL()
  - 6.4.4 Event handler; XMANAGER and \_event procedures
- 6.5 Data visualization / image handling routines
  - 6.5.1 WINDOW procedure
  - 6.5.2 Two- and three-dimensional plotting
  - 6.5.3 Shaded-surface visualization
  - 6.5.4 Image file formats
  - 6.5.5 Image handling routines
    - 6.5.5.1 QUERY\_\* routines
    - 6.5.5.2 READ\_\* routines
    - 6.5.5.3 WRITE\_\* routines
    - 6.5.5.4 TV and TVSCL for image display
      - 6.5.5.4.1 Displaying multiple images in a single window
  - 6.5.6 Zooming; ZOOM procedure
  - 6.5.7 Panning; SLIDE\_IMAGE procedure
  - 6.5.8 Getting information beneath a cursor; CURSOR

- 6.6 File input/output
    - 6.6.1 ASCII vs. binary data files
    - 6.6.2 Opening files; OPENR, OPENW
    - 6.6.3 Reading ASCII data from files; READ vs. READF procedures
    - 6.6.4 Writing ASCII data to files; PRINT vs. PRINTF procedures
    - 6.6.5 Reading binary data from files; READU procedure
    - 6.6.6 Writing binary data to files; WRITEU procedure
    - 6.6.7 Template-based data input from file
      - 6.6.7.1 READ\_ASCII()
      - 6.6.7.2 ASCII\_TEMPLATE()
  - 6.7 IDL direct graphics concepts
    - 6.7.1 Indexed vs. RGB color
      - 6.7.1.1 DEVICE, DECOMPOSED=0|1
    - 6.7.2 Working with color tables
      - 6.7.2.1 Loading color tables; LOADCT and XLOADCT procedures
      - 6.7.2.2 Modifying color tables; XPALETTE
      - 6.7.2.3 Specifying R,G,B colors directly
      - 6.7.2.4 COLOR keyword
- 7.0 INTENDED LEARNING OUTCOMES AND ASSOCIATED ASSESSMENT METHODS OF THOSE OUTCOMES:
- 7.1 Ability to use the IDL environment as an interactive problem solving tool (HOMEWORK ASSIGNMENTS)
  - 7.2 Ability to use IDL as a programming language to solve scientific problems (HOMEWORK ASSIGNMENTS)
  - 7.3 Ability to produce user-friendly, robust, graphical-user interface-driven codes for scientific problem solving (HOMEWORK ASSIGNMENTS)
  - 7.4 Gain an understanding of computer/imaging science concepts
    - 7.4.1 Binary, octal and hexadecimal numbering systems (EXAMS / QUIZZES)
    - 7.4.2 Programming control statements (HOMEWORK ASSIGNMENTS / EXAMS / QUIZZES)
    - 7.4.3 Data structures and arrays (HOMEWORK ASSIGNMENTS / EXAMS / QUIZZES)
    - 7.4.4 Image file formats (HOMEWORK ASSIGNMENTS)
    - 7.4.5 Basic image processing concepts (HOMEWORK ASSIGNMENTS)
    - 7.4.6 Vector and array-based processing (HOMEWORK ASSIGNMENTS)
- 8.0 PROGRAM OR GENERAL EDUCATION GOALS SUPPORTED BY THIS COURSE:
- 8.1 The student will have the programming and visualization tools necessary to solve problems in subsequent courses within the department including optics, color science, vision, digital image processing and all imaging-related mathematics

- 8.2 The student will have proficiency in a general-purpose programming language that is currently very popular in industry
- 8.3 The student will have the problem-decomposition skills that scientists need to solve complex problems

9.0 OTHER RELEVANT INFORMATION:

- 9.1 Course needs to be conducted in the Undergraduate DIP lab, 76-2155, so students have access to computers and IDL software for classroom exercises
- 9.2 High-resolution projector (1280x1024) for classroom instruction

10.0 SUPPLEMENTAL INFORMATION:

none