Introduction

Information Science 1052 is a course on human/computer interaction and interactive programming. More precisely, it is a course in which you will learn how to program consistent with the needs of applications that require significant amounts of human interaction. Video games, text, graphic, and software editors, page layout systems, database interfaces are all examples of interactive systems, i.e. they require a high degree of human interaction.

The development of interactive systems is taking place more and more on the C language. It is occurring on both microcomputer and workstation platforms. Interactive programs are being designed to run locally on these workstations and pc’s taking advantage of the ability to control high resolution bit mapped screens, either through local programming language control or through window management shells like X windows. For these reasons, this course on interactive programming assumes familiarity with the C language and encourages work on workstation and PC platforms. Because of the extensive help provided by micro based compilers, this course will focus on the use of turbo C.

This course requires IS 0020 as a prerequisite. Students who feel they can manage the demands of the course without having taken IS 0020 should discuss their plans with the instructor. Students who have not had IS 0020 or its equivalent should be aware that they may not take IS 0020 once they have completed IS 1052. Students for whom this is a problem should see Professor Ida Flynn.

Background

As we get into the course, you should be aware that our computer environment, particularly as it applies to interactive programming, is changing. Think about the following points:

- The programs written for 24-hour bank machines may be the best examples of interactive programs. Almost anyone can use them without reference to additional materials. The program is self explanatory. The systems seldom fail -- they may run out of money or lose the communications link to the mainframe, but the programs themselves are very robust.
- At the technology level, bit-mapped graphics screens, laser printers, and pointing devices such as the mouse have been introduced which allow a wide variety of entities to be constructed and reproduced with the computer as intermediary. Thus, we are no longer concerned simply with processing numbers and text, but with the text, numbers, graphics, images and any number of other definable entities. The degree of precision you control is amazing. It used to be that you could put any one of 70 or so characters at any one of 1920 positions on a screen or at any one of 8712 positions on a printed page. You can now put dots that can be combined to make just about any character or symbol at any one of 1,000,000 places on the screen or at any of 8,415,000 places on a sheet of paper.
This means you have a lot more flexibility. It also means you are faced with more decisions, and thus more coding and responsibility.

- At the software level, we are less and less concerned with the processing algorithms and more and more concerned with the user interface. Take for example the fact that in text processing we have evolved from line editors and printers, through screen editors, to page layout and document processing software and laser printers. You must not only know the sorting, searching, pattern matching algorithms, you must also know what screen colors will be best, how to form icons, what command words the user will understand, what likely mistakes the user will make, etc.

- At the standards level, work on window systems such as the X Window System and Microsoft Windows are making it essential for interactive system programmers to know the ins and outs of these very complex packages. Every day, a new standard is being introduced which will impact how you program. You will need to learn and use the standards in your programming.

- Conceptually, there is ongoing pressure to incorporate more and more task expertise in interactive systems. The systems need to be smarter about the domains in which they intend to assist the user. This is perhaps best illustrated by word processing systems where great amounts of energy are being expended to assist the user in creating good documents.
IS 1052 is an upper level course. This means students have responsibility to be proactive in their learning. The instructor’s role is less directive and more one of stimulating and guiding learning. If you have never experienced a course like this -- if your instructor always spoon fed the material to you -- you may be somewhat shocked by this course. You may have heard the saying that the way you really learn something is to teach it to someone else. To do that you must really come to grips with the material. If you think back to the things you remember from other courses you have taken, you will probably find that the things you remember best are the things you had to work hard to learn. It is simply true that the things you remember are the things you work hard to learn. For that reason, the instructor and students share the responsibility to make the course “work”. Students who feel they are not ready to begin taking responsibility for their own learning are encouraged to consider not taking this course. Much material can be learned by rote memorization. As indicated above, a different kind of learning takes place when students engage in the process actively. To this end, IS 1052 is structured to provide a variety of different kinds of learning experiences. Students are expected to be learning and reading about interactive systems on their own and to come to class prepared to ask questions and discuss topics pertaining to programming interactive systems.

Students will also be expected to learn and make use of the appropriate operating systems and application languages that are used in conjunction with a main language like C to make an interactive system work. While these additional languages, systems, and standards will be introduced in class, primary responsibility for learning the systems will reside with the students outside of class. It is true that a course covering everything you will be exposed to in this class could cover three or four terms. The instructor recognizes this, and understands that one of the things you should learn in this course is how much you don’t know about certain topics.

Workload

This course is an upper level course in information science. Students in this course are either majors in the department or serious students of programming and information system design. This means that you bring a genuine interest in learning to the course and a maturity in your approach. This means that you are going to work, like the instructor, three to four hours out of class for every hour that you are in class. That means that every week, beginning today, you will be allocating 9 - 15 hours of work to IS 1052.

For example, you may be introduced to the X Window System in this course if time permits. You will learn only a small portion of the languages -- but enough to do an assignment. You should also learn that if you are going to work in information science you should set as one of your long term goals the goal of learning about windowing systems.
Readings

The instructor will review the topics assigned for each class. The instructor will not make the classes extended tutorials reviewing the required readings. It is your responsibility to come to class prepared to discuss the concepts presented in the readings, not to come ready to learn with a clear head! The reading assignments for each class are to be read and understood prior to coming to class. They are not notes to review after class.

Assignments

The programming assignments will be as real as is possible within the constraints of an undergraduate course. In the real world, software systems are extremely complex and consist of thousands to hundreds of thousands of lines of code. You should anticipate many hours of work on some of your assignments to make them work. If you are assigned real problems, the real code you develop will be extensive.

To make it possible for you to work out some of the more complex assignments, you may be assigned to work on them on teams. This means you will need to coordinate your efforts. Grading will reflect both your work together and your individual effort. Be aware that you will be held accountable for your work in the group as well as the final product of that effort. Be smart about group work. Contribute in a timely fashion and be cooperative about meetings. The instructor will help the process by assigning group leaders. When your group leader gives you an assignment, it is just the same as if the instructor had given it. Your group leader will help the instructor to assess your contribution, so don’t think you can take a vacation and then take credit for the work of your teammates.
Goals, Requirements, and Grading

Course Goals for IS 1052

The goals of the course are as follows:

- To learn the basic characteristic of an interactive environment.
- To learn the basic requirements of programming an interactive system.
- To apply the C language to programming interactive environments.
- To learn some simple algorithms for error checking, editing, menu control, window control, etc.
- To understand the team nature of the development of real interactive systems and the ways in which modules may be interfaced.

Within these broad goals, students will define and submit to the instructor specific objectives for their own learning during the course.

Course Requirements

This course is a skills course. To be successful, you must exercise a concerted effort in the course. For this reason, the grading scheme is constructed to encourage you to keep on the schedule. It is as follows.

- Assignments submitted before the due date will be graded on a basis of 110% of the indicated value. Thus for a 10 point project, the maximum award will be 11 points.
- Assignments submitted after the date they are due will be graded on the basis of a 10% maximum point award reduction per week. Thus an assignment worth 10 points and due in week 3 will be graded on the basis of a maximum award of 9 points if received in week 4, 8 points in week 5, etc.
- The instructor will adjust the schedule as needed if the lectures fall off the announced schedule to allow time for you to work on the various assignments after the related issues have been covered in class.

Students who find themselves in situations where they will not be able to work on schedule for business or personal reasons should petition the instructor in advance and in writing providing a reason for the special schedule and a copy of the schedule. This schedule will become binding on them with the same restrictions as the schedule for the class.

Presentation Assignments

Students will be required to make two kinds of class presentations.

- Every student will be assigned one or two articles in the Laurel book and will be required to provide a five minute overview of the article and to lead a brief discussion of the article. In preparation for the assignment the student should prepare and hand in a three to four page paper that explores the topic of the article in some additional depth. This can be
done by reading one or two other articles on the same topic or by the same author. (0-15 points)

- Each week developments in the newsgroup netnews.comp.human.factors will be reviewed. Students should be prepared each week to demonstrate their understanding of the current discussion. Over the term, the participation in these discussions will be used to develop a participation grade for the course. (0-10 points)

**Programming Assignments**

- Complete four mini programming tasks that will be assigned during the term. The source code for each assignment will be submitted to the instructor electronically. (0-10 points/assignment) The mini modules will include:
  - A menu function that displays and returns the menu choice. (0-10 points)
  - A window function that opens a window saving the background and that allows a selection of display characteristics for the window. (0-10 points)
  - A context sensitive help function that uses an external text file for the appropriate information. (0-10 points)
  - A line input function that allows simple editing, window sizing and buffer sizing, and optional filtering -- e.g. filename constraints. (0-10 points)

- Write a hypertext/hypermedia system. (0-40 points) The system will include:
  - An editor for entering text, anchors and links
  - A browser for examining nodes and links
  - A processor to index nodes or combine nodes to form a document

  (Optional) Write a paint program using the X Window System (0-25 points).

**Exams**

There will be both a midterm and a final exam. Each exam will consist of two parts. The first part will consist of multiple choice, true/false, and short fill-in questions. The second part will consist of a programming problem to be resolved in approximately 20 lines of code.

- The mid term will be worth 20 points
- The final will be worth 25 points.

The instructor reserves the right to replace scheduled exams with pop quizzes if he feels students are not keeping up with the readings. Generally, quizzes will be for 5 points and will replace a portion of the midterm or final.

**Grading**

Based on actual points earned out of 150, grades will be awarded as follows:

A 135-150 points
B 100 - 134
C 75 - 100
F 0 - 75

Failure to submit a requirement or take an exam will result in a C as the highest awarded grade.
Required and Supplementary Materials

Required Materials

There is one required text for the course:


In addition, students will be provided a number of articles to read. As the course begins, the following articles are being considered (They have been put on reserve at the SLIS library):


Students are encouraged to look at current journals to find and review other articles that pertain to the course. Articles that are of particular interest should be brought to the attention of the instructor.

Supplementary Materials

Should you want a basic C text, you might want to consider.


The Turbo C reference is only one of a variety of different books in the Pitt bookstore. Pick up one that reads well to you. One that helps you better understand C.

Students working in ULTRIX may want to purchase a guide to Unix. There are 100’s of different guides to Unix. Pick up any one that helps you at your level.
# Course Schedule and Lecture Topics

## Course Schedule

Classes will meet from 6:00 to 8:50pm in Room 407 LIS building on Tuesday of each week.

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Dates</th>
<th>Topics/Assignments/Exams</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/7</td>
<td>Review of Syllabus; Intro to Interactive Programming</td>
</tr>
<tr>
<td>2</td>
<td>1/12;14</td>
<td>Input Devices and Event Loops</td>
</tr>
<tr>
<td>3</td>
<td>1/19;21</td>
<td>Command Parsing</td>
</tr>
<tr>
<td>4</td>
<td>1/26;28</td>
<td>Menus</td>
</tr>
<tr>
<td>5</td>
<td>2/2;4</td>
<td>Data Input Window Function Assignment</td>
</tr>
<tr>
<td>6</td>
<td>2/9;11</td>
<td>Data Display Menu Function Assignment</td>
</tr>
<tr>
<td>7</td>
<td>2/23;25</td>
<td>Review Midterm</td>
</tr>
<tr>
<td>8</td>
<td>3/2;4</td>
<td>Mouse Event Loops Help Function Assignment</td>
</tr>
<tr>
<td>9</td>
<td>3/9;11</td>
<td>Spring Break!!!</td>
</tr>
<tr>
<td>10</td>
<td>3/16;18</td>
<td>Screen Display &amp; Color Considerations</td>
</tr>
<tr>
<td>11</td>
<td>3/23;25</td>
<td>Error Control Editor Function Assignment</td>
</tr>
<tr>
<td>12</td>
<td>3/30;4/1</td>
<td>Windowing Systems</td>
</tr>
<tr>
<td>13</td>
<td>4/6;8</td>
<td>Direct Manipulation and Virtual Reality</td>
</tr>
<tr>
<td>14</td>
<td>4/13;14</td>
<td>Setup/Ini Files; Special Groups</td>
</tr>
<tr>
<td></td>
<td>4/20;22</td>
<td>Logging User Activities/Review Hypertext System</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Week of 4/26 Optional Window Assignment</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Final Exam -- Date/time to be determined</td>
</tr>
</tbody>
</table>

## Lecture Topics

Given the complexity of the issues addressed in this course, it may be the case that we will not be able to cover all the topics in the course. If we fall off this schedule the instructor will notify you of this and make any adjustments to the due dates of requirements.

One topic is listed for each week of the course. In general, the first lecture each week will be devoted to the readings for the week and the theory and principles of interactive programming. The second lecture each week will be devoted to related coding techniques.
Lecture Topic Outlines

Lecture 1: What is Interactive Programming?

Topics

- Course Overview and Requirements
  - Why use the C language
  - Review of the course syllabus and requirements
  - Discussion of course requirements
- Introduction of the class, instructor, and graduate student assistant
  - Assignment to groups (if appropriate)
- Description of the PC programming environment
  - TurboC debugger
  - TurboC Header Files: dos.h, conio.h, graphics.h, etc.
- Introduction to Interactive Systems
  - Interactive System Design
  - Principles of User Interface Construction
    - Metaphors
    - Direct Manipulation
    - Consistency
    - Flexibility
  - The Role of Toolkits

Readings

TurboC Manual -- Sections on Editing, Error Messages, and Debugging
The Art of Human-Computer Interface Design, Laurel, xi-xvi
The Art of Human-Computer Interface Design, Laurel, 91-94
The Art of Human-Computer Interface Design, Laurel, 187-190
The Art of Human-Computer Interface Design, Mountford, 247-250
The Art of Human-Computer Interface Design, Laurel, 345-346
The Art of Human-Computer Interface Design, Laurel, 481-482

Lecture 2: Input Devices and Keyboard Event Loops

Readings

Cotterman, William and Kumar, Kuldeep, User Cube: A Taxonomy of End
The Art of Human-Computer Interface Design, Rheingold, 5-10
The Art of Human-Computer Interface Design, Mountford, 17-30
The Art of Human-Computer Interface Design, Vertelney, 45-56
The Art of Human-Computer Interface Design, Wagner, 79-84
The Art of Human-Computer Interface Design, Gomoll, 85-90

**Topics**

- User Input to Define System Needs
  - Needs Analysis
  - Prototyping
  - Surveys and Questionnaires
  - Interviews: Structured and Unstructured
- User Input in the Program
  - Types of User Input
  - Input Devices
    - Physical Devices
    - Simulated Devices
  - Options in Control Input
  - Options in Handling Data Input
  - Providing Activity Feedback to the User

---

**Lecture 3: Command Parsing and Command-Based Systems**

**Readings**

The Art of Human-Computer Interface Design, Erickson, 11-16
The Art of Human-Computer Interface Design, Norman, 191-208
The Art of Human-Computer Interface Design, Shneiderman, 209-220

**Topics**

- Command-Driven Systems
  - Unix as an Example
  - Options for Commands
  - Overall Structure of a Command-Driven System
- Command Options
  - Command Undo
  - User-Defined Commands
  - Programming Command Input
    - Breaking Commands into Word Lists
    - Matching Words to Command Parts
Lecture 4: Menus

Readings

The Art of Human-Computer Interface Design, Nicol, 113-122
The Art of Human-Computer Interface Design, Kay, 191-208
The Art of Human-Computer Interface Design, Gervich, 131-134

Topics

• Examples of Menu-Driven Systems
• Programming Text Menus
  • Principles of Text Menus
  • Alternate Forms
  • Implementing Text Menus
  • A General Menu Approach

Lecture 5: Data Input

Readings

The Art of Human-Computer Interface Design, Clark, 95-102
The Art of Human-Computer Interface Design, Crawford, 103-112
The Art of Human-Computer Interface Design, Chen, 299-308
The Art of Human-Computer Interface Design, Kurtenbach, 309-318
The Art of Human-Computer Interface Design, Mountford, 319-334

Topics

• Different Forms of Data Input
• Controlling Data Input from the Program
• Techniques of Text Input
• Providing Default Values
• Techniques of Numeric Input
  • Getting an Integer Value
  • Getting a Real Value
  • Getting a Numeric Value from an Expression
• Managing "Fill in the Blank" Input
Lecture 6: Data Display

Readings

The Art of Human-Computer Interface Design, Vertelney 161-170
The Art of Human-Computer Interface Design, Grudin, 171-186
The Art of Human-Computer Interface Design, Baecker 251-268
The Art of Human-Computer Interface Design, Schmandt, 335-344

Topics

• General Principles
  • Planning
  • Readability
  • Meeting Expectations and Needs
  • Presenting Information
• Methods of Organizing Output
  • Major Components of the Screen
  • Title Line
  • Command or Status Line
  • Error or Warning Messages
  • The Screen Body
  • Considerations for Different Types of Screens
  • Menu
  • Data Entry Screens
  • Question and Answer Screen
  • Inquiry Screen
  • Information Screen
  • Control Screen
  • Highlighting Critical Parts
  • Presenting Tabular Summaries
  • Presenting Graphical Summaries

Lecture 7: Help Systems

Readings

The Art of Human-Computer Interface Design, Hakanson, 123-130
The Art of Human-Computer Interface Design, Swigart, 135-142
The Art of Human-Computer Interface Design, Sellen, 143-154
The Art of Human-Computer Interface Design, Cypher, 155-160

Topics

• Help Systems
  • TurboC as an Example
• Unix Man
• Make Help Work
  • Options for Help Systems
  • Pitfalls

**Lecture 8: Mouse Event Loops**

*Readings*


The Art of Human-Computer Interface Design, Brennan, 393-404

The Art of Human-Computer Interface Design, Buxton, 405-416

The Art of Human-Computer Interface Design, Blake, 289-298

*Topics*

• The Mouse Interrupt
• Checking for Mouse Software
• Checking for Mouse Hardware
• Simple Mouse Functions
  • Reset and button number
  • Cursor on/off
  • Cursor location and button status
  • Min and max x and y cursor locations
• A mouse event loop
  • Loop nesting
  • Keyboard/mouse loops
• Event queues

**Lecture 9: Screen Display and Color Considerations**

*Readings*

The Art of Human-Computer Interface Design, Salomon, 269-278

The Art of Human-Computer Interface Design, Kreuger, 417-422

*Topics*

• Interactive Screens
  • Data Inquiry
  • The Screen as a Dynamic Medium
• Displaying Information Properly
• Using Windows
Lecture 10: Error Control

Readings

The Art of Human-Computer Interface Design, Nelson, 235-245
The Art of Human-Computer Interface Design, Leary, 229-234
The Art of Human-Computer Interface Design, Negroponte, 245-246
The Art of Human-Computer Interface Design, Negroponte, 347-354
The Art of Human-Computer Interface Design, Laurel, 355-366
The Art of Human-Computer Interface Design, Oren, 367-382

Topics

• Input Errors
• Computation Errors
• File and Device Errors
• Error Messages
• Crash Recovery
Lecture 11: Window Systems and User Interface Management

Readings


The Art of Human-Computer Interface Design, Gygi, 279-288
The Art of Human-Computer Interface Design, Walker, 439-448
The Art of Human-Computer Interface Design, Oren, 467-480

Topics

• Window Systems Concepts
• Windows and Window Systems
• Bit versus Character-Mapped Windows
  • Curses
  • Microsoft Windows
  • The X Window System
• User Interface Management Systems

Lecture 12: Direct Manipulation Systems and Special Environments

Readings


The Art of Human-Computer Interface Design, Kim, 31-44
The Art of Human-Computer Interface Design, Vertelney, 57-64
The Art of Human-Computer Interface Design, Erickson, 65-74
The Art of Human-Computer Interface Design, Tognazzini, 75-78
The Art of Human-Computer Interface Design, Fisher, 423-438
The Art of Human-Computer Interface Design, Naimark, 455-460

Topics

• Direct Manipulation Concepts
• Programming Direct Manipulation
• Direct Manipulation Techniques for Data Input
• Experimental and Advanced Environments
  • The Interactive Image
  • Voice Recognition
  • Virtual Reality
Lecture 13: Setup/Ini Files & Special Groups

Readings

The Art of Human-Computer Interface Design, Gassee, 225-228

Topics

• Setup/Initialization Files
  • Textual Info
  • Screen Control
  • Menu Control and Function Pointers
• Special Groups
  • Special Input Devices
  • Menu Considerations
  • Managing Commands
  • Other Ways to Get Interactive Input
  • Considerations for Visually Impaired Users
  • Considerations for Reading Disabled Users

Lecture 14: Logging User Activity

Topics

• Daemons
  • Keyboard Daemons
  • Menu Daemons
  • Application Daemons
• Daemon Daemons