JEFFERSON COLLEGE

COURSE SYLLABUS

CIS129

Programming Logic

3 Credit Hours

Prepared by
Nancy Smith Walker

Revision Date: September 2005
by
Don LaPlante

CAREER & TECHNICAL EDUCATION
Alan C. Foster, Associate Dean
CIS129: Programming Logic

I. CATALOG DESCRIPTION

Prerequisite: none
Credit Hours: 3

This course develops analytical skills using structured programming design methods to solve practical business problems. This course is strongly recommended as a co-requisite or pre-requisite for programming languages.

II. GENERAL OBJECTIVES

A. The student will be able to use either flowcharts
B. The student will be able to design using pseudocode
C. The student will be able to employ subroutines in solving a problem
D. The student will be able to use page breaks
E. The student will be able to apply accumulation techniques to subtotals and final totals
F. The student will understand basic object-oriented techniques
G. The student will design user-friendly graphical user interfaces

III. COURSE OUTLINE

A. An Overview of Computers and Logic
B. Understanding Structure
C. Modules, Hierarchy Charts, and Documentation
D. Writing a Complete Program
E. Making Decisions
F. Looping
G. Control Breaks
H. Arrays
I. Object-Oriented Programming
J. Programming Graphical User Interfaces

IV. UNIT OBJECTIVES

A. An Overview of Computers and Logic
   1. Describe the data hierarchy
   2. Understand how to use flowchart symbols and pseudocode statements
   3. Use and name variables
   4. Use a sentinel, or dummy value, to end a program
   5. Use a connector
   6. Assign values to variables
   7. Recognize the proper format of assignment statements
   8. Describe data types
B. Understanding Structure
   1. Describe the features of unstructured spaghetti code
   2. Describe the three basic structures of sequence, selection, and loop
   3. Use a priming read
   4. Appreciate the need for structure
   5. Recognize structure

C. Modules, Hierarchy Charts, and Documentation
   1. Describe the advantages of modularization
   2. Modularize a program
   3. Understand how a module can call another module
   4. Explain how to declare variables
   5. Create hierarchy charts
   6. Understand documentation
   7. Create print charts
   8. Interpret file descriptions
   9. Understand the attributes of complete documentation

D. Writing a Complete Program
   1. Plan the mainline logic for a complete program
   2. Describe typical housekeeping tasks
   3. Describe tasks typically performed in the main loop of a program
   4. Describe tasks performed in the end-of-job module

E. Making Decisions
   1. Evaluate Boolean expressions to make comparisons
   2. Use the logical comparison operators
   3. Understand AND logic
   4. Combine decisions in an AND situation
   5. Understand OR logic
   6. Combine decisions in an OR situation
   7. Use selections within ranges

F. Accumulators and Counters
   1. Understand the use of accumulators and counters
   2. Use accumulators and counters

G. Control Breaks
   1. Understand control break logic
   2. Perform single-level control breaks
   3. Use control data within the control break module
   4. Perform control breaks with totals
   5. Perform page breaks

H. Arrays
   1. Understand how arrays are used
   2. Understand how arrays occupy computer memory
   3. Manipulate an array to replace using nested decisions
   4. Declare and initialize an array
5. Understand the difference between run-time and compile-time arrays
6. Load array values from a file
7. Search an array for an exact match
8. Use parallel arrays

I. Object-Oriented Programming
   1. Understand the principles of object-oriented programming
   2. Define classes
   3. Instantiate and use objects
   4. Understand inheritance

J. Programming Graphical User Interfaces
   1. Understand the principles of event-driven programming
   2. Describe the actions that GUI components can initiate
   3. Design graphical user interfaces
   4. Modify the attributes of GUI components
   5. List the steps to building an event-driven application

V. METHODS OF INSTRUCTION
   A. Lectures
   B. Teamwork
   C. Deskchecking designs

VI REQUIRED TEXTBOOK

VII REQUIRED MATERIALS
   A. Folder for submission of bonus homework assignments
   B. Flowcharting template recommended (available in bookstore)

VIII SUPPLEMENTAL REFERENCES
   A. Handouts provided
   B. VISIO 2000 for hierarchy charts and flowcharts available in TC103, TC107, TC109, JCA206, and JCA208
   C. MS Word (Office 2000) for typed assignments and pseudocode
   D. Print chart in Excel (psc.xls)

IX METHODS OF EVALUATION
   (points are approximate)
   A. Eight chapter tests
   B. In-class assignments
   C. Programming problems and review questions for each chapter
<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90-100%</td>
</tr>
<tr>
<td>B</td>
<td>80-89%</td>
</tr>
<tr>
<td>C</td>
<td>70-79%</td>
</tr>
<tr>
<td>D</td>
<td>60-69%</td>
</tr>
<tr>
<td>F</td>
<td>Below 60%</td>
</tr>
</tbody>
</table>