

Maryland in Europe Graduate Programs
Bowie State University

Software Structures
INSS 520

TERM V – 2002-2003

June 16, 2003 to July 25, 2003

Rota, Spain

Mondays, Tuesdays and Thursdays 1715 – 1950 hours

Instructor: Edmund I. Deaton, Ph. D.
Mailing Address: PSC 819 Box 54; FPO AE 09645
Email Address: edeaton@faculty.ed.umuc.edu
Consultation: Mondays, Tuesdays and Thursdays 1100-1200 and before class

Course Description: *Prerequisites: Undergraduate programming and college algebra, or permission of the instructor. Recommended prerequisite: INSS 510.* Provides an in-depth look at programming languages from a design and implementation perspective. Language semantics and syntax issues are explored. Specification and implementation of data structures are examined. Characteristics of non-procedural, heuristic and object-oriented languages are discussed. Current developments in software engineering methodologies are reviewed as well as research into the improvement of those practices. Software project management concepts and software quality issues are also addressed. *Students will be required to complete programming projects.*

Course Goals/Objectives:

Goals: Students who complete this course should have knowledge of:

1. Major attributes of several programming languages
2. Tradeoffs in programming language design and usage
3. Data types and abstract data types
4. Basic data structures
5. Structured programming
6. How a programming language can support good software engineering
7. Computational complexity and its relationship to software quality;
8. The principal programming paradigms: imperative/procedural, object-oriented, functional/applicative, logic, and concurrent programming
9. Current issues in programming languages

Objectives: At the conclusion of this course the student will be able to:

1. Describe and apply the fundamental criteria needed to evaluate and compare computer programming languages
2. Demonstrate understanding of the underlying concepts of programming languages such as: syntax, semantics, binding, type checking, scope, data types,

- expressions, control structures, and subprograms
3. Understand the major programming paradigms; recognize differences between imperative, object-oriented, functional and logic programming languages
 4. Compare and contrast the different capabilities of programming languages and evaluate languages for various programming problems
 5. Characterize a given program or algorithm in terms of its computational complexity and efficiency
 6. Compare and contrast different implementations of standard data structures such as lists, stacks, and queues
 7. Understand concepts of object-oriented programming such as encapsulation, inheritance, dynamic binding, and polymorphism
 8. Apply understanding of software engineering practices to software quality assurance
 9. Apply programming concepts in making software management decisions
 10. Research and discuss current issues in programming languages

Text: Sebesta, Robert W. (2002). *Concepts of Programming Languages*, (5th ed.). Boston: Addison-Wesley.

Grading Information: Grades for this course will be assigned as follows:

A	90% +	C	70 – 79%
B	80 – 89%	F	Below 70% F(a) or regular non-attendance F(n)

Course Requirements

Minor Assignments, exercises, class participation	10%
Programming Projects	30%
Midterm Examination:	20%
Final Examination:	25%
Group Research Project	15%

Projects Descriptions: Group Research Project: Students, in groups of 2 or 3, will select and evaluate a programming language of their choice. No two groups will have the same programming language to study. Details will be distributed in class.

Programming Projects: There will be three programming assignments to be done individually.

Tentative Course Schedule: *The instructor reserves the right to modify the syllabus when needed.*

Date	Module	Topics	Assigned readings/assignments due
16 June	1	Introduction to Course Procedures, Preliminaries	Ch 1
17 June	2	Evolution, History of Programming Languages	Ch 2
19 June	3	Programming Syntax and Semantics	Ch 3

23 June	4	Lexical and Syntax Analysis	Ch 4
24 June	5	Names, Bindings, Type Checking and Scopes	Ch 5
26 June	6	Data Types	Ch 6 Due Programming Project 1
30 June	7	Expressions and the Assignment Statement	Ch 7
1 July	8	Control Structures	Ch 8
3 July	9	Mid Term Exam Ch 1-8	
7 July	10	Subprograms	Ch 9
8 July	11	Implementing Subprograms	Ch 10
10 July	12	Abstract Data Types	Ch 11 Due Programming Project 2
14 July	13	Object-Oriented Programming	Ch 12
15 July	14	Concurrency	Ch 13
17 July	15	Exception Handling	Ch 14
21 July	16	Functional and Logic Programming Languages	Ch 15, Ch 16 Due Research Project, Group Work
22 July	17	Group Work Presentations Course Evaluation	Due Programming Project 3
24 July	18	Final Examination	

Academic Policies: Please refer to the UMUC Maryland in Europe Graduate Catalog, available online at http://www.ed.umuc.edu/visit/pubs/catalog/grad_02-03.pdf or from your local Education Center, for information on the following:

Academic Integrity
Course Load
Exception to Policy
Grade Appeal Process
Make-up Examinations
Nondiscrimination
Students with Disabilities

CODE OF CIVILITY

To promote a positive, collegial atmosphere among students, faculty, and staff, Maryland in Europe has developed the following Code of Civility:

Respect

Treat all students, faculty, and staff with respect and in a professional and courteous manner at all times and in all communications, whether in person or in written communication (including e-mail).

Kindness

Refrain from using profanities, insults, or other disparaging remarks.

Truth

Endeavor to cite only the truth and not knowingly misrepresent, mischaracterize, or misquote information received from others.

Responsibility

Take responsibility for our own actions instead of blaming others.

Cooperation

Work together with other students, faculty, and staff in a spirit of cooperation toward our common goals of seeking and providing quality education.

Privacy

Strive to uphold the right to privacy and not talk about others.

Nondiscrimination

Respect the differences in people and their ideas and opinions and reject bigotry.

Introduction to the Instructor: Edmund Deaton

Dr. Deaton received his Ph.D. in Mathematics from the University of Texas. He has been teaching and doing research in Computer Science since 1980. After many years at San Diego State University he retired in 1992. He was a visiting professor at Hope College, Holland, Michigan during 1993-1995. He spent two years at Oklahoma State University from 1980 to 1982 as a visiting professor and visited there again in 1992. He worked as a management consultant with a Southern California consulting firm for several years in the 1980's. He specialized in database design for governmental entities. He has been with the University of Maryland, European Division since 1995. He teaches in the graduate MIS program and also teaches undergraduate computer science courses. His academic specialty is data base design. His primary hobbies are hiking and Alpine climbing. Although based in Heidelberg, he calls Rota, Spain home and hopes to be assigned there for some time each year.

E-mail: edeaton@faculty.ed.umuc.edu Phone: 956 81 38 51 (h), 635 81 01 97 (c)