

Maryland in Europe Graduate Programs  
Bowie State University

**Software Structures**  
**INSS 520**

7 June – 27 July 2003

Kapaun Education Center, Germany

7/8 June, 21/22 June, 12/13 July, 26/27 July 2003, 0900 - 1600

**Instructor:** Kerry Painter  
**Mailing Address:** PSC 9, BOX 2297, APO AE 09123  
**Email Address:** kpainter@t-online.de  
**Consultation:** Lunchtime and immediately after class (weekends)

**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	92% +	C	70 – 79%
B	80 – 91%	F	Below 70% F(a) or regular non-attendance F(n)

**Course Requirements:** Grades for this course will be based on:

Test #1:	20%
Test #2:	20%
Test #3:	20%
Programming Project:	20%
Language Research Project	20%

### **Project Description:**

Two projects are required:

- a programming project, and
- a language research project.

Each student will be assigned to study groups consisting of three – five members each.

The group programming project can be implemented in a suitable language and the actual assignment and problems specifications will be announced and distributed in class during our first session.

Each study group will also be assigned a language to research. On the Sunday morning of our last meeting, each group will present their research to the class in the form of a slide presentation and a compiler or interpreter demonstration with an actual program written by the group. Some items to consider in the slide presentation: history of the language, author(s), notable characteristics of the language, syntax examples, and so on.

**Course Schedule:**

Module	Topics	Assigned readings/assignments due
1 Saturday AM June 7	Preliminaries: Introduction to Programming Languages	Chapter 1
2 Saturday PM June 7	Evolution of the Major Programming Languages	Chapter 2
3 Sunday AM June 8	Syntax and Semantics	Chapter 3
4 Sunday PM June 8	Lexical and Syntax Analysis	Chapter 4
5 Saturday AM June 21	<b>Test #1: Chapters 1 – 4 (1<sup>st</sup> Hour)</b>  Names, Bindings, Type Checking, and Scopes	<b>Due:</b> algorithm for programming project; progress report  Chapter 5
6 Saturday PM June 21	Data Types	Chapter 6
7 Sunday AM June 22	Expressions and the Assignment Statements	Chapter 7
8 Sunday PM June 22	Statement Level Controls	Chapter 8
9 Saturday AM July 12	<b>Test #2: Chapters 5 – 8 (1<sup>st</sup> Hour)</b>  Subprograms	<b>Due:</b> <ul style="list-style-type: none"><li>• programming project samples</li><li>• language research project samples</li></ul> Chapter 9
10 Saturday PM	Implementing Subprograms	Chapter 10

July 12		
11 Sunday AM July 13	Abstract data types  Support for OO programming	Chapter 11  Chapter 12
12 Sunday PM July 13	Exception Handling	Chapter 14
13 Saturday AM July 26	Functional Programming Languages	Chapter 15
14 Saturday PM July 26	Logic Programming Languages	Chapter 16
15 Sunday AM July 27	<b>Group Presentations</b>  Wrap-up and Review	
16 Sunday PM July 27	<b>Test #3: Chapters 9 – 12, 14 – 16 (last three hours)</b>	<b>Due:</b> <ul style="list-style-type: none"> <li>• Printed final copy of programming project code and documentation; copy of code on disk</li> <li>• Printed copy of final language research documents; copy of presentation and sample program and compiler or interpreter code on disk</li> </ul>

**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](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.

**About Your Instructor:** Kerry Painter earned his BA degree in Chinese-Vietnamese Language Studies from the University of Hawaii in Honolulu. Before attending the University of Hawaii, he studied electrical engineering at Clemson University, attended the 47-week North Vietnamese language course at the Defense Language Institute in Monterey, California, and worked as a linguist for the Army Security Agency in both Vietnam and Korea.

For six years after college graduation he worked in radio news as a reporter, writer, announcer, news director, and manager in Hawaii, Texas, and Delaware. He earned an MS degree in Technical and Science Communication and an MS in Computer Science from Drexel University in Philadelphia.

Kerry did doctoral studies at Drexel and has taught a variety of computer science and mathematics courses at Drexel University, Penn State University, Elizabethtown College, and Swarthmore College, all in Pennsylvania. He joined The University of Maryland European Division in January 1989 and has taught at SHAPE in Belgium, Soesterberg Air Base and AFNORTH in Holland, Aviano Air Base in Italy, and at several German locations: Augsburg, Bad Kreuznach, Bamberg, Baumholder, Berlin, Beuchel, Geilenkirchen, Giebelstadt, Hahn, Hanau, Heidelberg, Kapaun, Kitzingen, Mannheim, Ramstein, Schweinfurt, Spangdahlem, Wiesbaden, and Wuerzburg.