El Paso Community College
Syllabus
Part II
Official Course Description

SUBJECT AREA	**Computer Science**
COURSE RUBRIC AND NUMBER	**COSC 1437**
COURSE TITLE	**Programming Fundamentals II**
COURSE CREDIT HOURS	**4 4 1** **Credits Lec Lab**

I. Catalog Description

Focuses on the object-oriented programming paradigm, emphasizing the definition and use of classes along with fundamentals of object-oriented design. Includes basic analysis of algorithms, searching and sorting techniques, and an introduction to software engineering processes. Students will apply techniques for testing and debugging software. (This course is included in the Field of Study Curriculum for Computer Science.) **Prerequisite: INRW 0311 or ESOL 0340 (can be taken concurrently) or by placement exam or ENGL 1301 with a "C" or better or ENGL 1302 with a "C" or better. (4:1). Lab fee.**

II. Course Objectives

Upon satisfactory completion of this course, the student will be able to:

A. Unit I. Two-Dimensional Arrays
1. Describe how to declare and initialize a 2D array in terms of rows and columns.
2. Explain the process to transverse the 2D array by using nested loops.
3. Use method-based operations to manipulate rows and columns from a 2D array.
4. Accomplish security objectives by using 2D array to simulate a fingerprint or other biometric system.

B. Unit II. Linked Lists
1. Describe the difference between random access and sequential access.
2. Explain the notion of Node and the need to have an inner structure in a Linked List.
3. Identify the trade-offs of having a Node-based structure vs a Linked List structure.
4. Explain the syntax of how to define a Linked List and use methods to store information.
5. Extend the notion of single-linked to double-linked list.
6. Describe the recursive-linked list and translate the methods from single-linked list to recursive-linked list.
7. Accomplish security objectives by storing validated and verified data from the user.

C. Unit III. Recursion
1. Explain how to use recursive techniques in Java programming and avoid an "infinite" recursion logic.
2. Identify and explain the base case and recursive case from a recursive-based method.
3. Discuss the difference between head and tail recursion.
4. Illustrate the recursion activation records in a Stack.
5. Practice security objectives by recursive calls to methods to exhaust system's memory capacities.

D. Unit IV. Searching and Sorting Algorithms and Introduction of Complexity
1. Identify sequential, binary, and hash searches and explain the difference between these three approaches.

2. Describe bubble, insertion, selection, quick, and merge sort.
3. Explain the difference and complexity between iterative-based and recursive-based implementations.
4. Identify runtime analysis and Big-O notation.
5. Demonstrate brute-force algorithms vs divide-and-conquer algorithms to accomplish security objectives, e.g., attempting to break a password.

E. Unit V. Stacks and Queues
1. Identify the order of insertion and removal for both data structures, e.g., FIFO and FILO.
2. Implement push, pop, peek/enqueue, dequeue, and peek for both data structures.
3. Identify real-world applications for both data structures.
4. Accomplish security objectives by identifying bad input from the user according to the program design.

F. Unit VI. Trees
1. Identify binary search and AVL trees.
2. Provide implementation for both trees.
3. Explain complexity in the ideal and in the worst case scenarios for both implementations.
4. Accomplish security objectives by establishing different levels of security using binary search.

G. Unit VII. Hash Tables and Binary Heaps
1. Explain why a Hash Table is a hybrid data structure.
2. Explain and implement separate chaining and linear probing algorithms.
3. Compare complexities between hash tables and other data structures.
4. Explain binary heaps and the heapsort; relate the binary heap as a priority queue and discuss complexity.
5. Accomplish security objectives by showing the difference between hashing and encryption.

H. Unit VIII. Collections
1. Introduce the java.util.Collections and describe the three main families: List, Set, and Map.
2. Define ArrayList, Vector, LinkedList, HashSet, and TreeSet.
3. Describe Collections methods such as sort(), shuffle(), and search().
4. Accomplish security objectives such as integrity and confidentiality by using data structures in the Collection class.

III. THECB Learning Outcomes (ACGM)

Upon successful completion of this course, students will:
1. Identify and explain a programming development lifecycle, including planning, analysis, design, development, and maintenance.
2. Demonstrate a basic understanding of object-oriented programming by using structs and classes in software projects.
3. Use object-oriented programming techniques to develop executable programs that include elements such as inheritance and polymorphism.
4. Document and format code in a consistent manner.
5. Apply basic searching and sorting algorithms in software design.
6. Apply single- and multi-dimensional arrays in software.
7. Use a symbolic debugger to find and fix runtime and logical errors in software.
8. Demonstrate a basic understanding of programming methodologies, including object-oriented, structured, and procedural programming.
9. Describe the phases of program translation from source code to executable code.

IV. Evaluation

A. Pre-assessment
None

B. Points will be given for the coursework and exams that are made available during the course. The type of coursework and number of exams as well as the point value of each will be determined by the instructor.

C. Post-assessment
The final grade will be determined by the percentage obtained by dividing the total number of points earned by the student by the total number of points made available through the course of the semester.

D. Course Grade

The course grade will be based on the percentage of the total points earned:

Percentage	Letter Grade
90.00 – 100.00	A
80.00 – 89.99	B
70.00 – 79.99	C
60.00 – 69.99	D
0.00 – 59.99	F

V. Disability Statement (Americans with Disabilities Act [ADA])

EPCC offers a variety of services to persons with documented sensory, mental, physical, or temporary disabling conditions to promote success in classes. If you have a disability and believe you may need services, you are encouraged to contact the Center for Students with Disabilities to discuss your needs with a counselor. All discussions and documentation are kept confidential. Offices located: VV Rm C-112 (831-2426); TM Rm 1400 (831-5808); RG Rm B-201 (831-4198); NWC Rm M-54 (831-8815); and MDP Rm A-125 (831-7024).

VI. 6 Drop Rule

Students who began attending Texas public institutions of higher education for the first time during the Fall 2007 semester or later are subject to a 6-Drop limit for all undergraduate classes. Developmental, ESL, Dual Credit and Early College High School classes are exempt from this rule. All students should consult with their instructor before dropping a class. Academic assistance is available. Students are encouraged to see Counseling Services if dropping because exemptions may apply. Refer to the EPCC catalog and website for additional information.