El Paso Community College
Syllabus
Part II
Official Course Description

SUBJECT AREA	**Computer Science**
COURSE RUBRIC AND NUMBER	**COSC 1436**
COURSE TITLE	**Programming Fundamentals I**
COURSE CREDIT HOURS	**4 4 1** **Credits Lec Lab**

I. Catalog Description

Introduces the fundamental concepts of structured and object-oriented programming, and provides a comprehensive introduction to programming for computer science and technology majors. Includes topics on software development methodology, data types, control structures, functions, arrays, and the mechanics of running, testing, and debugging. This course assumes computer literacy. **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. Introduction to Programming and Java
1. Describe the techniques for problem solving
2. Identify programming concepts
3. Write pseudocode and flowcharts
4. Explain the differences between structured and object-oriented programming

B. Unit II. Fundamentals in Java
1. Identify the syntax of how to define and initialize variables of different types
2. Identify the primitive types in Java and the difference between them
3. Describe what a scope is and the visibility of a variable
4. Identify the correspondence between the class and the Java source code file name
5. Define what a String is and the difference between string and other primitive types
6. Explore the Java API for the Math, String, and Scanner library for usage of methods
7. Write programs that interact with the user using scanner and JOptionpane
8. Extract information from the user to handle primitive types or Strings
9. Secure programming objective
    i. Identify potential bad input from the user according to the program design

C. Unit III. Control and Decision Structures
1. Write flowcharts to understand decisions for a given problem
2. Explain Boolean expressions by using relational operators
3. Write programs that perform decisions by using if-else and else-if statements
4. Write programs that perform switch-case statements and decision operators (?:)
5. Translate problems that involve nested-if statements
6. Remove nested-if statements by using logical operators
7. Secure Programming Objectives
    i. Write programs that verify the user's input based on given restrictions
    ii. Identify invalid input from the user and notify the user

D. Unit IV. Looping Structures and File Manipulation

1. Identify control variables in code and translate it into a for, while, and do-while loop
2. Trace the state of variables based in a certain iteration
3. Apply sentinels and flags to a loop that control user's input
4. Write a Java source program which reads and writes to text files
5. Identify code that uses nested loops
6. Secure Programming Objective
   i. Write a code that loops the user's invalid input until data is valid
   ii. Write a code that finds a password of a given length

E. Unit V. Methods

1. Identify a piece of code and define a method based on its functionality
2. Define a method's signature based on a given description
3. Explain the difference between method definition and method calling
4. Secure Programming Objective
   i. Write a code that checks parameters based on certain restrictions on data
   ii. Validate return type of methods

F. Unit VI. Elementary Data Structure: Array

1. Access certain positions of an array
2. Use a loop to perform operations with the data of a given array
3. Use a method that takes an array as a parameter and returns a value based on the array's data
4. Secure programming objective
   i. Write a program that checks boundaries of an array as an example of a buffer over/under flow

G. Unit VII. Object Orientation Programming: Object Design

1. Implement simple ADTs incorporating multiple primitive instance variables and at least one reference instance variable, with appropriate getters and modification methods.
2. Create multiple objects and store them in an array of the same type, then calculate operations based on the instance variables from the objects

H. Unit VIII. Object Orientation Programming: Inheritance, Polymorphism, and Encapsulation

1. Create a subclass that extends a super class, then use methods from super class in the subclass
2. Redefine the toString() method in the sub-class
3. Use private and protected methods to understand the notion of en- capsulation
4. Secure Programming Objective
   i. Protect data from a super class by having private methods; however, use protected methods to provide accessibility for sub-class
   ii. Use the final keyword to avoid overwritten operations

I. Unit IX. Exception Handling

1. Use the exception-handlers: the try-catch-finally statement
2. Use the throws keyword to perform an exception for a given event
3. Define an exception class for a certain exception to be thrown
4. Secure Programming Objective
   i. report specific exception to the user based in the input the user provides

III. THECB Learning Outcomes (ACGM)

Upon successful completion of this course, students will:

1. Describe how data are represented, manipulated, and stored in a computer.
2. Categorize different programming languages and their uses.
3. Understand and use the fundamental concepts of data types, structured programming, algorithmic design, and user interface design.
4. Demonstrate a fundamental understanding of software development methodologies, including modular design, pseudo code, flowcharting, structure charts, data types, control structures, functions, and arrays.
5. Develop projects that utilize logical algorithms from specifications and requirements statements.

6. Demonstrate appropriate design, coding, testing, and documenting of computer programs that implement project specifications and requirements.
7. Apply computer programming concepts to new problems or situations.

IV. Evaluation

A. There will be four written/programming examinations, each worth 100 points.
B. Lab assignments will be assigned at the instructor's discretion and will be averaged on a 100-point scale. The total points accumulated by the lab assignments will be 300 points.
C. There will be quizzes averaged on a 100-point scale.
D. There will be a comprehensive final examination that will be worth 200 points.

E. 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.