COURSE DESCRIPTION

Department and Course Number	CS 242	**Course Coordinator**	Behzad Parviz
Course Title	C Programming	**Total Credits**	4

Current Catalog Description:

Introduction to program design and problem solving using the C programming language. Programming topics include control structures, functions, arrays, pointers, and file I/O.

Textbook:

Deitel and Deitel., *C++ How to Program, Fourth Edition.* Prentice Hall, 2003**.**

Deitel and Deitel., *C++ in the Lab, Fourth Edition.* Prentice Hall, 2003.

References:

Savitch, Walter., *Problem Solving with C++: The Object of Programming, 4th Ed.,* Addison Wesley**,** 2003.

Course Goals:

At the end of the course, students are able to

- Understand the basic programming concepts.
- Understand the use of arrays to store lists and tables of values.
- Use pointers and Strings.
- Understand the close relationships among pointers, arrays and strings.
- Understand the notion of data abstraction and ADTs
- Divide a problem into its logical set of components.
- Understand how a good program design can reduce coding and debugging time.
- Design and code most mid-level problems from the start.

This course is offered as a service course for non CS majors.

Prerequisites by Topic:

Algebra and Trigonometry.

Major Topics Covered in the Course:

- Introduction to Computers
  What is a Computer
  Computer Organization
  History of C and C++

  Arithmetic operators
  Equality and relational operators

- Control Statements
  Algorithms
  Pseudo-code
  if…else Selection Statement
  while Repetition Statement
  Formulating Algorithms
  Compound Assignment Operators
  Primitive Types
  for Repetition Statement
  do…while Repetition Statement
  switch Multiple-Selection Statement
  break and *continue* Statements
  Logical Operators

- Functions
  Program Modules in C
  Declaring and using Functions
  Passing arguments by values and by reference
  Recursive functions
  Argument Promotion and Casting
  Math library functions
  Scope of Declarations

- Arrays
  Declaring and Creating Arrays
  Examples Using Arrays
  Passing Arrays to Methods
  Multidimensional Arrays

- Pointers and Strings
  Introduction to pointers and pointer arithmetic
  Directly and indirectly referencing a variable
  Pointer operators & and *
  Pass-by-reference with pointer arguments
  Introduction to Strings and String manipulations
  Library string manipulation functions.

- Class and Data Abstraction
  Introduction to structures and classes
  Encapsulation and data hiding
  Data abstraction and abstract data type
  public and private members of the class
  Creating, using, and destroying class objects

Laboratory Projects (specify number of weeks on each):

Each week students complete a 3-hour lab project on a selected. In addition, they design and develop three to five large projects covering the material taught in the course.

Week 1: Introduction to computers and basic programming concepts and constructs

Week 2: Program development using simple control statements for decision making and repetition: *if…else* and *while* statements.

Week 3: Program development using control statements: *for, do…while, switch, break,* and *continue* statements.

Week 4: Problem development and design process using pseudo-code.

Week 5: Program modularity using functions.

Week 6: Program development using arrays

Week 7: Program development using strings

Week 8: Problem development using pointers

Week 9: Problem development using ADTs and objects

Week 10: More emphasis on design

Estimate Curriculum Category Content (Quarter Hours)

Area	Service*	Area	Service*
Algorithms	0.75	Data Structures	0.75
Software Design	1.0	Prog. Languages	1.5
Comp. Arch.		Other	

Service* : This course is offered as a service course for non CS majors.

Oral and Written Communications:

Written documentation of software built in labs and homework assignments.

Social and Ethical Issues:

No significant component.

Theoretical Content:

No significant component.

Problem Analysis:

In this course, students learn the basic concepts of the C and C++ programming languages. They learn algorithm development for structured programming, designing, coding, debugging, and documenting programs.

Solution Design:

Solution design in this course mostly involves generating pseudo-code for program development. Students also learn how good program design reduces coding and debugging time.