



**CREDIT COURSE OUTLINE**

**I. COVER PAGE**

(1) ENGR 40	(2) Programming for Scientists and Engineers	(3) 4
Number	Title	Units

(4) Lecture / Lab Hours:		(8) Classification:	
Course Hours			
	Weekly Lec hours:	3.00	Degree applicable: X
	Weekly Lab hours:	2.00	Non-degree applicable:
	Total Contact hours:	90.00	Basic skills:
Lec will generate __ hour(s) outside work.		(9)RC Fulfills AS/AA degree requirement: (area)	
Lab will generate __ hour(s) outside work.		Computer Familiarity	
(5) Grading Basis:	Grading Scale Only	General education category:	
	Pass/No Pass option	Major:	
	Pass/No Pass only	Certificate of:	
(6) Advisories:		Certificate in:	
<ul style="list-style-type: none"> <li>• Eligibility for English 125 or 126</li> <li>• <b>Eligibility for ENGL 125</b> Advisory (13) - Outcome to Objective (1) Requisite Outcomes:           <ul style="list-style-type: none"> <li>• an evaluation and analysis of ideas at the appropriate course level</li> <li>• complete sentences which includes correct capitalization, spelling, use of homophones, etc.</li> <li>• an avoidance of major grammatical errors including verb tense issues, subject-verb agreement, pronoun agreement problems, fragments, fused sentences and comma splices</li> <li>• appropriate use of academic language and descriptive vocabulary</li> <li>• writing that is free from plagiarism</li> </ul>           Current Objectives:           <ul style="list-style-type: none"> <li>• Write computer programs using an object-oriented programming language C++.</li> <li>• Write computer programs using selection and repetition control structures.</li> <li>• Write computer programs using functions.</li> <li>• Identify and correct syntax and logical errors in computer programs.</li> <li>• Write a total of 500 to 1000 lines of program code.</li> </ul> </li> <li>• <b>Eligibility for ENGL 126</b> Advisory (13) - Outcome to Objective (1) Requisite Outcomes:           <ul style="list-style-type: none"> <li>• apply a variety of vocabulary skills for increased comprehension during reading.</li> <li>• apply prereading and active reading strategies to increase success with and comprehension of unfamiliar texts.</li> <li>• analyze expository texts to determine explicit/implicit main ideas and logical support, leading to author's intended meaning.</li> </ul>           Current Objectives:           <ul style="list-style-type: none"> <li>• Write computer programs using an object-oriented programming language C++.</li> <li>• Write computer programs using selection and</li> </ul> </li> </ul>		(10)CSU Baccalaureate: X	
		(11)Repeatable: (A course may be repeated three times) 0	
		(12)C-ID:	
		Proposed Start Date: Fall 2012	

- repetition control structures.
- Write computer programs using functions.
- Identify and correct syntax and logical errors in computer programs.
- Create proper test cases to test computer programs.

(7) Pre-requisites (requires C grade or better):

- Mathematics 4A or 4C

Corequisites:

(12) Catalog Description:

This course introduces the use of C++ programming language to solve engineering and applied science problems. A systematic development of program structure, specification, testing and debugging.

## II. COURSE OUTCOMES:

*(Specify the learning skills the student demonstrates through completing the course and link critical thinking skills to specific course content and objectives.)*

Upon completion of this course, students will be able to:

- I. Describe the software development life-cycle.
- II. Explain what an algorithm is and its importance in computer programming.
- III. Formulate, represent, and solve problems using a high level programming language.
- IV. Demonstrate knowledge of high level language syntax, control structures, looping, arrays, files, and records.
- V. Demonstrate proper programming style, debugging and testing techniques.
- VI. Solve application problems in science and engineering.

## III. COURSE OBJECTIVES:

*(Specify major objectives in terms of the observable knowledge and/or skills to be attained.)*

In the process of completing this course, students will:

- I. Write computer programs using an object-oriented programming language C++.
- II. Write computer programs using selection and repetition control structures.
- III. Write computer programs using pointer and array data structures.
- IV. Write computer programs using functions.
- V. Write computer programs to get input from files and write output to files.
- VI. Compile and link C++ programs to create executable programs.
- VII. Identify and correct syntax and logical errors in computer programs.
- VIII. Create proper test cases to test computer programs.
- IX. Write a total of 500 to 1000 lines of program code.

## IV. COURSE OUTLINE:

### Lecture Content:

- I. Introduction to computing and engineering problem solving
  1. Historical perspective
  2. Computing systems
  3. Data representation and storage
  4. An engineering problem-solving methodology
- II. Simple C++ programs
  1. Program structure
  2. Constant and Variables
  3. C++ classes
  4. C++ operators
  5. Standard input and output
  6. Basic functions included in the C++ standard library
- III. Control structure: Selection
  1. Algorithm development
  2. Conditional expressions
  3. Selection statements: if, if/else statements
  4. Selection statements: switch statements
- IV. Control structure: Repetition
  1. while loop
  2. do/while loop
  3. for loop
  4. break and continue statements
  5. Counter-controlled loops/Sentinel-controlled loops/End-of-data loop
- V. Working with data files
  1. Defining file streams
  2. Reading data files
  3. Generating a data file
- VI. Modular programming with functions
  1. Modularity

- 2. Programmer-defined functions
- 3. Parameter passing
- 4. Random numbers
- 5. Defining class methods
- VII. One-dimensional arrays
  - 1. Definition and initialization
  - 2. Function arguments
  - 3. Sorting and searching algorithms
  - 4. Character strings
  - 5. string class
  - 6. vector class
- VIII. Two-dimensional array
  - 1. Definition and initialization
  - 2. Function arguments
  - 3. Matrices
- IX. An introduction to pointers
  - 1. Addresses and Pointers
  - 2. Pointers to array elements
  - 3. Dynamic memory allocation

**Lab Content:**

Program assignments practice following concepts and topics:

- A. C++ language overview
- B. Using C++ Standard Library functions
- C. User defined functions
- D. Condition Control Structures
- E. Repetition structures
- F. Fundamental data structure - Arrays
- G. Text File Processing

**V. APPROPRIATE READINGS**

**Reading assignments may include but are not limited to the following:**

- I. Sample Text Title:
  - 1. Recommended - Etter, Delores M. *Engineering Problem Solving with C++*, ed. 3 Pearson, 2012, Programming assignments are included in the text.
  - 2. Recommended - Savitch, Walter *Problem Solving with C++*, ed. 8 Addison Wesley, 2012,
  - 3. Recommended - Gaddis, Tony *Starting Out With C++ From Control Structures Through Objects*, ed. 7 Addison Wesley, 2012,
  - 4. Recommended - Friedman and Koffman *Problem Solving, Abstraction, and Design using C++*, ed. 6 Pearson, 2011,
  - 5. Recommended - Stroustrup, Bjarne *Programming: Principles and Practice Using C++*, Addison Wesley, 2009, Programming assignments are included in the text.
- II. Other Readings
  - 1. Recommended - *Lab Handouts by Instructor*

- Global or international materials or concepts are appropriately included in this course
- Multicultural materials and concepts are appropriately included in this course

If either line is checked, write a paragraph indicating specifically how global/international and/or multicultural materials and concepts relate to content outline and/or readings.

**VI. METHODS TO MEASURE STUDENT ACHIEVEMENT AND DETERMINE GRADES:**

Students in this course will be graded in at least one of the following four categories. Please check those appropriate. A degree applicable course must have a minimum of one response in category A, B, or C.

<b>A. Writing</b>			
Check either 1 or 2 below			
X	<b>1. Substantial writing assignments are required. Check the appropriate boxes below and provide a written description in the space provided.</b>		
	<b>2. Substantial writing assignments are NOT required. If this box is checked leave this section blank. For degree applicable courses you must complete category B and/or C.</b>		
	a) essay exam(s)	X	d) written homework
	b) term or other paper(s)		e) reading reports
	c) laboratory report(s)	X	f) other (specify) Computer Programs

**Required assignments may include but are not limited to the following:**

Writing Homework Example:

Formulate a pseudocode algorithm to obtain a series of positive numbers from the keyboard, and determine and display the sum of the numbers.

Writing Computer Program Example:

Write a C program to generate a table of conversions from inches to centimeters. Start the inches column at 0.0 and increment by 0.5 in. The last line should contain the value 20.0 in. (1 in = 2.54 cm)

<b>B. Problem Solving</b>			
Computational or non-computational problem-solving demonstrations, including:			
X	a) exam(s)		d) laboratory reports
	b) quizzes		e) field work
X	c) homework problems	X	f) other (specify): Computer Programs

**Required assignments may include but are not limited to the following:**

Exam/Homework Problems

1. True/False questions
2. Fill-in the blanks
3. Identify and correct the errors in C statements
4. Write section of C++ code to perform a certain task

Programming Assignments

Write a program to calculate student grade point average (GPA).

Write a program to print a temperature conversion table (from Fahrenheit to Celsius)

<b>C. Skill demonstrations, including:</b>			
	a) class performance(s)		c) performance exams(s)
	b) field work	X	d) other (specify) Computer Programs

**Required assignments may include but are not limited to the following:**

Programming skills:

- \* Develop algorithms through the process of top-down, stepwise refinement.
- \* Identify and debug syntax and logical errors in programs

<b>D. Objective examinations including:</b>			
X	a) multiple choice	X	d) completion
X	b) true/false	X	e) other (specify): identify and correct errors
	c) matching items		

**COURSE GRADE DETERMINATION:**

Description/explanation: Based on the categories checked in A-D, it is the recommendation of the department that the instructor's grading methods fall within the following departmental guidelines; however, the final method of grading is still at the discretion of the individual instructor. The instructor's syllabus must reflect the criteria by which the student's grade has been determined. (A minimum of five (5) grades must be recorded on the final roster.)

If several methods to measure student achievement are used, indicate here the approximate weight or percentage each has in determining student final grades.

Example #1 of a grading system

30% Exams

20% Final Projects

50% Weekly Homework and Programs

Example #2 of a grading system

50% Exams

10% Final Projects

20% Homework Assignments

20% Programming Lab Assignments

**VII. EDUCATIONAL MATERIALS**

For degree applicable courses, the adopted texts, as listed in the college bookstore, or instructor-prepared materials have been certified to contain college-level materials.

Validation Language Level (check where applicable):

- Textbook
- Reference materials
- Instructor-prepared materials
- Audio-visual materials

College-Level Criteria Met	
YES	NO
<u>X</u>	<u>      </u>
<u>      </u>	<u>X</u>
<u>      </u>	<u>X</u>
<u>      </u>	<u>X</u>

Indicate Method of evaluation:

- Used readability formulae (grade level 10 or higher)
- Text is used in a college-level course X
- Used grading provided by publisher
- Other: (please explain; relate to Skills Levels)

Computation Level (Eligible for MATH 101 level or higher where applicable)

	YES	NO
Content	<u>X</u>	<u>      </u>
Breadth of ideas covered clearly meets college-level learning objectives of this course	<u>X</u>	<u>      </u>
Presentation of content and/or exercises/projects:		
Requires a variety of problem-solving strategies including inductive and deductive reasoning.	<u>X</u>	<u>      </u>
Requires independent thought and study	<u>X</u>	<u>      </u>
Applies transferring knowledge and skills appropriately and efficiently to new situations or problems.	<u>X</u>	<u>      </u>
List of Reading/Educational Materials		
Recommended - Etter, Delores M. <i>Engineering Problem Solving with C++</i> , ed. 3 Pearson, 2012, Programming assignments are included in the text.		
Recommended - Savitch, Walter <i>Problem Solving with C++</i> , ed. 8 Addison Wesley, 2012,		
Recommended - Gaddis, Tony <i>Starting Out With C++ From Control Structures Through Objects</i> , ed. 7 Addison Wesley, 2012,		
Recommended - Friedman and Koffman <i>Problem Solving, Abstraction, and Design using C++</i> , ed. 6 Pearson, 2011,		
Recommended - Stroustrup, Bjarne <i>Programming: Principles and Practice Using C++</i> , Addison Wesley, 2009, Programming assignments are included in the text.		

Comments:

- This course requires special or additional library materials (list attached).
- X This course requires special facilities:  
Computer Lab / Computers and Printers

Attached Files:

**BASIC SKILLS ADVISORIES PAGE** The skills listed are those needed for eligibility for English 125, 126, and Math 201. These skills are listed as the outcomes from English 252, 262, and Math 250. In the right hand column, list at least three major basic skills needed at the beginning of the target course and check off the corresponding basic skills listed at the left.

Eligibility for ENGL 125  
(as outcomes for ENGL 252)

- |   |  |
|---|--|
| <ul style="list-style-type: none"> <li><u>      </u> Write an essay of at least 750 words with an introduction, at least two body paragraphs, and a conclusion. This paper will include:               <ul style="list-style-type: none"> <li><u>      </u> a thesis statement</li> <li><u>      </u> unified supporting details for each body paragraph which begin with a topic sentence</li> <li><u>      </u> an evaluation and analysis of ideas at the appropriate course level</li> <li><u>      </u> complete sentences which includes correct capitalization, spelling, use of homophones, etc.</li> <li><u>      </u> an avoidance of major grammatical errors including verb tense issues, subject-verb agreement, pronoun agreement problems, fragments, fused sentences and comma splices</li> <li><u>      </u> appropriate use of academic language and descriptive vocabulary</li> <li><u>      </u> correct usage of MLA format</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li><u>      </u> Write computer programs using an object-oriented programming language C++.</li> <li><u>      </u> Write computer programs using selection and repetition control structures.</li> <li><u>      </u> Write computer programs using pointer and array data structures.</li> <li><u>      </u> Write computer programs using functions.</li> <li><u>      </u> Write computer programs to get input from files and write output to files.</li> <li><u>      </u> Compile and link C++ programs to create executable programs.</li> <li><u>      </u> Identify and correct syntax and logical errors in computer programs.</li> <li><u>      </u> Create proper test cases to test computer programs.</li> <li><u>      </u> Write a total of 500 to 1000 lines of program code.</li> </ul> |
|---|--|

<input type="checkbox"/> writing that is free from plagiarism <input type="checkbox"/> Plan and revise with guidance, employing all stages of the writing process when necessary. <input type="checkbox"/> Write an in-class paper with a beginning, middle, and end that communicates a clear idea.	
Eligibility for ENGL 126 (as outcomes for ENGL 262)	
<input type="checkbox"/> apply a variety of vocabulary skills for increased comprehension during reading. <input type="checkbox"/> apply prereading and active reading strategies to increase success with and comprehension of unfamiliar texts. <input type="checkbox"/> analyze expository texts to determine explicit/implicit main ideas and logical support, leading to author's intended meaning. <input type="checkbox"/> determine basic organizational writing patterns to increase comprehension of expository texts. <input type="checkbox"/> distinguish between fact and opinion and determine author's tone and purpose in non-fiction writings.	<input type="checkbox"/> Write computer programs using an object-oriented programming language C++. <input type="checkbox"/> Write computer programs using selection and repetition control structures. <input type="checkbox"/> Write computer programs using pointer and array data structures. <input type="checkbox"/> Write computer programs using functions. <input type="checkbox"/> Write computer programs to get input from files and write output to files. <input type="checkbox"/> Compile and link C++ programs to create executable programs. <input type="checkbox"/> Identify and correct syntax and logical errors in computer programs. <input type="checkbox"/> Create proper test cases to test computer programs. <input type="checkbox"/> Write a total of 500 to 1000 lines of program code.

Check the appropriate spaces.

- Eligibility for Math 201 is advisory for the target course.  
 Eligibility for English 126 is advisory for the target course.  
 Eligibility for English 125 is advisory for the target course.

*If the reviewers determine that an advisory or advisories in Basic Skills are all that are necessary for success in the target course, stop here, provide the required signatures, and forward this form to the department chair, the appropriate associate dean, and the curriculum committee.*

**REQUISITES**

**Prerequisite -- MATH 4A TRIGONOMETRY**

- |  |  |
|--|--|
| 1. Students learn and use the concept of mathematical functions.<br>2. Students learn to use the six basic trigonometric functions of $\sin(x)$ , $\cos(x)$ , $\tan(x)$ , $\cot(x)$ , $\sec(x)$ and $\csc(x)$ ; and their inverse functions.<br>3. Students learn and use the concept of radian measure. | <ul style="list-style-type: none"> <li>• Write computer programs using a high level programming language (C Language).</li> <li>• Write computer programs using functions.</li> <li>• Create proper test cases to test computer programs.</li> </ul> |
|--|--|

**ESTABLISHING PREREQUISITES OR COREQUISITES**

Every prerequisite or corequisite requires content review plus justification of at least one of the seven kinds below. Prerequisite courses in communication and math outside of their disciplines require justification through statistical evidence. Kinds of justification that may establish a prerequisite are listed below.

Check one of the following that apply. Documentation may be attached.

- Significant statistical evidence indicates that the absence of the prerequisite course is related to unsatisfactory performance in the target course.  
Justification: Indicate how this is so.  
 The health or safety of the students in this course requires the prerequisite.  
Justification: Indicate how this is so.  
 The prerequisite course is part of a sequence of courses within or across a discipline.  
 The prerequisite is required in order for the course to be accepted for transfer to the UC or CSU systems.  
Justification: Indicate how this is so.  
 The prerequisite/corequisite is required by law or government regulations.  
Explain or cite regulation numbers:  
 The safety or equipment operation skills learned in the prerequisite course are required for the successful or safe completion of this course.  
Justification: Indicate how this is so.  
 The safety or equipment operation skills learned in the prerequisite course are required for the successful or safe completion of this course.  
Justification: Indicate how this is so.  
 Three CSU/UC campuses require an equivalent prerequisite or corequisite for a course equivalent to the target course:

**Justification:**

**Prerequisite -- MATH 4C TRIGONOMETRY/PRECALCULUS**

1. Students learn and use the concept of mathematical functions.
2. Students learn to use the six basic trigonometric functions of  $\sin(x)$ ,  $\cos(x)$ ,  $\tan(x)$ ,  $\cot(x)$ ,  $\sec(x)$  and  $\csc(x)$ ; and their inverse functions.
3. Students learn and use the concept of radian measure.

- Write computer programs using a high level programming language (C Language).
- Write computer programs using functions.
- Create proper test cases to test computer programs.

**ESTABLISHING PREREQUISITES OR COREQUISITES**

Every prerequisite or corequisite requires content review plus justification of at least one of the seven kinds below. Prerequisite courses in communication and math outside of their disciplines require justification through statistical evidence. Kinds of justification that may establish a prerequisite are listed below.

Check one of the following that apply. Documentation may be attached.

\_\_\_\_\_ Significant statistical evidence indicates that the absence of the prerequisite course is related to unsatisfactory performance in the target course.

Justification: Indicate how this is so.

\_\_\_\_\_ The health or safety of the students in this course requires the prerequisite.

Justification: Indicate how this is so.

\_\_\_\_\_ The prerequisite course is part of a sequence of courses within or across a discipline.

\_\_\_\_\_ The prerequisite is required in order for the course to be accepted for transfer to the UC or CSU systems.

Justification: Indicate how this is so.

\_\_\_\_\_ The prerequisite/corequisite is required by law or government regulations.

Explain or cite regulation numbers:

\_\_\_\_\_ The safety or equipment operation skills learned in the prerequisite course are required for the successful or safe completion of this course.

Justification: Indicate how this is so.

\_\_\_\_\_ The safety or equipment operation skills learned in the prerequisite course are required for the successful or safe completion of this course.

Justification: Indicate how this is so.

\_\_\_\_\_ Three CSU/UC campuses require an equivalent prerequisite or corequisite for a course equivalent to the target course:

**Justification:**

## JUSTIFICATION OF LIMITATION ON ENROLLMENT

Enrollment in courses or blocks of courses may be limited based on performance, honors, or other performance based criteria. Be mindful of the disproportionate impact the limitation will have on specific groups of students. It is important to determine if the limitation will disproportionately keep under-represented students from enrolling in the course or block of courses.

Describe the reasons for limiting the enrollment.

Course Designator: ENGR 40

Course Title(s): Programming for Scientists and Engineers

Rationale for Limiting Enrollment:

0