

CBM003 ADD/CHANGE FORM

APPROVED DEC 7 2011

Undergraduate Council
 New Course | Course Change
 Core Category: Math/Reason Effective Fall 2012
Core Code 90

or

Graduate/Professional Studies Council
 New Course Course Change
 Effective Fall 2012

1. Department: Engineering Technology College: TECH
 2. Faculty Contact Person: Driss Benhaddou Telephone: 713-743-5818 Email: dbenhaddou@uh.edu

3. Course Information on New/Revised course:
- Instructional Area / Course Number / Long Course Title:
ELET / 2300 / Introduction to C++ Programming
 - Instructional Area / Course Number / Short Course Title (30 characters max.)
ELET / 2300 / INTRODUCTION C++ PROGRAMMING
 - SCH: 3.0 Level: SO CIP Code: 15.1201.0019 Lect Hrs: 3 Lab Hrs: 0

RECEIVED OCT 14 2011

4. Justification for adding/changing course: To meet core curriculum requirements

5. Was the proposed/revised course previously offered as a special topics course? Yes No

If Yes, please complete:

- Instructional Area / Course Number / Long Course Title:
____ / ____ / _____
- Course ID: _____ Effective Date (currently active row): _____

6. Authorized Degree Program(s): CETEBS, ISTEBS, EPETBS, BIOTECHBS, METEBS

- Does this course affect major/minor requirements in the College/Department? Yes No
- Does this course affect major/minor requirements in other Colleges/Departments? Yes No
- Can the course be repeated for credit? Yes No (if yes, include in course description)

7. Grade Option: Letter (A, B, C ...) Instruction Type: lecture ONLY (Note: Lect/Lab info. must match item 3, above.)

8. If this form involves a change to an existing course, please obtain the following information from the course inventory: Instructional Area / Course Number / Long Course Title

- ELET / 2300 / Introductionn to C++ Programming
- Course ID: 20618 Effective Date (currently active row): 08/23/2004

9. Proposed Catalog Description: (If there are no prerequisites, type in "none".)

Cr: 3. (3-0). Prerequisites: MATH 1330. Description (30 words max.): Fundamentals of C++ programming, conditional and looping constructs, functions and function overloading, arrays, pointers and references, class concepts, and files. Emphasis on practical applications.

10. Dean's Signature: [Signature] Date: 10/13/11

Print/Type Name: Fred Lewallen, Associate Dean for Academic Affairs

U N I V E R S I T Y *of* H O U S T O N
CORE CURRICULUM REQUEST FOR COURSES NEW TO THE CORE

Originating Department/College: Engineering Technology/College of Technology

Person making request: Heidar Malki Telephone: 3-4075

E-mail: _malki@uh.edu

Dean's signature: _____ Date: _____

I. General Information:

Course number and title: ELET 2300: Introduction to C++ Programming

Catalog description must be included on completed CBM 003 form and attached to this document.

Category of Core for which course is being proposed (mark only one):

- Communication
- Mathematics
- Mathematics/Reasoning (IDO)
- American History
- Government
- Humanities
- Visual/Performing Arts Critical
- Visual/Performing Arts Experiential
- Natural Sciences
- Social/Behavioral Sciences
- Writing in the Disciplines (IDO)

II. Objectives and Evaluation (respond on one or more separate sheets):

Call ext. 3-0919 for a copy of "Guidelines for Requesting and Evaluating Core Courses" or visit the website at www.uh.edu/academics/corecurriculum

- A. How does the proposed course meet the appropriate Exemplary Educational Objectives (see **Guidelines**). Attach a syllabus and supporting materials for the objectives the syllabus does not make clear. Please see following document.
- B. Specify the processes and procedures for evaluating course effectiveness in regard to its goals. Please see following document.
- C. Delineate how these evaluation results will be used to improve the course.
Please see following document.

SVP. Effective 8/23/10. Replaces all previous forms, which may no longer be used.

COT Undergraduate academic committee
UH undergraduate Academic Subcommittee
University of Houston

Dear Committee Members:

The faculty of the Engineering Technology would like to have ELET 2300 (C++ programming), declared as mathematics/reasoning component of the core curriculum. ELET 2300 meets the requirements of institutionally designated option: Mathematics/reasoning through the following:

- Requirement A: ELET 2300 is a non-advanced course.
- Requirement B: The exemplary educational objectives are met by the course and are included in the attached syllabus. The following are the objectives included in the syllabus:
 - 1- Students will learn to demonstrate ability to apply sequential thinking to the development of appropriate programming algorithms such as loops, accessing array elements.
 - 2- Students will learn to be able to utilize mathematical and statistical formulas to produce tables and graphs such as histograms and formatted symbolic outputs .
 - 3- Students will learn to evaluate logical operators, relational operators, and control structures.
 - 4- Students will learn to be able to utilize the latest software in solving the real-world problems by writing structured C++ code.
 - 5- Students will learn to demonstrate the ability to use of mathematical functions such as factorial, quadratic, exponential functions to create tabulated output.
 - 6- Students will learn to be able to recognize the limit of various variable declarations, arrays size, memory space, and limitations of built in mathematical functions in C++ libraries.
- Requirement C: ELET 2300 has been taught the Engineering Technology department of many years and its objectives has been evaluated following Accrediting Board for Engineering and Technology accreditation procedure. The evaluation use three exams, quizzes, and programming projects to evaluate student learning outcomes. The following table summarizes the evaluation process:

Objective	Evaluation Technique
Objective 1	

Objective 2	Exams Quizzes Programming projects
Objective 3	
Objective 4	
Objective 5	
Objective 6	

**University of Houston
College of Technology
ELET 2300**

**Introduction to C++ Programming
Syllabus**

Fall 2011

Course: ELET 2300

Instructor: Dr. Heidar Malki

Phone: (713) 743-4075

Email: malki@uh.edu

Office: 304 T2

Textbook: A First Book of C++ by Gary J. Bronson.
My lecture notes will be posted on Webct.

Office Hours: T TH 10:30-11:30, T 4:00-5:00, W 2:00-3:00 or by an appointment

Course Objectives:

- **Students will demonstrate ability to apply sequential thinking to the development of appropriate programming algorithms such as loops, accessing array elements.**
- **Students will be able to utilize mathematical and statistical formulas to produce tables and graphs such as histograms and formatted symbolic outputs**
- **Students will learn to evaluate logical operators, relational operators, and control structures**
- **Students will be able to utilize the latest software in solving the real-world problems by writing structured C++ code**
- **Students will demonstrate the ability to use of mathematical functions such as factorial, quadratic, exponential functions to create tabulated output**
- **Students will be able to recognize the limit of various variable declarations, arrays size, memory space, and limitations of built in mathematical functions in C++ libraries.**

Course Outline

UNIT I. The Fundamental of C++ Programming (two weeks)

A. C++ programming basics

- B. Data types, Declarations, and Displays
- C. Declaring variables
- D. Shortcut arithmetic operators

UNIT II. Interactive Input and Formatted Output (one week)

- A. cout
- B. cin
- C. Assignment operator
- D. Incrementing and decrementing operators
- E. Mathematical library functions
- F. Casts
- G. The const qualifier
- H. Formatted output
- I. Void functions

UNIT III. Control Structures (two weeks)

- A. Relational expressions
- B. Logical operators
- C. If statement
- D. If else statement
- E. Compound statement
- F. Nested if statement
- G. The switch structure
- H. While loop
- I. Do while loop
- J. For loop
- K. Which loop to choose?

Exam 1: TBD

UNIT IV. Functions and Variable Scopes (two weeks)

- A. Function prototypes
- B. Function definition
- C. Function call
- D. Function templates
- E. Inline functions
- F. References and references parameters
- G. Storage classes
- H. Local variables
- I. Global variables
- J. Static variables
- K. Scope resolution operator
- L. Scope resolution operator (::)
- M. Recursion
- N. Generating random numbers
- O. Scaling random numbers

UNIT V. Arrays (two weeks)

- A. One-dimensional arrays
- B. Declaring arrays
- C. Passing arrays to functions
- D. Two-dimensional arrays
- E. Initializing two-dimensional arrays
- F. Bubble sort
- G. String fundamentals
- H. String input and output functions

Exam 2: TBD

UNIT VI. Pointers

- A. Pointer variable declaration and initialization
- B. Pointer operations
- C. References and pointers
- D. Array name and pointers
- E. Pointer expressions and pointer arithmetic
- F. String functions
- G. Advanced pointer notation
- H. Pointer arrays
- I. Initializing arrays of pointers to strings

UNIT VII. File Processing

- A. Files and streams
- B. Creating a sequential access file
- C. Reading data from a sequential access file
- D. Creating random access files
- E. Write () and read ()
- F. Random file access

UNIT VIII: Classes and Data Abstraction

- A. Introduction to classes
- B. Structure definitions
- C. Accessing members of structures
- D. Implementing a user-defined time with a structure
- E. Drawbacks of structures
- F. Implementing a time abstract data type with a class
- G. Class scope and accessing class members
- H. Separating interface from implementation
- I. Constructors
- J. Destructors
- K. Example: Constructing an elevator object

Final Exam: TBD

Assessments:

The objectives of the course will be measured by three methods outlined below. In each program, multiple objectives will be tested.

1. Three exams	60%
2. Quizzes	10%
2. Programs	30%
<hr/>	
Total	100

- Make up exams will not be given.
- Late programs will not be accepted.
- You are not allowed to work with other people in writing the programs.

Report:

1. On due date, please submit source file (HW1.CPP) through Webct.
2. Your file MUST BE virus FREE (Zero will be given if your disk is infected).
3. Your program must be fully documented.
4. Your program must run well and catch all illegal inputs as specified.
5. Include all necessary header files that make up your program.

**University of Houston
College of Technology
ELET 2300**

**Introduction to C++ Programming
Syllabus**

Fall 2011

Course: ELET 2300

Instructor: Dr. Heidar Malki

Phone: (713) 743-4075

Email: malki@uh.edu

Office: 304 T2

Textbook: A First Book of C++ by Gary J. Bronson.
My lecture notes will be posted on Webct.

Office Hours: T TH 10:30-11:30, T 4:00-5:00, W 2:00-3:00 or by an appointment

Course Objectives:

- **Students will demonstrate ability to apply sequential thinking to the development of appropriate programming algorithms such as loops, accessing array elements.**
- **Students will be able to utilize mathematical and statistical formulas to produce tables and graphs such as histograms and formatted symbolic outputs**
- **Students will learn to evaluate logical operators, relational operators, and control structures**
- **Students will be able to utilize the latest software in solving the real-world problems by writing structured C++ code**
- **Students will demonstrate the ability to use of mathematical functions such as factorial, quadratic, exponential functions to create tabulated output**
- **Students will be able to recognize the limit of various variable declarations, arrays size, memory space, and limitations of built in mathematical functions in C++ libraries.**

Course Outline

UNIT I. The Fundamental of C++ Programming (two weeks)

A. C++ programming basics

- B. Data types, Declarations, and Displays
- C. Declaring variables
- D. Shortcut arithmetic operators

UNIT II. Interactive Input and Formatted Output (one week)

- A. cout
- B. cin
- C. Assignment operator
- D. Incrementing and decrementing operators
- E. Mathematical library functions
- F. Casts
- G. The const qualifier
- H. Formatted output
- I. Void functions

UNIT III. Control Structures (two weeks)

- A. Relational expressions
- B. Logical operators
- C. If statement
- D. If else statement
- E. Compound statement
- F. Nested if statement
- G. The switch structure
- H. While loop
- I. Do while loop
- J. For loop
- K. Which loop to choose?

Exam 1: TBD

UNIT IV. Functions and Variable Scopes (two weeks)

- A. Function prototypes
- B. Function definition
- C. Function call
- D. Function templates
- E. Inline functions
- F. References and references parameters
- G. Storage classes
- H. Local variables
- I. Global variables
- J. Static variables
- K. Scope resolution operator
- L. Scope resolution operator (::)
- M. Recursion
- N. Generating random numbers
- O. Scaling random numbers

UNIT V. Arrays (two weeks)

- A. One-dimensional arrays
- B. Declaring arrays
- C. Passing arrays to functions
- D. Two-dimensional arrays
- E. Initializing two-dimensional arrays
- F. Bubble sort
- G. String fundamentals
- H. String input and output functions

Exam 2: TBD

UNIT VI. Pointers

- A. Pointer variable declaration and initialization
- B. Pointer operations
- C. References and pointers
- D. Array name and pointers
- E. Pointer expressions and pointer arithmetic
- F. String functions
- G. Advanced pointer notation
- H. Pointer arrays
- I. Initializing arrays of pointers to strings

UNIT VII. File Processing

- A. Files and streams
- B. Creating a sequential access file
- C. Reading data from a sequential access file
- D. Creating random access files
- E. Write () and read ()
- F. Random file access

UNIT VIII: Classes and Data Abstraction

- A. Introduction to classes
- B. Structure definitions
- C. Accessing members of structures
- D. Implementing a user-defined time with a structure
- E. Drawbacks of structures
- F. Implementing a time abstract data type with a class
- G. Class scope and accessing class members
- H. Separating interface from implementation
- I. Constructors
- J. Destructors
- K. Example: Constructing an elevator object

Final Exam: TBD

Assessments:

The objectives of the course will be measured by three methods outlined below. In each program, multiple objectives will be tested.

1. Three exams	60%
2. Quizzes	10%
2. Programs	30%
<hr/>	
Total	100

- Make up exams will not be given.
- Late programs will not be accepted.
- You are not allowed to work with other people in writing the programs.

Report:

1. On due date, please submit source file (HW1.CPP) through Webct.
2. Your file MUST BE virus FREE (Zero will be given if your disk is infected).
3. Your program must be fully documented.
4. Your program must run well and catch all illegal inputs as specified.
5. Include all necessary header files that make up your program.