Montgomery County Community College
EGR 111
Engineering Computations
3-2-2

CATALOG DESCRIPTION:
In engineering environments, it is critical to recognize when and how a computer can assist in the analysis of a problem. This course will introduce the key concepts of good programming practice and show how computer programming directly relates to solving engineering problems. Multiple programming environments are used, each one emphasizing the same core concepts, yet customized to support a specific category of engineering problem application. This course is subject to a course fee. Refer to http://mc3.edu/adm-fin-aid/paying/tuition/course-fees for current rates.

REQUISITES:
Previous Course Requirements
- MAT 100 Intermediate Algebra or MAT 100B Intermediate Algebra & Review

Concurrent Course Requirements:
None

LEARNING OUTCOMES Upon successful completion of this course, the student will be able to:	LEARNING ACTIVITIES	EVALUATION METHODS
1. Identify fundamental hardware and software components of modern engineering computing systems.	Lecture Demonstration and Practice Case Studies Student Projects	Student Projects
2. Apply the key stages of development of a computerized solution to an engineering problem- development of problem statement, algorithm design, testing plan, coding, debugging, user testing.	Lecture Demonstration and Practice Case Studies Student Projects	Student Projects

LEARNING OUTCOMES	LEARNING ACTIVITIES	EVALUATION METHODS
3. Write code in multiple programming environments to repeatedly emphasize the core ideas in a number of different settings.	Lecture Demonstration and Practice Case Studies Student Projects	Student Projects
4. Recognize the implications of modular design in software both in the use and design of custom libraries and in the context of object-oriented programming.	Lecture Demonstration and Practice Case Studies Student Projects	Student Projects
5. Apply techniques for algorithm development in both individual and group settings.	Lecture Demonstration and Practice Case Studies Student Projects	Student Projects
6. Apply to tasks the common components of any programming language- keywords, data types, operators, binary decisions, looping constructs, arrays and support for modularity.	Lecture Demonstration and Practice Case Studies Student Projects	Student Projects
7. Test and repair errors in programming logic and syntax.	Lecture Demonstration and Practice Case Studies Student Projects	Student Projects

LEARNING OUTCOMES	LEARNING ACTIVITIES	EVALUATION METHODS
8. Design programming solutions to engineering problems in three different application areas-array processing, robotics and numerical analysis.	Lecture Demonstration and Practice Case Studies Student Projects	Student Projects

At the conclusion of each semester/session, assessment of the learning outcomes will be completed by course faculty using the listed evaluation method(s). Aggregated results will be submitted to the Associate Vice President of Academic Affairs. The benchmark for each learning outcome is that *70% of students will meet or exceed outcome criteria*.

SEQUENCE OF TOPICS:

I. Introduction- Overview of hardware and software for modern computing systems
- System unit
- Ports
- Peripherals
- CPU operation

2. Introduction to Programming
- How does programming help an engineer?
- When does an engineer use programming?
- What is the program development life cycle?
- What are the common components of any programming language?
- Language control constructs
- Student presentations of final developed solutions

3. Environment 1 – C++
- Using a popular programming language, students will solve engineering problems in numerical analysis.
- Problem statement creation, including testing plan design
- Algorithm analysis
- Effective use of user-defined programming functions
- Data types and operators
- Language control constructs

4. Environment 2 - RobotC
- Using a machine control interface programming environment, students will solve engineering problems in robotics.
- Problem statement creation, including testing plan design
- Algorithm analysis
- Effective use of machine control programming and parallel task

processing
- Data types and operators

5. Environment 3 – C++
   - Using a popular object-oriented GUI development programming language, students will solve engineering problems in data acquisition and feedback control.
   - Problem statement creation, including testing plan design
   - Algorithm analysis
   - Effective use of GUI programming functions
   - Data types and operators
   - Language control constructs
   - Student – student presentations of final developed solutions

LEARNING MATERIALS:
C++, EXCEL, MATLAB & Basic Engineering Numerical Methods; Pearson
ISBN 978-0-13-612024-7; 2009

Instructor Handouts, text references (engineering library)

Other materials may be required and may be made available directly to the student or via the College's library reserve or its computer network.

COURSE APPROVAL:

Prepared by: H. Thomas Tucker, Jr.		Date: 10/10/2005
VPAA/Provost Compliance Verification:	Dr. John C. Flynn, Jr.	Date: 11/2009
Revised by: William H. Brownlowe		Date: 9/24/2013
VPAA/Provost or designee Compliance Verification: Victoria L. Bastecki-Perez, Ed.D.		Date: 12/3/2013
Revised by: Chengyang Wang		Date: 12/27/2017
VPAA/Provost or designee Compliance Verification:		Date: 1/10/2018

This course is consistent with Montgomery County Community College's mission. It was developed, approved and will be delivered in full compliance with the policies and procedures established by the College.