ECE 118 Introduction to Programming 3 credits Required for EE and CE

Contact hours: Three 50 minute lectures per week OR two 75 minute lectures per week

Course Instructor or Coordinator: Stephen Murrell

21st September 2017

Textbook: C++ programming: Program design including data structures, D. S. Malik, Course Technology inc., ISBN 1337102083, 2017

Other supplementary material:

a. Class web site, http://rabbit.eng.miami.edu/class/een118

Special for fall 2017:

Last class day: 19th December (sections R and RC) Final examination: 12th December (sections R and RC)

2017-2018 University of Miami Academic Bulletin Description: Introduction to computing, problem solving, program design, C++ language fundamentals, and software engineering principles. Software design projects are included. **Prerequisites or co-requisites:** None

Specific outcomes of instruction: The student will:

- 1. Understand the fundamental concepts of computer systems.
- 2. Know and understand the fundamentals of programming, algorithms, data, and software engineering.
- 3. Be able to program in C++.
- 4. Have hands-on experience in problem solving and software design.

Topics

- 1. Graphical programming in a windowing environment
- 2. Text-mode programming in a unix environment
- 3. C++: Functions, constants, local declarations, recursive design
- 4. C++: Strings, arrays, objects
- 5. C++: Variables and loops
- 6. C++: Input and output graphical, iostreams, files
- 7. Structured design: blocks, locality, pure functions
- 8. Modular design: abstraction, independence
- 9. Data visualization, interactive graphics
- 10. Simulation and modelling
- 11. Searching and sorting, managing data collections
- 12. Algorithms and specifications
- 13. Timing: function and algorithm speed estimation and analysis
- 14. Data representation, types, declarations, scope

Student outcomes strongly addressed by the course:

- (a) an ability to apply knowledge of mathematics, science, and engineering (3): Students must design and implement programs to solve, simulate, and investigate a variety of problems.
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data (3): Programming assignments require data to be presented clearly with graphics.
- (g) an ability to communicate effectively (4): Most of the student work involves presenting and investigating information graphically.
- (*i*) a recognition of the need for, and an ability to engage in life-long learning (4): An introductory programming course covering algorithms and data structures really makes it clear that there is no end to things to be learned, and that technology never stands still.
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice (4): Assignments make very heavy use of C++ in Visual Studio and unix.

Outcome		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
a	A1	Mathematics	1	2	2													
	A2	Science and Engineering	1	3	1													
b	B1	Conduct experiments	0	0	0	0	3	2	2	1	1	0						
c	C1	Fundamental design capabilities	2	1	1	2												
	C2	System design capabilities	2	1	0	0	1	1	0	1	4	2	0	0				
d	D1	Teamwork	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	E1	Identify, formulate, solve	1	2	1	0	2	0	0	1	3	1	3	1				
f	F1	Professional and Ethical	0	0	0	0	0	0	0	0								
g	G1	Oral Communications	0	0	0	0	0	0	0	0								
	G2	Written communications	0	0	0	0	0	0	0	0	0	0	0					
	G3	Graphical communications	0	0	3	3	3	3	4									
h	H1	Broad education	1	1	1	2	1											
i	I1	Life-long learning	4	4	4	0	0	1	1	1								
j	J1	Contemporary issues	2	0	2	0	0											
k	K1	Analog simulation	0	0	0	0	0											
	K2	Matlab	0	0	0	0	0	0	0	0								
	K3	Computer-aided digital design	0	0	0	0	0	0	0	0								
	K4	Test equipment	0	0	0	0	0	0	0	0								
	K5	Application development	4	2	1	0	0	0										
	K6	Programming Tools	4	4	1	1	1	4	2	0	0	0						
	K7	Development tools	0	0	0	0												

Course contributions to student outcomes