**Engineering Computation II**

**ECE 3822**

1. **Course number and name:** Engineering Computation II - ECE 3822
2. **Credits and contact hours:** 3 Credit Hours
3. **Instructor’s or course coordinator’s name:** Joseph Picone
4. **Text book, title, author, and year:** None
	1. **other supplemental materials**

Lynda.com, StackOverflow.com and many web-based tutorials are used extensively in lieu of a textbook.

1. **Specific course information**
	1. **brief description of the content of the course (catalog description)**

The primary goal for this course is to teach engineers how to solve problems of scale using a variety of computer tools. The three main goals for this course are: (1) introduce students to the hierarchy of software tools (e.g., scripting languages, interpreted languages, compiled languages) used to solve engineering problems; (2) introduce the basics of Python, a scripting language that is a dominant tool in engineering; and (3) introduce Java, object-oriented design, and a number of Java-related software tools that automate testing, documentation and cross-compilation into web applications. A common thread throughout these topics is the decomposition of large-scale problems into smaller problems that can be solved using reusable modules. Good software engineering practices will be stressed throughout the course. The latter part of the course will involve developing a significant computer simulation of a real-world engineering system that involves real data and utilizes both Python and Java.

* 1. **prerequisites or co-requisites** ECE 1111 -or- CIS 1057 and ENGR 2011
	2. **indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program** Elective
1. **Specific goals for the course**
	1. **specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.**
2. Modern multitasking operating systems. (PI 1.2)
3. Distributed computing. (PI 1.2)
4. Compiled languages and integrated development environments. (PI 1.1)
5. Scripting languages. (PI 1.2)
6. Software configuration management and revision control. (PI 1.2)
7. Object-oriented programming design and abstraction. (PI 1.2)
8. Relational and non-relational databases. (PI 1.2)
9. Dev Ops and other cloud-based software development technologies. (PI 1.1)
10. Computational and visualization challenges in big data. (PI 1.1)
	1. **explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.**

**SO (1)** An ability to identify, formulate and solve complex problems by applying principles of engineering, science and mathematics.

**PI (1.1)** Apply principles of engineering, science, and mathematics to formulate, model, analyze, and solve complex electrical engineering problems.

**PI (1.2)** Apply computational tools, design tools, and programming to solve complex electrical engineering problems.

1. **Brief list of topics to be covered**
2. The login environment, shells and file systems. (CLOs 1, 2)
3. Command line and regular expressions (CLOs 1, 3)
4. Processes, multitasking, multithreading and remote logins. (CLO 2)
5. Integrated Development Environments. (CLO 3)
6. Python Programming. (CLO 4)
7. Github and other revision control strategies. (CLO 5)
8. Object-oriented design, implementation and testing. (CLO 6)
9. Data structures and algorithms. (CLOs 3, 7, 9)
10. Git software revision control and management. (CLO 5)
11. Relational databases (mySQL). (CLO 7)
12. Non-relational databases and object-oriented stores (MongoDB). (CLO 7)
13. Big data computational and visualization challenges. (CLO 9)
14. Dev Ops and cloud-based development environments. (CLO 8)