SPRINGFIELD TECHNICAL COMMUNITY COLLEGE

ACADEMIC AFFAIRS

		Class/Lect.		Lab								
Course Number:	CSE-160	Hours:	3	Hours:	0	Credits:	3	Dept.:	CSE			
Course Title:	Introduction to Programming Using Python					Sen	nester:	Fall		Year:	2015	Y ₁ =

Course Description, Prerequisite, Corequisite:

In this course the student will learn the fundamentals of programming using the Python language. Python is used as a scripting language on Linux and as a scripting language for some applications that run on Linux. CSE-160 is also suitable to those that want some introductory Linux programming experience. This course is intended as an introduction to the core principles of programming. CSE-160 is recommended as a first course in programming for novice programmers before attempting C, C++, and Java. CSE-160 concentrates on basic principles of program design, structured programming, objects, and debugging. Using basic version control and testing for quality control will be covered. Some programs can be run on an embedded system such as the Raspberry Pi to demonstrate the principles surrounding the Internet of Everything (IOE).

Prerequisite: Basic computer experience. CSE-110 recommended but not required.

Three hours of lecture.

Course Number:

CSE-160 Intro To Prog Using Python

Course Objectives	Competencies					
Learn the theory behind modern computer programming.	Be able to place languages and tools into the appropriate problem context. To understand the program translation process. To develop debugging and testing methods. To develop programs using progressive refinement tactics. To properly structure a program.					
Obtain experience with a computer system in a development environment.	Use the Linux and open source development environment to develop programs. Ability to properly analyze debugger output to identify runtime exceptions. Ability to locate technical information on the development libraries, read and interpret the descriptions.					
Learn the theory behind modern computer programming.	Use decomposition to develop a problem specification. Convert the specification into a maintainable program. Use common debuggers and open source tools. Design programs and develop debugging strategies. Perform structure program design. Analyze program flow. Complete and evaluate program documentation.					