Objectives/Competencies

Course Objective	Competencies
1. To develop an understanding of software programming.	1. Use system software and resources to create, edit, save, and retrieve source files, to execute programs, and to obtain hard copy. 2. Understand the difference between an interpreter and a compiler. 3. Interpret generated error messages to debug code. 4. Appropriately apply control structures and the principles of structured programming. 5. Write programs that have well-documented source code (meaningful names, consistent indentation, comments).
2. To investigate common algorithms and techniques used in business programming applications.	1. Understand and use arithmetic, logical, and Boolean operators and operations of the Python programming language. 2. Learn, in depth, the application of sequence, repetition, and selections structures and statements; i.e., If-else if, Switch, While-do, Do-while, For, and Variations of I/O statements 3. Develop and write applications, which produce well-formatted and accurate output.
3. To declare, define and use the Object Oriented Programming concepts of Encapsulation, Inheritance and Polymorphism when writing Python code.	1. Define the scope of variables. 2. Design and implement classes. 3. Define appropriate constructors, fields and methods. 4. Organize a set of classes using inheritance. 5. Distinguish abstract and concrete classes. 6. Write polymorphic methods.
4. To encourage critical thinking skills.	1. Solve problems analytically.

Course Objective	Competencies
	2. Debug programs by resolving run-time and logic errors. 3. Evaluate existing code and revise it for efficiency and clarity. 4. Ability to transfer learned skills and concepts to the study of other programming languages.
5. To familiarize the student with the different Python Integrated Development Environment (IDE) tools.	1. Utilize the following sections of the IDE to efficiently develop Python programs 2. Create new or modify existing projects, forms, code modules and save them to disk. 3. Understand the types of files that make a C#.NET solution. 4. Create projects with multiple source files.
6. Instill in students the requirement to develop programs according to industry standards. Illustrate why the standards are important and how the use of standards actually makes the development of programs easier for the developer.	1. Code all programming assignments utilizing the comprehensive industry standards. 2. The incorporation of coding documentation to promote reuse and ease of maintenance.
7. Instill in the students the fact that the debugger is one of the most crucial skills of programming.	1. Utilize the debugger for every programming assignment. 2. Utilize breakpoints to stop the execution of the program at specific points within the program. 3. Examine the contents of variables after the program has started.