

LABETTE COMMUNITY COLLEGE BRIEF SYLLABUS

SPECIAL NOTE:

This brief syllabus is not intended to be a legal contract. A full syllabus will be distributed to students at the first class session.

TEXT AND SUPPLEMENTARY MATERIALS USED IN THE COURSE (if any):

Please check with the LCC bookstore <http://www.labette.edu/bookstore> for the required texts for this class.

<u>COURSE NUMBER:</u>	COMP 130
<u>COURSE TITLE:</u>	INTRODUCTION TO PROGRAMMING LOGIC
<u>SEMESTER CREDIT HOURS:</u>	3
<u>DEPARTMENT:</u>	Computer Science
<u>DIVISION:</u>	General Education
<u>PREREQUISITE:</u>	None
<u>REVISION DATE:</u>	11/2016

COURSE DESCRIPTION:

This class is an introduction to the program development and design process, including computer-based concepts of problem solving and use of tools such as flowcharts, structure charts, and pseudocode. The following is stressed in this course: basic constructs of programming including structured techniques, modular design, top-down design, coding, and testing.

The objective of this course is to provide not only an introduction to programming, but to lay the foundation for programming courses.

The goal is to develop your problem solving skills so that you will be able to create and understand program logic regardless of the programming language implemented.

COURSE OUTCOMES AND COMPETENCIES:

Students who successfully complete this course will be able to:

1. Describe the relationship between computers and logic.

- Identify and describe computer components and operations.
- Describe the steps involved in the programming process.
- Use and name variables.
- Assign values to variables.

- Explain data hierarchy and types.
- Explain program logic.
- Use programming vocabulary.

2. Describe, document, and follow the steps involved in the programming process.

- Use problem-solving skills to design and document a variety of computer programming problems.
- Use programming design tools.
- Describe and use the three basic programming structures—sequence, selection, and loop.
- Describe Modules, Subroutines, Procedures, and Methods.
- Use the logical comparison operators.
- Explain control break logic.
- Use arrays in program logic.
- Map logic for a modular program.
- Map logic for a structured program.
- Use flowchart diagrams and pseudocode statements.
- Understand the attributes of complete documentation.

3. Describe advanced programming topics.

- Define and discuss procedural and object-oriented programming theories.
- Use advanced modularization techniques.
- Explain event-driven programming.
- Use UML and relational database structuring.
-

4. Describe the use of GUI programming and relational databases.

- Define and discuss procedural and object-oriented programming theories.
- Use advanced modularization techniques.
- Explain event-driven programming.
- Use UML and relational database structuring.