

CIS 110 – Computer Programming Design

Course Description

This course involves extensive work in the development of the logic required in the development of application programs. The course applies the methods of program design and development, using a structured approach. Included in the course will be the learning of the following concepts: proper documentation techniques, sequence, selection, iteration, modules, and arrays. The student will demonstrate a fundamental understanding of these concepts by writing pseudocode and drawing flowcharts as a precursor to the writing of the programs. The demonstration will be made in examinations and in laboratory work.

Instructional Materials

Farrel, J. (2011). *Programming logic and design: Comprehensive*. (6th ed.). Boston: Cengage.

Course Learning Outcomes

1. Demonstrate the use of algorithms and pseudocoding to the problem-solving process.
2. Distinguish among the basic types, steps, and properties of programming.
3. Apply the techniques of functional decomposition, modularization techniques, and debugging strategies into program design.
4. Describe the features and fundamental data structures of programming design.
5. Select and create the appropriate conditional and iteration constructs for a given programming task.
6. Design and write programs using the appropriate data structure and fundamental programming constructs for a given problem.
7. Select and describe relational comparison operators, AND / OR logic and their precedence for a given problem.
8. Describe the use of arrays and subscripts and the steps involved in declaring, initializing, loading, and searching arrays.
9. Demonstrate an understanding of the data hierarchy for files, basic file operations, and sequential file processing.
10. Explain the types and uses of files on permanent storage devices.
11. Describe the process of sorting records and the bubble sort technique.
12. Explain and identify object-oriented concepts.
13. Identify object-oriented classes and also the attributes and methods they contain.
14. Explain the use and benefits of object-oriented programming and event-driven programming.
15. Develop design documents for an interactive event-driven program.
16. Use technology and information resources to research issues in computer programming design.
17. Write clearly and concisely about computer programming design topics using proper writing mechanics and technical style conventions.