

## **CIS 125 INTRO TO PROGRAMMING LOGIC W/L COURSE COMPETENCIES**

1. Examine the components of computer programming
  1. Describe the primitive data types in most computer languages
  2. Demonstrate initialization of variables with the assignment operator
  3. Determine appropriate test data to test a computer algorithm to verify anticipated output
  4. Use relational operators in the basic control structures
  5. Demonstrate the use of logical operators
2. Produce computer algorithms
  1. State the principle of the Structure Theorem
  2. Explain the three basic control structures
  3. Formulate an IPO chart in the development of algorithms
  4. Generate pseudocode to solve computer algorithms
  5. Construct flowcharts to communicate computer algorithms
3. Design computer algorithms that demonstrate appropriate use of the selection control structure
  1. Design an IF-THEN-ELSE selection control structure
  2. Understand the use of a CASE statement
4. Design computer algorithms that demonstrate appropriate use of the repetition control structure
  1. Solve an algorithm with a while loop
  2. Solve an algorithm with a for loop
  3. Solve an algorithm through the use of counters and accumulators
  4. Identify infinite loop conditions
5. Design computer algorithms that perform arithmetic operations
  1. List the order of precedence of arithmetic operators and logical operators
  2. Demonstrate the use of addition, subtraction, multiplication, and division operators in algorithms
6. Design computer algorithms to process arrays
  1. Demonstrate ability to define array structures
  2. Demonstrate the use of subscripts (indexes) are used to address array elements
  3. Perform a linear search of an array
  4. Create two dimensional arrays
7. Create functions/methods to organize programs into manageable code modules
  1. Define and call functions/methods
  2. Use techniques for passing values to and from functions/methods
8. Examine class organization and objects
  1. Differentiate between a class and an object
  2. Give examples of the relationships between classes and objects

9. Transform computer algorithms to a computer programming language using an editor and compiler to enter source code and generate object code
10. Explore the real-world programming environment
  1. Describe the steps in the program development process such as Agile and waterfall methodologies
  2. Discuss relevant development topics such as version control, test driven development, full-stack development, deployment, integration, relational databases, cloud computing