

Computer Programming

Computer Concepts 1

- 1 Describe the impact of software on society 1.1
- 2 Discuss legal/ethical issues related to computer programming and software development 1.2
- 3 Describe the categories and evolution of programming languages 1.3
- 4 Demonstrate an understanding of computer theory (e.g., bits, bytes, binary logic, memory, and storage) 1.4
- 5 Apply the application environment/interface for the specific language being covered (e.g., JCreator, BlueJay, and Visual Studio) 1.5
- 6 Explain the concept of security and its relationship to programming 1.6
- 7 Identify components of the information system model (input, process, output, storage) 1.7

Logical Problem-Solving Skills 2

- 1 Analyze a problem 2.1
- 2 Determine the steps needed to solve a problem 2.2
- 3 Create an algorithm to solve a problem 2.3
- 4 Illustrate the problem solution using a storyboard, flowchart or pseudocode 2.4
- 5 Build a program from a storyboard, flowchart, or pseudocode 2.5
- 6 Explain how to create and integrate reusable component into a program 2.6
- 7 Explain how a program is tested and accepted for release 2.7
- 8 Document code. 2.8

Program Algorithms 3

- 1 Use correct syntax and naming conventions of a given programming language 3.1
- 2 Create a program using internal documentation 3.2
- 3 Identify the use and limitations of different data types (integer, double, and constant) 3.3

4 Create programs that include;;; 3.4

- a Variables, data types, and constants 3.4.A
 - b Counters and accumulators 3.4.B
 - c Arithmetic expressions and assignment statements 3.4.C
 - d User input 3.4.D
 - e Input validation 3.4.E
 - f Boolean expressions 3.4.F
 - g Conditional statement 3.4.G
 - h Iterations 3.4.H
 - i Opening, writing, and reading from a data file 3.4.I
 - j Producing formatted output 3.4.J
 - k Use one or more student created functions that passes data to the function using parameters 3.4.K
 - l Modular programing and implement classes 3.4.L
-

5 Identify type of errors (e.g.syntax, run-time and logic) 3.5

6 Create a program with a standard graphic user interface (GUI) that includes: objects and menus; and a custom GUI 3.6

7 Modify an existing program 3.7

8 Create a program in collaboration with a team 3.8

9 Describe and practice steps of troubleshooting and debugging 3.9

Data Structure Concepts 4**1 Create a program using a list 4.1**

2 Create a program using one- and two-dimensional arrays 4.2

3 Create a program using a sort routine 4.3

4 Create file structures 4.4

5 Describe database structures (e.g., fields, records, files, and tables) 4.5

6 Write code to append, delete, and update a data structure 4.6

7 Write code to search, sort, and query a data structure 4.7

Additional Programming Concepts 5**1 Implement techniques for programming for efficiency (e.g., processing time, programmer time, etc.) 5.1**

2 create a user-friendly program (e.g., user testing, user experience, ect.) 5.2

3 Create programs with event-driven programming 5.3

4 Demonstrate error catching/handling 5.4

5 Compare object-oriented programming with structured programming 5.5

6 Describe considerations for mobile application programming 5.6

7 Identify accessibility considerations in programming 5.7

**Prepare for
Employment** 6

1 Demonstrate working as a team 6.1

2 Identify careers in the information technology field 6.2

3 Demonstrate communication skills 6.3

4 Discuss ethical behaviors in the workplace. 6.4

5 Demonstrate interpersonal skills 6.5

6 Exhibit leadership skills through a student organization (e.g.FBLA, PBL, ACM) 6.6
