

EMC Publishing

Textbook Correlation

State of South Carolina

AN INTRODUCTION TO PROGRAMMING USING MICROSOFT VISUAL BASIC .NET

A. SAFETY

The student will be able to:

1. Identify good work attitudes that affect safety on the job.

Text: Pages 1-14 – 1-16

2. Identify the major causes of work-related accidents.

Text: Page 1-16

3. Demonstrate knowledge of an emergency plan.

Text: Page 1-16

4. Describe the threat of viruses to a computer network, methods of avoiding attacks, and options in dealing with a virus attack.

Text: Pages 1-15 – 1-16, 1-23

Online, www.lvp.com/chapters/intro.pdf: Pages 21-23

5. Identify potential abuse and unethical uses of computers and networks.

Text: Pages 1-14 – 1-16, 1-23

Online, www.lvp.com/chapters/intro.pdf: Pages 21-23

B. STUDENT ORGANIZATIONS

The student will be able to:

1. Explain how related student organizations are integral parts of career and technology courses.

Text: Page 1-23

2. Explain the goals and objectives of related student organizations.

Text: Page 1-23

3. List opportunities available to students through participation in related student organization conferences and other activities.

Text: Page 1-23

4. Explain how participation in career and technology education student organizations can promote lifelong responsibility for community service and professional development.

Text: Page 1-23

Online, www.lvp.com/chapters/intro.pdf: Pages 22-25

C. COMPUTER SYSTEMS

The student will be able to:

1. Define what a computer is and its purpose.

Text: Chapter 1

2. Identify basic hardware and software components.

Text: Pages 1-8 – 1-10

3. Explain the flow of data and instructions through the computer system.

Text: Pages 1-9

4. Define a computer program.

Text: 1-6 – 1-7, 3-1 – 3-8

5. Define input and output.

Text: Pages 1-8 – 1-10, 2-9, 2-12 – 2-19

Online, www.lvp.com/chapters/intro.pdf: Pages 9-11

6. Categorize various types of storage.

Text: Pages 1-9, 1-13, 2-9 – 2-16

Online, www.lvp.com/chapters/intro.pdf: Pages 9-11

D. PROGRAM DOCUMENTATION

The student will be able to:

1. Describe the purpose and value of the program.

Text: 3-18 to 15-24

2. Define the input and source of input for the program.

Text: 3-18 to 15-24

3. Define the output and the destinations (report, database, etc).

Text: 3-18 to 15-24

4. Define variables and constants associated with the program (use descriptive names).

Text: 3-18 to 15-24

5. Describe the scope of the above variables.

Text: 4-8 to 15-24

6. Prepare specified report layout.

Text: 8-3 to 15-24

E. PROGRAMMING PROCEDURES

The student will be able to:

1. List in sequence the steps for developing a program.

Text: Chapter 3, Chapter Reviews and Case Studies

2. Analyze the task and develop an algorithm (pseudocode).

Text: 5-9, Case Studies

3. Design a program using a flowchart.

Text: Chapter Exercises

4. Write the program code from a flowchart or algorithm.

Text: Case Studies and Chapter Exercises

5. Key the program and save it.

Text: Chapter 3

6. Debug the program and verify the output of the program.

Text: 4-12 – 4-17

7. Design test data for use in program testing.

Text: 4-17, Case Studies and Chapter Exercises

8. Run the program to test the logical validity of an application program given appropriate data.

Text: 4-11, Case Studies and Chapter Exercises

F. PROGRAMMING

The student will be able to:

1. Describe the functions.

Text: 4-4 – 4-5, 7-17 - 7-42, 8-3 – 8-4

2. Identify correctly written statements.

Text: 3-1 – 15-24

3. Describe the use of commands.

Text: 3-1 – 15-24

4. Differentiate between commands and statements.

Text: 3-1 – 15-24

5. Write a program that will produce a formatted report.

Text: 3-17, 8-3

6. List and define arithmetic, relational, and logical operators.

Text: 4-1 – 5-36

7. Explain operator precedence.

Text: 3-19 – 3-20, 4-10

8. Write a program that will perform calculations on given data.

Text: Chapter 3 – Chapter 15 Reviews and Exercises

9. Write an interactive program.

Text: Chapter 3 – Chapter 15 Reviews and Exercises

10. Write a program using one or more subroutines/functions.

Text: 7-1 – 7-42

11. Write a program that passes arguments to another subroutine/function.

Text: 7-4 – 7-16, Chapter 7 Exercises

12. Write a program using arrays.

Text: 9-1 to 9-38

13. Write a program using accumulators/summing and counters to total.

Text: Page 5-16, Chapter 5 Reviews and Exercises

14. Write a program using the three decision structures: sequential, selection, and repetition.

Text: 5-1 to 15-24

15. Write a program using sort routines.

Text: Pages 13-1 – 13-18

16. Write a program to produce a report with subtotals and totals.

Text: Pages 4-1 – 13-18

17. Write a program to create a summary report.

Text: Pages 4-1 – 13-18

G. OBJECT-ORIENTED PROGRAMMING

The student will be able to:

1. Describe the concept of OOP (object-oriented programming).

Text: 1-7, 1-23, 3-1, 11-1 – 11-38

2. Describe Class.

Text: 3-1, 3-5 – 3-21, 11-1 – 11-38

3. Describe Objects.

Text: 3-1, 3-5 – 3-21, 11-1 – 11-38

4. Describe Pointers.

Text: 10-17

5. Describe Attributes, and Behaviors.

Text: 3-4 – 3-11, 11-2

H. FILE AND DATA MANAGEMENT

The student will be able to:

1. Describe file-naming conventions.

Text: 2-1 – 2-12

2. Design a file (records and fields) or database.

Text: 12-1 – 12-32

3. Create a file (sequential and random) or database).

Text: 12-1 – 12-32

4. Maintain a file (update, append, delete).

Text: Chapter 2, 12-1 – 12-32

5. Perform search, sort, and query (SQL) functions on a file/database.

Text: 13-1 – 13-18