



Experiential Learning Portfolio for 10152135 Program Logic

Student Contact Information:

Name: _____ Student ID#: _____

Email: _____ Phone: _____

Directions

Consider your prior work, military, volunteer, education, training and/or other life experiences as they relate to each competency and its learning objectives. Courses with competencies that include speeches, oral presentations, or skill demonstrations may require scheduling face-to-face sessions. You can complete all of your work within this document using the same font, following the template format.

1. Complete the Student Contact Information at the top of this page.
2. Write an Introduction to the portfolio. Briefly introduce yourself to the reviewer summarizing your experiences related to this course and your future goals.
3. Complete each "Describe your learning and experience with this competency" section in the space below each competency and its criteria and learning objectives. Focus on the following:
 - What did you learn?
 - How did you learn through your experience?
 - How has that learning impacted your work and/or life?
4. Compile all required and any suggested artifacts (documents and other products that demonstrate learning).
 - Label artifacts as noted in the competency
 - Scan paper artifacts
 - Provide links to video artifacts
 - Attach all artifacts to the end of the portfolio
5. Write a conclusion for your portfolio. Briefly summarize how you have met the competencies.
6. Proofread. Overall appearance, organization, spelling, and grammar will be considered in the review of the portfolio.
7. Complete the Learning Source Table. Provide additional information on the business and industry, military, and/or volunteer experiences, training, and/or education or other prior learning you mentioned in your narrative for each competency on the Learning Source Table at the end of the portfolio. Complete this table as completely and accurately as possible.

The portfolio review process will begin when your completed portfolio and Credit for Prior Learning Form are submitted and nonrefundable processing fees are paid to your local Credit for Prior Learning contact. Contact Student Services for additional information.

Your portfolio will usually be evaluated within two weeks during the academic year; summer months may be an exception. You will receive an e-mail notification regarding the outcome of

the portfolio review from the Credit for Prior Learning contact. NOTE: Submission of a portfolio does not guarantee that credit will be awarded.

You have 6 weeks to appeal any academic decision. See your student handbook for the complete process to appeal.

To receive credit for this course, you must receive “Met” on 6 of the 7 competencies.

10152135 Program Logic, 3 Associate Degree Credits

Course Description: In Program Logic, students learn to develop clear consistent strategies to create computer-programming solutions. Students analyze problems, review requirements, develop test plans, and then create solutions. Students learn to focus on understanding the logic behind each solution. Students also learn the proper use data types, variables and variable scope, as well as, methods, decision structures and repetition structures. Although this course emphasizes programming logic, students develop working computer programs.

Introduction: Briefly introduce yourself to the reviewer summarizing your experiences related to this course and your future goals.

Competency 1: Apply basic building blocks of a computer program

Criteria: Performance will be satisfactory when:

- the computer code is structured properly
- the computer code contains variables, modules, and functions
- the computer code contains correct variable initialization
- the computer code contains correct variable declaration
- the computer code contains proper naming conventions
- the computer code contains proper program documentation procedures
- the computer code contains user input prompts

Learning Objectives:

- a. Explain the Software Development Standards
- b. Explain why development standards are important
- c. Explain why descriptive names are important
- d. Explain the difference between various data types
- e. Explain the importance of initializing variables
- f. Explain how to initialize variables
- g. Explain flowchart symbols

Required Artifacts: None

Suggested Artifacts: Examples of source code. This could be done through screenshot or attached program.

Describe your learning and experience with this competency:

Met/ Not Met Evaluator Feedback:

Competency 2: Construct Modules and Functions

Criteria: Performance will be satisfactory when:

- the computer code includes a Module that has one primary purpose
- the Module is named according to Software Development Standards
- the Module correctly includes arguments
- the Module is properly called
- the computer code includes a Function that has one primary purpose
- the Function is named according to Software Development Standards
- the Function correctly includes arguments
- the Function correctly includes a return type
- the Function is properly called
- the computer code includes mathematical built-in functions
- the computer code include built-in functions for strings

Learning Objectives:

- a. Explain the design, purpose, and usage of modules
- b. Explain the design, purpose, and usage of Function
- c. Explain the design, purpose, and usage of built-in functions

Required Artifacts: None

Suggested Artifacts: Examples of source code. This could be done through screenshot or attached program.

Describe your learning and experience with this competency:

Met/ Not Met Evaluator Feedback:

Competency 3: Use Decision Structures and Boolean Logic

Criteria: Performance will be satisfactory when:

- the computer code includes a Single Alternative Decision Structure
- the computer code includes a Dual Alternative Decision Structure
- the computer code includes a Nested Decision Structure
- the computer code includes a Case Structure
- the computer code includes Logical Operators
- the computer code includes Boolean Operators
- the computer code includes Comparison Operators

Learning Objectives:

- a. Explain the design, purpose, and usage of Decision Structures
- b. Explain the design, purpose, and usage of Logical Operators
- c. Explain the design, purpose, and usage of Boolean Operators

Required Artifacts: None

Suggested Artifacts: Examples of source code. This could be done through screenshot or attached program.

Describe your learning and experience with this competency:

Met/ Not Met Evaluator Feedback:

Competency 4: Use Repetition Structures

Criteria: Performance will be satisfactory when:

- the computer code contains a Do-while loop
- the computer code contains a While loop
- the computer code contains a Do-Until loop
- the computer code contains a Count-Controlled loop
- the computer code contains a For loop
- the computer code does not contains an infinite loop
- the computer code contains Nested loops
- the computer code contains a Sentinel

Learning Objectives:

- a. explain the design, purpose, and usage of condition-controlled loops
- b. explain the design, purpose, and usage of count-controlled loops
- c. Explain infinite loops
- d. explain the design, purpose, and usage of nested loops
- e. explain the design, purpose, and usage of sentinels

Required Artifacts: None

Suggested Artifacts: Examples of source code. This could be done through screenshot or attached program.

Describe your learning and experience with this competency:

Met/ Not Met Evaluator Feedback:

Competency 5: Apply defensive programming techniques

Criteria: Performance will be satisfactory when:

- the computer code validates user input
- the computer code rejects invalid user input
- the computer code uses a validation loop
- the computer code checks for expected data types

Learning Objectives:

- a. Explain defensive programming
- b. Explain validating user input

Required Artifacts: None

Suggested Artifacts: Examples of source code. This could be done through screenshot or attached program.

Describe your learning and experience with this competency:

Met/ Not Met Evaluator Feedback:

Competency 6: Process Text

Criteria: Performance will be satisfactory when:

- the string length is correctly declared
- two or more strings are appended together
- the string is changed to upper case
- the string is changed to lower case
- a substring within a string is identified
- the “contains” function is correctly used
- the “stringToInteger” function is correctly used
- the “stringToReal” function is correctly used
- the “isInteger” expression is correctly used
- the “isReal” expression is correctly used

Learning Objectives:

- a. Explain common string functions
- b. Explain converting strings with built-in functions

Required Artifacts: None

Suggested Artifacts: Examples of source code. This could be done through screenshot or attached program.

Describe your learning and experience with this competency:

Met/ Not Met Evaluator Feedback:

Competency 7: Translate pseudocode into actual source code

Criteria: Performance will be satisfactory when:

- the source code will run without errors
- the source code is structured properly
- the source code contains variables, modules, and functions
- the source code contains correct variable initialization
- the source code contains correct variable declaration
- the source code contains proper naming conventions
- the source code contains proper program documentation procedures

Learning Objectives:

- a. Explain the Software Development Standards
- b. Explain the syntactical differences between pseudocode and source code
- c. Explain how to compile source code
- d. Explain how to run source code

Required Artifacts: None

Suggested Artifacts: Example of both pseudocode and source code. This could be done through screenshot or attached program.

Describe your learning and experience with this competency:

Met/ Not Met Evaluator Feedback:

Conclusion: Summarize how you have met the competencies of the course.

Learning Source Table

[illegible]