Experiential Learning Portfolio for 10152115 Beginning .NET Programming

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 8 of the 9 competencies.

10152115 Beginning .NET Programming, 3 Associate Degree Credits

Course Description: Introduction to the concepts and techniques of programming in the .NET environment using the C# language. Topics covered include requirement analysis, program design, coding, and debugging. Emphasis is placed on the techniques to create functional windows forms. ADO.NET will be used to connect to database.

COREQUISITE: 10152135 Program Logic.

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

Competency 1: Explain the Visual Studio .NET Development Environment

Criteria: Performance will be satisfactory when:

- programs are operational
- programs include documentation
- programs are modularized
- programs use identified standards

Learning Objectives:

a. Explain the unique features of C# programming language
b. Explain the concept of objects
c. Summarize the advantages and disadvantages of objects
d. Explain the tools in the C# Standard Version Toolbox
e. Use the tools in the C# toolbox

Required Artifacts: None

Suggested Artifacts: Windows form application using text boxes, labels, buttons, image lists, list box, and classes. Examples of code showing adherence to coding standards and techniques to reduce complexity

Describe your learning and experience with this competency:

Met/ Not Met Evaluator Feedback:

Competency 2: Create C# programs which contain a Form

Criteria: Performance will be satisfactory when:

- programs contain a Form
- programs are operational
- programs include documentation
- programs are modularized
- programs use identified standards

Learning Objectives:

a. Use forms within a C# program
b. Explain the Forms class
c. Explain the concept of Properties
d. Articulate the concept of properties within objects

Required Artifacts: None
Suggested Artifacts: Example of project/source code and be prepared to explain the source code

Describe your learning and experience with this competency:

Met/ Not Met Evaluator Feedback:

Competency 3: Create C# programs using multiple controls

Criteria: Performance will be satisfactory when:

- programs include multiple controls
- programs are operational
- programs include documentation
- programs are modularized
- programs use identified standards

Learning Objectives:

a. Use command buttons within a C# program
b. Find information with C# Help system as needed
c. Use the following controls: Label, TextBox, CheckBox, ComboBox, ListBox, Timer
d. Use the MsgBox and InputBox controls

Required Artifacts: None
Suggested Artifacts: Example of project/source code and be prepared to explain the source code

Describe your learning and experience with this competency:

Met/ Not Met Evaluator Feedback:

Competency 4: Use constants, enumerated types, global and local variables in programs

Criteria: Performance will be satisfactory when:

- programs include needed constants, global, and/or local variables
- programs are operational
- programs include documentation
- programs are modularized
- programs use identified standards

Learning Objectives:

a. Articulate the definition and use of constants, global variables, and local data within C#
b. Explain the various C#'s variable types
c. Use the various C#'s variable types

Required Artifacts: None
Suggested Artifacts: Example of project/source code and be prepared to explain the source code

Describe your learning and experience with this competency:

Met/ Not Met Evaluator Feedback:

Competency 5: Use IF, THEN, ELSE statements in C# programs
Criteria: Performance will be satisfactory when:

- programs include IF, THEN, ELSE statements
- programs are operational
- programs include documentation
- programs are modularized
- programs use identified standards

Learning Objectives:

a. Define IF, THEN, and ELSE statements
b. Use single-direction decision statements within C#
c. Use multi-direction decision statements within C#
d. Apply decision statements to multiple C# programs

Required Artifacts: None
Suggested Artifacts: Example of project/source code and be prepared to explain the source code

Describe your learning and experience with this competency:

Met/ Not Met Evaluator Feedback:

Competency 6: Use the CASE statement in C# programs

Criteria: Performance will be satisfactory when:

- programs include CASE statements
- programs are operational
- programs include documentation
- programs are modularized
- programs use identified standards

Learning Objectives:

a. Differentiate between IF, THEN, ELSE statements and CASE statements
b. Apply case structures to multiple C# programs

Required Artifacts: None
Suggested Artifacts: Example of project/source code and be prepared to explain the source code

Describe your learning and experience with this competency:

Met/ Not Met Evaluator Feedback:

Competency 7: Use FOR Next Loops in C# programs

Criteria: Performance will be satisfactory when:

- programs include FOR Next Loops
- programs are operational
- programs include documentation
- programs are modularized
- programs use identified standards

Learning Objectives:

a. Define FOR Next loops
b. Determine when FOR Next loops should be used
c. Apply FOR Next loops to multiple C# programs

Required Artifacts: None
Suggested Artifacts: Example of project/source code and be prepared to explain the source code

Describe your learning and experience with this competency:

Met/ Not Met Evaluator Feedback:

Competency 8: Use DO While Loops in C# programs

Criteria: Performance will be satisfactory when:

- programs include DO While Loops
- programs are operational
- programs include documentation
- programs are modularized
- programs use identified standards

Learning Objectives:

a. Define a Pretest DO Loop structure
b. Define a Post-test DO Loop structure
c. Differentiate between FOR Next and DO While Loop structures
d. Apply DO While Loops to multiple C# programs

Required Artifacts: None
Suggested Artifacts: Example of project/source code and be prepared to explain the source code

Describe your learning and experience with this competency:

Met/ Not Met Evaluator Feedback:

Competency 9: Use Arrays in C# programs

Criteria: Performance will be satisfactory when:

- programs include arrays
- programs are operational
- programs include documentation
- programs are modularized
- programs use identified standards

Learning Objectives:

a. Create an initialized one-dimensional array
b. Use specific values in a one-dimensional array
c. Create an initialized two-dimensional array
d. Use specific values in a two-dimensional array

Required Artifacts: None
Suggested Artifacts: Example of project/source code and be prepared to explain the source code

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

Learning Source (name of employer, training, military, volunteer organization, etc.)	**Supervisor**	**Start-End Date**	**Total Hours**	**Related Competencies**
Ex: XYZ Corporation	Bucky Badger	8/2012-9/2014	2000	#1, 2, 3, and 7