

## SELF - ASSESSMENT GUIDE

<b>Qualification:</b>	PROGRAMMING NC IV	
<b>Unit of competency:</b>	Develop object-oriented application	
<b>Units of competency covered:</b>	<ul style="list-style-type: none"> <li>• Design program logic</li> <li>• Apply program-development approach</li> <li>• Apply OOP language skills</li> </ul>	
Instruction: <ul style="list-style-type: none"> <li>• Read each of the questions in the left-hand column of the chart.</li> <li>• Place a check in the appropriate box opposite each question to indicate your answer.</li> </ul>		
<b>Can I?</b>	<b>YES</b>	<b>NO</b>
• Obtain design documentation and review and clarify requirements for the programs.		
• Determine design approach to be taken in coding and the modules and links required		
• Structure diagrams of program flow and modules according to project standards		
• Document program scope and limits according to project standards		
• Document or reference special routines or procedures according to project standards		
• Identify and revise references for tables, files, inputs, outputs, and other program functionalities according to program requirements		
• Use templates as applicable		
• Check program flow, states or conditions for interfaces and compliance to design documentation requirements		
• Gain/Obtain feedback/input from appropriate person as needed		
• Determine program activities and select appropriate program development approach		
• Create an initial plan to guide the program development process		
• Use documentation tool for program development		
• Draw program structure and organization		
• Define and use proper naming conventions		
• Document input and output forms accordingly		

• Document program flow and processes accordingly		
• Identify resources for coding and testing programs		
• Monitor and check program activities against plan		
• Review and document opportunities for improvement, any lessons learned and possible recommendations for future projects		
• Observe basic language syntax rules and best practices		
• Use language data-types, operators and expressions		
• Use appropriate language syntax for sequence, selection and iteration constructs		
• Use modular programming approach		
• Use arrays and arrays of objects		
• Implement class that contains primitive member/instance variables		
• Implement class that contains multiple options for object construction		
• Use class user defined aggregation		
• Implement inheritance to at least 2 levels of depth		
• Use polymorphism at a simple level through inheritance to enable easy code extension		
• Use an integrated development environment, particularly the language debugging facilities		
• Use program debugging techniques to detect and resolve errors		
• Follow guidelines for developing maintainable code adhering to a set of coding standard		
• Follow and use internal documentation standards and tools		
• Develop and conduct simple tests to confirm the coding process meets design specification		
• Document tests performed		
• Make corrections to the code and the documentation as needed		
<b>I agree to undertake assessment in the knowledge that information gathered will only be used for professional development purposes and can only be accessed by concerned assessment personnel and my manager/supervisor.</b>		
<b>Candidate's Signature:</b>	<b>Date:</b>	
<b>Assessor's Signature:</b>	<b>Date:</b>	