

## COMPUTER PROGRAM DEVELOPMENT

**UNIT CODE: IT/CU/ICT/CR/5/5**

### **Relationship to Occupational Standards**

This unit addresses the competency: **Develop computer program**

**Duration of Unit: 340** hours

### **Unit Description:**

This unit specifies competencies required to develop computer program. It involves identifying of programming concepts and approaches, identifying program development methodologies, identifying program design, identifying of programming languages, performing of basic structured programming and performing basic internet programming.

### **Summary of Learning Outcomes:**

1. Identify Programming concepts and approaches
2. Identify program development methodologies
3. Identify Program design
4. Identify computer programming languages
5. Perform Basic structured Programming using C language
6. Perform Basic Internet programming

### **Learning Outcomes, Content and Suggested Assessment Methods**

<b>Learning Outcome</b>	<b>Content</b>	<b>Suggested Assessment Methods</b>
1. Identify Programming concepts and approaches	<ul style="list-style-type: none"><li>• Definition of program and programming</li><li>• Language translators<ul style="list-style-type: none"><li>✓ Compiler</li><li>✓ Interpreter</li><li>✓ Editors</li><li>✓ Linker</li><li>✓ Loader</li></ul></li></ul>	<ul style="list-style-type: none"><li>• Oral questioning</li><li>• Written test</li><li>• Learner portfolio of evidence.</li></ul>

	<ul style="list-style-type: none"> <li>• Types of programming approaches</li> </ul>	
2. Identify program Development methodologies	<ul style="list-style-type: none"> <li>• Description of program specifications</li> <li>• Types of development methodologies <ul style="list-style-type: none"> <li>✓ Agile</li> <li>✓ Crystal</li> <li>✓ Rapid Application Development</li> </ul> </li> <li>• Program development cycle</li> <li>• Styles of programming <ul style="list-style-type: none"> <li>✓ Functional</li> <li>✓ Modular</li> <li>✓ Object oriented</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Observation</li> <li>• Written test</li> </ul>
3. Identify Program design	<ul style="list-style-type: none"> <li>• Define program design</li> <li>• Program Design Approaches <ul style="list-style-type: none"> <li>✓ Top – Down</li> <li>✓ Bottom – Up</li> <li>✓ Data-Driven <ul style="list-style-type: none"> <li>• Program Design Tools</li> </ul> </li> </ul> </li> <li>✓ Pseudo code</li> <li>✓ Decision Tree and tables</li> <li>✓ flow charts</li> </ul>	<ul style="list-style-type: none"> <li>• Oral questioning</li> <li>• Written test</li> </ul>
4. Identify computer programming languages	<ul style="list-style-type: none"> <li>• Define computer programming language</li> <li>• Computer programming languages <ul style="list-style-type: none"> <li>✓ High level</li> <li>✓ Low level</li> <li>✓ 4GL</li> <li>✓ Object Oriented</li> <li>✓ Visual</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Oral questioning</li> </ul>

	<ul style="list-style-type: none"> <li>• Factors to consider when choosing a programming language</li> <li>✓ Language domain match</li> <li>✓ Popularity</li> <li>✓ Project size</li> <li>✓ Tool support</li> <li>✓ Efficiency</li> <li>• Tools for program development</li> <li>✓ Pseudo code</li> <li>✓ flow charts</li> <li>✓ Data flow Diagrams</li> </ul>	
5. Perform Basic structured Programming using C language	<ul style="list-style-type: none"> <li>• C Concepts <ul style="list-style-type: none"> <li>✓ Characteristics</li> <li>✓ Pre-processor directives</li> <li>✓ C headers</li> </ul> </li> <li>• Fundamentals of C programming language <ul style="list-style-type: none"> <li>✓ Input and output statements</li> <li>✓ C key words</li> <li>✓ Variables</li> <li>✓ C operators</li> <li>✓ C Expressions</li> </ul> </li> <li>• Control Structures <ul style="list-style-type: none"> <li>✓ Sequence</li> <li>✓ Selection</li> <li>✓ Iteration</li> </ul> </li> <li>• Sub-programs <ul style="list-style-type: none"> <li>✓ Types</li> <li>✓ Scope of variables</li> <li>✓ Parameter passing</li> </ul> </li> <li>• C program format</li> </ul>	<ul style="list-style-type: none"> <li>• Oral questioning</li> <li>• Written test</li> </ul>
6. Perform Basic Internet programming	<ul style="list-style-type: none"> <li>• Concepts of Internet programming</li> <li>• Web programming approaches <ul style="list-style-type: none"> <li>✓ Server side</li> <li>✓ Client side</li> </ul> </li> <li>• Web programming languages</li> </ul>	<ul style="list-style-type: none"> <li>•</li> </ul>

	<ul style="list-style-type: none"> <li>✓ HTML</li> <li>• Web Programming Interfaces <ul style="list-style-type: none"> <li>✓ Common client interface</li> <li>✓ Common gateway interface</li> </ul> </li> <li>• HTML <ul style="list-style-type: none"> <li>✓ Tags</li> <li>✓ parcelling</li> <li>✓ Coding</li> </ul> </li> </ul>	
--	---	--

### Suggested Methods of Delivery

- Presentations and practical demonstrations by trainer;
- Guided learner activities and research to develop underpinning knowledge;
- Supervised activities and projects in a workshop;

The delivery may also be supplemented and enhanced by the following, if the opportunity allows:

- Visiting lecturer/trainer from the ICT sector;
- Industrial visits.

### Recommended Resources

<p><b>Tools</b></p> <p>Comprehensive set of tools.</p> <ul style="list-style-type: none"> <li>• Flow charts</li> <li>• Data flow diagram</li> <li>• Decision table</li> <li>• Decision tree</li> <li>• Web Authoring tools</li> <li>• Notepad</li> </ul>
<p><b>Equipment</b></p> <ul style="list-style-type: none"> <li>• Computer</li> <li>• Software</li> </ul>
<p><b>Materials and supplies</b></p> <p>Digital instructional material including DVDs and CDs</p>