






Level 4 Certificate in Computer Fundamentals (105) 115 Credits



Unit: QBASIC Programming	Total Qualification Time: 200
Exam Paper No.: 4	Number of Credits: 20
Prerequisites: Basic computing knowledge.	Corequisites: A pass or higher in Certificate in Information Systems or equivalence.
<p>Aim: This unit introduces the concepts of programming using the QBASIC language. The unit covers both theory and practical implementation of computer programming. The unit also analyses computer programming and problem solving in a programming environment including text editor, language translator and the role of the operating system. This unit gives learners an overview of the different components of the computer, different numbering systems and its conversions and problem-solving strategies. A computer is a tool for solving problems with information, hence writing a program without knowing what problem one is trying to solve is the equivalent of swinging a knife without knowing what it is that one wants to cut. Learners are taught basic goals of computer programming, common problems and solutions. Learners analyse how a program is organised, programming statements, use of variables, variable types, operator precedence and programming control structures. Also covered in this procedural high level language is the development of algorithms, flowcharting, documentation, testing and debugging and programming techniques.</p>	
Required Materials: Recommended Learning Resources.	Supplementary Materials: Lecture notes and tutor extra reading recommendations.
<p>Special Requirements: This unit has a required laboratory component. To understand programming concepts; learners should practice writing programs on their own at least 2-3 hours a day.</p>	
<p>Intended Learning Outcomes:</p> <ol style="list-style-type: none"> The QBASIC programming environment, menus and reserved words. The computer numbering systems and converting decimal numbers to binary; binary numbers to decimal; binary numbers to hexadecimal; hexadecimal numbers to binary; hexadecimal numbers to decimal and decimal numbers to hexadecimal. How the arithmetic and logic unit (ALU) performs arithmetic calculations by adding and shifting. The flowchart symbols and how to solve computer problems using flowchart diagrams. How to start Qbasic program; creating 	<p>Assessment Criteria:</p> <ol style="list-style-type: none"> Describe the different programming languages Identify how programs are written Define source code Describe syntax errors Define machine code Explain the layout of a QBASIC program Be able to start, manoeuvre and close QBASIC Produce a simple meaningful program Describe bits, bytes and words Describe memory measurements (bits, bytes, kilobytes, megabytes, gigabytes, terabytes etc.) Define decimal, binary, octal and hexadecimal numbering systems Demonstrate how to convert decimal to binary and vice versa Demonstrate how to convert octal to binary and vice versa Demonstrate how to convert hexadecimal to binary and vice versa Describe the components of the CPU Describe the functions of the ALU Describe how arithmetic calculations are performed Be able to draw CPU diagram Demonstrate using the flowcharting symbols Demonstrate drawing flowchart diagrams Describe flowcharting loops Define dummy values in programming Identify the main elements of the Qbasic program

<p>meaningful programs, executing; saving and retrieving Qbasic programs.</p> <p>6. Computer arithmetic algorithms; the main processing loop with the concept of input, processing and output.</p> <p>7. Understand decision making in programming using IF/THEN and CASE statements.</p> <p>8. Understand how files are created in Qbasic; the process of reading from a file and writing to a file.</p>	<p>screen</p> <p>5.2 Describe rules for defining variables</p> <p>5.3 Describe reserved words</p> <p>5.4 Identify program errors</p> <p>5.5 Define how to debug programs</p> <p>5.6 Describe the PRINT statement</p> <p>5.7 Use the CLS, OPEN and CLOSE statements</p> <p>6.1 Write arithmetic expressions for Qbasic program</p> <p>6.2 Identify how to use comments in programming</p> <p>6.3 Demonstrate how to use DO WHILE/LOOP and DO UNTIL/LOOP</p> <p>6.4 Discuss infinite loops</p> <p>6.5 Describe how to use READ/DATA statements</p> <p>6.6 Define a loop</p> <p>7.1 Evaluate decision-making structures</p> <p>7.2 Define how to write programs using IF/THEN and CASE structures</p> <p>7.3 Discuss the difference between different structures</p> <p>7.4 Explain the nested IF-THEN-ELSE logic structure</p> <p>7.5 Describe how to use the AND, OR and NOT logical operators</p> <p>7.6 Demonstrate how to identify program errors</p> <p>7.7 Demonstrate how to create programs using IF-THEN-ELSE and CASE statements</p> <p>8.1 Discuss file names</p> <p>8.2 Explain the syntax for opening disk files</p> <p>8.3 Describe how data is written (copied) from memory variables to a disk file</p> <p>8.4 Identify implementation of CLOSE, INPUT, EOF statements</p> <p>8.5 Be able to write data to a file</p> <p>8.6 Be able to read data from a file</p>
<p>Methods of Evaluation: A 2-hour written examination paper with Section A and Section B. Section A has 40 multiple choice questions. Section B has three essay questions, each carrying 20 marks. Candidates are required to answer all questions. Candidates also undertake project/coursework in QBasic Programming with a weighting of 100%.</p>	

Recommended Learning Resources: QBasic Programming

Text Books	<ul style="list-style-type: none"> • Qbasic by Example by Greg M. Perry. ISBN-10: 1565294394 • Programming in QBASIC for Engineering Technology by Kenneth Craven. • Easy Programming With Qbasic by Tory Stephen Toupin. • Qbasic Programming (Peter Norton Programming Series) by David I. Schneider. ISBN-10: 0136630227
Study Manuals 	BCE produced study packs
CD ROM 	Power-point slides
Software 	QBasic