Diploma in System Design (401) 135 Credits

Unit: Pascal Programming **Exam Paper No.:** 4	**Guided Learning Hours:** 240 **Number of Credits:** 24
Prerequisites: QBasic Programming knowledge	**Corequisites:** A pass or better in Certificate in Computer Fundamentals or equivalence.
Aim: Pascal contains some significant language features that allows it to be used as a powerful learning tool in introducing structured programming techniques to learners including: Built in Data Types - Pascal contains it's own built in data types of Integer, Real, Character, and Boolean; user defined Data Types - has the ability to define scalar types as well as subranges of those data types; a defined set of Data Structures - these data structures include Arrays, Records, Files and Sets; a strong data typing element - Pascal compliers can diagnose an incompatible assignment of one type to a variable to another type; Support for Structured Programming - this is accomplished through the use of subprograms called procedures and functions; simplicity and expressivity - because the language is simple and expressive in nature, it allows for effective teaching of computer programming techniques. This unit is designed to give the learners an introduction to Pascal Programming which will be referenced later in C Programming. Learners will utilise the commands, statements, and procedures of this language to develop computer programs. On completion of this unit, learners will be able to describe the syntax rules governing expressions and statements in Pascal.	
Required Materials: Recommended learning resources.	**Supplementary Materials:** Lecture notes and tutor extra reading recommendations.
Special Requirements: Building Pascal programming applications. This is a hands-on course, hence use of the computers is mandatory.	

Intended Learning Outcomes:	**Assessment Criteria:**
1. Design simple Pascal programs using variables, constant assignment statements and comments.	1.1 Describe structured programming 1.2 Describe how Pascal programs begin and end 1.3 Demonstrate the difference between ***Write*** and ***Writeln*** 1.4 Define how to display blank lines on the screen 1.5 Explain the different datatypes 1.6 Demonstrate how to get user's input 1.7 Describe the difference between ***Read*** and ***Readln*** 1.8 Demonstrate how to view and pause the screen after the program ran 1.9 Demonstrate how to declare variables in Pascal and be able to define variables 1.10 Be able to produce a basic program
2. Reserved words in Pascal and illustrating the use, meaning and purpose of reserved words.	2.1 Describe ***Clrscr*** 2.2 Define ***GotoXY(,)*** 2.3 Define ***Textbackground()*** 2.4 Define ***Textcolor()*** 2.5 Define ***Readkey*** 2.6 Define ***Delay()*** 2.7 Define ***Halt / Halt()***
3. Understand the concept of 'Program Control' by developing effective programs using the IF statement (decision making), FOR Loop and the REPEAT…UNTIL loop.	3.1 Demonstrate the **IF** statement (***IF…Then …Else, Nested IF*** statements) 3.2 Demonstrate the ***Repeat…Until Loop*** 3.3 Demonstrate the ***FOR Loop*** (Nested

		FOR loop)
	3.4	Demonstrate the ***WHILE…DO*** loop
	3.5	Explain the operation of the IF statement
	3.6	Explain the syntax of conditional statements
	3.7	Explain how to use **Nested IF** statements
	3.8	Use **Flowchart** diagrams to express IF statement
	3.9	Explain the characteristics of all loop syntaxes in Pascal and the differences between them
4. Ways of writing conditional statements using CASE …OF statement and the ability to diversify programming skills.	4.1	Define how to design simple *IF…ELSE* statement using *CASE…OF*
	4.2	Demonstrate how to design simple Menu programs using CASE…OF
	4.3	Describe how to design complex programs using CASE…OF
5. The use of logical operators; Boolean expressions and sample Pascal programs using logical expressions.	5.1	Define logical operators and how they can be used in Pascal programming
	5.2	Define the types of logical operators (AND, OR, NOT)
	5.3	Define Boolean expressions and how do they differ from logical operators.
	5.4	Create sample programs using logical operators
6. Understand the purpose of procedures; functions and recognise how large programs are broken down into small programs by using procedures and functions.	6.1	Explain the syntax of procedures and functions
	6.2	Explain the differences between procedures and functions
	6.3	Explain the differences between **pass by value** and **pass by reference**
	6.4	Explain the differences between global variable and local variable
	6.5	Explain the rule of calling procedures / functions.
7. Understand how files are implemented in Pascal programming including how to read, create and write from a file; append data to an existing file and delete files.	7.1	Describe how to create and write to a file
	7.2	Demonstrate how to read from a file
	7.3	Demonstrate how to write to a file
	7.4	Demonstrate how to append data to an existing file
	7.5	Demonstrate how to delete files
	7.6	Demonstrate how to create and remove sub-directory
	7.7	Illustrate how to find the size of a file using **FileSize()**
8. Understand the use and power of data structures in programming by analysing static and dynamic data in programming.	8.1	Describe the purpose of arrays
	8.2	Describe array declarations
	8.3	Understand how to design programs using arrays
	8.4	Describe the use of an array and the different arrays available
9. The concept of records as special types of data structures, differentiating records from arrays and how records collect different types of	9.1	Define records using the TYPE definition
	9.2	Define the WITH keyword

data.	9.3 Identify the use of array of records 9.4 Demonstrate how to design a database, which allows a user to: add new record, edit an existing record, view a particular record in a list, delete record and sort records
10. The process of producing a full working Pascal program by combining and outlining how all the modules of each process should be gathered.	10.1 Identify programming syntax errors and separate processes 10.2 Evaluate how to list the solution steps of each process and data requirements 10.3 Demonstrate how to determine the output requirements and construct an algorithm to do the process 10.4 Explore how to use creativity to expand the input, process and output 10.5 Demonstrate how to implement a program, step by step 10.6 Demonstrate how to generalise the input procedure to fit in each module to save energy and time by combining them using menus.
Methods of Evaluation: A 2½-hour essay written examination paper with 5 questions, each carrying 20 marks. Candidates are required to answer all questions. Candidates also undertake project/coursework in Pascal Programming with a weighting of 100%.	

Recommended Learning Resources: Pascal Programming

Text Books	• Fundamentals of Pascal,Understanding Programming and Problem Solving (Hardcover) by Douglas W. Nance. ISBN-10: 0314205543 • Schaum's Outline of Theory and Problems of Programming With Pascal (Paperback) by Byron S. Gottfried. ISBN-10: 0070238499 • Turbo Pascal: Programming and Problem Solving (Paperback) by Sanford Leestma and Larry Nyhoff. ISBN-10: 0023694114
Study Manuals 	BCE produced study packs
CD ROM 	Power-point slides
Software 	Pascal Programming