Form 5

Term 1

SECTION 6: DATABASE MANAGEMENT

1	explain the concept of a database	Definition of database: Purpose of database.	
	use terminology commonly associated with a database;	Database terminology: table, row (record), column (field), primary key, secondary key, candidate key, foreign key. Data types: numeric; text; logical; date /time; currency.	
	create a database;	Create and Modify a table structure: adding new fields, deleting fields, changing field definitions. Establish primary keys. Establish relationships: show the joins between tables (one-to-one and one-to-many).	
4	manipulate data in a database	(a) Forms: (i) Use of form wizard only; (ii) select suitable fields; (iii) use of sub-form.	
		(b) Queries: (i) more than one criterion; (ii) use of select; (iii) use of calculated field; (iv) two or more fields involving the use of relational and logical operators.	
		(c) Reports: (i) use of report wizard; (ii) use of sorting, grouping, statistical and summary features, for example, count, sum, and average; (iii) report generated to screen, printer and file;, (iv) renaming of report title.	

SECTION 7: PROBLEM-SOLVING AND PROGRAM DESIGN

5	outline the steps in problem-solving; use the divide-and-conquer approach define a problem	Steps in problem-solving: Basic treatment of the structured approach for solving complex problems. The components are: input; process; and output. A defining diagram (IPO Chart) may be used to delineate the components.	
	explain the concept of algorithms	Definition of algorithms.	
6	distinguish between variables and constants	Variables as an area of storage whose value can change during processing; the value of a constant never changes. Data types: integers, floating point (real), characters, Boolean, string.	
	represent algorithms in the form of flowchart and pseudocode;	Use of flowchart symbols: input/output, process, decision, directional arrows, start/stop. Pseudocode – Use of read, input, store, write, print, output, display, conditional branching (if-then, if-then-else, nested conditions); loops (for, while, repeat). Use of relational operators: <, >, =, <=, > =,< >. Logical operators: AND, OR, NOT; use of truth tables. Arithmetic operators: +, –, *, /, MOD, DIV.	
	test algorithms for correctness	Desk checks/dry run: construction and use of trace tables to verify results	

SECTION 8: PROGRAM IMPLEMENTATION

7	distinguish between low-level and high level programming languages;	Low-level language (Machine or Assembly) High-level language (For example, Visual Basic, Pascal, C).	
8	describe the sequence of steps in implementing a program;	Steps in implementing a program:	
9	perform checks and tests on programs to verify correctness;	Errors: syntax, logic, runtime. Testing (test data). Debugging techniques.	
10	declare variables and constants using elementary data types;	Data types: integer, real/double/float, character, string and Boolean/logical.	
11	translate algorithmic statements into high-level language syntax;	Assignment statements; input/output operations using standard input/output (reading data entered via keyboard, displaying data on monitor). Syntax for arithmetic, logic and relational operators. Syntax for conditional branching (for example, if-then, if-then-else, nested if-then-else or case). Syntax for Iteration (Loops): for, while, repeat.	
	effectively document programs	Importance of documentation. Features of internal documentation (use of mnemonic, variable names, use of comments, indentation, effective use of white space). Features of external documentation (user manual).	

Term 2

1-10	Revision and Exam Prep	Revision	
		Past Papers	