GCSE Computer Science Checklist

Unit 3 - Software Development
(Exam Condition Controlled Assessment: 20 hour - 80 Marks - 20% of Qualification)

Topic	Sub-Topic	Explanation	I can statement	Studied	R	A	G
Scope of the Problem	Analysis	Use a systematic approach to problem solving including the use of decomposition and abstraction.	I can break a complex problem into smaller component parts.				
			I can remove unnecessary detail from a given scenario.				
			I can simplify a given scenario.				
	Data Requirements	Identify the data requirements to create an effective solution.	I can list the fields that are needed to store data a solution requires.				
			I can identify appropriate data types for fields				
	Process Requirements	Identify the processing to be carried out by the solution.	I can explain the processes a solution will need to perform				
	Input and Output Requirements	Identify the required inputs and outputs required for the solution.	I can identify the inputs required to create a solution to a given problem				
			I can identify the outputs required to create a solution to a given problem				
	Objectives	Produce a detailed set of objectives, that are measurable, that define clearly the tasks required to create an effective and fully functional solution.	I can clearly explain appropriate measureable objectives for a solution based upon the tasks an effective, fully functional solution would perform.				
Design	Design	Produced a comprehensive design that would allow a competent third party to create a solution that covers all stated objectives.	I can produce comprehensive designs for all parts of the solution that tie to the objectives for the project.				
	Input and Outputs Facility's	Identified fully and described in detail the input and output facilities to be provided by the user interface which will be fit for purpose.	I can fully identify and describe the input facilities to be used in the solution.				
			I can explain how input facilities will be fit for purpose.				
			I can fully identify and describe the output facilities to be used in the solution.				

			I can explain how output facilities will be fit for purpose.				
	Data Structures	Described all data structures required to create an effective solution, using correct technical terminology.	I can produce data structures based off the data requirements for the solution.				
			I can identify appropriate data types for data within a data structure.				
	Validation	Described fully the validation routines required to ensure that only appropriate data can be entered into the solution.	I can identify validation routines that will be needed for each input within the system.				
	Authentication	Considered fully the need for authentication routines.	I can identify the authentication routines that will be included within the system.				
	Processing Stage	Described all data handling and processing routines for an effective solution as algorithms, using a standard convention such as pseudo code or flowcharts	I can produce clear and well-structured pseudo code or flowcharts to outline all procedures that will take place within the final solution.				
Development	**Creation**	Produce windows based forms utilising integrated objects.	I can create and save a new Visual Basic .Net (VB.Net) windows based application using Visual Studio.				
			I can add new windows forms into my application.				
			I can add Labels onto my windows forms				
			I can add TextBoxes onto my windows forms				
			I can add Buttons onto my windows forms				
			I can add ListBoxes onto my windows forms				
			I can add CheckBoxes onto my windows forms				
			I can add ComboBoxes onto my windows forms				
			I can add DataTimePickers onto my windows forms				
			I can add DataGridViews onto my windows forms				
			I can format all object to set; font size, type face, foreground and background colour.				
		Produce modules to reduce code reputation.	I can add new modules into my application				

		Name forms, modules and objects appropriately.	I can name all objects appropriately using a suitable naming system.				
			I can name all windows forms appropriately using suitable naming system.				
			I can name all modules appropriately using suitable naming system.				
		Produce efficient code	I can create local variables.				
			I can create global variables.				
			I can write code to receive information input by a user.				
			I can write code to compare data and/or variables.				
			I can write code to add, subtract, multiply and divide numeric values.				
			I can write code to open another form.				
			I can write code to hide a form.				
			I can write code to close a form.				
			I can write conditional statements (IF Statements).				
			I can write for loops.				
			I can write until/while loops.				
			I can code a data structure.				
			I can code public functions.				
			I can code public subroutines				
			I can write try, catch exceptions to catch file errors.				
			I can code a FilePut to store/overwrite information into a dat file.				
			I can code a FileGet to retrieve information from a dat file.				
			I can code a hashing algorithm to calculate record positions.				
			I can code a type check.				
			I can code a range check.				
			I can code a length check.				
			I can code a format check.				

			I can code a presence check.				
		Produce well-structured, self-documenting code.	I can structure my code appropriately using indents and spacing.				
			I can use self-documenting names for variables, subroutines and functions.				
		Debugging and error finding	I can execute my code				
			I can identify errors that stop my code from compiling and overcome the issues.				
			I can use breakpoints within my code to help with error checking and diagnostic.				
			I can use step through to run each line of code and help with error checking and diagnostic.				
			I can use variable watch to view the data a variable contains and help with error checking and diagnostic.				
Refinement Log	**Approach to Software Development**	Demonstrated a structured approach to developing the solution.	I can approach the development of my solution in a structured well organised manor.				
	Ordered and Logical Approach	Carried out activities in an appropriate order.	I can carry out the project in the correct order without moving back and forth between sections.				
	Development Progress	Evaluated effectively the progress made in each session.	I can document the progress that I make in each session.				
	Development Issues	Provided a full description of any problems encountered with good use of technical terminology	I can document the problems I encounter using appropriate technical vocabulary.				
	Design Alterations	Justified any changes that have been made to the original design demonstrating an informed understanding of the need for change.	I can identify issues during development that will require me to alter my original designs.				
			I can alter my original designs to overcome development issues.				
	Development Planning	Produced logical and prioritised actions for subsequent sessions.	I can plan tasks that need completing within my next sessions based off what I complete in each session.				
Effectiveness	**Functionality**	The solution achieves all the requirements of the given scenario.	I can explain how my solution archives all the requirements of the scenario.				
		The solution is efficient in use of resources.	I can explain how my solution is efficient.				

	User Interface	The solution is usable with a user interface that is intuitive and fit for audience and purpose.	I can explain how my user interface is intuitive.				
			I can explain how my user interface is fit for the audience and purpose.				
	Modular	The solution is well-structured and modular in nature.	I can demonstrate my code is well-structured.				
			I can demonstrate my code is modular in nature.				
	Authentication	The solution is secure with effective authentication routines.	I can demonstrate my solution can only be accessed by valid users.				
	Reliability and Robustness	The solution is reliable and robust.	I can explain how my solution is reliable and robust.				
Technical Quality	**Source Code**	The code for the solution is self-documenting, well-structured and modular in nature.	I can write a solution with self-documenting code.				
			I can write a solution that is well-structured				
			I can write a solution that is modular in nature.				
		The code uses a consistent programming style throughout, including indentation and the use of white space around operators and keywords.	I can write code that is correctly indented.				
			I can write code that is laid out with white space to make it easy to read.				
			I can write code using key words and operators.				
		The code has made full use of meaningful identifiers and appropriate use of constants.	I can write code and use meaningful identifiers throughout.				
			I can write code which uses appropriate constants.				
		Subroutines have been created.	I can create my own subroutines and functions.				
		The solution has a well-defined interface.	I can create a well-defined interface.				
	Variables	The code made effective use of local variables and minimised the use of global variables.	I can write code that uses mainly local variables.				
			I can write code that avoids the use of global variables.				
	Validation Routines	The solution has effective validation routines and created routines for exception handling.	I can write code that uses a range of validation techniques.				
			I can write code that allows for exception handling.				

	Annotated Code	The solution includes informed annotation of the code throughout.	I can fully annotate my code.				
Test Strategy	**Provide a Test strategy**	Test strategy considers fully the nature of the solution and is well-structured.	I can produce a well-structured test strategy that is appropriate for the solution.				
		Provided an informed description of the scope and range of the chosen test strategy.	I can fully explain the scope of the test strategy.				
		Fully explained the purpose of unit, integration and functional testing, taking into account the nature of the solution	I can fully explain the purpose of each test and how it will be used to test the system is working correctly.				
	Test Hypotheses	Considered in detail how the outcomes of the testing process will be used to influence any further development of the solution.	I can explain the expected results for each test.				
			I can explain what further development may be required within each area of the testing.				
	Produce detailed Plan	Produced a comprehensive plan for carrying out unit, integration and functional testing to cover all requirements of the given scenario.	I can create a comprehensive test plan that will fully test the solution.				
			I can create a test plan that will test the solution meets the requirements of the scenario.				
	Plan Test Data	Identified comprehensive test data to fully test the solution.	I can identify typical, extreme and erroneous test data.				
Testing	**Complete a full range of tests**	Followed the test plan in a logical and systematic manner.	I can follow my test plan and complete all tests in order.				
		Testing made effective use of typical, extreme and erroneous data.	I can make use of typical, extreme and erroneous data within my testing.				
	Present Testing Outcomes	Presented all testing outcomes with detailed and informed commentaries.	I can evidence all tests carried out.				
			I can explain each test in detail alongside the outcome.				
		Consider the outcomes of testing and identify areas of further development	I can identify areas of further development based upon the outcomes of my tests.				
Further Development	**Evaluate the Solution Based on the Objectives**	Considered fully the outcomes of the testing process in terms of the solution objectives.	I can identify which objectives the solution has met and not met.				
	Features	Fully described the successful features and areas for further development	I can fully describe the successful features of the solution.				

			I can fully describe the features that need further development.				
	Suggest Extensions to the Application	Proposed detailed and comprehensive suggestions for specific extensions to the solution.	I can explain in detail extensions that could be added to the solution with further development.				