Business and Computing

GLENIFFER HIGH SCHOOL

Name ....

Higher Computing Science
Progress and Achievement
LOGBOOK

Acceptance - Ambition - Nurture - Respect - Trust

Learning today for a better tomorrow

SOFTWARE DESIGN & DEVELOPMENT

Content	Green	Amber	Red
Describe and compare the development methodologies: iterative development process and agile methodologies			
Identify the purpose, scope, boundaries and functional requirements including inputs, processes and outputs of a program			
Identify the data types and structures required for a program and be able to describe, exemplify and implement them (1D arrays, records and arrays of records)			
Describe, identify and be able to read and understand structure diagrams, flowcharts and pseudocode (showing top level design, data flow and refinements)			
Describe, exemplify and implement user-interface design using a wireframe showing input and output			
Describe, exemplify and implement appropriate constructs: parameter passing (formal and actual)			
Describe, exemplify and implement appropriate constructs: the scope of local and global variables			
Describe, exemplify and implement appropriate constructs: sub-programs (with arguments – inputs and outputs): functions and procedures			
Describe, exemplify and implement appropriate constructs: pre-defined functions: substrings (mid), convert to ascii (asc), convert to character (char), convert to floating point (int), find the remainder (mod)			
Describe, exemplify and implement appropriate constructs: file handling (open, create, read, write, close)			
Describe, exemplify and implement standard algorithms using 1D arrays or arrays of records: linear search, find minimum and maximum, count occurrences			
Describe, identify and exemplify and implement a comprehensive final test plan that shows functional requirements are met			
Describe and identify syntax, execution and logic errors			
Describe and exemplify debugging techniques: dry runs, trace tables/tools, breakpoints and watchpoints			
Describe, identify and exemplify the evaluation of a solution in terms of: fitness for purpose, efficient use of coding constructs, usability, maintainability, robustness			

COMPUTER SYSTEMS			
Content	**Green**	**Amber**	**Red**
Describe and exemplify the use of binary to represent positive and negative integers using two's complement, including the range of numbers that can be represented using a fixed number of bits.			
Conversion of two's complement numbers from binary to denary and vice versa.			
Describe and exemplify floating-point representation of positive and negative real numbers, using the terms mantissa and exponent.			
Describe the relationship between the number of bits assigned to the mantissa/exponent, and the range and precision of floating-point numbers.			
Describe Unicode used to represent characters and its advantage over extended ASCII code (8-bit) in terms of numbers of characters.			
Describe the relative advantages and disadvantages of bit-mapped graphics versus vector graphics.			
Describe the concept of the fetch-execute cycle.			
Describe the factors affecting computer system performance: ♦ number of processors (cores) ♦ width of data bus ♦ cache memory ♦ clock speed			
Describe the environmental impact of intelligent systems: ♦ heating systems ♦ traffic control ♦ car management systems			
Describe and identify the implications for individuals and businesses of the Computer Misuse Act 1990: ♦ unauthorised access to computer material ♦ unauthorised access with intent to commit a further offence ♦ unauthorised modification of programs or data on a computer			
Describe and identify the security risks of: ♦ tracking cookies ♦ DOS (Denial of Service) attacks: — symptoms o slow performance, inability to access — effects o disruption to users and businesses — costs o lost revenue, labour in rectifying fault — type of fault o bandwidth consumption, resource starvation, Domain Name Service (DNS) — reasons o financial, political, personal			
Describe how encryption is used to secure transmission of data: ♦ use of public and private keys ♦ digital certificates ♦ digital signatures			

DATABASE DESIGN & DEVELOPMENT

Content	Green	Amber	Red
Identify the end-user and functional requirements of a database problem that relates to the implementation at this level.			
Describe and exemplify entity-relationship diagrams with three or more entities, indicating: entity name attributes name of relationship cardinality of relationship (one-to-one, one-to-many, many-to-many)			
Describe and exemplify an instance using an entity-occurrence diagram.			
Describe and exemplify a compound key.			
Describe and exemplify a data dictionary with three or more entities: entity name and attribute name primary and foreign key attribute type: — text — number — date — time — Boolean attribute size validation: — presence check — restricted choice — field length — range			
Exemplify a design of a solution to a query: tables and queries fields search criteria sort order calculations grouping			
Describe, exemplify and use SQL operations for pre-populated relational databases, with three or more linked tables: UPDATE, SELECT, DELETE, INSERT statements making use of: — wildcards			
UPDATE, SELECT, DELETE, INSERT statements making use of: — aggregate functions (MIN, MAX, AVG, SUM, COUNT)			
UPDATE, SELECT, DELETE, INSERT statements making use of: — computed values, alias			
UPDATE, SELECT, DELETE, INSERT statements making use of: — GROUP BY — ORDER BY — WHERE			
Describe and exemplify testing: SQL operations work correctly at this level			
Evaluate solution at this level in terms of: fitness for purpose accuracy of output			