**Computing Science – National 5**

Name:

**Pupil Tracking Booklet**

**Other Subjects**

|  |  |  |
| --- | --- | --- |
| Subject | U3 Exam Result | U4 Prelim Result |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**My targets**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | My Target for U3 | My U3 Result | | My Target for U4 | My U4 Prelim result | | My Target for my assignment |  | |

What am I going to do to ensure I meet or exceed my target……………

***Review of Target 1***

|  |  |
| --- | --- |
| ***Date*** |  |
| ***Strengths*** |  |
| ***Targets*** |  |
| **Pupil signature** |  |
| **Teacher signature** |  |
| **Parent signature** |  |

***Review of Target 2***

|  |  |
| --- | --- |
| ***Date*** |  |
| ***Strengths*** |  |
| ***Targets*** |  |
| **Pupil signature** |  |
| **Teacher signature** |  |
| **Parent signature** |  |

***Review of Target 3***

|  |  |
| --- | --- |
| ***Date*** |  |
| ***Strengths*** |  |
| ***Targets*** |  |
| **Pupil signature** |  |
| **Teacher signature** |  |
| **Parent signature** |  |

**Pupil Tracking sheets**

The following pages contain the full content required for Higher Computer Science. To help you plan your next steps in learning and independent study you will shade in the boxes beside each topic item. If you are fully confident you can explain this topic in detail and fully answer exam questions to the required standard you shade all 5 boxes. If you are not at all confident in the topic and feel you do not really understand the topic, you shade just 1 box. Boxes 2, 3 and 4 are appropriate scales between 1 and 5.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SOFTWARE DESIGN AND DEVELOPMENT | | | | | | |
|  | Details | 1 | 2 | 3 | 4 | 5 |
| Development Methodologies | Describe and implement the phases of an iterative development process within general programming problem-solving |  |  |  |  |  |
| * Analysis | x |  |  |  |  |
| * Design | x |  |  |  |  |
| * Implementation | x |  |  |  |  |
| * Testing | x |  |  |  |  |
| * Documentation | x |  |  |  |  |
| * Evaluation | x |  |  |  |  |
| Analysis | Identify the purpose and functional requirements of a problem that relates to the design and implementation at this level, in terms of: |  |  |  |  |  |
| * Inputs | x |  |  |  |  |
| * Processes | x |  |  |  |  |
| * Outputs | x |  |  |  |  |
| Design | Describe, identify and be able to read and understand: |  |  |  |  |  |
| * Structure Diagrams | x |  |  |  |  |
| * Flowcharts |  |  |  | x |  |
| * Pseudocode |  |  | x |  |  |
| Exemplify and implement **one** of the above design techniques to design efficient solutions to a problem. | x |  |  |  |  |
| Describe, exemplify, and implement user-interface design, in terms of input and output, using a wireframe. | x |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SOFTWARE DESIGN AND DEVELOPMENT | | | | | | |
|  | Details | 1 | 2 | 3 | 4 | 5 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Implementation (Data types and structures) | Describe, exemplify, and implement appropriately the following data types and structures: |  |  |  |  |  |
| * Character |  |  |  | x |  |
| * String |  |  |  | x |  |
| * Numeric (integer and real) |  |  |  |  | x |
| * Boolean |  |  |  |  | x |
| * 1-D Array |  |  | x |  |  |
| Implementation (computational construct) | Describe, exemplify, and implement the appropriate constructs in a high-level (textual) language: |  |  |  |  |  |
| * Expression to assign values |  |  |  | x |  |
| * Expressions to return values using arithmetic operations (addition, subtraction, multiplication, division, and exponentiation) |  |  |  | x |  |
| * Expressions to concatenate strings |  |  |  | x |  |
| * Selection constructs using simple conditional statements with <, >, ≤, ≥, =, ≠ operators |  |  |  |  | x |
| * Selection constructs using complex conditional statements |  | x |  |  |  |
| * logical operators (AND, OR, NOT) |  |  |  | x |  |
| * Iteration and repetition using fixed and conditional loops |  |  | x |  |  |
| * Predefined functions (with parameters): |  |  |  |  |  |
| * Random | x |  |  |  |  |
| * Round | x |  |  |  |  |
| * length | x |  |  |  |  |
| Read and explain code that makes use of the above constructs |  |  | x |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SOFTWARE DESIGN AND DEVELOPMENT | | | | | | |
|  | Details | 1 | 2 | 3 | 4 | 5 |
| Implementatiion (Algorithm Specification) | Describe, exemplify, and implement standard algorithms: |  |  |  |  |  |
| * input validation | x |  |  |  |  |
| * running total within loop | x |  |  |  |  |
| * traversing a 1-D array | x |  |  |  |  |
| Testing | Describe, identify, exemplify, and implement normal, extreme, and exceptional test data for a specific problem, using a test table | x |  |  |  |  |
| Describe and identify syntax, execution, and logic errors |  | x |  |  |  |
| Evaluation | Describe, identify, and exemplify the evaluation of a solution in terms of: |  |  |  |  |  |
| * fitness for purpose | x |  |  |  |  |
| * efficient use of coding constructs | x |  |  |  |  |
| * robustness | x |  |  |  |  |
| * readability: | x |  |  |  |  |
| — internal commentary | x |  |  |  |  |
| — meaningful identifiers | x |  |  |  |  |
| — indentation |  |  | x |  |  |
| — white space |  |  | x |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SOFTWARE DESIGN AND DEVELOPMENT – COMPUTER SYSTEMS | | | | | | |
|  | Details | 1 | 2 | 3 | 4 | 5 |
| Data Representation | Describe and exemplify the use of binary to represent positive integers. |  |  |  |  | x |
| Describe floating point representation of positive real numbers using the terms mantissa and exponent. | x |  |  |  |  |
| Convert from binary to denary and vice-versa. |  |  |  |  | x |
| Describe extended ASCII code (8-bit) used to represent characters. | x |  |  |  |  |
| Describe the vector graphics method of graphic representation for common objects: |  |  |  |  |  |
| * rectangle | x |  |  |  |  |
| * ellipse | x |  |  |  |  |
| * line | x |  |  |  |  |
| * polygon | x |  |  |  |  |
| With attributes: |  |  |  |  |  |
| * co-ordinates | x |  |  |  |  |
| * fill colour | x |  |  |  |  |
| * line colour | x |  |  |  |  |
| Describe the bit-mapped method of graphics representation. | x |  |  |  |  |
| Computer Structure | Describe the purpose of the basic computer architecture components and how they are linked together: |  |  |  |  |  |
| * processor (registers, ALU, control unit) | x |  |  |  |  |
| * memory locations with unique addresses | x |  |  |  |  |
| * buses (data and address) | x |  |  |  |  |
| Explain the need for interpreters and compilers to translate high-level program code to binary (machine code instructions). | x |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SOFTWARE DESIGN AND DEVELOPMENT – COMPUTER SYSTEMS | | | | | | |
|  | Details | 1 | 2 | 3 | 4 | 5 |
| Environmental Impact | Describe the energy use of computer systems, the implications on the environment and how these could be reduced through: |  |  |  |  |  |
| * settings on monitors |  | x |  |  |  |
| * power down settings |  | x |  |  |  |
| * leaving computers on standby |  | x |  |  |  |
| Security Precautions | Describe the role of firewalls | x |  |  |  |  |
| Describe the use made of encryption in electronic communications | x |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DATABASE DESIGN AND DEVELOPMENT | | | | | | |
|  | Details | 1 | 2 | 3 | 4 | 5 |
| Analysis | Identify the end-user and functional requirements of a database problem that relates to the implementation at this level. | x |  |  |  |  |
| Design | Describe and identify the implications for individuals and businesses of the Data Protection Act 1998: |  |  |  |  |  |
| * prior consent of data subject | x |  |  |  |  |
| * accuracy of data | x |  |  |  |  |
| * data used for limited, specifically stated purposes | x |  |  |  |  |
| * data kept safe and secure | x |  |  |  |  |
| Describe and exemplify entity-relationship diagrams with two entities indicating: |  |  |  |  |  |
| * entity name | x |  |  |  |  |
| * attributes | x |  |  |  |  |
| * relationship (one-to-many) | x |  |  |  |  |
| Describe and exemplify a data dictionary: |  |  |  |  |  |
| * entity name | x |  |  |  |  |
| * attribute name | x |  |  |  |  |
| * primary and foreign key | x |  |  |  |  |
| * attribute type: |  |  |  |  |  |
| — text | x |  |  |  |  |
| — number | x |  |  |  |  |
| — date | x |  |  |  |  |
| — time | x |  |  |  |  |
| — Boolean | x |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DATABASE DESIGN AND DEVELOPMENT | | | | | | |
|  | Details | 1 | 2 | 3 | 4 | 5 |
| Design | * attribute size | x |  |  |  |  |
| * validation: |  |  |  |  |  |
| — presence check | x |  |  |  |  |
| — restricted choice | x |  |  |  |  |
| — field length | x |  |  |  |  |
| — range | x |  |  |  |  |
| Exemplify a design of a solution to the query: |  |  |  |  |  |
| * multiple tables | x |  |  |  |  |
| * fields | x |  |  |  |  |
| * search criteria | x |  |  |  |  |
| * sort order | x |  |  |  |  |
| Implementation | Implement relational databases with two linked tables, to match the design with referential integrity. | x |  |  |  |  |
| Describe, exemplify and implement SQL operations for pre-populated relational databases, with a maximum of two linked tables: |  |  |  |  |  |
| * select: |  |  |  |  |  |
| — from | x |  |  |  |  |
| — where: | x |  |  |  |  |
| * AND, OR, <, >, = | x |  |  |  |  |
| * order by with a maximum of two fields | x |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Implementation | * insert | x |  |  |  |  |
| * update | x |  |  |  |  |
| * delete | x |  |  |  |  |
| * equi-join between tables | x |  |  |  |  |
| Read and explain code that makes use of the above SQL. | x |  |  |  |  |
| Testing | Describe and exemplify testing: |  |  |  |  |  |
| * SQL operations work correctly at this level | x |  |  |  |  |
| Evaluation | Evaluate solution in terms of: |  |  |  |  |  |
| * fitness for purpose | x |  |  |  |  |
| * accuracy of output | x |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| WEB DESIGN AND DEVELOPMENT | | | | | | |
|  | Details | 1 | 2 | 3 | 4 | 5 |
| Analysis | Identify the end-user and functional requirements of a website problem that relates to the design and implementation at this level. | x |  |  |  |  |
| Design | Describe and exemplify the website structure with a home page, a maximum of four linked multimedia pages, and any necessary external links. | x |  |  |  |  |
| Describe, exemplify and implement, taking into account end-user requirements, effective user-interface design (visual layout and readability) using wire-framing: |  |  |  |  |  |
| * navigational links | x |  |  |  |  |
| * consistency across multiple pages | x |  |  |  |  |
| * relative vertical positioning of the media displayed | x |  |  |  |  |
| * file formats of the media (text, graphics, video, and audio) | x |  |  |  |  |
| Describe and identify the implications for individuals and businesses of the Copyright, Designs and Patents Act 1988 relating to: |  |  |  |  |  |
| * web content (text, graphics, video, and audio) | x |  |  |  |  |
| Compare a range of standard file formats: |  |  |  |  |  |
| * audio standard file formats WAV and MP3 in terms of compression, quality, and file size | x |  |  |  |  |
| * bit-mapped graphic standard file formats JPEG, GIF, and PNG in terms of compression, animation, transparency, and colour depth | x |  |  |  |  |
| Describe the factors affecting file size and quality, relating to resolution, colour depth, and sampling rate. | x |  |  |  |  |
| Describe the need for compression. | x |  |  |  |  |
| Describe, exemplify and implement prototyping (low-fidelity) from wireframe design at this level. | x |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| WEB DESIGN AND DEVELOPMENT | | | | | | |
|  | Details | 1 | 2 | 3 | 4 | 5 |
| Implementation (CSS) | Describe, exemplify and implement internal and external Cascading Style Sheets (CSS): |  |  |  |  |  |
| * selectors, classes and IDs | x |  |  |  |  |
| * properties | x |  |  |  |  |
| — text: | x |  |  |  |  |
| * font (family, size) | x |  |  |  |  |
| * color | x |  |  |  |  |
| * alignment | x |  |  |  |  |
| — background colour | x |  |  |  |  |
| Read and explain code that makes use of the above CSS. | x |  |  |  |  |
| Implementation (HTML) | Describe, exemplify and implement HTML code: |  |  |  |  |  |
| * HTML |  | x |  |  |  |
| * head |  | x |  |  |  |
| * title |  | x |  |  |  |
| * body |  | x |  |  |  |
| * heading |  | x |  |  |  |
| * paragraph |  | x |  |  |  |
| * DIV | x |  |  |  |  |
| * link |  | x |  |  |  |
| * anchor | x |  |  |  |  |
| * IMG |  | x |  |  |  |
| * audio | x |  |  |  |  |
| * video | x |  |  |  |  |
| * lists — ol, ul and li | x |  |  |  |  |
| Implementation (HTML) | Describe and implement hyperlinks (internal and external), relative and absolute addressing. | x |  |  |  |  |
| Read and explain code that makes use of the above HTML. | x |  |  |  |  |
| Implementation (Javascript) | Describe and identify Javascript coding related to mouse events: |  |  |  |  |  |
| * Onmouseover | x |  |  |  |  |
| * Onmouseout | x |  |  |  |  |
| Testing | Describe and exemplify testing: |  |  |  |  |  |
| * matches user-interface design | x |  |  |  |  |
| * links and navigation work correctly | x |  |  |  |  |
| * media (such as text, graphics, and video) display correctly | x |  |  |  |  |
| * consistency | x |  |  |  |  |
| Evaluation | Evaluate solution in terms of: |  |  |  |  |  |
| * fitness for purpose | x |  |  |  |  |

**Assessment Log**

|  |  |  |
| --- | --- | --- |
| Date | Assessment | Feedback |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |