Step-Ups

| **Level 1** | **Level 2** | **Level 3** |
| --- | --- | --- |
| **Database** |  |  |
| ● using appropriate tools and techniques to structure, organise, query and present  data for a purpose and end user  ● applying appropriate data integrity and testing procedures  ● describing relevant implications. | ● designing the structure of the data  ● using appropriate tools and advanced techniques to organise, query and present  data for a purpose and end users  ● applying appropriate data integrity and testing procedures  ● explaining relevant implications. | ● designing the structure of the data  ● using appropriate tools and appropriate techniques to organise, query and  present data for a purpose and end users  ● applying appropriate data integrity and testing procedures  ● addressing relevant implications. |
| Basic Techniques EN4   * The data may be stored in a flat file or relational database. | Advanced techniques will include a selection from:  ● linking data in related tables or nodes using queries or keys  ● writing custom queries to filter and/or sort data  ● using logical, mathematical and/or wildcard operators  ● customising presentation of the data  ● using custom forms to add user input to the database  ● setting validation rules for data entry. | Complex techniques will include a selection from:  ● structuring the data using multiple tables or nodes  ● creating queries which insert, update or delete to modify data  ● creating customised data displays from multiple tables or nodes (e.g. reports,  PDFs, web pages, dashboards, program interfaces)  ● dynamically linking data between the database and a front-end display  ● applying data access permissions as appropriate to the outcome. |
| **Programming** |  |  |
| ● writing code for a program that performs a specified task using a suitable  programming language  ●setting out the program code clearly  ●documenting the program with comments  ●testing and debugging the program to ensure that it works on a sample of expected  cases. | ● writing code for a program that performs a specified task  ● using advanced techniques in a suitable programming language  ● setting out the program code clearly and documenting the program with  comments  ● testing and debug | ● writing code for a program that performs a specified task  ● using complex techniques in a suitable programming language  ● setting out the program code clearly and documenting the program with  comments  ● testing and debugging the program to ensure that it works on a sample of  expected cases. |
| A computer program uses:  ● variables storing at least two types of data (e.g. numeric, text, Boolean)  ● sequence, selection and iteration control structures  ● input from a user, sensors or another external source  and one or more of:  ● data stored in collections (e.g. lists, arrays, dictionaries)  ● user-defined methods, functions or procedures. | Level 1 Plus:  ● produces output  ● uses two or more advanced programming techniques.  Examples of advanced programming techniques include:  ● modifying data stored in collections (e.g. lists, arrays, dictionaries)  ● storing multidimensional data in collections  ● creating methods, functions, or procedures that use parameters and/or return  values  ● responding to events generated by a graphical user interface (GUI)  ● using non-trivial string manipulation  ● using additional non-core libraries. | Level 2 Plus  ● produces output  ● uses two or more complex programming techniques.  Examples of complex programming techniques include:  ● programming or writing code for a graphical user interface (GUI)  ● reading from, or writing to, files or other persistent storage  ● object-oriented programming using class(es) and objects defined by the student  ● using types defined by the student  ● using third party or non-core API, library or framework  ● using complex data structures (e.g. stacks, queues, trees). |