**Project Submission Form (Secondary)**

|  |  |
| --- | --- |
| Name | William Lau |
| Title of project | Form tutor management system with GUI |
| Description of the project (max 500 words) – include the underlying design and functionality | The project is based on a sample Controlled Assessment task from NEA which involves sequence, selection, iteration, file reading, writing and updating (see attached PDF).  The task involves the creation of a simple tutor management system for a client (Mr Leeman) at Tree Road School. Students are automatically assigned a unique ID number, tutor group and unique school email address. The system will not need to worry about generating these as they will already be provided. However the system should check that they are actually unique and do not already exist.  The system will have a user friendly text-based (command-line) interface. There will not be a need to make a graphical user interface. The program should allow the client, Mr Leeman to log in and enter student details which are stored on his system. Through a simple menu system, Mr Leeman will also be able to perform other simple operations such as retrieving student details based on a student’s unique ID and produce a series of useful reports. Example reports might be an alphabetical register, a contact list with student names and their parents’ phone numbers.  I want to develop this project further by including a GUI. A GUI is not necessary at GCSE, but is necessary at A-Level. I would like my high ability GCSE students to be able to create a simple GUI using (say) GuiZero or PyQT and I will investigate both of these modules and develop a coded solution as well as some resources to help students develop their own GUI for any given project.  I could explore further enhancements such as generating a uniqueID as opposed to using a set of existing ID numbers. |
| Programming language and environment | Python 3  GuiZero or PyQT |
| Who will use this program? | Students will use this program as a template/reference to help with their controlled assessment. The task above is for a sample controlled assessment which is not “live” therefore malpractice is not an issue. |
| What test data can be used to test this program? | I will create a bank of Valid, invalid and erroneous test data for categories such as uniqueID, email address, DOB, password and home phone number and demonstrate that these tests have been performed with either success or meaningful error messages in each case. |

Highlight the statements below which this project demonstrates

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Algorithms | Programming | Data | Computers and Social Informatics | Communication and the Internet |
| 1 | A1 Explain that an algorithm is a precise way of solving a problem which can be followed by humans and computers. | P1 Code competently in at least two programming languages, which may both be ‘visual’;; at least one of these must allow the use of programming concepts such as selection, repetition, procedures, variables and relational operators. acteristics for the same task. | D1. Explain how computers represent all data in binary, with a variety of examples: unsigned integers, text representation (e.g. ASCII), different sound file data/types, and different graphics data/file types. | C1. Explain what a computer is and give examples of devices that include computers. | I1. Explain what the World Wide Web and the Internet are, and the difference |
| 2 | A2 Give examples of algorithms met in everyday life | P2 Explain and use programming concepts such as selection, repetition, procedures, variables, and relational operators. | D2. Explain how the same binary data can be interpreted in different ways e.g. an 8-bit value could be a character or a number. | C2. Explain and describe the key characteristics of basic computer architecture (eg CPU, memory, hard disk, mouse, display etc) . | I2. Outline the key features of the World Wide Web and their relationships– eg browsers, URLs, navigation methods |
| 3 | A3 Explain that computers need more precise instructions than humans and the need for precision to avoid errors. | P3 Review and assess the quality of code. Find and correct errors in syntax and meaning. | D3. Explain how the same information can be represented in a computer in a variety of ways e.g. sound as mp3 or MIDI. | C3. Explain why there are sometimes different operating systems and application software for the same hardware. ture (eg CPU, memory, hard disk, mouse, display etc) . | I3. Outline how data are transported on the Internet, including packets and the notion of a protocol. eg browsers, URLs, navigation methods |
| 4 | A4 Explain and show how algorithms can use selection (if), repetition (loops), procedures (sub- algorithms within an algorithm). | P4 Explain that computers are controlled by sequences of precise instructions known as programs meaning. | D4. Explain that data can have errors, how this might affect results and decisions based on the data and how errors can be reduce. | C4. Explain and use common troubleshooting techniques. ation software for the same hardware. ture (eg CPU, memory, hard disk, mouse, display etc) . | I4. Explain the role of search engines and what happens when a user requests a web page in a browser. |
| 5 | A5 Explain the need for accuracy of algorithms. | P5 Explain that computers follow instructions/ blindly; hence the need for care and precision. | D5. Explain the need for and content of the Data Protection Act, Computer Misuse Act and Copyright legislation (and other relevant legislation). | C5. Explain Moore’s Law and multitasking by computers. | I5. Explain the technological perspective on safety and security. |
| 6 | A6 Distinguish between an algorithm and the programs that implements that algorithm | P6 Represent algorithmic steps in multiple programming languages (e.g. logo, scratch). | D6. Explain the difference between data and information. | C6. Discuss social and ethical issues raised by the role of computers in the world. | I6. Explain the concepts of: client/server models; MAC addresses, IP addresses and domain names; and cookies. |
| 7 | A7 Explain how the choice of an algorithm should be influenced by the data | P7 Explain how and use programs to simulate environments to test hypothesis. | D7. Explain the need for and use of hexadecimal, two’s complement, signed integers, and string manipulation. | C7. Explain the importance of human-computer interface design. world. | I7. Explain a ‘real protocol’ e.g. using telnet to interact with an HTTP server. |
| 8 | A8 Be able to explain and use several key algorithms (e.g. sorting, searching, shortest path). | P8 Explain and show how programs can be planned, tested and corrected and documented. | D8. Explain the need for data compression, and be able to describe simple compression methods. | C8. Discuss career paths for those studying Computing. | I8. Explain routing; redundancy and error correction; encryption and security. |
| 9 | A9 Explain how algorithms can be improved, validated, tested and corrected. | P9 Explain how HTML constructs the rendering of a web page | D9. Explain the need for analogue to digital conversions and how this works. | C9. Explain the use of logic gates and registers. |  |
| 10 | A10 Explain that a single problem could be solved by more than one algorithm. | P10 Program competently in a least two programming languages, at least one of which must be ‘textual’. | D10. Explain the limitations of using binary representations – eg rounding errors, sampling frequency and fractional numbers. | C10. Explain Von Neumann architecture |  |
| 11 | A11. Explain and show how different algorithms can have different performance characteristics for the same task. | P11. Explain and use programming concepts such as selection, repetition, procedures, constants, variables, relational operators, logical operators and functionsProgram competently in a least two programming languages, at least one of which must be ‘textual’. | D11. Explain how structured data can be represented in tables in a relational database, and simple database queries | C11. Explain the fetch- execute cycle. |  |
| 12 | A12 Successfully apply algorithms in solving GCSE and A level type problems characteristics for the same task. | P12. Explain and use truth tables and Boolean valued variables. |  | C12. Explain and use low level instruction sets and assembly code. |  |
| 13 |  | P13. Explain and use two- dimensional arrays (and higher). |  | C13. Explain what compilers and interpreters are and do and give some examples of when they are used. |  |
| 14 |  | P14. Explain and use nested constructs (e.g. a loop that contains a conditional, and vice versa) |  | C14. Explain the main functions of operating systems. |  |
| 15 |  | P15. Explain the concept of procedures that call procedures |  |  |  |
| 16 |  | P16. Explain how low level languages work and when they are used, being able to give simple examples. |  |  |  |
| 17 |  | P17. Explain that a program can be written to satisfy requirements and that they should be corrected if they do not meet these. |  |  |  |
| 18 |  | P18. Successfully apply programming in solving Computing/Computer Science GCSE and A level type problems |  |  |  |