Project Log

Analysis (9 marks)

1. Research
    a. Start with good explanation of the system
        i. Use of sketches in addition to textual information to build the storyline.
        ii. DB: same but can include DFD OR ER-Diagram
    b. Examples:
        i. so a DB project should have DFD / ER or similar
        ii. but a game might have sketch layouts (annotated) and a story line
        iii. a projectile simulator should have a sketch of what it might look like annotated with formulas
        iv. a chess game should have a sketch of the game and perhaps discussion around the AI (or features like 'move checker')
        v. a quiz would have example questions and how they would be answered / marked by the system…
2. Use examples while modelling parts of the system wherever possible
    a. E.g. a quiz creator system can give examples of different types of questions to include in the system like sample of multiple choice, image based and simple questions
3. Identification of end-user
4. Acceptable limitations if any
5. Good to talk about programming language/ tools which will be used to build the system.
    a. List any libraries to install and use for program
6. Well defined set of measurable objectives which must be split into sub-objectives
7. Evidence of analysis (Interview/questionnaire)

Design (12 marks)

1. Start with a good 'textual' overview where focus should be on core requirements of the system with some details
2. Use of system flowchart/ Hierarchy chart with description
    a. Class diagram – for games project where there is relationship between the classes e.g inheritance, composition etc.
3. Describe the entire proposed system in a sequence using the UI.
    a. UI sketch
        i. If DB project is using data table in C#, can make box next to UI to label with data to use in this datatable
    b. Annotate each element of UI sketch
    c. Purpose of this UI
    d. Class definition if used
    e. Algorithm for this section & Data structures for complex parts only
        i. Explain algorithm in detail with example data
        ii. Explain the role of data structures used.
        iii. DB projects: also explain the query used in the algorithm
    f. Validation list
        i. Mostly required for DB projects as these projects need input from user while storing the data etc.

        ii. May be required for non-DB if input is given by the user to the system for some processing
    g. Repeat the cycle for each UI sketch
4. Database project:
    a. Normalised final tables with example data

Technical solution (42 marks)

Completeness of solution (15 marks)

Techniques used (27 marks)

The descriptions in table below are cumulative, i.e. for a program to be classified as excellent it would be expected to exhibit the characteristics listed as excellent, good and basic not just those listed as excellent.

Coding style	**Characteristics**
Excellent	Modules (subroutines) with appropriate interfaces. Loosely coupled modules (subroutines) – module code interacts with other parts of the program through its interface only. Cohesive modules (subroutines) – module code does just one thing. Modules (collections of subroutines) – subroutines with common purpose grouped. Defensive programming. Good exception handling.
Good	Well-designed user interface Modularisation of code Good use of local variables Minimal use of global variables Managed casting of types Use of constants Appropriate indentation Self-documenting code Consistent style throughout File paths parameterised

Basic	Meaningful identifier names Annotation used effectively where required

Testing (8 marks)

1. Bottom up testing
    a. To test normal, erroneous and boundary data
2. Top down testing
    a. To test navigations of program
3. System testing (DB)
    a. To test the Insert, Updates and Deletes correctly work and reflect the changes.
4. Black box testing (non-DB)
    a. To test the program gives correct output for given input.
5. White box testing (non-DB)
    a. To test the flow of the code and its values using debug
    b. Evidences of manual checking of the formula
6. Instead of carrying out all these testing strategies and give evidences for each tests, one video can be made to demonstrate all aspects of the system.

Evaluation (4 marks)

1. Start with a paragraph to reflect on the project as a whole.
2. Comparison of project performance against numbered general and specific objectives
    a. If and wherever possible, critically evaluate the objectives
    b. If not, write at the end of table giving a reference to the "Possible extension" heading to reflect on these objectives i.e. critical evaluation of these objectives.
3. Authenticated feedback letter from end user
4. Analysis of feedback letter
5. Possible Extensions:
    a. End user's extensions
    b. Student's extensions
    c. This sections must be critically evaluated e.g.
        i. if currently using linear search and is not very efficient to use, would like to try with binary search
        ii. Or would like to introduce networking options to my game so it can be played from any location