

A LEVEL COMPUTING (9691)

MODULE 4: COMPUTING PROJECT GUIDE

Complies with 2010 Syllabus

1. Introduction

This guide is based on the instructions given in the A Level Computing syllabus and Module 4 of the B&L Publishing's A Level Computing CD. It is an attempt to help students understand the requirements and structure of Paper 4, the Computing Project.

Best grades in Computing Project are possible if the students ensure not only the completeness of the project report but also the quality of its contents and presentation. It is thus very important to read carefully what is required and produce a professional-looking documentation that complies with the standards set in this guide, the syllabus and the Module 4 book.

2. The Project

The Computing Project is a major part of the course. The volume of work is considerable and students should not underestimate the time required to produce a high quality work. Students are at liberty to decide the difficulty level of their projects. There is a list of possible project topics in this guide, but it is by no means exhaustive and students are free to choose other topics. An important point here is not to choose a project topic that is self-penalizing. A topic that is too complex will be difficult for the student to complete in time, if at all. On the other hand, a topic that is trivial will not be able to fulfill syllabus requirements.

It is very important the students choose a real end user with a real-life problem. This will ensure two things; it will enable the real interaction that is required in some sections of the project, and it will allow the student to experience the actual project development life cycle.

Although Microsoft Access and Sun OpenOffice Base are recommended, students are free to choose the development tool for their projects. It is, however, important to understand that the teacher/resource person cannot be expected to be an expert on every development tool available in the market today.

3. The Project Report

The project report is a document that is produce as an evidence of a successful system implementation. The student develops this report for the examiner to demonstrate his/her abilities and understanding of the concepts involved in producing a fully working system for an end user.

Important Points

1. The project report must be word processed with consistent formatting used throughout the document.
2. You must spell and grammar check the document.
3. The tone of the report must be formal. Do not use casual language, emoticons or chat lingo in your report.
4. Choose A4 as your document size.
5. Use only non-comic and non-script fonts such as Arial, Times New Roman or Garamond. Do not use more than two fonts in your report document (one for headings and one for text).
6. For Interview scripts use Italics to show conversions.
7. Recommended font sizes are:
 - a. Section Headings: 25 – 30 points in bold.
 - b. Main Headings: 18 – 20 points in bold.

- c. Sub Headings: 15 – 18 points in bold.
- d. Document Text: 11 or 12 points.

Example of the font selection is given below.

Section 1: Analysis

Requirements Definition

a) User Requirements

The user requires an easy to learn software that fulfills the following requirements.

8. The document margins (top, bottom, left and right) should not be more than 1 inch.
9. Documentation must be in single line spacing.
10. The entire document must have page numbers (except the title page).
11. Each section and sub section must be separated visibly by proper headings/separators.
12. Documentation text must be either Left Justified or Justified. Only headings and Tables can be Center Justified.
13. Avoid one line paragraphs and very long sentences.
14. Make a proper title page at the beginning of the documentation.
15. Make a Table of Contents with page numbers for reference.
16. All printing must be clear and visible. The screenshots must be clear and large enough for the contents to be readable.
17. Print your report document on both sides of paper. This will reduce the size of the printed report and make it easier to mark. The final printed report must have a title page listing the following:
 - a. Candidate Name
 - b. Candidate Number
 - c. Center Name
 - d. Center Number
 - e. Project Title
18. Leave at least four inch by three inch (width, height) space on the title page for identification sticker.
19. Use only ring or spiral binding. Please do NOT use tape or hard bindings.

4. Plagiarism

Computing Project is a complex and comprehensive undertaking. It is appreciated that students may seek help and guidance from sources other than their teacher during the course of the project development. However, it is imperative that students understand the consequences of unlawful copying of someone else's work. This is plagiarism and it is considered a serious offense on student's part. In case a student is found committing plagiarism, he/she will be asked to redo the section(s) that are identified as copied work. In extreme cases, entire project could be cancelled and/or a letter from the School will accompany the project stating that the coursework could not be identified as 'original work' and the school/teacher has elected not to mark it. In the later two cases, the student will receive a 'U' grade in the Computing Project (component 4) resulting in an overall 'U' grade.

5. Explanation of Project Stages

Note: Sections in *Italics* require end user involvement, preferably on company's letterhead signed by the end user.

Stage	Description	Marks	
1	Definition, Investigation & Analysis		11
1.1	Introduction and Nature of The Problem	3	
	<ol style="list-style-type: none"> 1. Introduction of the organization along with an organizational chart. 2. Concise description of the problem, its origin and the current solution being utilized by the organization. 3. Description of the kind of data in the current system. 4. Origin of the data and in what form is it recorded/used. 		
1.2	Investigation and Analysis	8	
	<ol style="list-style-type: none"> 1. Plan of information gathering. Questions covering the problem area and a list of people selected for the interviews. 2. <i>Complete transcript of the interviews.</i> Description of information gathered by other methods, e.g. Direct Observation and Document Analysis. 3. Critical evaluation of the information gathered. Analysis of the current solution, its merits and demerits. 4. Schematic representation of the current solution. Identification of the problem areas. 5. <i>Identification of alternative solutions and their critical analysis (merits and demerits).</i> 6. <i>Proposition of the New Solution and justification of its selection.</i> 7. <i>Specification of requirements along with justification of each requirement.</i> <ol style="list-style-type: none"> a. <i>Functional specifications.</i> b. <i>Input and Output Specifications.</i> c. <i>Software specifications.</i> d. <i>Hardware specifications.</i> 		
2	Design		11
2.1	Nature of The Solution	7	
	<ol style="list-style-type: none"> 1. Data structure design along with any schematic to explain relationship among data. 2. <i>Input (Forms) design. Hand drawn mockups or prototyping in the development tool being used along with a description of validation rules on each input field.</i> 3. <i>Output (Report) design. Prototyping or hand drawn mockups.</i> 4. <i>Menu/module design. Schematic in Visio or hand drawn diagram.</i> 5. Schematic representation of Processes in the proposed system. Flow charts, DFD etc. 		
2.2	Intended Benefits	2	
	<ol style="list-style-type: none"> 1. Justification for the proposed system by identifying benefits to the end user, the organization. 		
2.3	Limits of The Scope of The Solution	2	
	<ol style="list-style-type: none"> 1. <i>Limitations encountered due to hardware/software or lack of skills in required software.</i> 2. Estimation of the amount of data that will be generated by the new system. 3. Estimation of database/file size. 		
3	Software Development, Testing & Implementation		18
3.1	Software Development	9	
	<ol style="list-style-type: none"> 1. Software listing of original code written by the student. Must be annotated with description of each module/form. 2. Screen dumps of database structure with and without sample data. Must be annotated. 3. Screen dumps of input forms with sample data. Must be annotated. 4. Screen dumps of reports with sample data. Must be annotated. 5. Screen dumps of queries (if any) in Design, SQL and Data View modes. Must be annotated. 		

3.2	Testing		
	<ol style="list-style-type: none"> 1. Development of a clear and accurate Test Strategy describing the purpose of testing. 2. Test Plan that allows testing for Valid, Extreme and Invalid data (minimum 15 different tests). 3. Test results with annotated screen dumps of each test conducted. 		
3.3	Implementation	6	
	<ol style="list-style-type: none"> 1. Identification of an implementation plan. 2. Identification of any user training required. 3. Evidence of the new system being used by the end user (User acceptance). 4. Evidence of user testing. 		
3.4	Appropriateness of Structure And Exploitation of Available Resources	3	
	<ol style="list-style-type: none"> 1. Discussion of the resources (hardware and software) that were available for developing the new system and how they were used. 2. A Log of problem encountered/milestones throughout the development of the solution. 		
4	Documentation		12
4.1	Technical Documentation	6	
	<ol style="list-style-type: none"> 1. Proper presentation of the technical documentation with Table of Contents and section description. 2. Sections included: <ol style="list-style-type: none"> a. Specification of software used for system development and system implementation. b. Specification of hardware used for system development and system implementation. c. Data structure with annotation. d. Table/Entity Relationship Diagram. e. Input Forms with validation rules and field description. f. Reports with field description. g. List of Formulas used in the system. h. System Schematics (flow charts, DFD, module diagram etc.) i. Program listing fully annotated. 		
4.2	User Documentation	6	
	<ol style="list-style-type: none"> 1. Proper presentation of the user documentation with Table of Contents and section description. 2. Sections include: <ol style="list-style-type: none"> a. System Introduction. b. Installation guide. c. Security procedure (login and passwords, if any). d. Input procedures along with validation rules guide. e. Reporting procedures. f. Backup and restore procedures. g. Troubleshooting guide with a list of common errors. h. A glossary of terms. 		
5	Evaluation		8
5.1	Discussion of The Degree of Success In Meeting The Original Objectives	3	
	<ol style="list-style-type: none"> 1. Evaluation of Requirement Objectives in section 1.2 (7) along with reasons why (if) some of them were not met. Each evaluation point should be cross-referenced with the original objective and the section of project where the evidence can be found to support the claim. 		
5.2	Evaluation of User's Response To The System	3	
	<ol style="list-style-type: none"> 1. Evaluation of user's comments on his/her experience of using the system. 2. Problems in the system identified by the user. 		
5.3	Desirable Extensions	2	
	<ol style="list-style-type: none"> 1. Clear and impartial identification of merits and demerits of the new system. 2. Any possible extensions/improvements in the system. 		

6. Headings for The Project Report Document

Students can choose any **ONE** of the following styles. Please note that you ‘*may*’ choose your own heading style but it **must** comply with the documentation structure mentioned in the syllabus.

Headings without Numbered List	Headings with Numbered List
<p>Analysis</p> <ul style="list-style-type: none"> Introduction and Nature of The Problem Investigation and Analysis <ul style="list-style-type: none"> Plan of Investigation Interviews Direct Observation Document Analysis Description of The Existing System Critical Evaluation of The Existing System Alternative Solutions Proposed Solution Requirements Specification <ul style="list-style-type: none"> Functional Requirements Input Requirements Output Requirements Software Requirements Hardware Requirements <p>Design</p> <ul style="list-style-type: none"> Nature of Solution <ul style="list-style-type: none"> Data Structure Design Input Design Output Design Menu/Module Design Schematics Intended Benefits Limits of The Scope of The Solution <ul style="list-style-type: none"> System Limitations File Size Estimation <p>Software Development, Testing & Implementation</p> <ul style="list-style-type: none"> Software Development <ul style="list-style-type: none"> Code Listing Development of Database Development of Input Methods Development of Output Methods Development of Database Queries/Filters Testing <ul style="list-style-type: none"> Test Strategy Test Plan Testing Implementation <ul style="list-style-type: none"> Implementation Plan User Training 	<ul style="list-style-type: none"> 1. Analysis <ul style="list-style-type: none"> 1.1. Introduction and Nature of The Problem 1.2. Investigation and Analysis <ul style="list-style-type: none"> 1.2.1. Plan of Investigation 1.2.2. Interviews 1.2.3. Direct Observation 1.2.4. Document Analysis 1.2.5. Description of The Existing System 1.2.6. Critical Evaluation of The Existing System 1.2.7. Alternative Solutions 1.2.8. Proposed Solution 1.2.9. Requirements Specification <ul style="list-style-type: none"> 1.2.9.1. Functional Requirements 1.2.9.2. Input Requirements 1.2.9.3. Output Requirements 1.2.9.4. Software Requirements 1.2.9.5. Hardware Requirements 2. Design <ul style="list-style-type: none"> 2.1. Nature of Solution <ul style="list-style-type: none"> 2.1.1. Data Structure Design 2.1.2. Input Design 2.1.3. Output Design 2.1.4. Menu/Module Design 2.1.5. Schematics 2.2. Intended Benefits 2.3. Limits of The Scope of The Solution <ul style="list-style-type: none"> 2.3.1. System Limitations 2.3.2. File Size Estimation 3. Software Development, Testing & Implementation <ul style="list-style-type: none"> 3.1. Software Development <ul style="list-style-type: none"> 3.1.1. Code Listing 3.1.2. Development of Database 3.1.3. Development of Input Methods 3.1.4. Development of Output Methods 3.1.5. Development of Database Queries/Filters 3.2. Testing <ul style="list-style-type: none"> 3.2.1. Test Strategy 3.2.2. Test Plan 3.2.3. Testing 3.3. Implementation <ul style="list-style-type: none"> 3.3.1. Implementation Plan 3.3.2. User Training

User Testing User Acceptance Appropriateness of Structure and Exploitation of Available Resources Use of Available Resources Problem Log Documentation Technical Documentation User Documentation Evaluation Degree of Success Evaluation of User's Response Desirable Extensions	3.3.3. User Testing 3.3.4. User Acceptance 3.4. Appropriateness of Structure and Exploitation of Available Resources 3.4.1. Use of Available Resources 3.4.2. Problem Log 4. Documentation 4.1. Technical Documentation 4.2. User Documentation 5. Evaluation 5.1. Degree of Success 5.2. Evaluation of User's Response 5.3. Desirable Extensions
--	--

7. Suggested Project Topics

- a. **Library Management System**
System covering a database of books, members, book issuance and return.
- b. **Stock Management System**
Management of products, stock in and out.
- c. **Order Processing System**
A system managing orders received and tracking of these orders till the time they are filled.
- d. **Membership Management System**
Managing information of members of any club/society/association including member joining, leaving, membership fee etc.
- e. **Event Management**
Management of events including booking, resource acquisition and billing.
- f. **Billing System**
System managing a database of consumers and producing periodic bills based on utility or service usage.
- g. **Sale/Purchase Management System**
Management of products and their sale or purchase.
- h. **Car Rental Management System**
Management of information of cars, their rental and billing.
- i. **Inquiry Management System**
System assisting in recording and tracking of product/service inquires from customers/consumers.
- j. **Customer Relationship Management System**
System managing complete information of customer for effective relationship management and loyalty building.
- k. **Video/Music Store Management System**
Management of sale/rental of video/music products.
- l. **Computerization of Sport/Science Department of School**
Computerizing any department of your school. This may include event and finance management.
- m. **School Admission System**
Management of new admissions in a school/institute.
- n. **Boutique/Salon Management System**
Management of customer and/or services/products.
- o. **Clinic Management System**
Management of patient records, appointments and medical/visit histories.

8. List of Recommended Software

Although students are free to choose any software for developing the system and the report, the following are recommended.

For development of software: Microsoft Access (2003 or 2007) OR OpenOffice Base.

For producing project report document: Microsoft Word (2003 or 2007) OR OpenOffice Write

For producing schematics: Microsoft Visio

9. Suggested Points To Cover In Project Stages

Analysis

Introduction and Nature of The Problem

- Name of the company and why you chose this company for your project?
- Who started it? Who owns it?
- Location of the company's office.
- What is the primary business of the company?
- How many departments are there and which department are you computerizing?
- What is the nature of the problem? Give an overview without going into details.
- What type of data is being produced in the current system? (example: alphanumeric data is produced from the Admission Form such as student name, phone number, address etc).
- Where does the data originate from? (example: the data originates from Student Admission Form, Student Exam Result Card etc.)

Investigation and Analysis

Plan of Investigation

- What is the aim of your investigation?
- Which methods will you employ for your investigation and a brief justification of choosing each method.
- Names of people you will contact for your investigation with brief reasons for selecting each one of these people.

Interviews

- Name of the person you are interviewing along with the date of the interview.
- Interview script. Relate your questions to the problem and its solutions. Do not ask irrelevant questions.

Direct Observation

- The department/business process (example: preparation of bills, admission of a student) that you have observed along with the a brief description of your findings. Relate your observations with the problem and its solutions.

Document Analysis

- Titles of the documents that you have analyzed (example: invoice, admission form).
- Brief description of each document and the nature of the data present on each document.

Description of The Existing System

- IMPORTANT: Do NOT criticize the existing system in this stage.
- Where does the business cycle start? (example: the admissions start the school's business cycle)
- What happens at each stage of the business cycle and who is responsible for each stage of the business cycle? Give special attention to the department/stage that has the problem.
- What type of data is produced at each stage and how is it recorded?

Critical Evaluation of The Existing System

- Description of the problems identified.
- Critical discussion (brief) of the cause of each problem.
- Merits of the existing system (every system has some merits).

Alternative Solutions

- Suggestion of (preferably) three alternative solutions to the one that is currently being used.
- Merits and demerits of each alternative solution mentioned above. Relate your discussion to the problem.

Proposed Solution

- Which solution (from the above list) was chosen by you?
- How did your user/client agreed to it? (example: my client was interested in a low-cost solution. my client wanted the new system to be implemented within 3 months).

Requirements Specification**Functional Requirements**

- What functionality is required from the new system?
- What type of data entry methods are required by the user? (example: user wants the new system to provide 'point and click' data entry for fields such as Date, Gender, Payment Method).
- Does the user require any data backup facility?
- What type of output is required by the user?
- What type of interface is required? (Character based-for faster access, GUI based for better look and feel and low learning curve).
- Any special feature(s) that are required? (example: user wants the facility to export his data in .CSV format).
- Evidence of user approving the requirements.

Input Requirements

- Name of each input category and the information that is to be input in each category.
(example: Category: Student
Fields:
Student Name
Address
Phone)
- Evidence of user accepting the input requirements.

Output Requirements

- Name of each report and the information that is to be displayed on each report.
(example: Report: Daily Attendance Report
Fields:
Date
Student ID
Student Name)
- Evidence of user accepting the output requirements.

Software Requirements

- Which software are required to develop the new system?
- A brief description of each software chosen and justification of the choice.
- IMPORTANT: Please identify which software is required by the user to run the system and which software will be used only by you for developing the system.

Hardware Requirements

- A complete list of minimum hardware that is required to run the new system. This is the base requirement without which the system will not be able to run at all.
- A complete list of recommended hardware that is required to run the new system giving optimal performance. Be careful not to go too far.

- TIP: Do some research on the Internet to find out the base and recommended requirements for the software that you are using.

Design

Nature of Solution

- How will you proceed with the designing of the new system?
- TIP: Its the stages given below (up to Schematics)!

Data Structure Design

- Design a table that gives all the data structure that will be used in the new system.
example:

Key	Field	Data Type	Size(bytes)	Sample Data	Constraints
PK	Student ID	Long Integer	4	27	Numbers only
	Student Name	String	45	Ali Akbar Khan	Letters and space only

Input Design

- Hand-drawn or computer generated mockups of each input form along with annotation of each control/field on the form.
- Evidence of user accepting the input design.

Output Design

- Hand-drawn or computer generated mockups of each report along with annotation of each field on the report.
- Evidence of user accepting the output design.

Menu/Module Design

- Menu and/or module structure of the new system.

example: Module Design

Student Information Module
Student Fee Module
Student Examinations Module

example: Menu Design

Main Menu
 Data Entry Forms Menu
 Student Information Form
 Student Fee Form
 Reports Menu
 Student Attendance Report
 Student Fee Report
Exit System

Schematics

- Diagrammatic representation of different aspects of the new system.
- Flow Chart (for all forms and reports in the new system)
- Entity Relationship Diagram (ERD) (for all tables in the new system)
- Data Flow Diagram (DFD) (to show the data flow in the new system)
- Process Diagram (to show how the processes will work in the new system)

Intended Benefits

- What benefits will the new system bear for the user?
- TIP: Cost benefit, time saving, increase in productivity, increase in efficiency, customer satisfaction etc.

Limits of The Scope of The Solution**System Limitations**

- Things/processes that the new system will not do. (example: system will not handle accounts, system will not be able to handle more than 100,000 records etc.)

File Size Estimation

- Estimation of data files generated by the system.
example:
Total number of bytes for 1 record = 250
10% slack = 25
Estimated number of records per year = 500
Total number of bytes for 500 records = $250 + 25 \times 500 = 1250$

Software Development, Testing & Implementation**Software Development****Code Listing**

- Complete listing of the code which must be annotated. Use comments and indentation throughout the code to make it readable.

Development of Database

- Show screen dumps in Design View and Data Sheet View of each table in the new system.

Development of Input Methods

- Show screen dumps in Design View and Run View of each form in the new system and describe what each control/field on the form does.

Development of Output Methods

- Show screen dumps in Design View and Run View of each report in the database and describe what each field on the report does.

Development of Database Queries/Filters

- Show screen dumps in Design View and SQL View of each query in the database and describe what each query does.

Testing**Test Strategy**

- What is the aim of testing the new system?
- What particular features of the new system do you intend to test?
- What kind of testing will you be performing yourself? (dry run, variable dumps, black box).
- Testing strategy should relate to the problem and the proposed solution.

Test Plan

- A table showing the data that will be used for testing, its expected results, the actual results and the actions that were taken if the actual results didn't match the expected results.

example:

Test #	Field	Test Type	Test Data	Expected Results	Actual Results	Action	Evidence
1	Student Name	Valid	Abid Ali	Abid Ali	Abid Ali	Successful	Page 97
2	Student Name	Extreme	Abid Ali Khan	Abid Ali Khan	Abid Ali Kh	Failed. I adjusted the field size to accommodate the data.	Page 98
3	Student Name	Invalid	Abi45@d	Abid	Abid	Successful. Numbers were not entered.	Page 99

Testing

- Provide screen dumps of at least 15 tests from the above table. Make sure these are cross referenced to the above table and annotated.

Implementation**Implementation Plan**

- Which implementation method have you chosen? (example: parallel running) Give reasons for your choice.
- What resources/preparations, if any, are needed for the implementation?
- How did the implementation go? Were there any problems? If yes, document them.
- Evidence of system implementation.

User Training

- Is there any need for user training?
- If yes, what methods have you identified for user training?
- Evidence of user training.

User Testing

- Prepare a set of tests that user can perform on the new system.
- The user provides you feedback on whether the tests were successful or not.
- In case there were some problems, you must fix the problems with the system and communicate to the user that these have been fixed.

User Acceptance

- A letter from the user, on the company letterhead, accepting the software stating that the software works as (or close to) what was originally decided.
- In case there are gaps in the original requirements and the new finished system these must be outlined.

Appropriateness of Structure and Exploitation of Available Resources**Use of Available Resources**

- Briefly mention what type of hardware and software were available to you for developing the system. These are NOT the hardware and software that you mentioned in the Hardware/Software Requirements. This is the hardware/software at your home or school where you developed the system.
- Briefly discuss whether the available hardware and software was adequate and if you were able to take full advantage from it.

Problem Log

- You need to clarify any problems that you have faced during the development of the new system.
- These problems must be mentioned in form of a chronologically arranged list.
example:

January 12, 2009: I started working on the user interface but the TAB control on my Access were not properly installed. I had to take it to the school lab's administrator to get it fixed. Wasted two days.

January 27, 2009: Database is complete but now I am finding a lot of validation errors in the coding. I used the original specifications to check the validation one by one and it took me four days worth of work to finally fix all validation errors.

Documentation

NOTE: The documentation section is supposed to be a separate section/booklet, but preferably should be with the rest of the documentation. However, you can use separators to separate it from the rest of the documentation.

Technical Documentation

- Table of Content
- Introduction to the Technical Documentation
- Complete Requirements Section
- Complete Design Section
- Complete Schematic Section
- Complete Software Coding
- Complete List of Formulas/Validation Checks

User Documentation

- Table of Content
- Introduction to the System
- How to Install the System?
- Detail description of the Menu System
- Detail description of Input Forms and Procedure (including validation restrictions)
- Detail description of Output Reports and their purpose
- How to take backup of data and how to restore it later on.
- A Troubleshooting Guide
- Glossary of Terms

Evaluation

Degree of Success

- Evaluation of each of the original requirement objectives and whether it has been successfully achieved.
example:
Objective: System should be able to input data while ensuring that invalid data is rejected.
Result: Successfully achieved.
Evidence: See Page 9 for Input Form Design. See Page 67 for validation code that restricts invalid data in the fields of the form.

Evaluation of User's Response

- Impartial evaluation of what the user feels about the system.
- Talk about both good and bad reviews (if any).

Desirable Extensions

- What other features would you like to see in the new system? You do not have to be able to develop these features.
- Feature list must be practical and related to the user's requirements.