

AS Projects load sharing:

1. Development (Student and teacher)
2. Completion of Program Comments. Making sure nobody copies (Students and lab assistant)
3. Completion of documentation according to headings (page 2) and required evidences (Page 3) provided (Students and lab assistant)
4. Creation of TOC as per headings on page 2. (Students)
5. Formatting document as per rules set (students)
6. Taking initial submissions. soft copies only (lab assistant)
7. Checking of finalized soft copies for points 2 to 5 (Round 1: lab assistant, notes taking)
8. Checking of finalized soft copies for points 2 to 5 (Round 2: teacher, notes taking)
9. Giving soft copies back to students with notes (7-8) for final corrections (teacher and lab assistant)
10. Taking final submissions CD ; page 7 (lab assistant)
11. Printing and staple documents (lab assistant)
12. Manual code annotation (student)
13. Marking (teacher)
14. Submissions to British council (school)

AS Project Headings:

Problem /Task Identification

Program Design

- Hardware requirements

- Input/Output design

- Data structures (file format)

- Processes (main and separate algos and DFD)

Program Development

- Implementing the Program

- Using good Programming Style

- Programming skills

Testing

- Test strategy & Test data Grid

- Pictorial evidences of tests

Implementation

- Technical documentation

- Installation instructions

AS Project Required Evidences:

Problem /Task Identification

Evidence required

- A statement of who has the problem, briefly what the problem is and the reason a computerised solution would be a good idea
- Very brief description of how the data will be input and what form the output might take

Program Design

Evidence required

- there needs to be a file to store data as this is one of the requirements of the programming, this file needs to be detailed here. Ensure that you have included details of name, size, data types and the data that will be held.
- a diagrammatic representation of your proposed solution, this will probably be a flow diagram to illustrate the logic of your solution
- simple diagrams of what the input and output screens will look like. These should be drawn rather than being printouts as, like the rest of this section, they should be done before the programming is attempted.
- the hardware requirements for the system. Don't go into too much detail, keep it simple. Nobody wants to know about the type of mother board or the size of the memory (after all I'm only going to be using a file with 20 students in it!)
- a written description of how you intend to solve the problem. Keep it short, no more than a side of A4.

Program Development

Implementing the Program

Evidence required

- program code which is annotated by hand to show how it relates to the design section
- printout to show the structure (probably with the data included) of the file that has been created
- printout to show input screens and output screens (perhaps a number on a single page). These outputs should be hand annotated to explain their purpose and under what circumstances they will be used.

Using good Programming Style

Evidence required

- the code must have a recognised and accurate header containing appropriate detail. It is strongly suggested that, unless there is a good reason to the contrary, the list in the syllabus should be used
- the code should illustrate the following techniques
 - data names, procedure and function names should all be sensible and relate to the data being stored
 - data, procedures and functions should be explained when they are declared
 - procedures and functions should be clearly delineated from each other and from the main program
 - indentation should be clearly used to distinguish selection and loop constructs clearly
 - comments should be used widely to explain meanings in the code in order to make the code understandable to a third party observer

Programming skills

Evidence required

For each of the eight techniques

- a valid use within the context of the problem being solved
- sensibly and clearly annotated so that the techniques can be clearly identified within the code.

Testing

Evidence required

- A comprehensive test plan, in table form, specifying the reason for the test, the test data to be used, the expected outcome and the actual outcome.
- Differences between the expected outcome and the actual outcome should be explained, and the measures taken to overcome any problems should be outlined. Students should be encouraged to accept discrepancies as a good thing because it shows that the problem was not puerile and it allows the student to demonstrate their ability to analyse problems that arise and find a way around them. If everything works perfectly first time this is not possible.
- Evidence that the tests have been carried out. This will probably be very simple outputs, at the most complicated it will only be simple screen dumps which can be fitted 2 or 4 to a page.

Implementation

Technical documentation

Evidence required

Note that the evidence for different projects will tend to differ because they will have different requirements.

- descriptions of the data to include reasons for inclusion and the data types/lengths that will be used
- details of the security measures used to protect the data in the file, including passwords and back-up procedures
- diagrams to illustrate the solution (in my case the sensible one is a flowchart, which already exists so I will simply give a page reference)
- annotated program listings (again this already exists)
- hardware and software specifications for use of the system

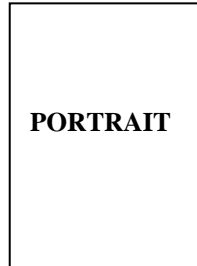
Installation instructions

Evidence required

- instructions how to transfer the code onto the user's computer (probably from a floppy disk or a CD)
- how to set up a password to protect the data (if a password system has been included in the code)
- how to run the program once it has been installed.

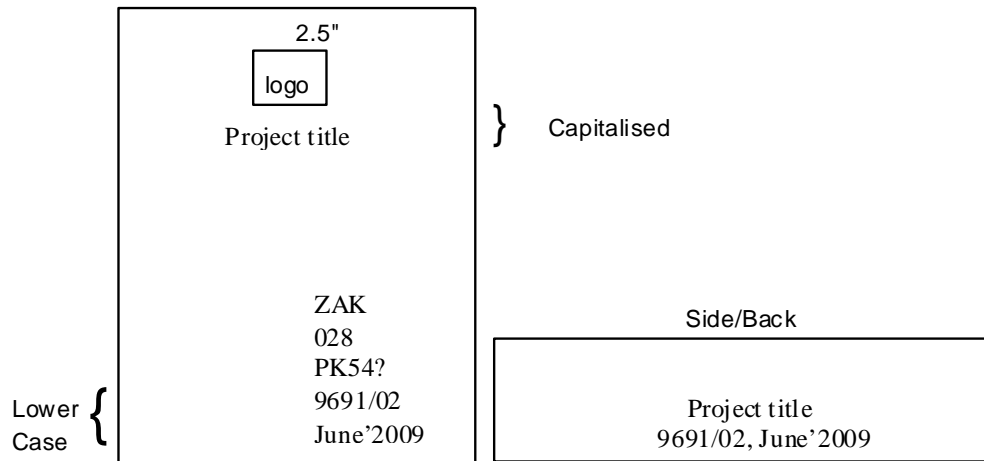
Project documentation and formatting:

Project documentation is required to be prepared on standard **A4** sized paper (210mm X 297mm). Presentation of material within the documentation should, as far as possible, be presented in the portrait form. Landscape forms are allowed mainly for illustrative documentation such as screens design or report design.



- i) Paper weight should be anything equivalent to (or above) 80grams as the paper lighter than this weight tends to be very “transparent”.
- ii) Documentation must be prepared using white colored paper only. Ensure that the documentation is properly and professionally prepared, placing emphasis on language clarity, flow of thoughts, neatness, accurate information, useful information and general report presentation.
- iii) The documentation is required not to be bound in any form as detailed below:

Cover colour	:	White
Size	:	A4
Front cover	:	The Logo and name of <i>The City School</i> - centralized at top (2.5" from top edge)
	:	Approved title - centralized below in capitals
	:	candidate name - at bottom right corner)
		(See below)
	:	All lettering in black (both front cover and side cover)
		Lettering size - 0.25"
		Lettering - see diagram below for upper and lower case letterings
Side/Back Cover	:	Approved Title - centralised along the side
	:	Course and year
	Format: 9691/01	: Lettering size = 0.25"
		All lettering are capitalized



Text Paper Margins	:	top 1", bottom 0.5', left 1", right 0.5"
Font & size	:	Office 2007: Calibri 12 for normal Courier New 12 for coding (Comments font can be changed to Calibri 12)
		Any Office: Times New Roman 12 for normal Courier New 12 for coding (Comments font can be changed to T.N.R 12)
Images (Captured or drawn)	:	Should be centralized and thoroughly annotated
Headings	:	Standard Heading 1, 2, 3 must be used
Header and Footer	:	As required.
Alignment	:	All text, code and headings must be left aligned
Bullets and Numbering	:	standard 1,2,3,... and normal bullet mark
Table of Content and page Numbers:		All pages (not including cover or T.O.C) must be numbered As "09/40" and a proper automatic TOC must be generated and shown after cover page.
Printing	:	No printing will be done by students. Documents will be printed once the final documentation is ready and would not be handed over to students to take away.
Manual Annotation	:	Once printed, document will be handed over to the students-- for one period max. for hand annotating the code for eight 9691/02 CIE requirements.

Submission

: Only a CD, will be turned as below.
Root Folder named after candidate name and id; like "ZAK-028"
Two subfolders namely "Documentation" & "RAW"
"Documentation" folder holds single word document named
after project name
"RAW" folder holds Program and data files and no sub folders.