

MARK SCHEME for the October/November 2006 question paper

9691 COMPUTING

9691/02 Paper 2 (Practical Programming Project),
maximum raw mark 50

This mark scheme is published as an aid to teachers and students, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began.

All Examiners are instructed that alternative correct answers and unexpected approaches in candidates' scripts must be given marks that fairly reflect the relevant knowledge and skills demonstrated.

Mark schemes must be read in conjunction with the question papers and the report on the examination.

The grade thresholds for various grades are published in the report on the examination for most IGCSE, GCE Advanced Level and Advanced Subsidiary Level syllabuses.

- CIE will not enter into discussions or correspondence in connection with these mark schemes.

CIE is publishing the mark schemes for the October/November 2006 question papers for most IGCSE, GCE Advanced Level and Advanced Subsidiary Level syllabuses and some Ordinary Level syllabuses.

Page 2	Mark Scheme	Syllabus	Paper
	GCE A/AS LEVEL - OCT/NOV 2006	9691	

Practical Programming Projects are assessed as follows:

Problem/Task identification	[6 marks]
Program Design	[6 marks]
Program Development	[28 marks]
Testing	[8 marks]
Implementation	[6 marks]

(a) Problem/Task Identification **[Total: 2 marks]**

A candidate should not expect the Examiners to be familiar with the problem/task that has been chosen. There should be a brief description of the problem/task and a clear statement of the form of data input should be given together with the required output.

- 1 Outline of the problem to be solved.
- 2 Description of the problem to be solved including the data input and the desired output.

(b) Program Design **[Total: 6 marks]**

A detailed program design (including diagrams as appropriate) should be produced. Proposed record, file and data structures should be described. Design of input formats (with examples of screen layouts) and output formats should be included here. A detailed description of processes should also be included. The hardware requirements must be stated.

- 1 – 2 Some vague discussion of what the program will do with a brief diagrammatic representation.
- 3 – 4 There is an outline of a design specification, including mock ups of inputs and outputs, process model described (including a diagram: structure diagram, data flow diagram or system flowchart). However there is a lack of completeness with omissions from the process model, inputs and outputs. Data structures have been identified but there may be inadequate detail. Or there may be some errors or logical inconsistencies, for example validation specified may be inadequate or field lengths incorrect.
- 5 – 6 A detailed and complete design specification, which is logically correct. There are also detailed written descriptions of any processes/modules and a clear, complete definition of any data structures.

(c) Program Development **[Total: 28 marks]**

(i) Implementing the program **[6 marks]**

There is evidence that the program produces the desired results. The finished program should relate clearly to the design work.

- 1 – 2 Program listings are provided in the form of printouts. The developed solution does not fulfil the design specification. A teacher may award up to 2 marks if they have been shown the system working satisfactorily and there is no hard evidence in the project report.
- 3 – 4 Program listings are provided in the form of printouts. Data structures are illustrated as part of the listings where appropriate, detailing their purpose. The developed solution has logical flaws and is only slightly related to the design.
- 5 – 6 Program listings are provided in the form of printouts. Data structures are illustrated as part of the listings where appropriate, detailing their purpose. There is a full set of printouts showing input and output as well as data structures. The program is clearly related to the design. All hardcopy is fully annotated and cross-referenced.

Page 3	Mark Scheme	Syllabus Paper
	GCE A/AS LEVEL - OCT/NOV 2006	9691

(ii) Using Good Programming Style

Program listings should be easily readable. There should be a 'header' identifying the program. The header should contain the program name, author, school or college, programming language used, revision number, date and purpose. The program should be self-documenting. All data declarations should be accompanied by explanatory comments; identifiers should have meaningful variable names; programs, functions and procedures should be clearly named, well separated and fully commented; suitable indentation should be used to set out the programming constructs used.

Program listings must contain all the code written by the candidate. If any library routines or automatically generated code is included this must be clearly identified and not taken into account for assessment purposes.

- 1 – 2 Program listings are not easily readable and have few comments or comments are handwritten on the listing.
- 3 – 4 The program listing shows some attention to good style but not all elements are included.
- 5 – 6 The program listing is easily readable and shows considerable attention to good style.

(iii) Programming Skills

[16 marks]

Candidates must demonstrate their use of the following programming skills.

- arrays and/or records
- different data types
- selection
 - iteration
- procedures
 - functions
- searching techniques
- files

For each of the above skills:

- 1 mark for a valid use
- 1 mark for correct annotation within the code

(d) Testing

[Total: 8 marks]

It is the responsibility of the candidates to produce evidence of their development work and for producing a test plan for the system. It is vital to produce test cases and to show that they work. To do this it is necessary, not only to have test data, but to know what the expected results are with that data.

An attempt should be made to show that all parts of the program have been tested, including those sections dealing with unexpected or invalid data as well as extreme cases. Showing that many other cases of test data are likely to work - by including the outputs that they produce - is another important feature. Evidence of testing is essential. Comments by teachers and others are of value, but the test plan must be supported by evidence in the report of a properly designed testing process. The examiner must be left in no doubt the program actually works.

- 1 – 2 A collection of hardcopy test run outputs with no test plan, or a test plan with no hardcopy evidence may also be present. A teacher may award up to 2 marks if they have been shown the program working satisfactorily and there is no hard evidence in the project report.
- 3 – 4 There is little evidence of testing with a badly developed test plan with clear omissions. There is no description of the relationship between the structure of the development work and the testing in evidence.
- 5 – 7 The developed solution partially fulfils the design specification. There should be at least eight different test runs together with a test plan and hardcopy evidence. However, the test plan has omissions in it and/or not all cases have been tested.
- 8 A comprehensive test plan, with evidence of each test run is present in the report, together with the expected output. The test plan should cover all aspects of the programming designed to cover the topics in c (iii) and demonstrate their effective use within the boundaries of the solution.

Page 4	Mark Scheme	Syllabus	Paper
	GCE A/AS LEVEL - OCT/NOV 2006	9691	

(e) Implementation

(i) Technical Documentation

Much of the documentation will have been produced as a by-product of design and development work and also as part of writing up the report to date. The following should be included: record, file and data structures used; data dictionary; data flow (or navigation paths); annotated program listings; detailed flowcharts; details of the algorithms and formulae used. These should be fully annotated since this is important for subsequent development of the system. The specifications of the hardware and software on which the system was developed should be included.

Since the contents in the technical documentation will differ from one project to another, professional judgement as to what would be necessary for another analyst to maintain and develop the program has to be made.

- 1 – 2 Some items are present with some annotation attempted.
- 3 One or two omissions, but the rest is present and annotation is used sensibly.
- 4 No major omissions, with all parts fully annotated.

(ii) Installation Instructions

[2 marks]

Clear guidance, as friendly as possible, should be given to the user on how to install the program ready for use.

- 1 Sensible instructions on how to install the program for use.
- 2 Comprehensive, well illustrated instructions on how to install the program for use.