



UNIVERSITY OF CAMBRIDGE INTERNATIONAL EXAMINATIONS  
 General Certificate of Education  
 Advanced Subsidiary Level and Advanced Level

CANDIDATE NAME

CENTRE NUMBER 

--	--	--	--	--

CANDIDATE NUMBER 

--	--	--	--

\* 1 4 5 9 1 3 0 1 1 5 \*

**COMPUTING** **9691/23**  
 Paper 2 **October/November 2011**  
**2 hours**

Candidates answer on the Question Paper.  
 No additional materials are required.

**READ THESE INSTRUCTIONS FIRST**

Write your Centre number, candidate number and name on all the work you hand in.  
 Write in dark blue or black pen.  
 You may use a soft pencil for any diagrams, graphs or rough working.  
 Do not use staples, paper clips, highlighters, glue or correction fluid.  
**DO NOT WRITE IN ANY BARCODES.**

Answer **all** questions.

At the end of the examination, fasten all your work securely together.  
 The number of marks is given in brackets [ ] at the end of each question or part question.

This document consists of **12** printed pages.



1 Nathan is designing a software solution for stock control in a computer shop. He has a colleague, called Andre, who will help him write the program. Nathan decides to modularise the solution.

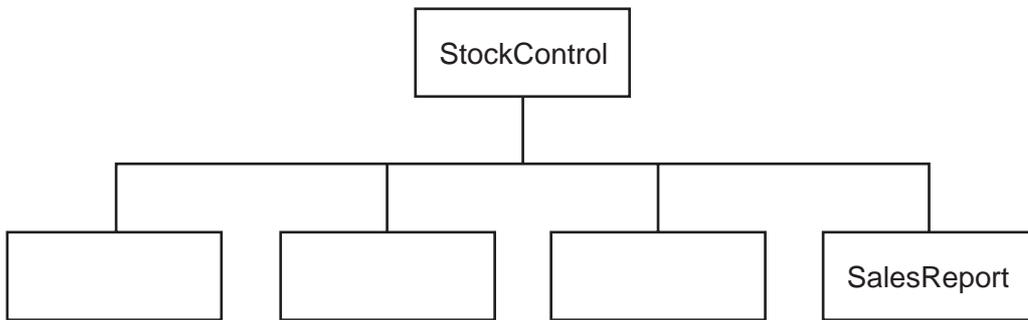
(a) State why modularisation is a good idea.

.....  
..... [1]

(b) As the first step in his design he splits the solution into the following main areas:

Initialisation, StockOrdering, Sales, SalesReport.

Complete the following structure diagram.



[1]

(c) SalesReport is made up of two modules, MonthlySalesReport and AnnualSalesReport.

Add them to the structure diagram in (b). [2]

(d) Nathan will write the MonthlySalesReport module and Andre will write the AnnualSalesReport module. Nathan will use the identifier *Profit* for the monthly profit, and Andre will use the identifier *Profit* for the annual profit.

Explain how they can both use the same identifier and not cause a problem when the program is run.

.....  
.....  
.....  
..... [2]

(e) (i) Explain why there are some words that cannot be used as identifiers.

.....  
..... [1]

(ii) Both programmers need to choose other identifiers that they use.

State **three** rules of a high-level programming language that restrict the choice of identifiers.

Language .....

Rule 1 .....

.....

Rule 2 .....

.....

Rule 3 .....

..... [3]

(iii) Give an example of an **invalid** identifier.

Language .....

Invalid identifier ..... [1]

(f) While they are working on the SalesReport modules, they use the expression:

$$\text{PercentageIncrease} = a+b/100$$

(i) Work out the value of PercentageIncrease when a is 600 and b is 400.

.....  
..... [1]

The programmer has made an error in the code. When a is 600 and b is 400, the expected result is 10.

(ii) Rewrite the line of code with added parentheses to give the expected result.

.....  
..... [1]

(iii) Name the type of testing strategy which identified this error.

..... [1]

(g) One type of test data is invalid data.

(i) Name the other **two** types.

1 .....

2 ..... [2]

(ii) Andre has written the Sales module, which now needs testing.

- InvoiceNumber has the format yy-nnnn e.g. 11-0035 is the 35th invoice of the year 2011
- PromotionCode can be 'gold', 'silver' or 'bronze' only

Give **six** different items of test data, other than invalid data, which thoroughly test the two rules given above. Give a reason for each choice.

InvoiceNumber	PromotionCode	Reason
130092		Invalid data for InvoiceNumber
	glod	Invalid data for PromotionCode

[6]

**(h) (i)** Write a Boolean expression to validate PromotionCode.

.....  
.....  
.....  
..... [2]

**(ii)** Describe the problems that could occur if PromotionCode is not validated.

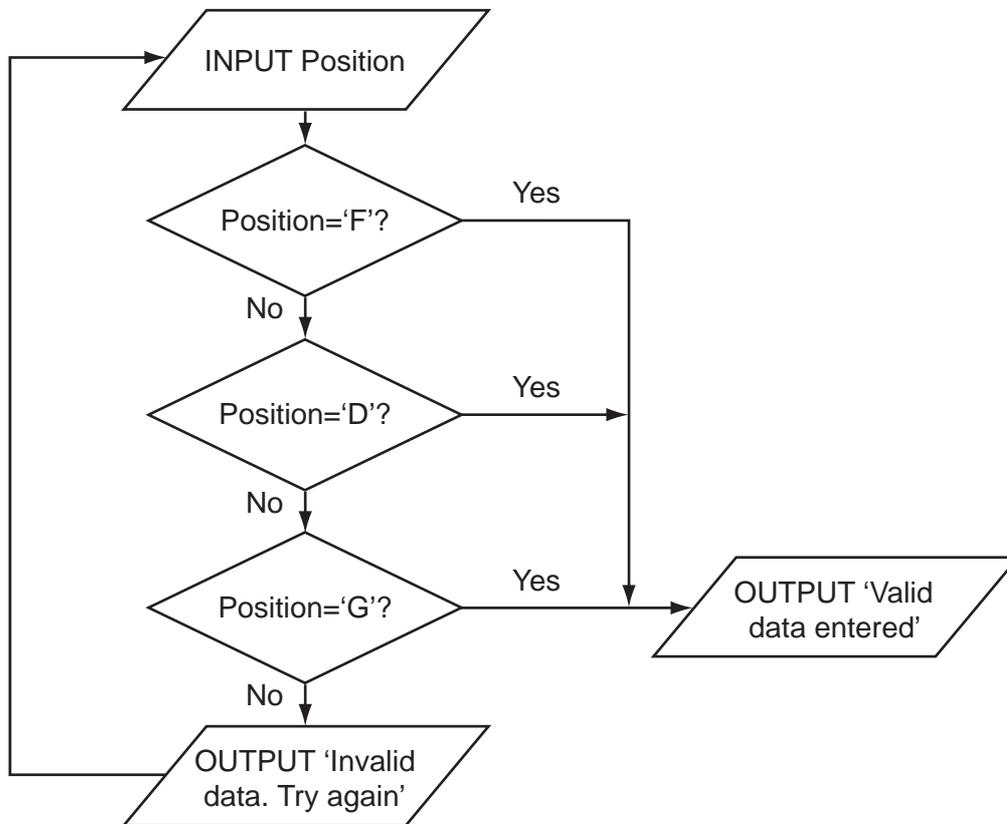
.....  
.....  
.....  
..... [2]

**(iii)** Design a screen which lets the user choose which type of sales report they would like to view.

[2]

- 2 Ahmed is writing a program to record the data of members of the school hockey club. The input data will be validated. One input will be the playing position of each member. Is he/she a forward, defender or goalkeeper? These will be entered as F, D or G. The flowchart for the validation of the playing position is shown below.

For  
Examiner's  
Use



- (a) (i) What is the output when the input is 'G'?

..... [1]

- (ii) What is the output when the input is 'K'?

..... [1]

(b) In a high-level language, write the code that will produce the validation process shown in the flowchart.

Language .....

Code .....

.....

.....

.....

.....

.....

.....

..... [5]

(c) The three basic programming constructs used to control the flow of information are: sequence, selection and iteration.

State the **two** constructs that are used in your code.

.....

..... [1]

(d) Describe what is meant by iteration.

.....

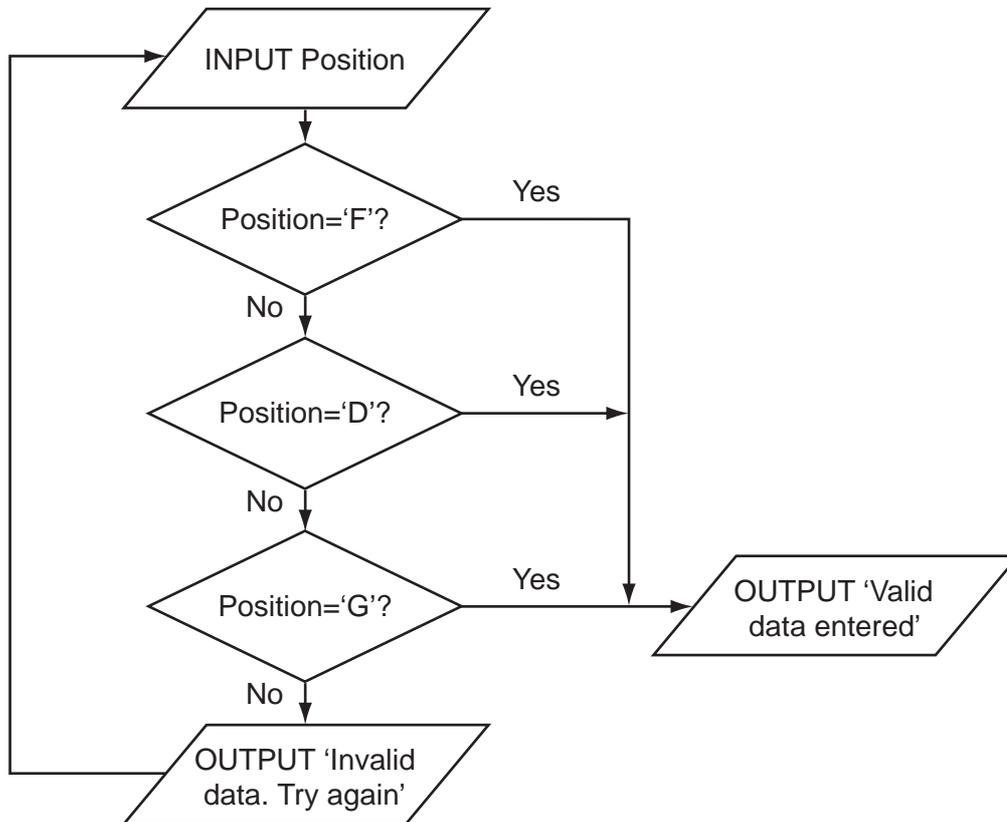
.....

.....

..... [2]

- (e) Ahmed thinks it will be a good idea to allow only four attempts at getting the input data correct. If it is not a valid entry after four attempts a message 'Please enter D, F or G' should be output. Modify the flowchart to include this additional check.

For  
Examiner's  
Use



[5]

- (f) Ahmed needs to store more information about the players. He creates a record structure that contains PlayerID (a whole number between 1 and 50), Sex (m or f), PlayerName, Position (F, D or G), and DateOfBirth. Complete the table.

Field Name	Data Type	Field Size (bytes)
PlayerID		
Sex		
PlayerName		
Position		
DateofBirth		

[10]

(g) The club has 45 members. Ahmed stores the records in an array called Club. To calculate the number of goalkeepers he designs this pseudocode:

For  
Examiner's  
Use

```

Gtotal ← 0
Index ← 1
REPEAT
    IF Club[Index].Position = 'G'
        THEN
            Gtotal ← Gtotal + 1
            Index ← Index + 1
UNTIL Index = 45

```

This will only look at the first 44 records in the array.

(i) State the name of this type of error.

..... [1]

(ii) State the line that needs changing.

..... [1]

(iii) Rewrite the line to ensure the pseudocode considers all 45 records.

..... [1]

(h) Write this updated pseudocode using a FOR loop ensuring that it will check all records in the array.

.....

.....

.....

.....

.....

.....

.....

.....

..... [3]

- 3 Aisha is learning about manipulating strings in a high-level programming language. She has an idea that she wants to try. She produces the following design in pseudocode:

```

FUNCTION Surprise(s : string): string
x ← 1
REPEAT
q[x] ← s[x]
x ← x + 1
UNTIL s[x] = ' '
Surprise ← q
ENDFUNCTION
    
```

For  
Examiner's  
Use

- (a) Complete the trace of this function for the function call Surprise('CHO JABA').

s	x	q[1]	q[2]	q[3]	q[4]	Surprise
CHO JABA						
	1					

[3]

- (b) State the purpose of this function.

.....  
 ..... [1]

(c) Describe what `Surprise ← q` does.

.....

.....

.....

..... [2]

(d) Describe the features of any function.

.....

.....

.....

.....

.....

..... [3]

(e) Aisha's pseudocode is not easily understood.

(i) She could have annotated her code with suitable comments.

Write a comment to annotate the line:

```
UNTIL s[x] = ' '
```

.....

.....

..... [2]

(ii) State **two** techniques, other than annotation, that she should use to improve the understanding of her pseudocode.

1 .....

.....

2 .....

..... [2]

(f) Aisha uses an expression:

```
String1 > String2
```

Explain how strings are compared by the processor.

.....

.....

.....

.....

.....

.....

..... [3]

For  
Examiner's  
Use

Permission to reproduce items where third-party owned material protected by copyright is included has been sought and cleared where possible. Every reasonable effort has been made by the publisher (UCLES) to trace copyright holders, but if any items requiring clearance have unwittingly been included, the publisher will be pleased to make amends at the earliest possible opportunity.

University of Cambridge International Examinations is part of the Cambridge Assessment Group. Cambridge Assessment is the brand name of University of Cambridge Local Examinations Syndicate (UCLES), which is itself a department of the University of Cambridge.