



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

CANDIDATE  
NAME

CENTRE  
NUMBER

--	--	--	--	--

CANDIDATE  
NUMBER

--	--	--	--

**COMPUTING**

**9691/22**

Paper 2

**October/November 2013**

**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 **14** printed pages and **2** blank pages.



1 Jemma is designing a program that will work out the end of year bonuses for her employees. The main steps are:

- input employee's data
- calculate the bonus
- calculate deductions
  - tax
  - optional contribution to charity
- print out the bonus

**(a)** Draw a structure diagram to show the modules that will be needed.

*For  
Examiner's  
Use*

[3]

(b) Each module is programmed by Jemma.

Describe **three** program coding techniques she should use to help Freddie who will maintain the code.

*For  
Examiner's  
Use*

Technique 1 .....

.....

.....

.....

Technique 2 .....

.....

.....

.....

Technique 3 .....

.....

.....

..... [6]

- (c) (i) Jemma is using the variable `Percentage` in one module and a different variable called `Percentage` in another module.

Explain the feature of a high-level programming language that avoids any possible conflict.

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

- (ii) Some modules require data values that originate from another module.

Explain the feature of a high-level programming language that enables this.

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

*For  
Examiner's  
Use*

(d) Jemma is designing a range validation check for the input of an employee's pay. The pay range depends on the employee's job type, which may be P(part-time), F(full-time) or C(commission only).

*For  
Examiner's  
Use*

- A part-time employee will earn between \$100 and \$10 000 a year.
- A full-time employee will earn between \$5 000 and \$50 000 a year.
- A commission only employee will earn between \$0 and \$80 000 a year.

Complete the table showing **five** more rows of test data. Give a different reason for each, describing what is being tested.

Job type	Pay	Reason
F	25 000	Normal data – within pay range for full-time

[5]

2 James plans to store the titles and authors of his favourite ebooks in an array. He has 56 ebooks at the moment and expects to reach 150.

A two-dimensional array `MyEbooks` is to be used.

(a) In a high-level programming language, write the array declaration statement.

Programming language .....

Declaration ..... [3]

(b) (i) Why is it good programming practice to initialise the array `MyEbooks`?

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

(ii) Write program code to initialise the array `MyEbooks`.

Programming language .....

Code .....

.....  
.....  
.....  
.....  
.....  
.....  
.....  
..... [3]

(c) At present when the program terminates all the ebook data are lost.

How can these data be stored and made available every time the program is run?

.....  
.....  
.....  
.....  
.....  
..... [3]

- (d) James' friend, Jatinder, uses his idea, but decides to store more information about each of her ebooks.

Complete the following table. Use a single numerical value for Field size.

Identifier	Description	Data type	Field size (Bytes)
EbookID	a unique ebook ID, a whole number between 1 and 500		
BookTitle	the title of the ebook	STRING	
Author	the author of the ebook		
DateBought	date bought		
Cost	price paid		
Fiction	fiction or non-fiction		

[8]

For  
Examiner's  
Use

- 3 Aisha wants to write a program that checks the password to her personal computer. The program should check each attempt to enter the password correctly and should terminate after three wrong attempts.

She wants the log-in screen to display:

- a prompt to enter the password
- space to enter the password
- how many attempts have been made
- if the log-in has been successful or not
- a means of cancelling the log-in process

(a) Draw a suitable layout for the screen.

Aisha's log-in screen

[5]

Aisha writes her first try at designing the code in pseudocode. She wants the password to be "Aisha", her name.

```

1  Attempt ← 0
2  REPEAT
3      INPUT Password
4      Attempt ← Attempt + 1
5  UNTIL (Password = "Aisha") OR (Attempt > 3)
6  IF Password = "Aisha"
7      THEN
8          OUTPUT "Password correct"
9      ELSE
10         OUTPUT "No valid password entered"
11  ENDIF

```



(b) (i) Complete the trace table using "Aisha" as input.

Attempt	Password	Password = "Aisha"	Attempt > 3	Password = "Aisha" OR Attempt > 3	Output
0					
	Aisha				

[3]

(ii) Complete the trace table using the following sequence of inputs:

"aisha", "Asha", "AISHA"

Attempt	Password	Password = "Aisha"	Attempt > 3	Password = "Aisha" OR Attempt > 3	Output
0					

[5]

For  
Examiner's  
Use

(c) This piece of code does not do what Aisha intended. There is an error.

(i) State the type of error.

..... [1]

(ii) There are several ways to correct this. One is to change line 5.

Rewrite line 5.

..... [1]

(d) State why someone's name is not a good password.

..... [1]

(e) Describe how to choose a more secure password.

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

**Question 3 continues on page 12.**



- (ii) Rewrite your IF statements using a CASE/SELECT statement to implement the menu options.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

*For  
Examiner's  
Use*

4 Aisha writes the following pseudocode for a recursive function that works out the greatest common divisor of two positive integers:

```

0  FUNCTION Divisor(x, y)
1      IF y = 0
2          THEN
3              RETURN x
4          ELSE
5              x ← x MOD y
6              RETURN Divisor(y, x)
7      ENDIF
8  ENDFUNCTION
    
```

(a) (i) Trace the call `Divisor(8, 2)`.

Line number	x	y	RETURN
0	8	2	

[2]

(ii) Trace the call `Divisor(38, 7)`.

Line number	x	y	RETURN
0	38	7	

[4]

(b) (i) State the purpose of line 1.

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

(ii) State what would happen if line 1 were not present.

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

(c) What happens if  $x$  is less than  $y$ ?

.....  
.....  
.....  
.....  
.....  
..... [3]

**BLANK PAGE**

---

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.