Cambridge International A Level

Cambridge International Examinations
Cambridge International Advanced Level

CANDIDATE NAME

CENTRE NUMBER

CANDIDATE NUMBER

COMPUTER SCIENCE **9608/42**

Paper 4 Further Problem-solving and Programming Skills **October/November 2015**

2 hours

Candidates answer on the Question Paper.

No Additional Materials are required.

No calculators allowed.

READ THESE INSTRUCTIONS FIRST

Write your Centre number, candidate number and name in the spaces at the top of this page.
Write in dark blue or black pen.
You may use an HB pencil for any diagrams, graphs or rough working.
Do not use staples, paper clips, glue or correction fluid.
DO **NOT** WRITE IN ANY BARCODES.

Answer **all** questions.
No marks will be awarded for using brand names of software packages or hardware.

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.

The maximum number of marks is 75.

This document consists of **15** printed pages and **1** blank page.

CAMBRIDGE International Examinations

[Turn over

Throughout the paper you will be asked to write either **pseudocode** or **program code**.

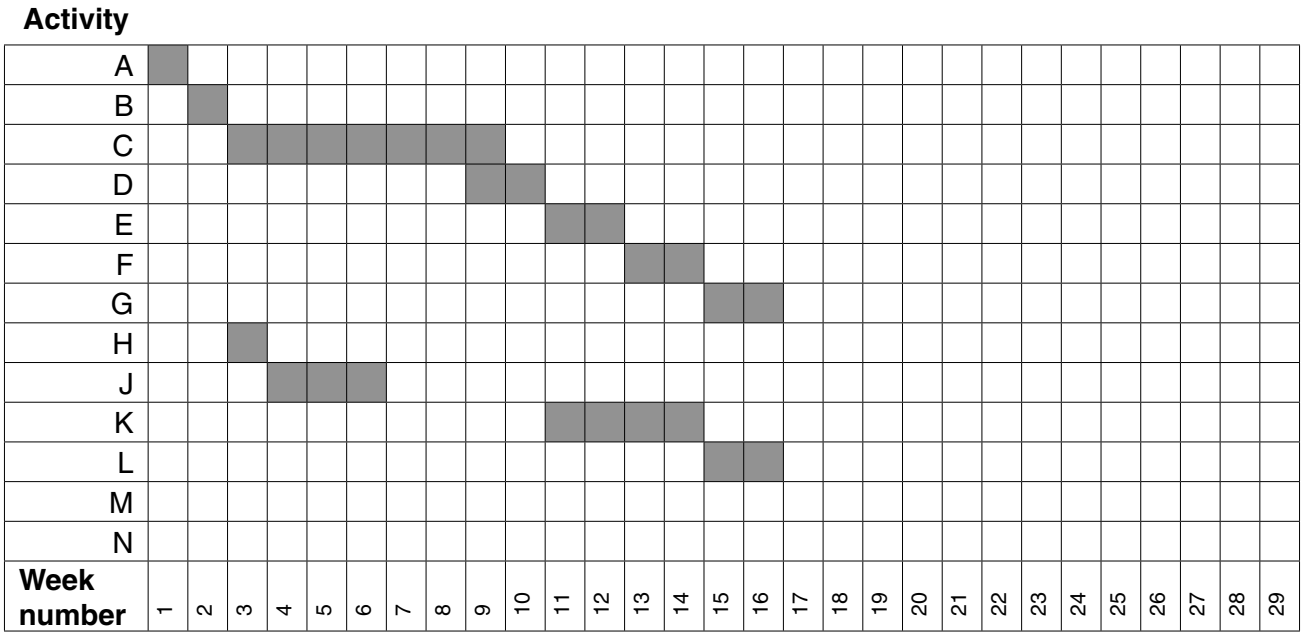
Complete the statement to indicate which high-level programming language you will use.

Programming language .....

1 A large software house has been asked to supply a computerised solution for a business. The project manager has drawn up a list of activities and their likely duration.

Activity	Description	Weeks to complete
A	Write requirement specification	1
B	Produce program design	1
C	Write module code	7
D	Module testing	2
E	Integration testing	2
F	Alpha testing	2
G	Install software and carry out acceptance testing	2
H	Research and order hardware	1
J	Install delivered hardware	3
K	Write technical documentation	4
L	Write user training guide	2
M	Train users on installed hardware and software	1
N	Sign off final system	1

(a) From this data a GANTT chart is constructed.



(i) Complete the GANTT chart by adding activities M and N. [2]

(ii) State the earliest completion date.

Week number ..... [1]

(b) There are problems with the progress of the project:

- Activity E showed that the code contained major errors. The senior programmer now estimates that:
  - further module coding will require another 2 weeks
  - further module testing will require another 2 weeks
  - further integration testing will require another 2 weeks
- The hardware delivery is delayed by 16 weeks.

A revised GANTT chart is now required.

(i) Complete the chart in the grid below.

Activity

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
A																													
B																													
C																													
D																													
E																													
F																													
G																													
H																													
J																													
K																													
L																													
M																													
N																													
Week number																													

[9]

(ii) State the new estimated completion date.

Week number ..... [1]

[Turn over

2 A declarative programming language is used to represent the following facts and rules:

```
01  male(ahmed).
02  male(raul).
03  male(ali).
04  male(philippe).
05  female(aisha).
06  female(gina).
07  female(meena).
08  parent(ahmed, raul).
09  parent(aisha, raul).
10  parent(ahmed, philippe).
11  parent(aisha, philippe).
12  parent(ahmed, gina).
13  parent(aisha, gina).
14  mother(A, B) IF female(A) AND parent(A, B).
```

These clauses have the following meaning:

Clause	Explanation
01	Ahmed is male
05	Aisha is female
08	Ahmed is a parent of Raul
14	A is the mother of B if A is female and A is a parent of B

(a) More facts are to be included.

Ali and Meena are the parents of Ahmed.

Write the additional clauses to record this.

15 .....

16 ..... [2]

(b) Using the variable C, the goal

```
parent(ahmed, C)
```

returns

```
C =  raul, philippe, gina
```

Write the result returned by the goal

```
parent(P, gina)
```

P = ... [2]

(c) Use the variable `M` to write the goal to find the mother of Gina.

..... [1]

(d) Write the rule to show that `F` is the father of `C`.

```
father(F, C)
```

`IF` .....

..... [2]

(e) Write the rule to show that `X` is a brother of `Y`.

```
brother(X, Y)
```

`IF` .....

.....

.....

.....

..... [4]

3 A college has two types of student: full-time and part-time.

All students have their name and date of birth recorded.

A full-time student has their address and telephone number recorded.

A part-time student attends one or more courses. A fee is charged for each course. The number of courses a part-time student attends is recorded, along with the total fee and whether or not the fee has been paid.

The college needs a program to process data about its students. The program will use an object-oriented programming language.

(a) Complete the class diagram showing the appropriate properties and methods.

Student
StudentName: STRING ....
ShowStudentName() ....

FullTimeStudent
Address: STRING ....
Constructor() ShowAddress() ....

PartTimeStudent
....
....

[7]

(b) Write **program code**:

(i) for the class definition for the superclass `Student`.

Programming language .....

.....

.....

.....

.....

.....

.....

.....

..... [2]

(ii) for the class definition for the subclass `FullTimeStudent`.

Programming language .....

.....

.....

.....

.....

.....

.....

.....

..... [3]

(iii) to create a new instance of `FullTimeStudent` with:

- identifier: `NewStudent`
- name: A. Nyone
- date of birth: 12/11/1990
- telephone number: 099111

Programming language .....

.....

.....

.....

.....

.....

..... [3]

4 A dictionary Abstract Data Type (ADT) has these associated operations:

- Create dictionary `(CreateDictionary)`
- Add key-value pair to dictionary `(Add)`
- Delete key-value pair from dictionary `(Delete)`
- Lookup value `(Lookup)`

The dictionary ADT is to be implemented as a two-dimensional array. This stores key-value pairs.

The pseudocode statement

```
DECLARE Dictionary : Array[1:2000, 1:2] OF STRING
```

reserves space for 2000 key-value pairs in array `Dictionary`.

The `CreateDictionary` operation initialises all elements of `Dictionary` to the empty string.

(a) The hashing function `Hash` is to extract the first letter of the key and return the position of this letter in the alphabet. For example `Hash("Action")` will return the integer value 1.
(Note: The ASCII code for the letter A is 65.)

Complete the pseudocode:

```
FUNCTION Hash (.....) RETURNS .....
   DECLARE Number : INTEGER
   Number ←.....
   .....
ENDFUNCTION
```

[5]

(b) The algorithm for adding a new key-value pair to the dictionary is written, using pseudocode, as a procedure.

```
PROCEDURE Add(NewKey : STRING, NewValue : STRING)
   Index ← Hash(NewKey)
   Dictionary[Index, 1] ← NewKey          // store the key
   Dictionary[Index, 2] ← NewValue        // store the value
ENDPROCEDURE
```

An English-German dictionary of Computing terms is to be set up.

(i) Dry-run the following procedure calls by writing the keys and values in the correct elements of `Dictionary`.

```
Add("File", "Datei")
Add("Disk", "Platte")
Add("Error", "Fehler")
Add("Computer", "Rechner")
```

Dictionary

Index	Key	Value
1		
2		
3		
4		
5		
6		
7		
8		
:		
:		
1999		
2000		

[2]

(ii) Another procedure call is made: `Add("Drive", "Laufwerk")`

Explain the problem that occurs when this key-value pair is saved.

.....

.....

.....

..... [2]

(iii) Describe a method to handle the problem identified in **part (b)(ii)**.

.....

.....

.....

.....[2]

(iv) Write **pseudocode** to implement the method you described in **part (b) (iii)**. Choose line numbers to indicate where your pseudocode should be inserted in the given pseudocode.

```
10  PROCEDURE Add(NewKey : STRING, NewValue : STRING)
20      Index ← Hash(NewKey)
30      Dictionary[Index, 1] ← NewKey      // store the key
40      Dictionary[Index, 2] ← NewValue    // store the value
50  ENDPROCEDURE
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

Question 5 begins on page 12.

5 The table shows assembly language instructions for a processor which has one general purpose register – the Accumulator (ACC).

Instruction		Explanation
Op Code	**Operand**	
LDM	#n	Immediate addressing. Load the number n to ACC
LDD	<address>	Direct addressing. Load the contents of the given address to ACC
STO	<address>	Store the contents of ACC at the given address
ADD	<address>	Add the contents of the given address to the ACC
INC	<register>	Add 1 to the contents of the register
CMP	<address>	Compare the contents of ACC with the contents of <address>
JPN	<address>	Following a compare instruction, jump to <address> if the compare was False
END		Return control to the operating system

(a) (i) Dry-run this assembly language program using the trace table.

500	LDD 512
501	ADD 509
502	STO 512
503	LDD 511
504	INC ACC
505	STO 511
506	CMP 510
507	JPN 500
508	END
509	7
510	3
511	0
512	0

Trace table

	Memory address			
Accumulator	**509**	**510**	**511**	**512**
	7	3	0	0

[5]

(ii) Explain the role address 511 has in this assembly language program.

.....

.....

.....

..... [2]

(b) Using opcodes from the given table, write instructions to set the value at address 509 to 12.

.....

.....

.....

..... [2]

6 A company keeps details of its stock items in a file of records, `StockFile`.

(a) The record fields are the `ProductCode`, the `Price` and the `NumberInStock`.

Write the **program code** to declare the record structure `StockItem`.

Programming language .....

.....

.....

.....

.....

.....

.....

..... [4]

(b) Before records can be read from file `StockFile`, the file needs to be opened.

(i) Complete the pseudocode.

```
01 TRY
02     OPENFILE .....
03 EXCEPT
04     .....
05 ENDTRY
```

[2]

(ii) Explain the reason for including lines `01`, `03`, `04`, `05`.

.....

.....

.....

..... [2]

(c) A stock report program uses a variable of type `StockItem` declared as follows:

```
DECLARE ThisStockItem : Stockitem
```

The program reads each record in the file `StockFile` in turn.

The program outputs the fields `ProductCode` and `NumberInStock` for each record.

Write **pseudocode** for this.

.....

.....

.....

.....

.....

.....

.....

..... [4]

BLANK PAGE