Cambridge International A Level

Cambridge International Examinations
Cambridge International Advanced Level

COMPUTER SCIENCE 9608/41

Paper 4 Further Problem-solving and Programming Skills May/June 2016

PRE-RELEASE MATERIAL

3291757689

No Additional Materials are required.

This material should be given to the relevant teachers and candidates as soon as it has been received at the Centre.

READ THESE INSTRUCTIONS FIRST

Candidates should use this material in preparation for the examination. Candidates should attempt the practical programming tasks using their chosen high-level, procedural programming language.

This document consists of **10** printed pages and **2** blank pages.

DC (NH/SG) 106700/5

CAMBRIDGE
International Examinations

This material is intended to be read by teachers and candidates prior to the June 2016 examination for 9608 Paper 4.

Candidates will also benefit from using pre-release materials from previous examinations. These are available on the teacher support site.

Reminders
The syllabus states:

- there will be questions on the examination paper which do not relate to this pre-release material
- you must choose a high-level programming language from this list:
  - Visual Basic (console mode)
  - Python
  - Pascal / Delphi (console mode)

Note: A mark of zero will be awarded if a programming language other than those listed is used.

The practical skills covered in Paper 2 are a precursor to those required in Paper 4. It is therefore recommended that the high-level programming language chosen for this paper is the same as that for Paper 2. This allows for sufficient expertise to be acquired with the opportunity for extensive practice.

Questions on the examination paper may ask the candidate to write:

- structured English
- pseudocode
- program code

A program flowchart should be considered as an alternative to pseudocode for the documenting of an algorithm design.

Candidates should be confident with:

- the presentation of an algorithm using either a program flowchart or pseudocode
- the production of a program flowchart from given pseudocode (or the reverse)

Declaration of variables

The syllabus document shows the syntax expected for a declaration statement in pseudocode.

```
DECLARE <identifier> : <data type>
```

It is appreciated that candidates who use Python as their chosen language will not be familiar with the concept of declaring all variables with their data type before they are used.

However, answers using Python will be required, instead of a declaration statement, to include a comment statement, documenting the identifier name with its intended data type.

The question will clarify this with a wording and answer layout such as:

> **(i)** Write **program code** for the new design.
>
> *Visual Basic and Pascal: You should include declaration statements for variables.*
> *Python: You should show a comment statement for each variable used with its data type.*
>
> Programming language ....
>
> ....
>
> ....

Structured English – Variables

An algorithm written in pseudocode requires that all variables have been identified. This may not be the case if the initial attempt at the algorithm design is in structured English. The candidate will then be required to identify the variables from the question.

Jackson Structured Programming (JSP)

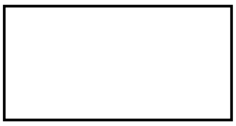
Key focus: **JSP data structure diagrams**

A JSP data structure diagram is used to represent the data used by a problem.

The tasks that follow introduce you to the symbols used.

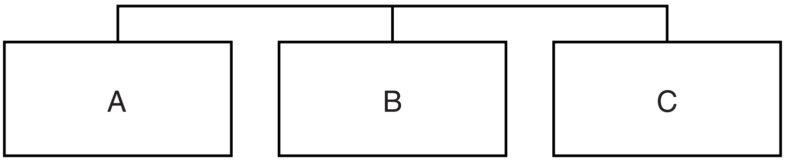
Task 1

1.1 This symbol is used in a JSP data structure diagram. What is it used to represent?



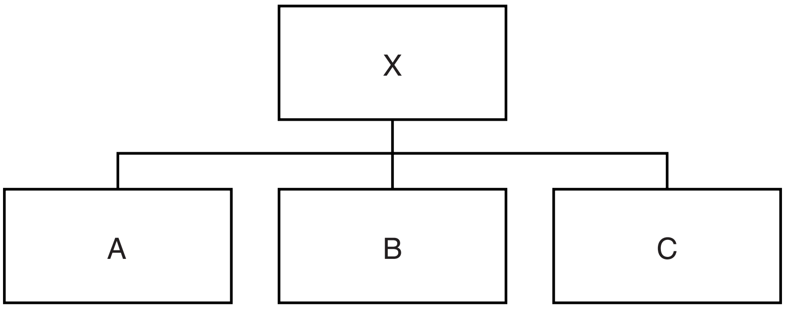
- a data item or data component
- a subroutine
- a procedure
- a function

1.2 Which **one** statement correctly describes this diagram?



- It shows a sequence of data items which can be presented in any order.
- It shows alternative data items.
- It shows a sequence of data items structured in the order A, then B, then C.

1.3 Which are the **three** true statements for the JSP data structure diagram shown below?



1 Data structure X is made up of items A, then B, then C.
2 Data structure X is made up of a combination of items A, B and C.
3 The JSP data structure diagram illustrates sequence.
4 The JSP data structure diagram shows stepwise refinement.

1.4 A customer transaction, `CustomerTransaction`, consists of:

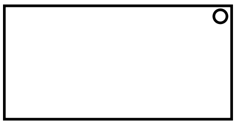
- a customer number, `CustomerNumber`
- a product code, `ProductCode`
- the quantity, `Quantity`

in this given order.

Key focus: **Using 'sequence' …**

Draw the JSP data structure diagram.

1.5 This symbol is used in a JSP data structure diagram. What is it used to represent?



- sequence
- selection
- a procedure
- iteration

Task 2

A train, `Train`, consists of an engine, `Engine`, followed by a carriage, `Carriage`. Sometimes there is a second `Engine` at the rear of the train.

Draw the JSP data structure diagram.

Task 3

Key focus:
Using ‘sequence’ …

An employee record `Record` consists of:

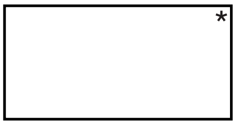
- the employee name `EmployeeName`
- the address `EmployeeAddress`
- a pay code `EmployeePayCode`, which is either Full, Part Casual

in this given order.

Draw the JSP data structure diagram.

Task 4

4.1 This symbol is used in a JSP data structure diagram. What is it used to represent?



- a process, action or operation on a data item
- iteration of data item
- selection of a data item
- a procedure

4.2 A `Train` consists of an `Engine` followed by one or more `Carriage`s. Sometimes there is a second `Engine` at the rear of the train.

Draw the JSP data structure diagram.

Key focus:
Using ‘iteration’ …

4.3 An examination paper, `ExamPaper`, consists of a front page, `Front`, followed by at least one question, `Question`.

Draw the JSP data structure diagram.

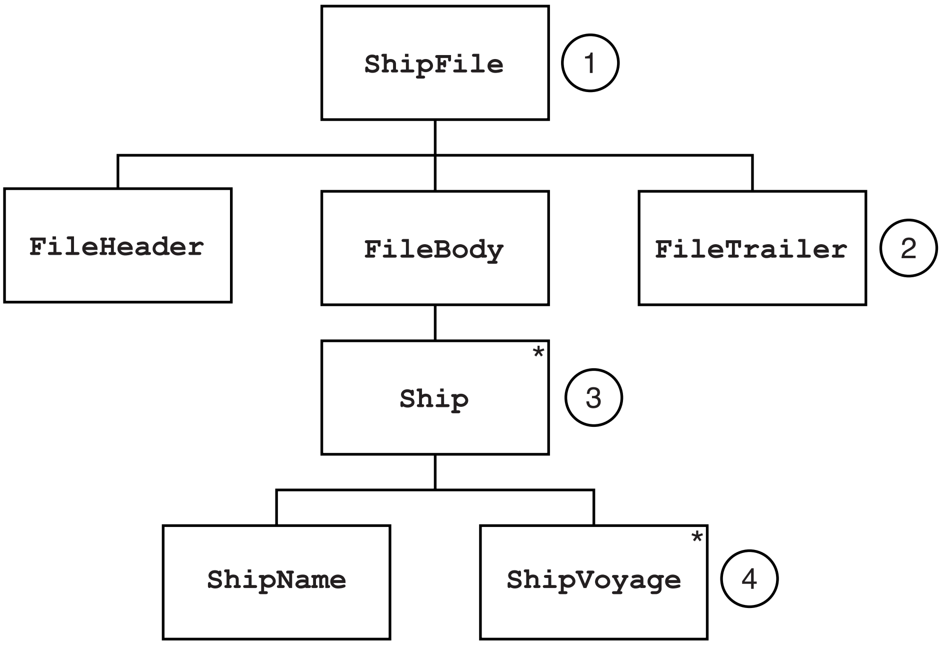
Task 5

A shipping company has a number of merchant ships. Data are recorded for all the voyages made by each ship.

The data are stored in a file `ShipFile`. The file has:

- a file header, `FileHeader`, which shows the date the file was last updated
- a file body, `FileBody`, which contains data for all the ships
- a file trailer, `FileTrailer`, which marks the end of the file

The data for each ship are made up of the name of the ship followed by the data records for each voyage made. There may be a ship record for a ship which has not made any voyages so far.



To build up the JSP data structure diagram:

① the file has the name `ShipFile`.

② the file is made up of the **sequence** of data **components** stated in the bulleted list above the diagram.

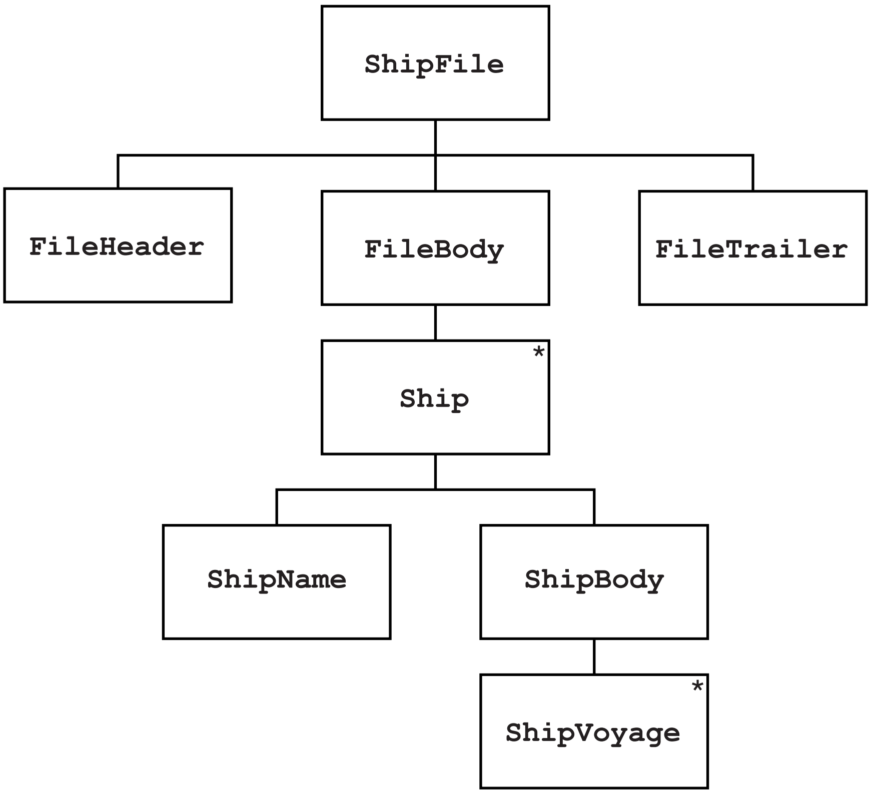
Note:

- A component which has no derived lower level component(s) is called an **elementary component**. FileHeader and FileTrailer are both elementary components.

- It may have been tempting to have the iterated data item Ship one line above (which would seem to avoid the use of the FileBody item. However, this would have resulted in a sequence of 'mixed-type components' at the same level. **<u>This must always be avoided</u>**.

③ The component Ship is an **iteration**, as it consists of the data for several ships.

④ We have ‘mixed-type components’ at the same level. That is ShipName (elementary) and ShipVoyage (iterated). The solution therefore is the same as earlier. This gives the final diagram below:



Key focus:
A data structure diagram for output

Task 6

A JSP data structure diagram can also be used to describe the data requirements for a printed report.

This was hinted at in Task 4.3.

A report, Report, is made up of a number of pages, Page. Each page consists of a possible heading, Heading, followed by a number of lines, Line.

Draw the JSP data structure diagram.

Logic Programming

Task 7

Key focus:
Lists

Logic programming can use a data structure called a **list**. The list items are enclosed inside square brackets. Each item is separated by a comma. The list can be given an identifier name.

The following is a list with a sequence of elements.

`[budapest,paris,london,singapore,dhaka,oslo,brussels,amsterdam,cairo]`

`[]` denotes an **empty list**.

A variable can be used to represent a list.

`Cities` represents `[budapest,paris,london,singapore,dhaka,oslo,brussels,amsterdam,cairo]`

Key focus:
The '|' list operator

The operator `|` is used with a list.

For example, `[X|Y]` denotes a list with:

`X` a variable representing the first list element, the **head**
`Y` a variable consisting of the remaining list elements, the **tail**

Example:

If `Cities` represents `[X|Y]`

- `X = budapest`
- `Y = [paris,london,singapore,dhaka,oslo,brussels,amsterdam,cairo]`

Note:
The head is always a list item
The tail is always a list
Predicates can be defined and used to manipulate and interrogate a list.

Assume the predicate `isEmpty` is defined in the knowledge base to return `TRUE` or `FALSE`.

`isEmpty(X)` returns:

`TRUE` when `X = []`
`FALSE` when `X` has one or more elements

Key focus:
Predicates with lists

Example:

`isEmpty(Cities)` would return `FALSE`.

7.1

The list `[gibraltar,hanoi]` is denoted as `[A|B]`

State the values of `A` and `B`.

`A` = ...

`B` = ...

7.2

The following two lists are identical:

```
[kingston,rome,riga|C] = [kingston,rome,riga,bangkok,minsk]
```

State the value of `C`.

`C` = ...

7.3

The following two lists are identical:

```
[suva,stanley] = [D,E|F]
```

State the value of `F`.

`F` = ...

Consider this second predicate:

```
removeAndCompare(A,B,C)
```

Remove any elements from list `A` which are also present in list `B`.
Compare the amended list `A` with list `C`.

If the lists are identical, the predicate returns `TRUE`, otherwise returns `FALSE`.

7.4.1

```
removeAndCompare([moscow,ottowa,monaco],
                                    [monaco,london],
                                               [moscow,ottowa,monaco])
```

What is returned by this predicate?

....

7.4.2

```
removeAndCompare([orange,banana],[ ],[orange,banana])
```

What is returned by this predicate?

....

BLANK PAGE

BLANK PAGE