Cambridge International AS & A Level

Cambridge Assessment International Education
Cambridge International Advanced Subsidiary and Advanced Level

COMPUTER SCIENCE **9608/23**

Paper 2 Fundamental Problem-solving and Programming Skills **October/November 2019**

PRE-RELEASE MATERIAL

8918955785

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.

DC (MM) 182207
© UCLES 2019

Teachers and candidates should read this material prior to the November 2019 examination for 9608 Paper 2.

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.

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 and vice versa

Some tasks may need one or more of the built-in functions or operators listed in the **Appendix** at the end of this document.

There will also be a similar appendix at the end of the question paper.

Declaration of variables

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

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

If Python is the chosen language, each variable's identifier (name) and its intended data type must be documented using a comment statement.

Structured English – Variables

An algorithm in pseudocode uses variables, which should be declared. An algorithm in structured English does not always use variables. In this case, the candidate needs to use the information given in the question to complete an identifier table. The table needs to contain an identifier, data type and description for each variable.

© UCLES 2019

TASK 1 – Scenario description

A software developer has been asked to write a program to manage membership information for a sports club. Membership information will be stored in a text file. Each line of the file will represent information about one member.

The information that needs to be stored is as follows:

- membership ID
- name
- email address
- month joined
- membership active

Information about a member will be held as a single `STRING`. Each `STRING` will occupy one line in the file.

TASK 1.1

Many pieces of information will have fixed formats. For example, membership active will have one of two possible values as follows:

Value	Meaning
`"TRUE"`	Membership still active
`"FALSE"`	Membership expired

1. Decide on a format for the membership ID, for example, four digits “1234”.

2. Decide on a format for the month joined. This could be the full name of the month, first three letters or a two-digit code, for example, 03 to represent the month of March.

TASK 1.2

Design a suitable format for each single `STRING` to allow the program to extract the different pieces of information, when reading from the file.

TASK 1.3

Produce program flowcharts to represent the algorithms that could be used to validate the formats of the different pieces of information, as decided in TASK 1.1.

TASK 2

Use a text editor to create a text file and add five example membership information strings.

Print the file to check the contents.

© UCLES 2019

TASK 3

A member joins the club for a period of one year, starting from the beginning of the current month.

TASK 3.1

Design a program in **pseudocode** that has a menu-driven interface and will perform the following tasks.

1. Add a new member to the text file. Include validation of the different pieces of information.

2. Search for the details of a given member using membership ID and output their information in a suitable format, for example:

   Name: Manish Modee
   Email address: Manish@Modee.com

3. Output a list of members whose membership ends in a given month.

TASK 3.2

Expand the program to refine step 2 from TASK 3.1 to allow the user to search using either the membership ID or the name.

TASK 4

Write **program code** to implement the program design.

TASK 5

Design and implement test data that can be used to prove the design.

TASK 6

The manager of the sports club wants a list of members whose membership has expired.

A new file will be created containing the name and email address for each member whose membership has expired.

Write **program code** to create the new file.

© UCLES 2019

Appendix

Built-in functions (pseudocode)

Each function returns an error if the function call is not properly formed.

`MID(ThisString : STRING, x : INTEGER, y : INTEGER) RETURNS STRING`
returns a string of length `y` starting at position `x` from `ThisString`

Example: `MID("ABCDEFGH", 2, 3)` returns `"BCD"`

`LENGTH(ThisString : STRING) RETURNS INTEGER`
returns the integer value representing the length of `ThisString`

Example: `LENGTH("Happy Days")` returns `10`

`LEFT(ThisString : STRING, x : INTEGER) RETURNS STRING`
returns leftmost `x` characters from `ThisString`

Example: `LEFT("ABCDEFGH", 3)` returns `"ABC"`

`RIGHT(ThisString : STRING, x : INTEGER) RETURNS STRING`
returns rightmost `x` characters from `ThisString`

Example: `RIGHT("ABCDEFGH", 4)` returns `"EFGH"`

`INT(x : REAL) RETURNS INTEGER`
returns the integer part of `x`

Example: `INT(27.5415)` returns `27`

`NUM_TO_STRING(x : REAL) RETURNS STRING`
returns a string representation of a numeric value.
Note: This function will also work if `x` is of type `INTEGER`

Example: `NUM_TO_STRING(87.5)` returns `"87.5"`

`STRING_TO_NUM(x : STRING) RETURNS REAL`
returns a numeric representation of a string.
Note: This function will also work if `x` is of type `CHAR`

Example: `STRING_TO_NUM("23.45")` returns `23.45`

Operators (pseudocode)

Operator	Description
`&`	Concatenates (joins) two strings Example: `"Summer" & " " & "Pudding"` produces `"Summer Pudding"`
`AND`	Performs a logical `AND` on two Boolean values Example: `TRUE AND FALSE` produces `FALSE`
`OR`	Performs a logical `OR` on two Boolean values Example: `TRUE OR FALSE` produces `TRUE`

© UCLES 2019

BLANK PAGE

© UCLES 2019

BLANK PAGE

© UCLES 2019

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.

To avoid the issue of disclosure of answer-related information to candidates, all copyright acknowledgements are reproduced online in the Cambridge Assessment International Education Copyright Acknowledgements Booklet. This is produced for each series of examinations and is freely available to download at www.cambridgeinternational.org after the live examination series.

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

© UCLES 2019