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.
Teachers and candidates should read this material prior to the June 2018 examination for 9608 Paper 4.

**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 for Paper 4 build on the practical skills covered in Paper 2. We therefore recommend that candidates choose the same high-level programming language for this paper as they did for Paper 2. This will give candidates the opportunity for extensive practice and allow them to acquire sufficient expertise.

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

- structured English
- pseudocode
- program code

A program flowchart or the use of Jackson Structured Programming (JSP) should be considered as an alternative to pseudocode for documenting a high-level 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
- the production of a JSP structure diagram from a given scenario

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

**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.
TASK 1

Staff salaries are read from a file. The role of each member of staff is determined by their salary:

- if they earn 50 or less, they are a Manager
- if they earn more than 50 but less than 90 they are a Lead Developer
- if they earn 90 or more, they are a Project Manager.

The following JSP structure diagram shows the structure of a solution for this problem.

**Key focus: JSP structure diagrams**

**TASK 1.1**

Research the purpose of JSP structure diagrams.

**TASK 1.2**

Find out the meaning of the different symbols inside the boxes and how these affect the structure of the program.
TASK 1.3

Complete the following pseudocode algorithm for the JSP structure diagram on the previous page.

WHILE ..............................................................................................................................................

    CALL ReadSalary()

    IF ..............................................................................................................................................

        THEN

            IF ......................................................................................................................................

                THEN

                    Role ← ....................................................................................................................

                ELSE

                    Role ← ....................................................................................................................

                ENDIF

        ELSE

            Role ← ..............................................................................................................................

        ENDIF

ENDWHILE

TASK 1.4

A new role is introduced. Any member of staff earning more than 70 and less than 90 is now called a Consultant.

Edit the JSP structure diagram to include the Consultant role.

TASK 1.5

Edit the pseudocode algorithm to add the Consultant role.

TASK 1.6

Write a function in program code to determine the role of a staff member.

The function will take salary as its parameter, and return the role title.
TASK 2

A toy shop needs a computer system to store information about its toys.

The developer writes a program using object-oriented programming.

- A toy has the properties:
  - Name (for example, Train engine)
  - ID (for example, TE11)
  - Price (for example, 0.99)
  - Minimum age (for example, 4)

- A computer game is a type of toy; it has the properties of toy, and:
  - Category (for example, Car racing)
  - Console (for example, Camstation)

- A vehicle is a type of toy; it has the properties of toy, and:
  - Type (for example, Car)
  - Height (for example, 4)
  - Length (for example, 15)
  - Weight (for example, 0.2)

TASK 2.1

Use the following incomplete class diagram to specify the three classes. Make sure you include the constructor, and get and set methods for each class.

Key focus: Object-oriented programming

Key focus: Class diagrams
TASK 2.2

Find out what is meant by inheritance.

The ComputerGame and Vehicle classes inherit from the Toy class.

Add arrows to represent the inheritance relationships in your class diagram.

TASK 2.3

Write program code to create the Toy class. Make sure:

- the properties are declared as private
- you have a constructor
- you have get and set methods for each property.

Key focus: Writing classes

TASK 2.4

Write program code to create the ComputerGame and Vehicle classes. Make sure:

- the properties are declared as private
- you have a constructor
- you have get and set methods for each property
- the classes inherit the properties from the Toy class.

Key focus: Writing classes

TASK 2.5

Add validation (where appropriate) to the constructor and set methods for all three classes. For example, age range may be limited to between 0 and 18.

TASK 2.6

Create instances of ComputerGame and Vehicle with appropriate data, in the main program. Store the objects in arrays. The data for one instance of Vehicle is given to get you started.

Name = "Red Sports Car"
ID = "RSC13"
Price = 15.00
Minimum age = 6
Type = "Car"
Height = 3.3
Length = 12.1
Weight = 0.08

Key focus: Instantiating objects

TASK 2.7

A procedure prompts the user to input the ID for a toy. The program finds that toy and outputs the values of its properties in an appropriate format.

Write the procedure and test it with appropriate data.
TASK 2.8

A procedure allows the user to input a number to be used as a discount rate. The price of all the toys will be reduced by that number. For example, 10 would reduce the price of all toys by 10%.

Write the procedure and test it with appropriate data.

TASK 2.9

Research the differences between a bubble sort and insertion sort.

TASK 2.10

The shop would like the program to sort the objects in the ComputerGame class in ascending order of price, and output the values of their properties.

Write at least one algorithm to sort and output the arrays in ascending order of price.

Test the algorithm with appropriate data.

TASK 3

A declarative language is used to create a set of facts and rules for characters in a computer game.

01 character(jim).
02 character(jenny).
03 character_type(jim, prince).
04 character_type(jenny, princess).
05 has_skill(jim, fly).
06 has_skill(jenny, invisibility).
07 pet(jim, horse).
08 pet(jenny, bird).
09 animal(horse).
10 animal(bird).
11 skill(fly).
12 skill(invisibility).
13 has_skill(X, Y) if character(X) and skill(Y).

TASK 3.1

Add the following facts:

- Habib is a character, who is an explorer.
- Habib has a pet fish and the skill of time travel.

TASK 3.2

Write a rule to state that a character can only have a pet, if they are a character, and the pet is an animal.
TASK 3.3

Add additional characters, animals, pets and skills.

TASK 3.4

State what would be returned if the following goals were added.

- `character(jim)`.
- `character_type(jenny, X)`.
- `character_type(X, prince)`.
- `has_skill(jenny, X)`.
- `has_skill(X, fly)`.

TASK 3.5

Write goals to find the following information.

- all Jim’s pets
- all the characters who can fly
- all the skills
- all the pets of characters who are princesses