Cambridge International Examinations
Cambridge International Advanced Subsidiary and Advanced Level

Cambridge International AS & A Level	

COMPUTER SCIENCE **9608/04**

Paper 4 Further Problem-solving and Programming Skills **For Examination from 2015**

SPECIMEN PRE-RELEASE MATERIAL

2 hours

No Additional Materials are required.

This material should be given to candidates up to six months before the examination.

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 **3** printed pages and **1** blank page.

Candidates' preparation for the examination should include attempting the following practical **program coding** tasks.

Key focus: object-oriented programming

In a school, the Personnel Department is responsible for storing details about all staff who work for the school. There are four categories of staff:

- management
- teacher
- support staff
- maintenance and kitchen staff

All staff have a StaffID and a Name. The Name has three parts: title (Mr/Mrs), first name and surname. In addition, the salary for each member of staff is stored.

A teacher has a classroom in which they do all their teaching. A teacher is also responsible for a form group.

Support staff are either part-time or full-time. They are also allocated to one academic department.

Software, using object-oriented programming, is to be written to process data about the staff.

For this solution, details about maintenance and kitchen staff are not included.

Management are classified as staff but are not teachers or support staff or maintenance and kitchen staff.

(a) Complete a class diagram showing the three classes:
- Staff
- Teacher
- SupportStaff

Include all the properties described above. In addition, include a constructor method that is called when an object of the class is created. The method should allow the object properties to be initialised. There should also be a show method that displays all the property values of the object.

(b) Implement the classes designed in **(a)**. Use inheritance to create appropriate relationships between the classes.

(c) Test the code by creating objects from each class and then displaying their property values.

(d) Create a new class, Name, which has three properties: Title, FirstName and Surname. Include a method, DisplayName that has two parameters:

1 – Name displayed in the format: <Title>space<FirstName>space<Surname>
2 – Name displayed in the format: <Surname>space<FirstName>

(e) Test the class Name and the DisplayName method.

(f) Use containment to replace the title, first name and surname properties in your original Staff class with the new Name class. Test your changes.

(g) Lunches are provided for all staff. The cost is 1/80 of their salary. Write a method that will display the cost for each member of staff. Test your solution.

(h) Changes are made to the cost of the lunches so that the teaching staff are charged only 1/100 of their salary, and for support staff it is 1/120 of their salary. Use polymorphism to implement methods in the Teacher and SupportStaff classes for displaying the cost of lunches. Test your changes.

(h) If all management staff were classified as either a teacher, or a member of the support staff, implement changes to your solution so that the staff class can be only used as a base class. Consequently an object cannot be created directly from the Staff class. Test your changes.

4

BLANK PAGE