OCR
Oxford Cambridge and RSA

GCSE (9–1) Computer Science

J276/03 Programming project – Task 1

Non-examined assessment (NEA) task

June 2018 series

7034910059

- Please check on **OCR Interchange** that you have the Non Examined Assessment material valid for the appropriate assessment series.

INSTRUCTIONS TO TEACHERS

- Please refer to Section 3f of the GCSE (9–1) Computer Science specification for instructions on completing the Non-Exam Assessment tasks.
- The marking criteria should be available to candidates whilst completing the task.
- The quality of extended response will be assessed in the 'Development and Testing' and 'Evaluation and Conclusions' sections.
- Teachers are responsible for ensuring that the Non-Exam Assessment Material completed is valid for the appropriate assessment series.
- The deadline for submitting NEA marks will be 31 March 2018.

INFORMATION FOR CANDIDATES

- The total mark for this component is **40**.
- This document consists of **4** pages.

Candidates should complete the task and provide evidence to meet all the marking criteria.

For the following scenario analyse the detailed requirements and, using suitable algorithms, design a solution to be coded in a suitable high-level programming language.

Show the iterative development of the individual solutions with suitable testing throughout the process.

Test the final product and evaluate your solution against the detailed requirements you identified in the analysis. The non-exam assessment must be done using a suitable high level language such as:

- Python
- C family of languages (for example C# C++ etc.)
- Java
- JavaScript
- Visual Basic/.Net
- PHP
- Delphi
- SQL
- BASH

You may use a combination of programming languages to produce a solution to the task.

Teachers **may**:

- explain the task
- advise on resources
- provide the support described within the 'Permitted Support' section of the Specification
- interrogate learners to ensure that the work is their own
- provide a copy of the mark scheme to candidates.

Teachers **must not**:

- give detailed advice and suggestions as to how the work may be improved in order to meet the assessment criteria. This includes indicating errors or omissions and personally intervening to improve the presentation or content of the work
- practise the task with the learners
- practise tasks which are similar in nature with the learners
- provide templates, model answers or feedback on drafts
- produce templates or model answers and publish them online.

Teachers **must** ensure that:

- learners do not access the internet*
- learners are not allowed to take the NEA tasks home with them
- all work presented for submission must have been completed under supervised conditions
- accounts associated with the NEA tasks must be locked between sessions to ensure that learners cannot access them outside of the supervised conditions
- learners do not access online file storage accounts or email during the supervised conditions in order to prevent learners from completing work at home and bringing it into the supervised conditions.

*unless the centre is using an online IDE, in which case, only access to the IDE website is allowed.

Scenario

Fergus is creating a quiz that tests students' knowledge on different topics*, such as:

- History
- Music
- Computer Science.

Students must register before they can take the quiz, choosing a unique username and a password for the account. The username and password are saved into an external text file. They must then enter the following details about themselves:

- Name
- Age
- Year Group.

A student can choose which topic they want to answer a series of questions on. The student can then choose a difficulty rating of 'Easy', 'Medium' or 'Hard'. Each question will have a set of possible answers that the user can choose from. The number of answers the student chooses from changes based on the difficulty of the quiz. The user must select an answer to a question before the next question is displayed.

At the end of the quiz, the game will output the number of questions they got correct and a grade depending on the percentage of questions the user has got correct. You must create a suitable grade system, and the percentages needed to achieve each grade. There must be at least four possible grades.

The program stores information about every quiz each student has taken, including the topic, score and difficulty rating.

Analyse the requirements for this system and design, develop, test and evaluate a program that:

1. creates a unique username for each user. The username is made up from the first 3 letters of their name and their age. E.g. Gemma Smith, age 17 would have a username of 'Gem17'. It then asks the user to enter a password for their account.
2. stores the username and password of the user, the details about each user, and the topic, score and difficulty rating of all quizzes each student has taken. These do not have to be stored in the same file.
3. allows a user to select a topic and difficulty rating (Easy, Medium or Hard) and asks five questions on that topic:
    a. 'Easy' mode has a choice of two answers for each question
    b. 'Medium' mode has a choice of three answers for each question
    c. 'Hard' mode has a choice of four answers for each question
4. loads the questions and answers from a file stored externally to the game.
5. displays the user's score, percentage and grade achieved for that quiz.
6. gives Fergus the option to generate and output the following reports:
    a. a report that allows Fergus to choose a username, and outputs all of the quizzes that they have taken, and the grade for each of those quizzes.
    b. a report that outputs for a selected topic and difficulty: the average score achieved, the highest score achieved, and the user details of the person that achieved the highest score.

NOTE FOR CANDIDATES:
*The system only needs to be tested using **two topics**.