Oxford Cambridge and RSA

GCSE (9–1) Computer Science

J276 Programming Project Tasks 1, 2 and 3

June 2019 and June 2020 series

20 timetable hours

771730930 3

- Please check on the OCR website that you have the material valid for the appropriate assessment series.

INSTRUCTIONS

- Please refer to Section 4d of the GCSE (9–1) Computer Science specification for instructions on completing the Programming Project.
- Candidates must choose **one** of the enclosed tasks.
- The work is to be completed by 15 May for the relevant series.
- This document consists of **8** pages.

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

Candidates are required to select **one** scenario.

For the chosen scenario candidates must:
- analyse
- develop
- test and evaluate.

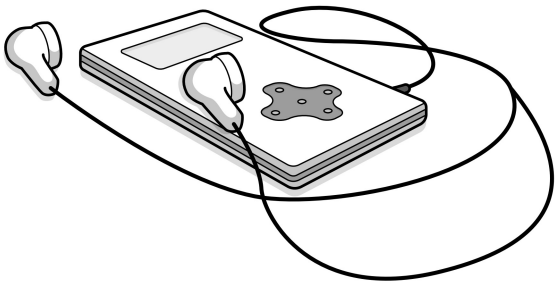
Test the final product and evaluate your solution against the detailed requirements you identified in the analysis. This can be done using a suitable high level language such as:

- 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.

Please refer to the **Programming Project Guidance** on the OCR website for guidance on completing the task.

TASK 1



Noel is creating a music quiz game.

The game stores a list of song names and their artist (e.g. the band or solo artist name). The player needs to try and guess the song name.

The game is played as follows:

- A random song name and artist are chosen.
- The artist and the first letter of each word in the song title are displayed.
- The user has two chances to guess the name of the song.
- If the user guesses the answer correctly the first time, they score 3 points. If the user guesses the answer correctly the second time they score 1 point. The game repeats.
- The game ends when a player guesses the song name incorrectly the second time.

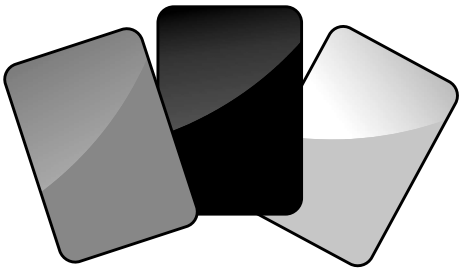
Only authorised players are allowed to play the game.

Where appropriate, input from the user should be validated.

Design, develop, test and evaluate a system that:

1. Allows a player to enter their details, which are then authenticated to ensure that they are an authorised player.
2. Stores a list of song names and artists in an external file.
3. Selects a song from the file, displaying the artist and the first letter of each word of the song title.
4. Allows the user up to two chances to guess the name of the song, stopping the game if they guess a song incorrectly on the second chance.
5. If the guess is correct, add the points to the player's score depending on the number of guesses.
6. Displays the number of points the player has when the game ends.
7. Stores the name of the player and their score in an external file.
8. Displays the score and player name of the top 5 winning scores from the external file.

TASK 3



Louise is creating a card game for two players.

The game uses a deck of cards. There are 30 cards in a deck. Each card has one colour (red, black or yellow). Each card has a number (1, 2, 3, 4, 5, 6, 7, 8, 9, 10) for each colour. Each card is unique.

The 30 cards are shuffled and stored in the deck.

The rules are:

- Player 1 takes the top card from the deck.
- Player 2 takes the next card from the deck.
- If both players have a card of the same colour, the player with the highest number wins.
- If both players have cards with different colours, the winning colour is shown in the table.

Card	Card	Winner
Red	Black	Red
Yellow	Red	Yellow
Black	Yellow	Black

- The winner of each round keeps both cards.
- The players keep playing until there are no cards left in the deck.

Only authorised players are allowed to play the game.

Where appropriate, input from the user should be validated.

Design, develop, test and evaluate a program that:

1. Allows two players to enter their details, which are then authenticated, to ensure that they are authorised players.
2. Shuffles the 30 cards in the deck.
3. Allows each player to take a card from the top of the deck. Play continues until there are no cards left in the deck.
4. Calculates the winner and allocates both cards to the winner.
5. Displays which player wins (the player with the most cards).
6. Lists all of the cards held by the winning player.
7. Stores the name and quantity of cards of the winning player in an external file.
8. Displays the name and quantity of cards of the 5 players with the highest quantity of cards from the external file.