

# Pearson Edexcel Level 1/Level 2 GCSE

## Computer Science

### Practical Programming

Sample Controlled Assessment Material

Paper Reference

**1CP0/2A**

**You do not need any other materials.**

#### Instructions

- You must use a computer to complete these tasks.
- You may **not** have access to the internet.
- You may have access to a printer.
- You may use a calculator.
- You will be given access to relevant files during the controlled assessment.
- All files must be saved and kept secure in the folder specified by your centre between sessions. You will not have access to these files between sessions.
- You must save the final versions of your files in the **CA folder** indicated by your centre.

#### Information

- The total mark for this controlled assessment task is 50.
- The marks for **each** task are shown in brackets.
- The total time for the controlled assessment is 15 hours.
- A guide to how much time you should spend is shown at the end of each task.
- Tasks labelled with an **asterisk** (\*) are ones where the quality of your written communication will be assessed  
– *you should take particular care on these tasks with your spelling, punctuation and grammar, as well as the clarity of expression.*

#### Advice

- Read each task carefully before you start to answer it.
- Attempt every task.

Turn over ►

S45205A

©2014 Pearson Education Ltd.

1/1/1/1



**PEARSON**

## Controlled assessment tasks

The controlled assessment tasks should be completed using Python, Java or any C-derived language.

### Context: Developing a Game

You will complete tasks to produce parts of a game. The aim of the game is to promote online safety.

### Features

- The game will have 5 levels.
- Players will be given a score out of 20 for each level.
- Players must register their details to play the game.
- Players must choose a 'player name' consisting of no more than 12 characters.

You have been asked to write programs to:

- register a new player
- produce a quiz for level 1 of the game
- display a scoreboard for the game.

**You must complete the following tasks.**

Task	Features	Marks	Suggested time
<b>1 Registering a new player</b> Program with comments	<ul style="list-style-type: none"><li>• Prompt player to enter registration data</li></ul>	8	1½ hours
<b>2 Creating a quiz</b> Design, program, test, evaluate	<ul style="list-style-type: none"><li>• Player interface</li><li>• Generate quiz</li><li>• Scoring system</li></ul>	18	5½ hours
<b>3 Creating a scoreboard</b> Design, program, test, evaluate	<ul style="list-style-type: none"><li>• Input scores from a file</li><li>• Generate leader board displays</li><li>• Create error log</li></ul>	24	8 hours

### Task 1: Registering a new player

To register for the game, players must enter their age, gender, an email address and a player name.

#### Registration requirements

Here is an outline design for the program.

*Display a welcome message for the game*

*Ask the player to enter their age, gender, email address and a player name*

*Display this information on the screen and ask the player to confirm that it is correct*

*If the information is incorrect, ask the player to re-enter the information*

*Display the correct information on the screen*

#### Your task

- 1 Write a program to implement this design. It must be easy to read and include comments to explain how it works.

Save your program in a file called **registration** in the **CA folder**.

**You are advised to spend no more than 1½ hours on Task 1.**

**(Total for Task 1 = 8 marks)**

---

## Task 2: Creating a quiz

Level 1 of the game is a quiz. Players must answer questions about online safety.

Players must be presented with a set of questions to test their knowledge of online safety. **quizQuestions.txt** *Appendix B* contains some questions and answers for you to use or you can create your own.

### Quiz requirements

- The player interface for the quiz must be text based.
- Players should be asked to enter a player name.
- Players must be told how to play the game.
- Validation should be used to ensure that all user input is valid.
- Questions on online safety must be displayed to the player.
- Questions must be presented in a different order each time the quiz is run.
- Players must be able to select an answer.
- A score between 0 and 20 must be assigned for each correct answer.
- You must decide how many points to assign to each correct answer and how many questions to display.
- Once the quiz has been completed, the player name, level number and total score must be displayed on the screen.

### Your task

- 2 (a) (i) Develop a program to meet these requirements. It must be easy to read and include comments to explain how it works. The quality of your program design will be assessed using the program code you submit.
- (ii) Write and implement a plan to test that the program works correctly and that the requirements have been met, using the **test\_plan.doc** *Appendix C* template provided.

\*(b) Write an assessment of your program that:

- evaluates how well your program meets the quiz requirements
- explains how your program selects which question to ask.

Your answer should use the correct technical terms where appropriate.

Save the final version of your source code in a file called **quiz** in the **CA folder**.

Save your test plan in a file called **quiz\_test\_plan.doc** in the **CA folder**.

Save your evaluation in a file called **quiz\_evaluation.doc** in the **CA folder**.

**You are advised to spend no more than 4½ hours on developing and testing the program and 1 hour on the evaluation.**

**(Programming and testing = 12 marks)**

**(Evaluation = 6 marks)**

**(Total for Task 2 = 18 marks)**

### Task 3: Creating a scoreboard

The game must have a scoreboard.

When a player completes any level, their player name, level and score are appended to a file called **playerScores.txt**. The information is separated by commas and is not ordered. Records are in the format 'player name, level, score'.

'superfred299,2,19' is an example of a record from the file. This shows that the player 'superfred299' has completed level 2 with a score of 19.

The scoreboard program will sort and search the file **playerScores.txt** to select and display information about players and their scores.

#### Scoreboard requirements

- Three options should be displayed for players to select:

<b>Option A</b>	Scores for a player
<b>Option B</b>	Highest scorer for each level
<b>Option C</b>	Highest scorer in the game

**Enter Q** to quit the scoreboard

- Players must be able to select an option or enter 'Q' to quit the scoreboard
- Data on player, level and score must be read from the **playerScores.txt** file. If there is no record for a player for a level, a score of 0 must be used for that level.

**Option A** must provide a prompt to enter the name of the player.

- It must find the scores for each level for the selected player.
- It must display in this format:

The level scores for *player\_name* are:

Game level	Score
Level 1	11
Level 2	12
Level 3	14
Level 4	15
Level 5	0

If the player has not completed a level a score of 0 should be displayed.

**Option B** must find the player with the highest score in each level.

- It must display in this format:

The highest scorers for each level are:

Level	Player name	Score
Level 1	Star	15
Level 2	Star	12
Level 3	Lion	19
Level 4	Kev	15
Level 4	Planet	15
Level 5	Star	16

- If more than one player has the same highest score, each player with the highest score should be displayed.

**Option C** must find the highest scorer. The highest scorer is the player with the highest combined score for each of the five levels.

- It must display in this format:

The highest scorer in the game is Lion236 with a score of 87.

- If more than one player is the highest scorer, each player and their score should be displayed.
- All data input from the screen must be validated.
- Any records with score values outside the normal range (0 to 20) must not be included in the scoreboard.
- Any records with level values outside the normal range (1 to 5) must not be included in the scoreboard.
- An error log file called **scoreboard\_errorlog.text** must be written. It must contain a list of any records with invalid score or level values.
- The scoreboard program must work for any number of records in the **playerScores.txt** file.

### Your task

- 3 (a) (i) Develop a program to meet the above requirements. It must be easy to read and include comments to explain how it works. The quality of your program design will be assessed using the program code.
- (ii) Write and implement a test plan to ensure that the program works correctly and that the requirements have been met. Use the **test\_plan.doc** Appendix C template provided.
- \*(b) Write an assessment of your program that:
- evaluates how well your program meets the quiz requirements
  - explains how your program found the correct information for each of the three options.

Your answer should use the correct technical terms where appropriate.

Save the final version of your source code in a file called **scoreboard** in the **CA folder**.

Save your test plan in a file called **scoreboard\_test\_plan.doc** in the CA folder.

Save your error log in a file called **scoreboard\_error\_log.txt** in the CA folder.

Save your evaluation in a file called **scoreboard\_evaluation.doc** in the CA folder.

**You are advised to spend no more than 6½ hours on developing and testing the program and 1½ hours on the evaluation.**

**(Programming and testing = 15 marks)**

**(Evaluation = 9 marks)**

**(Total for Task 3 = 24 marks)**

---

**TOTAL FOR PAPER = 50 MARKS**

### Files supplied to candidates

Purpose	File name	File type
Quiz questions	quizQuestions	.txt
Player scores	playerScores	.txt
Test plan template	test_plan	.doc

### Checklist of files for the candidate to submit:

Task	File name	File type
1 program	registration	program source code
2 program	quiz	program source code
2 test plan	quiz_test_plan	document
2 evaluation	quiz_evaluation	document
3 program	scoreboard	program source code
3 error log	scoreboard_error_log	text file
3 test plan	scoreboard_test_plan	document
3 evaluation	scoreboard_evaluation	document



## Appendix A

You have been given the file **playerScores.txt**. A copy of the file is shown here.

The data is shown as: **player name, level, score**

Starplayer,1,19  
Ben,1,20  
Kev,0,20  
Cris,1,19  
Runner,4,19  
Cris,2,15  
Heart,5,16  
Cris,3,16  
Kitty,1,19  
Jade,4,20  
Moon,5,9  
Gem,3,13  
Mo,1,19  
Kev,2,18  
Moon,3,19  
Kitty,2,17  
Ant,4,5  
Moon,1,5  
Sword,1,10  
Runner,1,0  
Ant,1,10  
Kitty,4,20  
Jade,3,5  
Crazy-horse,1,15  
Crazy-horse,2,18  
Heart,2,19  
Ellie,3,10  
Fish,6,5  
Crazy-horse,3,9  
Kitty,3,19  
Mo,2,17  
Starplayer,2,5  
Moon,21,18  
Ellie,4,-5  
Ellie,5,11  
Crazy-horse,4,17  
Sword,2,20  
Ant,3,16  
Heart,1,15  
Runner,5,98  
Cris,5,17  
Heart,3,17  
Sword,3,6  
Mo,3,18  
Kitty,5,18  
Heart,4,20  
Starplayer,3,12

Mo,44,16

Cris,4,13

Fish,1,18

Ben,2,15

Starplayer,14,5

Starplayer,5,14

Ben,3,18

Mo,5,21

Runner,3,18

Ben,5,16

## Appendix B

You have been given the file **quizQuestions.txt**. A copy of the file is shown here.

The data is grouped as follows:

- the quiz question
- the correct answer
- two incorrect alternatives.

What is CEOP?

Child Exploitation and Online Protection

Criminal Exploration and Online Protection

Child Exploitation and Organised Protectors

When you get an email from someone you do not know, what should you do?

Delete it and mark as spam

Reply and say hello

Forward to your friends

How secret should you keep your passwords?

Never give out passwords except to your parents

Give them only to your best friends

Give them to strangers

When an online contact who frightens you asks to meet you in person what should you do?

Report to CEOP

Arrange to meet them

Arrange to meet them with your best friend

If an email asks you to enter your bank account details because of a problem with your account what should you do?

Contact the bank to check if they sent the email

Reply to the email

Enter your bank account details

## Appendix C

You have been given the file **test\_plan.doc**. A copy of the file is shown here.

Test number	Description of test	Test data	Expected result	Actual result	Pass/Fail