

Examiners' Report/
Principal Examiner Feedback

Summer 2015

Pearson Edexcel GCSE
In Computer Science (1CP0)
Paper 2A Practical Programming

Edexcel and BTEC Qualifications

Edexcel and BTEC qualifications are awarded by Pearson, the UK's largest awarding body. We provide a wide range of qualifications including academic, vocational, occupational and specific programmes for employers. For further information visit our qualifications websites at www.edexcel.com or www.btec.co.uk. Alternatively, you can get in touch with us using the details on our contact us page at www.edexcel.com/contactus.

Pearson: helping people progress, everywhere

Pearson aspires to be the world's leading learning company. Our aim is to help everyone progress in their lives through education. We believe in every kind of learning, for all kinds of people, wherever they are in the world. We've been involved in education for over 150 years, and by working across 70 countries, in 100 languages, we have built an international reputation for our commitment to high standards and raising achievement through innovation in education. Find out more about how we can help you and your students at: www.pearson.com/uk

Summer 2015

Publications Code UG041188

All the material in this publication is copyright

© Pearson Education Ltd 2015

General administration

Most Assessor Record Sheets were completed to a good standard including the signature of both the assessor and the student. However, some were inadequately completed with incorrect student numbers, incorrect student marks and missing student signatures. It is important that all information is present and correct in future series.

Most Centres made good annotations on the Assessor Record Sheet by providing thorough comments, which greatly assisted the moderation process.

Some centres provided comments that did not relate well to the work submitted.

Most centres followed the correct naming convention for files and folders; however a limited number of centres did not follow the specified format, which hindered the moderation process.

General comments

It was clear that the majority of students had planned the programming tasks well in the scheduled task preparation sessions. Work produced by students demonstrated good knowledge of the brief and it was very pleasing to see high-quality programming skills incorporated into programming tasks.

Insufficient program testing was an issue in most cases, with programs only working when valid data was input. It is important to test using valid, invalid and erroneous data to ensure that programs run correctly in all circumstances.

Students often limited themselves from accessing the high level marks for the evaluation tasks by submitting a commentary of the program code rather than a critical evaluation of how they met the requirements of the task. Most students simply stated that they had met the requirements when they should have given an evaluation of the selected program code and constructs they used to fulfil the each requirement. Some students did not answer the second part of the evaluation task, where they explain a specific aspect of the program; thus demonstrating the importance of reading the question thoroughly before attempting the task.

There were cases of over-generous assessment and high marks being awarded which were not supported by the evidence submitted. This was particularly noticeable in programming tasks 2a and 3a, where programs were not fully tested and therefore crashed easily.

Program code submitted for tasks 2a and 3a demonstrated complex programming skills, it was unfortunate that, in most cases, testing using invalid and erroneous data had not taken place.

Many centres were inconsistent in their awarding of marks. Evaluations tended to be marked generously, high marks were awarded by centres when students had only provided a commentary of the program and offered limited, or no, critical evaluation.

Individual reports are written for centres at the time of moderation; it is important that centres review and consider the points raised.

Comments on Task 1

Overall, Task 1 was completed to a high standard. Most students produced well-structured programs that used appropriate programming constructs and met most requirements of the brief.

Validation for this task was not necessary, as stated in the brief. There was evidence of students ensuring that the '@' was included in the input for email address, and a number of students differentiated the data types of input using 'int', however this was unnecessary for this task and often resulted in errors with invalid data input. Although testing was not a requirement of this task, programs that resulted in major errors were unable to not access the higher Level 3 marks.

Comments on Task 2a

In Task 2a, it was sufficient to create lists to hold the relevant data and to read directly from these lists. It was not a requirement of the task for the program to read directly from the text file 'clientIntensity'. File handling is a requirement of Task 3. Students who chose to use file handling in this task were not penalised in any way and could access the full range of marks available.

Most programs showed thorough, well planned decomposition into sub-programs. Some students submitted extremely user-friendly interfaces that went above and beyond the requirements of the task. Many students understood the need to work with numbers but did not understand that casting non-numeric input as integers, without accompanying validation, would result in errors.

Overall, the quality of programs submitted for Task 2a were high and demonstrated excellent computational thinking skills. Some students submitted detailed test plans which enabled the program to run error-free when valid data was input. Programs that resulted in major errors with the input of invalid or erroneous data were unable to access the high Level 3 marks. It is important that students test their programs with a range of data to ensure that it works in all circumstances.

Comments on Task 2b

Many of the evaluations were commentaries and contained very few evaluative comments. Many students did not answer the second part of the question; it is important that students read each question carefully.

Some of the evaluations gave clear and extensive explanations as to why certain methods had been used. Most students demonstrated a good understanding of the task and included appropriate specialist technical terminology.

Comments on Task 3a

The majority of students submitted high-quality programs that demonstrated effective decomposition, use of appropriate programming constructs and efficient naming, layout and structure that made the whole program easy to read.

Some students annotated program code with sufficient comments to explain how the program worked, whereas some students submitted program code with limited annotation.

There was evidence of excellent use of escape sequences to create well-structured output.

Menus were often very user-friendly and included various methods of validation to prevent data entry errors. Many students had validated the menu input well, but had failed to provide appropriate validation for the client ID or intensity level within the options, causing the program to crash easily with invalid data input.

Some students programs did not include iteration after each option, meaning programs did not loop and simply exited after running any of the options.

Comments on Task 3b

The majority of students produced a commentary of how they achieved the programming tasks rather than giving a critical evaluation of how they completed them. In some cases, the evaluations were not attempted or were incomplete, where the student had not allowed themselves enough time to complete the task.

Grade Boundaries

Grade boundaries for this, and all other papers, can be found on the website on this link:

<http://www.edexcel.com/iwantto/Pages/grade-boundaries.aspx>

