# Examiner's Report /
# Principal Moderator's Feedback

# Summer 2016

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

## General administration

As in the previous series most assessor record sheets were completed to a good standard, including the signature of both the assessor and the student. However, some were completed with incorrect student numbers, or incorrect student marks and missing student signatures. It is important that all information is present and correct.

Most centres added detailed comments in support of their assessments on the record sheets. These comments generally assisted the moderation process, although some of the comments did not relate well to the work submitted.

Most centres followed the correct naming convention for files and folders.

## General comments

Many students demonstrated good knowledge of the brief and were able to demonstrate well developed programming skills for this level.

Program testing was an issue in several cases, where the content of the test plans limited the extent of testing and validity of test data used.  In many examples testing should have been extended to check outputs after selection of a re-run option. Many of the implemented plans would have benefited from screen shots of test results.

Students often limited themselves from accessing the high level marks for the evaluation tasks by submitting a description of the program code rather than a critical evaluation of their program design in relation to the requirements of the tasks.

Most students addressed the second part of the evaluation tasks and provided detailed explanations of how specific aspects of the programs worked.

There were some cases of over-generous assessment, with 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 did not produce accurate output.

## Option 1CP0_2A: Python

Comments on Task 1

Overall, Task 1 was completed to a high standard. Variable names and suitable comments generally made the programs easy to read and most students produced well-structured code that used appropriate programming constructs and met the requirements of the brief.

The data types selected were generally suitable, although telephone number set as integer was a frequent error and, whilst validation for this task was not necessary, the required input options (yes/no, or Y/N etc.) should have been clarified for the user, where not handled in the code.

Comments on Task 2a

This task involved the production of cost estimates for decorating a variable number of rooms, each containing a variable number of walls. Additional variables for removal of wallpaper and level of qualification of decorator were also included.

Most programs showed at least partial decomposition into sub-programs and addressed most requirements of the brief. Variable names, constructs, program layout and structure were generally consistent with the band 3 assessment criteria, although comments tended to become less clear as the extent of the programs increased.

Most programs provided the iterations required to handle varying room and wall numbers, although logical errors in area calculations, which should have been recorded and corrected during testing, were commonplace. Several solutions that would not accept decimal values for room dimensions were seen. Some of these crashed on input of real values, whereas others handled the input using a type check to generate an error message.

Successful programs also handled the wallpaper removal and qualification level variables accurately and re-set all values when the option for a further estimate was selected prior to closing. Less successful programs provided a running total between estimates for at least one variable and often included the wallpaper option outside the room loop.

Some students submitted detailed test plans which demonstrated accurate output for a range of valid input data. Many other examples of test plans completed with 'as expected' results for all tests were submitted. Test plans, that indicate a fully functional solution, for a program that does not run without major errors are of limited value and are consistent with the band 1 assessment criteria.

Comments on Task 2b

Many of the evaluations were descriptions of work done and contained few evaluative comments.

Some of the evaluations gave clear and extensive explanations as to why certain methods had been used. In these examples students often demonstrated a good understanding of the task and included appropriate technical terminology, particularly in their explanation of how a specified aspect of their program works.

Comments on Task 3a

Many 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 and consistent with the band 3 assessment criteria.

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

Menus were often user-friendly and many programs included options to anticipate various input formats.

Most programs read the data file provided. The most successful programs handled the data correctly to produce accurate output from the required searches for each of the three options, although several students had difficulty presenting the tabbed output from option B as required.

Less successful programs did not return to the menu, but quit after running each option and tended to produce accurate output for option A only.

Centres are requested to ensure that the text file for task 3 is retained in the students' folders for testing and moderation purposes.

Comments on Task 3b

Comments generally as for task 2b, although in some cases, probably due to timing issues, the evaluations were not attempted or were substantially incomplete.

## Option 1CP0_2B and 2C: Java and C

Comments on completion of tasks all as above for Option 1CP0_2A.

When presenting work for language options 2B and 2C centres are requested to provide compiled versions of the source code to enable program testing for moderation.

**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.as