Pearson Edexcel Level 1/Level 2 GCSE (9–1)

Computer Science

Programming Project

For use from September 2018
Time: 20 hours

You must have:
TestPlan_Template.rtf

Instructions to teachers

- Students should use **one** of the following programming languages:
  – Python
  – Java
  – Pascal/Object Pascal
  – Visual Basic.NET
  – C-derived languages.
- You must adhere to the instructions as specified in the specification.
- Internet access is allowed.
- The materials submitted for assessment must include:
  – evidence of the development of the solution
  – the program code including any necessary solution files
  – a completed programming project authentification form (PPA) – available in the specification – see Appendix 3
  – a completed Head of Centre declaration form which should be signed by the head – available in the specification – see Appendix 4.

Information to students

- The work you submit must be your own.

Turn over ▶

W57332A

1/1/1/1/1

W 5 7 3 3 2 A

Pearson

Coin Count

A local youth club has organised several coin collections for a charity during the year.

People have been very generous and a lot of money has been collected. The coins now need to be counted and paid into the charity's bank account.

Six club members have volunteered to count the coins. The bank has supplied plastic bags for the coins. The coins must be bagged in set amounts as shown.

Coin	Bag Value
£2	£20.00
£1	£20.00
50p	£10.00
20p	£10.00
10p	£5.00
5p	£5.00
2p	£1.00
1p	£1.00

Each bag must hold exactly the value of coins shown in the table and contain only one type of coin. The volunteers sort the coins by type before counting and bagging them.

The club leader wants to check that the coins have been counted correctly by weighing each bag. At this stage, any bag containing the wrong number of coins is corrected. This checking process will take place over several sessions.

The weight of each coin, in grams (g), is shown in the table.

Coin	Weight (g)
£2	12.00
£1	8.75
50p	8.00
20p	5.00
10p	6.50
5p	3.25
2p	7.12
1p	3.56

The youth club leader wants a computer program to check the coin count. The program must:

- allow the user to input the volunteer's name, type of coin and weight of bag
- validate the coin type
- indicate the number of coins to be added or removed to correct an inaccurate bag weight
- maintain running totals of the number of bags checked and total value
- provide an option to display the total number of bags checked and total value
- monitor the accuracy of the volunteers counting the coins
- provide an option to display a list of the volunteers, sorted by accuracy, showing:
  - the total number of bags they have counted
  - the number of bags they counted correctly, as a percentage of their total.
- Save the data in a text file called CoinCount.txt
- Load CoinCount.txt at the beginning of each session.
- Update CoinCount.txt at the end of each session.

Your task is to analyse this problem and to design, implement, test and evaluate a programmed solution.

You must create test data to check that your program works as intended.

Some sample test data is shown in the table.

Volunteer	Coin	Bag weight (g)	Correct
Abena	5 pence	325.00	Y
Malcolm	1 pence	356.00	Y
Jane	2 pound	120.00	Y
Andy	1 pound	166.25	N
Sandip	50 pence	160.00	Y
Liz	20 pence	250.00	Y

The Report

Create a folder called Report. Save all your evidence for assessment in this folder.
Save your evidence as instructed at each stage.

Stage 1 Analysis (It is recommended that you spend 2 hours on Stage 1)

You should include an introduction summarising the overall problem.

The problem should be broken down into sub-problems. You should write a description of each sub-problem you identify and explain your selection of sub-problems.

State any assumptions you have made.

Save your work in the Report folder as a file called Analysis.

Stage 2 Design (It is recommended that you spend 6 hours on Stage 2)

Algorithms

Design algorithms, using pseudo-code or flowcharts, that show a logical solution to each sub-problem. You should include inputs, processes, outputs, validation checks and the programming constructs that you will use when you produce your program.

You should show how the algorithms will link together and lead to an overall solution.
Save your algorithms in the Report folder as a file called Design.

Initial test plan

You should complete the relevant sections of the test plan template provided to produce an initial test plan that will demonstrate your strategy for testing your solution.

Save your initial test plan in the Report folder as a file called TestPlan.

Save a copy of TestPlan in the Report folder as a file called TestTable, to be used in stages 3 and 4.

Stage 3 Implementation (It is recommended that you spend 10 hours on Stage 3)

You should translate your design into a program. Ensure that your program is clear and easy to understand.

Add the results of any tests carried out during the implementation stage to the TestTable file.

Save the updated TestTable file.

You should provide evidence showing how you debugged your program. Save your evidence in the Report folder as a file called Debugging.

Create a subfolder called Implementation in the Report folder. Save your source code and all the files required to execute the program, including your CoinCount.txt file, in the subfolder.

Stage 4 Testing, refining and evaluation (It is recommended that you spend 2 hours on Stage 4)

You should complete the TestTable file by adding any further tests carried out at this stage, including the results of retesting following the correction of any errors.

Save the completed TestTable file.

Evaluate your solution by explaining how well your program meets each of the requirements that you identified in your analysis and describing any refinements that you made to your program during design and implementation.

Save your evaluation in the Report folder as a file called Evaluation.