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

Computer Science

Programming Project

For use from September 2019
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 ▶

©2020 Pearson Education Ltd.

Pearson

Tim's Diner

Tim's Diner is a small fast food restaurant. It has 10 tables, each accommodating up to 4 diners. The restaurant has a menu with 12 different items, as shown.

TIM'S DINER

BREAKFAST	
1. All day (large)	£5.50
2. All day (small)	£3.50
MAINS	
3. Hot dog	£3.00
4. Burger	£4.00
5. Cheese burger	£4.25
6. Chicken goujons	£3.50
EXTRAS	
7. Fries	£1.75
8. Salad	£2.20
DRINKS	
9. Milkshake	£2.20
10. Soft drinks	£1.30
11. Still water	£0.90
12. Sparkling water	£0.90

Tim wants a computer program that will help him manage the business and make it easier to take and calculate orders correctly. The program should:

- Allow the user to input order details comprising table number and a string of numbers corresponding to the menu items chosen e.g. 6, 4, 4, 7, 8, 10, 10 would be an order from table 6 for 2 burgers, 1 fries, 1 salad and 2 soft drinks.
- Validate the input data.
- Calculate the total cost of the order.
- Display the order details for printing (actual printing not required).
- Loop for input of the next order.

File handling

In addition, the program should:

- Allow the menu to be saved to a file.
- Allow the user to add, amend and delete menu items, and save menu changes.
- Maintain a running total of order values.
- Maintain running totals of the quantity of each menu item ordered.
- Save the running totals to a file.
- Provide options to display the menu and running totals.

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

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 pseudocode 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 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.

BLANK PAGE

BLANK PAGE

BLANK PAGE