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

Computer Science

Programming Project

For use from September 2020

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 authentication 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 ▶

Pearson

Parents' Evenings

Crawdale Primary School holds parents' evenings over three days in November each year. During these evenings, parents can discuss their child's progress with the class teacher.

The evenings are organised using an appointment system which allows for 9 x 20 minute appointments on each day, starting at 5.00 pm and finishing at 8.00 pm.

Parents are invited to send in their two preferred appointment times. If possible, parents will be allocated one of their preferred appointment times. If both preferences are already booked, they will be offered an alternative appointment.

The school wants a computer program to help manage the parents' evening appointments. The program should:

- Use a file to record appointments for one class of 24 pupils. The file should cover 3 days and allow for a maximum of 9 appointments per day.
- Accept input of parents' details and appointment preferences.
- Allocate appointments according to stated preferences.
- Allocate alternative appointments** when preferences cannot be met.
- Display a timetable of appointments for each evening for printing (actual printing not required).
- Display details of individual appointments for parents.

****** You will need to devise a set of rules for the allocation of alternative appointments.

Refinements

Adapt the working system to:

- Allocate, where possible, consecutive appointments for parents of twins or siblings.
- Allow for the cancellation and reallocation of appointments according to parents' stated preferences.

Your task is to analyse this problem and to design, implement, test, refine and evaluate a programmed solution for the appointment system.

The Report

Create a folder called Report.
Save your work 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.