



GCSE

C500U20-1



TUESDAY, 20 MAY 2025 – AFTERNOON

COMPUTER SCIENCE – Component 2
Computer Programming

2 hours

The decryption password for the encrypted media files is chips-d8b96

C500U201
01

ADDITIONAL MATERIALS

You will require the WJEC supplied prototype Python files.

Your computer should be pre-installed with a word processing package and a functional copy of Python using the version specified in the pre-release materials.

INSTRUCTIONS TO CANDIDATES

Some questions should be answered in a word-processed document. All other questions will require the use of the Python IDE.

Save your work regularly.

INFORMATION FOR CANDIDATES

The total number of marks available for this examination is 80.

The number of marks is given in brackets at the end of each question or part-question.

You are reminded of the need for good English and orderly, clear presentation in your answers. The quality of your written communication, including appropriate use of punctuation and grammar, will be assessed in your answers.

Create a new word-processed document called ExamAnswers.

Open the PVMuseum.py file and the Payroll.py file and familiarise yourself with the contents.

Investigation

1. (a) Using the PVMuseum.py file:
 - (i) Provide a screenshot of the error message displayed following a failed login attempt. State the incorrect username AND incorrect password used. [3]
 - (ii) Provide a screenshot of the message displayed following a successful login attempt. State the username and password used. [3]
- (b) Using the Payroll.py file, provide a screenshot of the calculation results when a Gross Pay of 2000 is entered. [1]

Save your screenshots in your ExamAnswers document.

2. Identify **one** example of each of the following from the Payroll.py file:
 - (a) setting a variable to 0.0 [1]
 - (b) creating a button [1]
 - (c) copying data from a textbox into a variable [1]

Copy your selected examples into your ExamAnswers document.

3. Annotate the PVMuseum.py code to describe **one** example of each of the following:
 - (a) Setting a caption on a form/window [2]
 - (b) code to generate a button [2]
 - (c) code to return a 50 character left justified text string [2]
 - (d) on screen error message code. [2]

Provide your answers as code in the PVMuseum.py Python file.

Design

Parkwood Vale Museum would like you to design additional features for the system.

4. Design an additional feature that will calculate the cost of lighting a display.

Each light installation costs £200 plus a running cost of £4 per hour.

Design an algorithm that:

- allows the user to input the number of:
 - installations required.
 - hours lighting required.
- calculates and outputs the:
 - overall installation cost.
 - overall running cost of lighting.
 - total cost (installation + running cost).

You may assume all light installations are to be lit for the same amount of time.

Your algorithm should be written using pseudo-code and self-documenting identifiers.

[8]

Enter your algorithm into your ExamAnswers document.

Implementation

5. Parkwood Vale Museum would like to create a new form to implement the algorithm designed in question 4.
- (a) Create a new form. [1]
 - (b) Insert a title on the form "Calculate display lighting costs". [2]
 - (c) Create text boxes to allow a user to input each of the following:
 - Cost of each lighting installation.
 - Running cost of lighting per hour.
 - Number of installations required.
 - Number of hours lighting required. [4]
 - (d) Provide an appropriate label for each of the text boxes in part (c). Ensure that the layout is clear. [5]
 - (e) Create a text box to display the output of the calculations and add a suitable label. [3]
 - (f) Create a button and include code which calculates the total cost of lighting installations and outputs the result in the text box created in part (e). [4]
 - (g) Create a "Save" button and add code to the Python file to enable the saving of the latest calculation result in a file called results.txt displaying a "saved successfully" message. [4]
 - (h) Create a "Back" button to return the user to the main menu. [3]
 - (i) Explain how your program works by annotating the code you have written in your Python file. [5]

Provide your answers as code in a new Python file called lighting.py.

Testing

Parkwood Vale Museum requires you to test the program.

6. These details are to be stored, in a results.txt file, using your Python program:

- Cost of installation: 200
- Cost of lighting per hour: 4
- Number of installations: 3
- Number of hours lighting required: 12

Test the functionality of the Python program by providing screenshots showing:

- | | |
|--|-----|
| (a) the form completed with the above details | [1] |
| (b) a message confirming that the details have been stored | [1] |
| (c) the results.txt file open with the above details stored. | [2] |

Save your screenshots in your ExamAnswers document.

Refinement

Parkwood Vale Museum has asked you to carry out refinements to change the function and improve the accuracy of its code.

7. Changes to the tax system. Refine the code within Payroll.py to take account of the following changes.
- (a) (i) Change the Tax Rate to 21.5%. [2]
 - (ii) Describe the refinements you have made to your code for the change in Tax Rate by annotating your code. [2]
 - (b) (i) Change the National Insurance Rate to 10.5%. [2]
 - (ii) Describe the refinements you have made to your code for the change in National Insurance Rate by annotating your code. [2]

Provide your answers as code in Payroll.py.

Refinement testing

8. Provide a screenshot of your updated Payroll form showing all outputs when a gross pay of £2000.00 is input. [5]

Save your screenshot in your ExamAnswers document.

Evaluate

9. Evaluate how your final program meets Parkwood Vale Museum's requirements. You should consider:
- **two** refinements that your program succeeds in implementing
 - how the code achieves those refinements
 - areas for improvement in your final program. [6]

Enter your evaluation into your ExamAnswers document.

END OF PAPER

BLANK PAGE

BLANK PAGE