Candidate Name	Centre Number					Candidate Number				
						0				

GCSE

COMPUTER SCIENCE

COMPONENT 2

Computer Programming

SAMPLE ASSESSMENT MATERIAL

2 Hours

ADDITIONAL MATERIALS

You will require the prototype Python file: ParkWoodSchool.py with supporting files which should be pre-installed on your examination account.

Your computer should be pre-installed with a word processing package and a functional copy of Python 3.8.0.

INSTRUCTIONS TO CANDIDATES

Questions 1, 2, 4, 5, 7, 10 and 11 should be answered in a word-processed document. All other questions will require the use of the Python 3.8.0 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 answer to question 11.

Create a new word-processed document called ExamAnswers.

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

Investigation

1. Using the ParkWoodSchool.py file:

   (a) Provide a screenshot of the error message displayed following a failed login attempt. State the username and password used. [2]

   (b) Provide a screenshot of the message displayed following a successful login attempt. State the username and password used. [2]

Enter your answers in your ExamAnswers document.

2. Identify **one** example of each of the following from the ParkWoodSchool.py file. Copy each example identified into your word-processed document:

   (a) a Boolean value; [1]

   (b) iteration; [1]

   (c) selection. [1]

Enter your answers in your ExamAnswers document.

3. Describe **one** example of each of the following using annotation in the ParkWoodSchool.py file:

(a) a user-defined function or subroutine; [2]

(b) assignment; [2]

(c) writing information to a file; [2]

(d) code to generate a button on a form. [2]

Enter your answers as code in the ParkWoodSchool.py Python file.

Design

Park Wood School would like you to design additional features for the system.

4. Design an algorithm that accepts the input of a pupil's first name. Your algorithm should output a suitable error message if the data entered contains any numeric values.

Your algorithm should be written using pseudo-code and self-documenting identifiers. [6]

Enter your answers in your ExamAnswers document.

5. Park Wood School requires an additional feature that will calculate the student loan repayment amounts which need to be deducted from staff salaries.

You may assume that student loan repayments are 10% of gross pay.

Design an algorithm which:

- allows the user to input the employee's gross pay;
- calculates the student loan repayment;
- calculates the net pay by deducting the student loan from the gross pay;
- outputs the result of the calculations.

Your algorithm should be written using pseudo-code and self-documenting identifiers. [5]

Enter your answers in your ExamAnswers document.

Implementation

6. Park Wood School would like to create a new form to store pupil details.

(a) Create a new form. [1]

(b) Insert a title on the form "Add Pupil". [2]

(c) Create a text box and provide appropriate labels to allow a user to input each of the following:

- Pupil ID
- First name
- Surname
- Form class
- Date of birth [3]

(d) Create a functioning "Back" button that returns the user to the main menu. [2]

(e) Add code to implement a presence check on Pupil ID. [2]

(f) Create a "Save" button and add code to the Python file to enable the saving of the above details in a file called "pupilDetails.txt" displaying a confirmation message. [3]

(g) Implement the validation check that you designed in question 4 to check that a pupils first name does not contain a numeric character. [3]

(h) Explain how all the new functionality works by annotating the code you have added within your Python file. [6]

Enter your answers as code in a new Python file called Pupils.py.

Testing

Park Wood School requires you to carry out the following tests on the program.

7. (a) Test the functionality of the presence check on the Pupil ID field and provide a screenshot of the error message. [1]

(b) The following pupil details are to be stored using your Python program:

- Pupil ID: 1101
- First name: Kwai
- Surname: Oer
- Form class: 4B
- Date of birth: 6/1/2015

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

(i) the form completed with the above details; [1]

(ii) a message confirming that the pupil details have been stored; [1]

(iii) the pupilDetails.txt file open with the above details stored; [2]

Enter your answers in your ExamAnswers document.

Refinement

Park Wood School has asked you to carry out the following code refinements to change the function and improve the efficiency of their code.

8. Park Wood School is aware of changes to the tax system that they will have to implement. Refine the code within ParkWoodSchool.py to take account of the following changes.

(a) (i) Increase the Tax Rate from 20% to 22%. [1]

(ii) Describe the refinements you have made to your code for the increase in Tax Rate by annotating your code. [2]

(b) (i) Change the National Insurance Rate to 8.5% [1]

(ii) Describe the refinements you have made to your code for the change in National Insurance Rate by annotating your code. [2]

(c) (i) Refine your program, using your design work from Question 5, to include this calculation in your payroll form. [2]

(ii) Describe the refinements you have made to include the student loan deduction by annotating your code. [2]

Enter your answers as code in Payroll.py.

9. The code within the ParkWoodSchool.py program makes use of many nested if statements which are used to display an error.

(a) Refine this code to make it more efficient. [3]

(b) Describe the refinements you made by annotating the code. [2]

Enter your answers as code in ParkWoodSchool.py.

Refinement testing

10. Provide a screenshot of your Payroll form showing all outputs when a gross pay of £1,850.00 is input. [6]

Enter your answer in your ExamAnswers document.

Evaluate

11. Discuss how your final program meets the revised 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 answers in your ExamAnswers document.

End of Paper.