Pearson Edexcel Level 1/Level 2 GCSE

Computer Science

Practical Programming

Controlled Assessment Material
For use from January 2016 to May 2016

Paper Reference
1CP0/02

You must have:
- paintingJobs.txt
- test_plan.doc

Instructions

- You must complete all tasks.
- You must use a computer to complete these tasks.
- You may **not** have access to the internet.
- You may have access to a printer.
- You may use a calculator.
- You will be given access to relevant files during the controlled assessment.
- All files must be saved and kept secure in the folder specified by your centre between sessions. You will not have access to these files between sessions.
- You must save the final versions of your files in the **CA folder** indicated by your centre.

Information

- The total mark for this controlled assessment is 50.
- The marks for **each** task are shown in brackets.
- The total time for the controlled assessment is 15 hours.
- A guide to how much time you should spend is shown at the end of each task.
- Tasks labelled with an **asterisk** (*****) are ones where the quality of your written communication will be assessed
  – *you should take particular care on these tasks with your spelling, punctuation and grammar, as well as the clarity of expression.*

Advice

- Read each task carefully before you start to answer it.
- Attempt every task.

Turn over ▶

PEARSON

Controlled assessment tasks

The controlled assessment tasks must be completed using Python, Java or any C-derived language.

Context

Caroline runs a painting and decorating company. The company specialises in indoor painting. Customers can call, email, or visit her store to enquire about the cost of having their rooms painted.

Caroline needs a system that can:

- hold employee information
- generate an estimate to show the cost of painting a customer's rooms
- monitor any outstanding customer payments.

Features of the system

- The user interface must be text based.

- Caroline must be able to:
  - enter employee information
  - enter customer information
  - enter the dimensions of each room
  - generate an estimate.

- To generate an estimate the system needs to:
  - calculate the surface area of the walls in square metres (height x width)
  - add £15 per square metre for paint
  - add £70 per room if wallpaper needs to be removed
  - add on the employee charge:
    - £250 for a fully-qualified painter
    - £100 for an apprentice.

- Caroline must be able to:
  - search for an estimate
  - view outstanding payments
  - view how much money has been paid to the company (revenue).

You must complete the following tasks.

Task	Features	Marks	Suggested time
1 Entering employee details Program with comments	• Prompt user to enter employee details	8	1½ hours
2 Generating an estimate Design, program, test, evaluate	• Design interface • Input house details • Generate estimate	18	5½ hours
3 Tracking jobs and revenue Design, program, test, evaluate	• Input painting jobs from a file • Generate financial information	24	8 hours

A summary of the files supplied to you and a checklist of files you should submit is given on **page 11**.

Task 1: Entering employee details

Caroline needs to enter employee details before she can assign someone to a job.

Requirements

Here is an outline design of your program.

- Display a title stating "Employee details".
- Ask the user to enter the employee's ID, name, telephone number.
- Ask the user to enter the employee's qualification level using the codes:
  - AP for apprentice
  - FQ for fully-qualified.
- Display the employee details on the screen:
  - If the user entered AP in the qualification level, it should display it as "Apprentice", if they entered FQ it should display as "Fully-qualified". If any other data is entered it should display an error message.
- Ask the user to confirm the information is correct:
  - If any information is incorrect, ask the user to re-enter all of the data again and then display the employee details again.
- Return to the title message.

The employee data does not need to be saved.

Your task

1 Write a program to implement this design. It must be easy to read and include comments to explain how it works.

Save the final version of your source code as a file called **employee** in the **CA folder**.

You are advised to spend no more than 1½ hours on Task 1.

(Total for Task 1 = 8 marks)

BLANK PAGE

Task 2: Generating an estimate

Caroline must be able to enter details of the customer and of the house that requires painting. She also needs to charge customers VAT at 20%.

The system must then calculate how much the job will cost.

Requirements to generate an estimate

- The user interface must be text based.
- Display a welcome message.
- Ask the user for this information:
  - Customer number, date of estimate, number of rooms that require painting:
    - For each room, ask for the room name (e.g. lounge, dining room) and the number of walls in the room:
      - Ask if wallpaper needs to be removed – if yes, charge £70 per room
    - For each wall, ask the height and width of the wall in metres.
- Calculate the total surface area of the walls in square metres.
- Multiply the total surface area of the walls by £15 per square metre for paint.
- Ask what type of employee should be assigned to the job.
- Add on the cost of the employee – if apprentice add on £100, if fully-qualified add on £250.
- Calculate and display the total.
- Add 20% VAT to the total.
- Calculate and display the final total.
- Ask the user if they wish to generate another estimate:
  - If yes, return to the welcome message
  - If no, display an exit message.

The data output from this activity does not need to be saved. There is no need to validate the user input in this task.

Your task

2 (a) (i) Develop a program to meet these requirements. It must be easy to read and include comments to explain how it works. You are expected to decompose your program into subprograms. The quality of your program design will be assessed using the source code you submit.

(ii) Write and implement a plan to test that the program works correctly and that the requirements have been met. Use the **test_plan.doc** template provided. A copy of this is shown in *Appendix B*.

Save the final version of your source code as a file called **estimate** in the **CA folder**.

Save your test plan in a file called **estimate_test_plan.doc** in the **CA folder**.

*(b) Write an evaluation of your program that:

- assesses how well your program meets the system requirements
- explains how your program calculates the final total.

You should use correct technical terms where appropriate.

Save your evaluation in a file called **estimate_evaluation.doc** in the **CA folder**.

You are advised to spend no more than 4½ hours on developing and testing the program and 1 hour on the evaluation.

(Programming and testing = 12 marks)

(Evaluation = 6 marks)

(Total for Task 2 = 18 marks)

Task 3: Tracking jobs and revenue

Caroline needs to keep track of job and revenue information. You will write a program to enable this to happen.

The program will sort and search the file **paintingJobs.txt** to select and display information about estimates and jobs.

The **paintingJobs.txt** file shows Estimate Number, Estimate Date, Customer ID, Final Total (in £ pounds), Status (E for Estimate, A for Accepted job or N for Not accepted), and Amount Paid (in £ pounds). A copy of this file is shown in *Appendix A*.

Requirements

- Display four options for the user to select from.

Option A	Search for an estimate
Option B	Display outstanding payments
Option C	Display total revenue

Enter Q to quit

- The user must be able to select an option or enter 'Q' to quit the program.
- All data must be read from the **paintingJobs.txt** file.
- The program must work for all of the records in the **paintingJobs.txt** file.

Option A must provide a prompt for the user to enter the estimate number.

- It must find the customer ID and estimate details for the entered estimate number.
- Here is a suggested format for displaying the data.

```
Estimate Number: <the estimate number>
Customer ID: <the customer ID>
Estimate Amount: <the final total>
Estimate Date: <the estimate date>
Status: <the status>
```

Option B must show **all** jobs with outstanding payments. This only applies to estimates with a status of A (Accepted). Payments are outstanding if the amount paid is less than the final total.

- It must only show outstanding payments for jobs with a status of ‘A’.
- It must calculate and display the total of the outstanding payments.
- Here is a suggested format for displaying the data.

Estimate Number	**Customer ID**	**Final Total**	**Amount Paid**	**Amount Outstanding**
Estimate Number	*Customer ID*	*Final Total*	*Amount Paid*	*Amount Outstanding*
Estimate Number	*Customer ID*	*Final Total*	*Amount Paid*	*Amount Outstanding*
			Total Outstanding:	*Total Outstanding*

Option C must calculate the total revenue for jobs that have been accepted **and** fully paid.

- The calculation must only include jobs with a status of ‘A’.
- The calculation must only include jobs that have been fully paid.
- Here is a suggested format for displaying the data.

Total Revenue

The company’s total revenue so far is <*the total revenue*>

Your task

3 (a) (i) Develop a program to meet these requirements. It must be easy to read and include comments to explain how it works. You are expected to decompose your program into subprograms. The quality of your program design will be assessed using the source code you submit.

(ii) Write and implement a test plan to ensure that the program works correctly and that the requirements have been met. Use the **test_plan.doc** template provided. A copy of this is shown in *Appendix B*.

Save the final version of your source code as a file called **tracking** in the **CA folder**.

Save your test plan as a file called **tracking_test_plan.doc** in the **CA folder**.

*(b) Write an evaluation of your program that:

- assesses how well your program meets the requirements
- explains which method you used to extract information for each option from the data file.

You should use correct technical terms where appropriate.
Save your evaluation as a file called **tracking_evaluation.doc** in the **CA folder**.

You are advised to spend no more than 6½ hours on developing and testing the program and 1½ hours on the evaluation.

(Programming and testing = 15 marks)
(Evaluation = 9 marks)
(Total for Task 3 = 24 marks)

TOTAL FOR PAPER = 50 MARKS

Files supplied to candidates

Purpose	File name	File type
Painting jobs	paintingJobs	.txt
Test plan template	test_plan	.doc

Checklist of files for the candidate to submit

Task	File name	File type
1 program	employee	program source code
2 program	estimate	program source code
2 test plan	estimate_test_plan	document
2 evaluation	estimate_evaluation	document
3 program	tracking	program source code
3 test plan	tracking_test_plan	document
3 evaluation	tracking_evaluation	document

Appendix A

You have been given the file **paintingJobs.txt**. A copy of the file is shown here.

The data is shown as Estimate Number, Estimate Date, Customer ID, Final Total (in £ pounds), Status (E for Estimate, A for Accepted job or N for Not accepted), and Amount Paid (in £ pounds).

```
E5341,21/09/2015,C102,440,E,0
E5342,21/09/2015,C103,290,A,290
E5343,21/09/2015,C104,730,N,0
E5344,22/09/2015,C105,180,A,180
E5345,22/09/2015,C106,815,A,400
E5346,23/09/2015,C107,970,N,0
E5347,23/09/2015,C108,1050,E,0
E5348,23/09/2015,C109,370,A,200
E5349,25/09/2015,C110,480,A,250
E5350,25/09/2015,C111,330,A,330
E5351,25/09/2015,C112,1750,E,0
E5352,28/09/2015,C113,1500,N,0
E5353,28/09/2015,C114,272,A,200
E5354,29/09/2015,C115,560,E,0
E5355,29/09/2015,C116,530,A,450
E5356,29/09/2015,C117,860,E,0
E5357,29/09/2015,C118,650,E,0
E5358,29/09/2015,C119,380,A,380
E5359,29/09/2015,C120,980,N,0
E5360,30/09/2015,C121,1375,E,0
E5361,01/10/2015,C122,374,A,374
E5362,01/10/2015,C123,867,E,0
E5363,01/10/2015,C124,930,A,500
E5364,02/10/2015,C125,915,A,800
E5365,02/10/2015,C126,1486,A,1486
E5366,03/10/2015,C127,576,E,0
E5367,03/10/2015,C128,427,A,350
```

```
E5368,05/10/2015,C129,1346,A,1346

E5369,07/10/2015,C130,1756,N,0

E5370,07/10/2015,C131,432,E,0

E5371,10/10/2015,C132,289,A,289

E5372,10/10/2015,C133,923,E,0

E5373,10/10/2015,C134,1023,A,550

E5374,10/10/2015,C135,2070,E,0

E5375,10/10/2015,C136,687,N,0

E5376,13/10/2015,C137,297,A,297

E5377,13/10/2015,C138,829,E,0

E5378,13/10/2015,C139,1503,E,0

E5379,14/10/2015,C140,850,E,0

E5380,15/10/2015,C141,425,A,425
```

Appendix B

You have been given the file **test_plan.doc**. A copy of the file is shown here.

Test number	Description of test	Test data	Expected result	Actual result	Pass/Fail