Coursework Task

Higher Computing Coursework Task 2009–2010

Part 1

Click-Kit, a company which sells photographic equipment, decides to commission a software development company to produce a stock control program that will:

- store the names of items held in stock
- calculate product codes of held items
- produce a list of products stocked, with their initial stock levels
- allow information on products held in stock to be found
- allow products to be purchased
- produce a list of final stock levels, indicating which items need to be re-stocked.

The names of five digital cameras and their stock levels which should be used in testing the program are given below.

Product Name	Stock Level
Simpsun GN120	1
Sonic Lux10	2
Ultimax G42	4
Antalpha A200	3
Nickov N230	2

You may use any suitable method available in your programming environment to enter this data into the system.

How the program should work

Storing the Product Names

The program should take in the product names and initial stock levels. It should store them in a suitable data structure.

Calculating and Storing the Product Codes

The program should calculate the product codes using the first three and last three characters of the product name. It should store them in a suitable data structure.

Displaying the Initial Stock List

The program should display a list of the product names, their product codes and the initial stock levels (see page 8 for an example layout).

Displaying a Menu

The program should then offer the user the following three menu options:

- **Finding a Product**

  The program should ask the user for a **product code** then find the product. If the product code is **not** found then an appropriate message should be displayed. Otherwise it should display the product name, product code and stock levels.

- **Purchasing an Item**

  The program should ask the user to enter the product code of an item. If the product is **not** in stock a suitable error message should be displayed. Otherwise a suitable message should be displayed confirming it is in stock and confirming the purchase. The stock level for that item should then be decreased by one.

  Note: this part of the program should be tested by the user purchasing a **Sonic Lux10** and **Nickov N230**.

- **Quit**

  The program should display a suitable message.

Displaying the Final Stock List

After quit is chosen, the program should display a list of the product names, their product codes and the final stock levels. The program should also display a re-order message for each product where the number in stock is less than two.

Displaying the Results

The output from the program should be in columns, similar to the examples shown below.

- The **initial stock list** should look something like this . . .

```
Product Name        Product Code        Initial Stock Level
Simpsun GN120       Sim120              1
Sonic Lux10         Sonx10              2
Ultimax G42         UltG42              4
Antalpha A200       Ant200              3
Nickov N230         Nic230              2
```

- The output of **finding a product** should look something like this . . .

```
Product to be found:  UltG42

Product Name        Product Code        Stock Level
Ultimax G42         UltG42              4
```

- The output of **purchasing a product** should look something like this . . .

```
Product to be purchased:  Sonx10

Product Name        Product Code        Status
Sonic Lux10         Sonx10              In stock
Purchase confirmed
```

- The **final stock check** should look something like this . . .

```
Product Name        Product Code        Stock     Action
Simsun GN120        Sim120              1         Re-order
Sonic Lux10         Sonx10              1         Re-order
Ultimax G42         UltG42              4
Antalpha A200       Ant200              3
Nickov N230         Nic230              1         Re-order
```

Algorithm

```
1.    Enter and store product names and initial stock levels
2.    Calculate and store product codes
3.    Display product names and codes
4.    Start conditional loop
5.        Display menu
6.        Get option from user
7.        Where option is F, perform Find a product
8.        Where option is P, perform Purchase a product
9.        Where option is Q, perform Quit
10.   End conditional loop when Q is chosen
11.   Display final stock check
```

What you have to do:

Tasks		Evidence required	Marks
1	Indicate data flow on the algorithm.	Algorithm with data flow.	3
2	Refine steps 2, 7 and 8 of the algorithm.	Pseudocode for steps 2, 7 and 8.	7
3	Using a software development environment of your choice, implement the algorithm. Use separate sub-programs where appropriate. Use parameter passing where appropriate.	Listing of implemented program.	16
4	Test the program with the data provided to ensure that it is fit for purpose.	Hard copy of test results.	1
5	Evaluate the test results.	Brief report on test results.	3