Centre Number						Candidate Number				
Surname										
Other Names										
Candidate Signature										

GCSE
For submission in 2015

Computer Science 4512 4512/CB4

Unit **4512/1 – Practical Programming**

Scenario 4: Traditional Application

AQA Code Puzzle

For candidates entering for the 2015 examination
To be issued to candidates on or after 4 June 2013

This scenario is one of four available. Each of the four scenarios is available in a separate candidate booklet. You must complete ***two*** *of the four scenarios.*

- You have approximately 25 hours in which to complete this scenario.
- Before starting work on the problem, read the whole of this Candidate Booklet thoroughly. You can ask your teacher to explain anything in this booklet, except Computer Science specific terms, that you do not understand.
- There are restrictions on when and where you can work on this problem. Your teacher will explain them to you. For example, you should only do work that you intend to hand in for marking when a teacher is present, so that he or she can confirm that the work is your own. The Candidate Booklet must **not** be taken outside your school/college.
- You may need to use the Internet to research certain parts of the problem. This does not have to be within the 25 hours recommended time.
- You will need to complete and sign a Candidate Record Form which your teacher will provide.

Information

You will also be marked on your use of English. It is important to:

- make sure that all your work is legible
- use correct spelling, punctuation and grammar
- use a style of writing which suits the person you are writing for
- organise your information clearly, so that you make yourself understood
- use Computer Science terms where they are needed.

Scenario 4: AQA Code Puzzle

AQA Code Puzzle is a logic puzzle. In the puzzle, ten words are shown to the user in a coded form. The user has to work out what the ten words are by cracking the code.

The words have been coded by using symbols instead of letters. Each occurrence of a particular letter in the ten words has been replaced by the same symbol. Each symbol used in the coded words represents a unique letter.

Example

If A is represented by the symbol `#`, `M` by `*`, and `N` by `%`, then the word `MANNA` would be represented by the symbols `*#%%#`

An external file **words.txt** will be provided by your teacher. This file contains a list of ten coded words that you should use to test that your program works correctly. The contents of this file are shown in **Figure 1**. Each coded word appears on a separate line.

Figure 1

```
#+/084&"
#3*#%#+
8%203:
,1$&
!-*%
.#7&33&
#*#71%
&-&641'2
#))85
9&330*
```

An external file **clues.txt** will also be provided by your teacher. This file contains three letter and symbol pairings. The contents of this file are shown in **Figure 2**. Each pairing appears on a separate line.

Figure 2

```
A#
M*
N%
```

When the data in the file **clues.txt** is used to substitute letters for symbols in the list of ten words from the file **words.txt** the result will be as shown in **Figure 3.**

Figure 3

```
A+/084&"
A3MANA+
8N203:
,1$&
!-MN
.A7&33&
AMA71N
&-&641'2
A))85
9&330M
```

The user will attempt to solve the puzzle by replacing symbols with letters. For example, if the user has decided that the letter E is represented by the symbol & the result of making this substitution would be as shown in **Figure 4**.

Figure 4

```
A+/084E"
A3MANA+
8N203:
,1$E
!-MN
.A7E33E
AMA71N
E-E641'2
A))85
9E330M
```

The user will then continue to replace symbols with letters until all symbols have been replaced. An external file **solved.txt** will be provided by your teacher. This file should be used to check the user's solution to the puzzle. The contents of this file are shown in **Figure 5**.

Figure 5

```
ACQUIRED
ALMANAC
INSULT
JOKE
HYMN
GAZELLE
AMAZON
EYEBROWS
AFFIX
VELLUM
```

Tasks

1. Develop the part of the program that loads the list of ten coded words from the external file **words.txt** that your teacher has given you, and displays the content of this file in an appropriate way. **Figure 1** shows the content of this file.

2. Develop the part of the program that loads the three letter and symbol pairings from the external file **clues.txt** your teacher has given you, and displays the content of the file in an appropriate way. **Figure 2** shows the content of this file.

3. Develop the part of the program that displays the list of ten words with the substitutions made based on the letter and symbol pairings. An example is shown in **Figure 3** for the substitutions made for the letters `A`, `M` and `N`.

4. Develop the part of the program that allows the user to enter a letter and symbol pairing.

    a. If the **letter** entered has already been matched to a **symbol** then they are told that the letter is already matched and they are asked to enter another letter and symbol pairing.

    b. If the **symbol** entered has already been matched to a **letter** then they are told that the symbol is already matched and they are asked to enter another letter and symbol pairing.

5. Develop the part of the program that allows the user to delete a letter and symbol pairing.

6. Develop the part of the program that allows **Tasks 3, 4 and 5** to be repeated continuously until all symbols have been substituted by letters for each word in the list.

7. Develop the part of the program that checks the user's list of words against the external file **solved.txt** and displays an appropriate 'success' or 'fail' message. If the puzzle has not been completed correctly, the user should be allowed to continue to try to solve the puzzle or exit the program.

8. When trying to solve puzzles of this type, it may help the user to know the frequency with which each symbol is used in the code words. For example, the symbol # is used eight times, * is used four times and % is used four times in the code words shown in **Figure 1**.

    Extend the program so that it calculates the frequency with which the symbols are used in the code words. Display the results of this calculation in an appropriate format.

In addition

1. Your Portfolio

Remember that you are looking to provide an application that will allow a person to fully complete the puzzle.

You are free to use whatever software tools and techniques are available to you.

What your teacher will be looking for and how to provide that evidence for your Portfolio

In preparing you for this unit of work, your teacher will have provided you with more information about the section headings below.

Part 1 – Design of solution

Design of solution (0–9 marks available)
What you must do
• Show an understanding of what the problem involves with reference to the user's needs. • Produce an overview plan that shows how the problem is to be solved. • Produce pseudo code (or suitable alternative) showing the main blocks within the proposed solution.

Part 2 – Solution development

Solution development (0–9 marks available)
What you must do
• Show evidence of an understanding of how the final solution meets the needs of the user. • Produce annotated code that demonstrates an understanding of the programming techniques used.

Part 3 – Programming techniques used

Programming techniques used (0–36 marks)
What you must do
• Show an understanding of the programming techniques used and how the different parts of the solution work together. • Explain/justify the choice of programming techniques used to create a solution that has been coded efficiently. • Show evidence for the purpose and use of data structures. • Show the techniques used (appropriate to the language used) within the code to make the solution robust.

Part 4 – Testing and evaluation

Testing and evaluation (0–9 marks available)
What you must do
• Produce a test plan that shows the expected tests, test data and expected results. • Show that the planned tests have been carried out and provide a record of the actions taken. • Evaluate how the final solution meets the needs of the user.

Turn over ►

2. Organising your Portfolio of work

Your Portfolio is where you keep the evidence that you have produced.

You should imagine that the Portfolio is to be used by another person who is interested in how you produced your solution. It is to help them to do something similar. It is important that you organise work for the Portfolio as shown below.

- You must keep all the work you produce for the organiser in hard copy in a Portfolio (or save your work electronically in folders which you will later copy onto a CD or DVD). Your teacher will have instructed you what to do.
- If you are putting hard copy printouts in your Portfolio make sure that you number each page and fasten it all together. Take your work out of any plastic sleeves before you hand it in to your teacher for marking.
- Each page should have your name, centre number and candidate number clearly shown on it.
- When you have completed this scenario, if you are putting your work on a CD or DVD, put the work for each heading on page **5** of this booklet in a separate folder. Each folder must be clearly named *(for example, 'Design of solution', 'Solution development' etc)*. Inside each folder the work must have a filename (*for example, 'What the problem involves', 'Control statements' etc*) which should be a final version for each heading. The CD or DVD should have your name, centre number and candidate number clearly shown on it. Your teacher will have advised you what to do.

END OF CANDIDATE BOOKLET