Released September 2014
For Assessment Submission
June 2015 to June 2016

GCSE COMPUTING

A453 Programming Project

CONTROLLED ASSESSMENT MATERIAL 3

This assessment may be periodically reviewed. Please check on OCR Interchange that you have the Controlled Assessment material valid for the appropriate assessment session.

A453

4868409792

INSTRUCTIONS TO TEACHERS

- Please refer to Section 4 of the GCSE Computing specification for instructions on completing this controlled assessment task.
- The marking criteria should be available to candidates whilst completing the tasks.
- The quality of written communication will be assessed in the testing section.
- The total number of marks for this unit is **45**.

INFORMATION FOR CANDIDATES

- This document consists of **4** pages. Any blank pages are indicated.

Teachers are responsible for ensuring that assessment is carried out against the Controlled Assessment set for the relevant examination series (detailed above).

Assessment evidence produced that does not reflect the relevant examination series will not be accepted.

© OCR 2014 [D/600/3257]
DC (LEG) 104066/1

OCR is an exempt Charity

Turn over

This assessment consists of three tasks.

Candidates should complete all tasks and provide evidence to meet all the marking criteria.

For the following scenario analyse the detailed requirements for each situation and, using suitable algorithms, design a solution to be coded in a suitable high-level programming language. Show the iterative development of the individual solutions with suitable testing throughout the process. Test the final products and evaluate your solutions against the detailed requirements you identified in the analysis.

The results for a task may be used without further testing in any subsequent task, or each of the tasks may be solved as a separate system.

Encryption

Task 1

One simple encryption technique is the Caesar cipher. This involves taking the letters of the alphabet and offsetting them by a fixed number of positions to replace letters in the original message (plaintext) by the offset characters.

For example, an offset of 5 gives the following:

Original Alphabet	`ABCDEFGHIJKLMNOPQRSTUVWXYZ`
	→
Offset by 5 we get	`FGHIJKLMNOPQRSTUVWXYZABCDE`

The message
becomes
	`COMPUTING IS FUN`
	`HTRUZYNSL NX KZS`

Analyse the requirements for this program and design, develop, test and evaluate a program to enter, encrypt and decrypt messages. You must use a high-level language in order to implement the Caesar cipher.

Task 2

A more secure method of encryption would be to use a keyword. A keyword is the information needed in order to transform the plaintext into the encrypted message and is also used by the recipient to decrypt the message. For example, a keyword could be the letters `GCSE`. The keyword is repeated enough times to match the length of the plaintext message. The alphabet value (position of the letter in the alphabet) of each letter of the key phrase is added to the alphabet value of each letter of the plaintext message to generate the encrypted text.

For example, if we use the keyword `GCSE` then we make up the key phrase for `COMPUTINGISFUN` by repeating the letters within `GCSE` to make a 14 letter code, `GCSEGCSEGCSEGC`, where G is the 7th letter of the alphabet, C the 3rd, S the 19th and E the 5th.

Message	C	O	M	P	U	T	I	N	G	I	S	F	U	N
Alphabet value	3	15	13	16	21	20	9	14	7	9	19	6	21	14
Keyword	G	C	S	E	G	C	S	E	G	C	S	E	G	C
Alphabet value of keyword	7	3	19	5	7	3	19	5	7	3	19	5	7	3
Add the alphabet value of the letter to the alphabet value of the keyword	10	18	32	21	28	23	28	19	14	12	38	11	28	17
Encrypted message	J	R	F	U	B	W	B	S	N	L	L	K	B	Q

Analyse the requirements for this program and design, develop, test and evaluate a program to enter, encrypt and decrypt messages. You must use a high-level language to implement this encryption technique. The program must be able to accept keywords of any length. Consider how you will deal with spaces in the original message.

Task 3

It is more secure to use two keywords to encrypt data. Whole files may need to be encrypted.

Analyse in detail how to use two keywords to encrypt and save a text file you have created and to decrypt this saved text file. Develop, test and evaluate a program to encrypt and decrypt text files. You must use a high-level language to implement your solution.

Copyright Information

OCR is committed to seeking permission to reproduce all third-party content that it uses in its assessment materials. OCR has attempted to identify and contact all copyright holders whose work is used in this paper. To avoid the issue of disclosure of answer-related information to candidates, all copyright acknowledgements are reproduced in the OCR Copyright Acknowledgements Booklet. This is produced for each series of examinations and is freely available to download from our public website (www.ocr.org.uk) after the live examination series.

If OCR has unwittingly failed to correctly acknowledge or clear any third-party content in this assessment material, OCR will be happy to correct its mistake at the earliest possible opportunity.

For queries or further information please contact the Copyright Team, First Floor, 9 Hills Road, Cambridge CB2 1GE.

OCR is part of the Cambridge Assessment Group; Cambridge Assessment is the brand name of University of Cambridge Local Examinations Syndicate (UCLES), which is itself a department of the University of Cambridge.

© OCR 2014