



**Released June 2012  
For Assessment Submission  
January 2013 to June 2016**

**GCSE COMPUTING**

**A453/01** Programming Project

**CONTROLLED ASSESSMENT MATERIAL 5**

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



**INSTRUCTIONS TO TEACHERS**

- Please refer to Section 4 of the GCSE Computing specification for instructions on completing this controlled assessment task.
- Each task can be contextualised appropriately to suit facilities available in your centre.
- 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.**

**This assessment consists of three tasks.**

**Candidates should complete all tasks.**

The tasks are set to enable techniques identified in the specification to be demonstrated in the solution. The tasks provide opportunities to demonstrate a range of skills and all three tasks contribute to the overall mark awarded for this assessment. Marks are awarded for using the appropriate skills and techniques effectively and efficiently to produce a solution to these three tasks. Not all techniques will be required for each of the subtasks. You must identify the requirements for each task, design a solution using appropriate techniques, code the solution, and test and evaluate this solution against the identified success criteria.

**Task 1 – Binary / Decimal converter**

Design, code, test and evaluate a program (or programs) that will convert between binary and decimal.

For the decimal to binary converter the program should accept a positive value and output the binary equivalent. The system need only be tested for values up to 255.

One method for converting from decimal to binary is shown below.

To convert the decimal value 43 to binary:

$43 \div 2 = 21$	Remainder 1
$21 \div 2 = 10$	Remainder 1
$10 \div 2 = 5$	Remainder 0
$5 \div 2 = 2$	Remainder 1
$2 \div 2 = 1$	Remainder 0
$1 \div 2 = 0$	Remainder 1

Once the result is 0 the remainder values form the binary equivalent of our decimal number  
43 decimal = 101011 binary

For the binary to decimal converter the program should only accept valid inputs (a series of 0s and 1s), and output the decimal equivalent. The program need only be tested for inputs of up to 8 binary digits.

One method for converting from binary to decimal is shown below.

To convert the binary value 1101 to decimal:

Take each value in the binary string starting at the right-hand end, the least significant bit, and multiply by 1 then  $1 \times 2$ , then  $1 \times 2 \times 2$ , then  $1 \times 2 \times 2 \times 2$  etc.

$1 \times 1$	=1
$0 \times 1 \times 2$	=0
$1 \times 1 \times 2 \times 2$	=4
$1 \times 1 \times 2 \times 2 \times 2$	=8

Add the decimal values to get the decimal equivalent of 1101, which is 13.

**Task 2 – Adding binary numbers**

Design, code, test and evaluate a program that will accept two binary values (up to 8 binary digits) and output their total in binary. The output should not contain any leading zeros.

**Task 3 – Binary logic**

A food vending machine accepts 10p, 20p, 50p and £1 coins.

One or more coins are inserted and the current credit is calculated and displayed.

A product is selected from those available. The system checks to see if there is enough credit to purchase the product chosen.

If there is not enough credit the system displays an error message.

If there is enough credit it dispenses the product, updates the credit available and displays the remaining credit.

Further selections can be made if there is enough credit.

The vending machine simulation should have five products and prices.

Design, code, test and evaluate a program for this simulation.

**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](http://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.