**Lesson 1**

**Activity 2 – Engagement**

**Assembly Language:**

Operation

Input 0001 ‘takes input into an Accumulator

Output 0010 ‘Outputs the contents of Accumulator

Add 1010 ‘Adds an 8 bit value to the Accumulator

Add (memory) 1011 ‘Adds a memory location to the Accumulator

Sub 1001 ‘Subtracts an 8 bit value from the Accumulator

Sub (memory) 1101 ‘Subtracts a memory location from the Accumulator

Store 1100 ‘Stores the contents of accumulator in a memory location

**Machine Code Example:**

0001 ‘This line of code allows the user to input a value into the Accumulator

1100 00000100 ‘This uses the operation Store to put the contents of the Accumulator into

‘memory address number 4 – identified by the 8 bit number 00000100

0001 ‘This line of code allows the user to input a value into the Accumulator

1011 00000100 ‘This will Add the contents of the memory location signified by 00000100

‘to the value stored in the accumulator

Imagine if the user input the value 10 – this would be placed into the accumulator and then a copy would be transferred to memory address 4. If the user then input the value 6 – this would replace 10 in the accumulator. The final line of code would add the contents of memory address 4 (the value 10) to the value held in the accumulator (6) – therefore the final value held in the accumulator would be 16.

E.g of invalid code:

**1111** ‘There is no instruction in our in our instruction set for 1111

0001 ‘This would be valid as it would allow the user to enter a value into the Accumulator

**1010 1011** ‘This is invalid as it is adding a 4 bit value rather than 8 bit as specified above

0001 ‘This would allow a new value to replace the existing in the Accumulator

1011 10111101 ‘This correctly adds the contents of a memory location to the Accumulator.

The code above is invalid in two places – first it uses an instruction that is not provided in the instruction set and therefore would cause the assembling to stop. Secondly a line of code is used to add the contents of a memory location to the Accumulator – the code given is only 4 bits as oppose to 8 bits, therefore this would not be accepted. The rest of the code is valid.

**Find the errors in this code:**

### Task 1

The program should allow the user to input a value and then store it in memory address 25

0001

1100

### Task 2

The following program allows the user to enter a value – the number 3 is added to the value and then output. Another value is then entered and the number 1 is added to the value – this is then also output.

0001

1010 000011

0010

0001

1010 00000001

0001 00000001

### Task 3

This program allows the user to enter 3 values – all of the values are added together and output.

0001

1100 00010000

0001

0011 00010001

0001

1011 00010001

1010 00010000

0010

### Task 4

Using the instructions given above – write a program that allows the user to enter two values which are added together and subtracted from a third value that is entered. This is then output to the user.

## Lesson2 - Answers

### Task 1

0001 ‘This line is fine as it allows the user to enter a value into the Accumulator

**1100** ‘This line of code is not valid as it does not specify the memory address

### Task 2

0001 ‘This is valid

1010 **000011** ‘This is meant to add the value 3 – but the value is not an 8 bit Binary Number

0010 ‘This correctly outputs the contents of the Accumulator

0001 ‘This correctly allows input into the Accumulator

1010 00000001 ‘This correctly adds the value 1 to the contents of the Accumulator

0001 **00000001** ‘This does not correctly output the contents of the Accumulator

### Task 3

0001 ‘Correct – allows input into the Accumulator

1100 00010000 ‘Correctly stores the value in location 16

0001 ‘Correct – allows input into the Accumulator

**0011** 00010001 ‘Incorrect operation code

0001 ‘Correct – allows input into the Accumulator

1011 00010001 ‘Adds memory location 17 to the Accumulator

**1010** 00010000 ‘Adds the value 16 to the Accumulator

0010 ‘Outputs the result

### Task 4

0001 ‘Value entered into the Accumulator

1100 00010000 ‘Value is stored in memory address 16

0001 ‘Another value is entered into the Accumulator

1011 00010000 ‘The value in memory address 16 is added to the accumulator

1100 00010001 ‘The result is stored in address 17

0001 ‘A new value is entered into the Accumulator

1101 00010001 ‘The contents of location 17 is subtracted from the accumulator

0010 ‘The result of the accumulator is output.

We’d like to know your view on the resources we produce. By clicking on ‘[Like](mailto:resources.feedback@ocr.org.uk?subject=I%20liked%20this%20GCSE%20(9-1)%20Computer%20Science%20Learner%20Activity%202)’ or ‘[Dislike](mailto:resources.feedback@ocr.org.uk?subject=I%20disliked%20this%20GCSE%20(9-1)%20Computer%20Science%20Learner%20Activity%202)’ you can help us to ensure that our resources work for you. When the email template pops up please add additional comments if you wish and then just click ‘Send’. Thank you.

If you do not currently offer this OCR qualification but would like to do so, please complete the Expression of Interest Form which can be found here: [www.ocr.org.uk/expression-of-interest](http://www.ocr.org.uk/expression-of-interest)

**OCR Resources**: *the small print*OCR’s resources are provided to support the teaching of OCR specifications, but in no way constitute an endorsed teaching method that is required by the Board, and the decision to use them lies with the individual teacher. Whilst every effort is made to ensure the accuracy of the content, OCR cannot be held responsible for any errors or omissions within these resources.   
© OCR 2016 - This resource may be freely copied and distributed, as long as the OCR logo and this message remain intact and OCR is acknowledged as the originator of this work.

OCR acknowledges the use of the following content: n/a

Please get in touch if you want to discuss the accessibility of resources we offer to support delivery of our qualifications: [resources.feedback@ocr.org.uk](mailto:resources.feedback@ocr.org.uk)