# End of Unit Quiz – Unit 2.2 Programming Techniques

**a.** Compare the use of **variables** and **constants** in a computer program, giving one similarity and one difference.

|  |
| --- |
|  |

**bi.** A programmer creates the following code:

|  |  |  |
| --- | --- | --- |
|  | 01 input  02 x = y MOD  03 if x = 0 then  04 print “True”  05 end if |  |

How is the = sign on line 02 and line 03 used differently?

|  |
| --- |
|  |

**bii.** What is one input that would case the program to output **True** and explain why this is the case?

|  |
| --- |
|  |

**biii.** What are **two** basic programming constructs that have been used in the code above?

|  |
| --- |
|  |

**biv.** What is the name of **one** programming construct that has **not** been used in the code above and give an example of how this construct could be used?

|  |
| --- |
|  |

1. Create an algorithm that will allow the user to enter a word and then count how many times the letter A appears in that word. Both upper case (“A”) and lower case (“a”) letters must be counted. The algorithm should repeat until a word is entered that has 3 or more letter As.

|  |
| --- |
|  |

* 1. The following algorithm prints out the times table of the number entered using a **count controlled** loop.

|  |  |  |
| --- | --- | --- |
|  | 01 input b  02 for x = 1 to 100  03 print b \* x  04 next |  |

Rewrite the algorithm to produce the same result using a **condition controlled** loop.

|  |
| --- |
|  |

* 1. Write an algorithm that will ask the user for their age (in years) and then print the message “happy birthday” that many times.

|  |
| --- |
|  |

* 1. Complete the data type column on the below table to show the **most appropriate** data type for each:

|  |  |  |
| --- | --- | --- |
| **Data recorded** | **Example data** | **Data type** |
| Number of goals scored | 2 |  |
| Training venue | Bycars Park |  |
| Session completed (True / False) | True |  |
| Best sprint time (seconds) | 12.7 |  |

* 1. What is meant by the term **casting** in relation to data types?

|  |
| --- |
|  |

* 1. The data from part (a) is stored in an array called trainingdata*.* The training sessions are stored in a text file called allsessions.txt

Complete the algorithm below to add the new trainingdata to the text file.

**trainingdata = [2, “Bycars Park”, True, 12.7]**

|  |
| --- |
|  |

* 1. Using the trainingdata array from the previous question, give the pseudocode that a programmer would use to output just the training venue details(“Bycars Park”) from this array. You may assume that the array is zero-indexed.

|  |
| --- |
|  |

* 1. How could a 2 dimensional array be used to allow a programmer to hold details of multiple training sessions?

|  |
| --- |
|  |

* 1. A database table called *songs* is used to store details of music that is played on an Internet radio station.

The *songs* table is shown below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MusicID** | **BandName** | **SongTitle** | **Length** | **Fade** |
| 0001 | Penguin Steak | Along The Way | 3.27 | False |
| 0002 | Faustus | Round At Jessica’s | 2.55 | False |
| 0003 | Scholar Green | The Mule | 3.12 | True |
| 0004 | Penguin Steak | Insomnia | 3.06 | False |
| 0005 | Faustus | The Last Train Home | 4.19 | False |
| 0006 | Elvis Fontenot | Dear Love | 4.07 | True |

What is meant by the term **database** **record**?

|  |
| --- |
|  |

* 1. Write SQL statements to display the following data from the *songs* table:

1. Show the SongTitle and Length for all songs by the band Penguin Steak

|  |
| --- |
|  |

1. Show the SongTitle for all songs that are over 3 minutes in length.

|  |
| --- |
|  |

**a.** Procedures and functions are both examples of subroutines. What are two advantages of producing modular code using subroutines?

|  |
| --- |
|  |

1. What **two** ways in which procedures differ from functions?

|  |
| --- |
|  |

**ci.** A password must have at least 8 characters to be valid.

Using pseudocode, create a function which will accept a password string as a parameter passed into the function, returning True if the password is a valid length or False if it is not valid.

|  |
| --- |
|  |

**cii.** Use the function defined in part (i) above to check whether “HELLO123” is a valid password, printing out True or False as appropriate. **You must use the function defined in part (i).**

|  |
| --- |
|  |

**Answers**

**a.** Compare the use of **variables** and **constants** in a computer program, giving one similarity and one difference.

|  |
| --- |
| Similarity :  Both refer to a memory location  Both are given an identifier  Both are used to store data whilst the program is running  Difference:  Variable’s value can be changed / Constant’s value cannot be changed whilst the program is running |

**bi.** A programmer creates the following code:

|  |  |  |
| --- | --- | --- |
|  | 01 input  02 x = y MOD  03 if x = 0 then  04 print “True”  05 end if |  |

How is the = sign on line 02 and line 03 used differently?

|  |
| --- |
| Line 02 = is an assignment operator / assigns a value to x  Line 03 = is a comparison operator / compare the value of x with 0.  Do not award for simply rewording the line (eg “on line 03, if x is equal to zero then it…” |

**bii.** What is one input that would case the program to output **True** and explain why this is the case?

|  |
| --- |
| Any integer value that is divisible by 5 (so ends in 5 or 0). Eg 35, 90, 5.MOD produces the remainder when y is divided by 5……this has to be zero to allow the output to be True. |

**biii.** What are **two** basic programming constructs that have been used in the code above?

|  |
| --- |
| Sequence, Selection. |

**biv.** What is the name of **one** programming construct that has **not** been used in the code above and give an example of how this construct could be used?

|  |
| --- |
| **1biv.** Iteration. Any reasonable example (eg use of a FOR / WHILE loop, any reference to looping or repeating code). |

1. Create an algorithm that will allow the user to enter a word and then count how many times the letter A appears in that word. Both upper case (“A”) and lower case (“a”) letters must be counted. The algorithm should repeat until a word is entered that has 3 or more letter As.

|  |
| --- |
| **Example:**  word = input(“Enter a word”) count = 0  while count < 3  for x = 1 to word.length  if x.upper = “A” then  count = count + 1  endif  next x  endwhile  Inputting word from the user  Initialising a counter variable to 0 at the start  Use of a count controlled loop…  ….to loop around until the correct number of times (until 3 As are entered  Dealing with upper and lower case (eg by converting all to upper case)  Checking each individual letter for an A…  …and adding 1 to the counter if an A is found  Alternative solution (especially for students familiar with Python) would be to use .split or treat the string as an array of characters rather than using the FOR loop. This should be credited under bullet point 6 if done correctly. |

* 1. The following algorithm prints out the times table of the number entered using a **count controlled** loop.

|  |  |  |
| --- | --- | --- |
|  | 01 input b  02 for x = 1 to 100  03 print b \* x  04 next |  |

Rewrite the algorithm to produce the same result using a **condition controlled** loop.

|  |
| --- |
| **Example:**  input b  x = 1  while x <=10  print b \* x  x = x + 1  endwhile  Repeating the input b line as in line 01  Initialising a variable to use a counter (x used here)  Correct use of condition controlled loop (eg WHILE), with the counter being < or <= to 9 or 10 (depending on use – either could be correct)  Repeating the print b \* x line as in line 03.  Manually incrementing the counter variable  Ending the loop correctly (eg ENDWHILE) |

* 1. Write an algorithm that will ask the user for their age (in years) and then print the message “happy birthday” that many times.

|  |
| --- |
| **Example:**  input age  for x = 1 to age  print “happy birthday”  next x  Inputting the age from the user  Repeating this many times  …Printing out “happy birthday” that many times |

* 1. Complete the data type column on the below table to show the **most appropriate** data type for each:

|  |  |  |
| --- | --- | --- |
| **Data recorded** | **Example data** | **Data type** |
| Number of goals scored | 2 | Integer |
| Training venue | Bycars Park | String |
| Session completed (True / False) | True | Boolean |
| Best sprint time (seconds) | 12.7 | Real / Float |

* 1. What is meant by the term **casting** in relation to data types?

|  |
| --- |
| Changing how a variable’s data type is interpreted / a temporary conversion of data type.  Suitable example – eg str(123) will treat 123 as a string, not an integer / so a string and an integer can be concatenated. |

* 1. The data from part (a) is stored in an array called trainingdata*.* The training sessions are stored in a text file called allsessions.txt

Complete the algorithm below to add the new trainingdata to the text file.

**trainingdata = [2, “Bycars Park”, True, 12.7]**

|  |
| --- |
| **Example:**  **Trainingdata = [2, “Bycars Park”, True, 12.7] (already given)**  open allsessions.txt…  …for append  write trainingdata  close file |

* 1. Using the trainingdata array from the previous question, give the pseudocode that a programmer would use to output just the training venue details(“Bycars Park”) from this array. You may assume that the array is zero-indexed.

|  |
| --- |
| print trainingdata[1] |

* 1. How could a 2 dimensional array be used to allow a programmer to hold details of multiple training sessions?

|  |
| --- |
| 2D array has rows and columns / treated like a table / accessed via two indexes.  First index / rows / columns can hold data for one training session.  Second index / subsequent rows / columns can hold other training sessions.  Suitable example. |

* 1. A database table called *songs* is used to store details of music that is played on an Internet radio station.

The *songs* table is shown below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MusicID** | **BandName** | **SongTitle** | **Length** | **Fade** |
| 0001 | Penguin Steak | Along The Way | 3.27 | False |
| 0002 | Faustus | Round At Jessica’s | 2.55 | False |
| 0003 | Scholar Green | The Mule | 3.12 | True |
| 0004 | Penguin Steak | Insomnia | 3.06 | False |
| 0005 | Faustus | The Last Train Home | 4.19 | False |
| 0006 | Elvis Fontenot | Dear Love | 4.07 | True |

What is meant by the term **database** **record**?

|  |
| --- |
| A collection of fields / data about one person / thing / entity.  Suitable example from the table (could be a whole record from the table or a new record that could be entered into the table). |

* 1. Write SQL statements to display the following data from the *songs* table:

1. Show the SongTitle and Length for all songs by the band Penguin Steak

|  |
| --- |
| SELECT SongTitle, Length  FROM songs  WHERE BandName = “Penguin Steak” |

1. Show the SongTitle for all songs that are over 3 minutes in length.

|  |
| --- |
| SELECT SongTitle  FROM songs  WHERE Length > 3 |

**a.** Procedures and functions are both examples of subroutines. What are two advantages of producing modular code using subroutines?

|  |
| --- |
| Can reuse code / can use pre-built or external subroutines. Easier to debug / maintain. Work can be split between programmers / programmers can concentrate on their areas of expertise. |

1. What **two** ways in which procedures differ from functions?

|  |
| --- |
| Procedures are called by their name / functions are called by assign their return value to something. Procedures do not return values / Functions always return a single value. |

**ci.** A password must have at least 8 characters to be valid.

Using pseudocode, create a function which will accept a password string as a parameter passed into the function, returning True if the password is a valid length or False if it is not valid.

|  |
| --- |
| **Example:**  function checkpassword(password)  if password.length >= 8 then  return True  else  return False  endfunction  Correct function definition with a single value passed in as a parameter  Check if the length is >= 8…  ….return True if it is  …return False if it is not.  Does not matter what the function is called or the parameter is called, but this must logically work. |

**cii.** Use the function defined in part (i) above to check whether “HELLO123” is a valid password, printing out True or False as appropriate. **You must use the function defined in part (i).**

|  |
| --- |
| print checkpassword(“HELLO123”) |

**OCR Resources**: *the small print*OCR’s resources are provided to support the delivery of OCR qualifications, 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 2017 - 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)

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%20the%20GCSE%20Computer%20Science%20Quiz%202.2%20Programming%20Techniques) or [‘Dislike’](mailto:resources.feedback@ocr.org.uk?subject=I%20disliked%20the%20GCSE%20Computer%20Science%20Quiz%202.2%20Programming%20Techniques) 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.

Whether you already offer OCR qualifications, are new to OCR, or are considering switching from your current provider/awarding organisation, you can request more information by completing 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)

Looking for a resource? There is now a quick and easy search tool to help find free resources for your qualification:   
[www.ocr.org.uk/i-want-to/find-resources/](http://www.ocr.org.uk/i-want-to/find-resources/)

This formative assessment resource has been produced as part of our free GCSE teaching and learning support package. All the GCSE teaching and learning resources, including delivery guides, topic exploration packs, lesson elements and more are available on the qualification webpages.

If you are looking for examination practice materials, you can find Sample Assessment Materials (SAMs) on the qualification webpage: [Computer Science (9-1)](http://www.ocr.org.uk/qualifications/gcse-computer-science-j276-from-2016/)