



Computing

GCSE

Computing

Exemplar Candidate Work

J275

Unit A453 Sample Material C3

Version 1

Disclaimer on use of Sample Material

Confidentiality

These tasks are taken from legacy Controlled Assessment tasks, undertaken and submitted by candidates. Where possible, we have removed all identifying information from these assessments. Should any data remain, you are requested to treat this confidentially and inform OCR as soon as possible highlighting the data concerned.

Use of URS Sheets and Sample Material

These tasks have all been moderated as part of the relevant exam series in which they were submitted and the marks submitted have all been allowed to stand. However, schools should bear in mind that this only indicates that the **overall assessment** of the Controlled Assessment is within tolerance and not necessarily each individual mark band. There may be instances where the mark scheme has been applied too generously, or similarly too harshly. This would have been identified in the reports to the centre – but will not be evident from URS alone. The spirit of the release of these samples is to give teachers better understanding of what High, Medium and Low graded coursework would feel like as an entity, rather than exact definitions of requirements for mark bands independently.

The provision of high graded work should **not infer** that this is the only, or best way of writing up a Controlled Assessment Task. Candidates are encouraged to map their personal journey through the tasks. Writing frames, or 'guides' for documentation are against the spirit of the coursework and constitute malpractice.

Each set of materials released contains a High, Middle and Low grade band. This should allow teachers to gain good understanding of the general standard of work quality required for each mark band, and as a whole – especially when comparing each set side by side.

Teachers are encouraged to seek further support when they feel clarification is needed in applying the mark scheme. We would also recommend regular CPD in respect of Controlled Assessment delivery and marking.

Accuracy

All work has, where possible, remained unaltered from the original submission. There may well be grammatical errors and poor layout in diagrams. This is to allow better matching of mark band criteria, where specific bullet points refer to quality of Spelling, Punctuation and Grammar, and also ease of navigation etc. Any significant changes are clearly marked. Some data that is perceived sensitive may be blocked out in black.

3476666671



GCSE Computing Controlled Assessment

Unit A453 Programming project Unit Recording Sheet

Please read the instructions printed on the other side of this form. One of these Unit Recording Sheets, suitably completed, should be attached to the assessed work of each candidate.

Unit	A453	Year
Centre Name	Centre Number	
Candidate Name	Candidate Number	

	Guidance		Teacher Comment	Mark
Use of programming techniques	There is an attempt to solve parts of the tasks using few of the techniques identified. 0 = no response or responses not worthy of credit [0 - 2]	There is an attempt at most parts of the tasks using several techniques. [3 - 4]	There is an attempt to solve all of the tasks using most of the techniques listed.	The candidate has made an attempt to solve most of the tasks. The candidate has used several techniques in his pseudo-code/actual code: loops, selection, variables, file handling 4
Efficient use of programming techniques	The techniques used produce partially working solutions to a small part of the problem. 0 = no response or responses not worthy of credit [0 - 4]	The techniques are used appropriately giving working solutions to most of the parts of the problem. Some sections of the solution are inefficiently coded. [5 - 8]	The techniques are used appropriately in all cases giving an efficient, working solution for all parts of the problem. [9 - 12]	The candidate has evidenced part of the recipe task completed. 4
				Max 6
				Max 12

URS666 Revised May 2014

A453/URS

Oxford Cambridge and RSA Examinations

<p>Design</p>	<p>There are comments on what the task involves and a limited outline describing the intended approach to some parts of the problem. There are brief comments on how this might be tested but with no mention of success criteria. 0 = no response or responses not worthy of credit</p> <p>[0 - 3]</p>	<p>There is a brief analysis of the tasks indicating what is required for each of the tasks. There is a set of basic algorithms outlining a solution to most parts of the problem. There is some discussion of how this is tested and how this compares to the identified outcomes in the tasks. There is discussion of the variables to be used and some general discussion of validation.</p> <p>[4 - 6]</p>	<p>There is a detailed analysis of what is required for these tasks justifying their approach to the solution. There will be a full set of detailed algorithms representing a solution to each part of the problem. There is detailed discussion of testing and success criteria. The variables and structures are identified together with any validation required</p> <p>[7 - 9]</p>	<p>The candidate has an analysis of the task which indicates what is required for each task. The candidate has flowcharts and pseudocodes for all tasks. The candidate has brief test tables for each task and expected outcomes. The candidate has briefly explained some of the possible variables in each task.</p> <p>5</p> <p>Max 9</p>
<p>Development</p>	<p>There is some evidence to show a solution to part of the problem with some evidence to show that it works. Code is presented with little or no annotation, the variable names not reflecting their purpose and with little organisation or structure. 0 = no response or responses not worthy of credit</p> <p>[0 - 3]</p>	<p>There is evidence to show how the solutions were developed. There is some evidence of testing during development showing that many parts of the solution work. The code is organised with sensible variable names and with some annotation indicating what sections of the code does.</p> <p>[4 - 6]</p>	<p>There is detailed evidence showing development of the solution with evidence of systematic testing during development to show that all parts work as required. The code is well organised with meaningful variable names and detailed annotation indicating the function of each section.</p> <p>[7 - 9]</p>	<p>Very little evidence of development and testing outcomes.</p> <p>1</p> <p>Max 9</p>

347666671

<p>Testing and evaluation</p>	<p>There is evidence to show that the system has been tested for function but the test plan is limited in scope with little structure. There is limited evidence to show how the result matches the original criteria. The evidence of written communication is limited with little or no use of specialist terms. Errors in spelling, punctuation and grammar may be intrusive. Information may be ambiguous or disorganised. 0 = no response or responses not worthy of credit</p> <p>[0 - 3]</p>	<p>There is a test plan covering many parts of the problem with some suggested test data. There is evidence that the system has been tested using this data. There is some evidence to show how the results of testing have been compared to the original criteria. There is a brief discussion of how successful or otherwise the solutions are. The quality of written communication is good using some specialist terms. There are few errors in spelling, grammar and punctuation. Information for the most part is presented in a structured format.</p> <p>[4 - 6]</p>	<p>The test plan covers all major success criteria for the original problem with evidence to show how each of these criteria have been met, or if they have not been met, how the issue might be resolved. There is a full evaluation of the final solution against the success criteria. A high level of written communication is obvious throughout the task and specialist terms/ technology with accurate use of spelling will have been used. Grammar and punctuation are used correctly and information is presented in a coherent and structured format.</p> <p>[7 - 9]</p>	<p>Limited testing.</p> <p>1</p> <p>Max 9</p>
<p>Total/45</p>				

Guidance on Completion of this Form

- 1 One sheet should be used for each candidate.
- 2 Please ensure that the appropriate boxes at the top of the form are completed.
- 3 Using the guidance identify the most appropriate mark range for the work and enter the mark awarded for each element in the mark column.
- 4 Add appropriate comments to assist the moderator in the 'Teacher Comment' column.
- 5 Add the marks for the strands together to give a total out of 45. Enter this total in the relevant box.

A453

Task 3

1: Requirements:

- The program should allow the user to choose a number.
- The program should allow the user to add, take away, multiply or divide.
- The program should only give an answer up to the value 999.
- The program output the correct answer.
- The program, after doing its first sum, should ask the user if they want to do another sum.

2: Design:

- The program should allow the user to press the preferred number.
- The program should allow the user to work out sums for any questions, making sure the value only goes up to 999.
- The program should display a picture of a calculator.
- The program should allow the user to only do addition and subtraction.
- The program should allow the user clear their work after they are finished with the first sum.

3: How to approach the task:

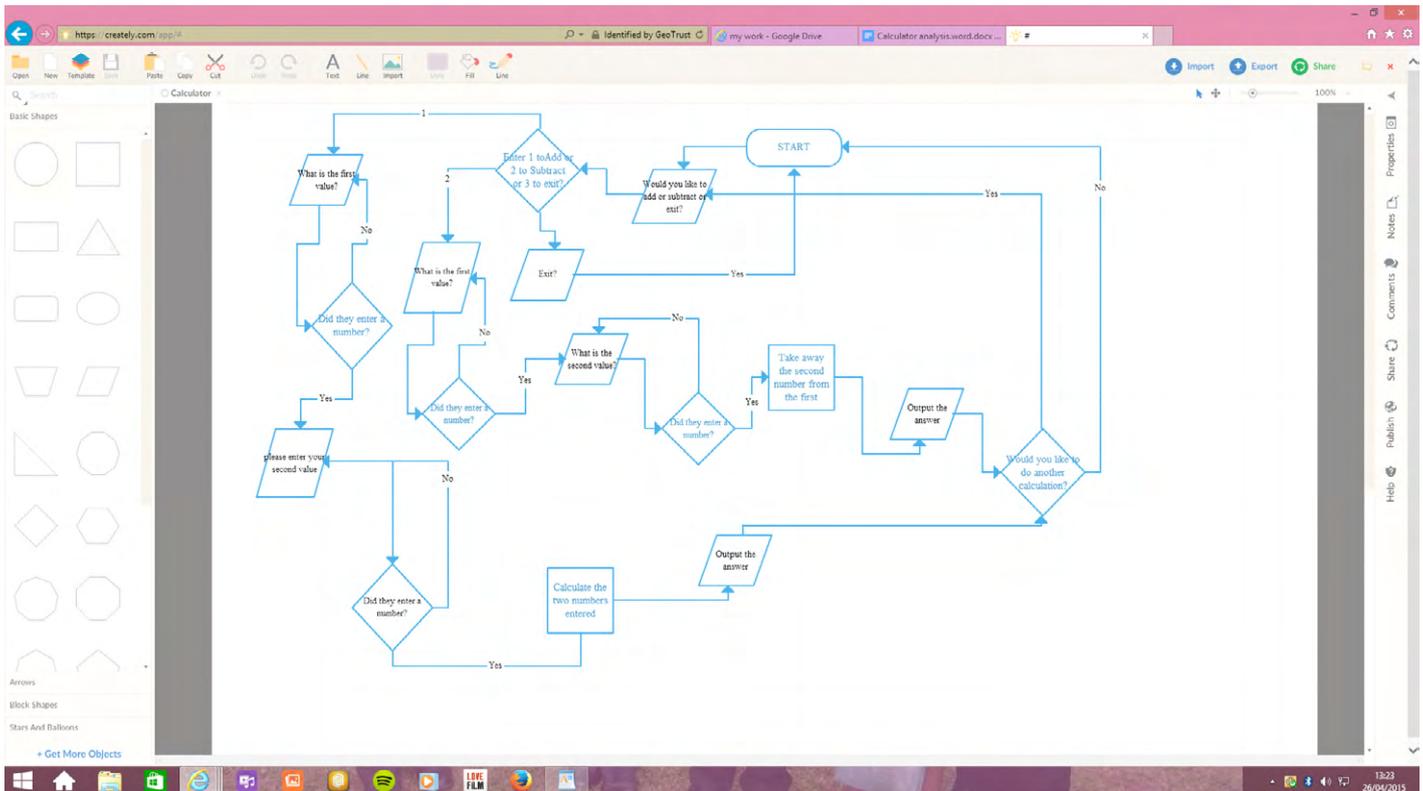
- The program should allow the user to press the preferred number.
- The program should allow the user to work out sums for any questions, making sure the value only goes up to 999.
- The program should display a picture of a calculator.
- The program should allow the user to only do addition and subtraction.
- The program should allow the user clear their work after they are finished with the first sum.

4: Variables:

- Because I used scratch there won't be any variables used, however, if I used python or any other programming language the variables would have been:
- Integer- Because it consists of whole numbers.
- String- To ask the user their number they are going to input.

5: Flow Chart:

This flowchart will help my coding be easier because it helps to keep track of what I have done and what I haven't done.

**6: Pseudocode:**

```

Input FirstNumber
Input Operation
Input SecondNumber
  IF Operation is plus THEN
Add FirstNumber and SecondNumber
Output Answer
  ELSE THEN
  IF Operation is minus THEN
Take away SecondNumber from FirstNumber
Output Answer
  ELSE THEN
  IF operation is times THEN
Times FirstNumber and SecondNumber together
Output Answer
  ELSE THEN
  IF operation is divide THEN
Divide FirstNumber and SecondNumber
Output Answer
  END ELSE
  END IF
END

```

7:

Test number	Success Criteria	Input	Expected Outcome
1	Creating the operations	The operations for each of the calculations for example plus will be '+' minus will be '-'	The program should allow the user to add their numbers or take it away from each other.
2	Drawing the numbers	Because I used scratch to produce my final product, I drew the numbers on the calculator.	The user can click on the numbers and choose what he/she wants to do with them.

8:

Test number	Date	Purpose	Input	Expected Outcome	Actual Outcome	Did it work? – if not how was it fixed
1	23/01/15	Creating the block for addition.	I choose scratch to produce my final product, which meant I had to think about the correct operation block for each operation.	When the user for example to add something, the first number and second number should both be added.	The user asked for 12 and 20 to be added and the answer came out as 32 which was right.	Yes
2	07/02/15	Creating the block for subtraction.	The subtraction block has to be done so the user can take away numbers as well.	When the user asks for the first number to be taken away from the second, the program should allow this to happen and give the right answer.	The user asked for 20 minus 11 and the program gave the answer 9.	Yes

10: Evaluation

My final project allows the user to add two numbers together, also allows the user to take away, times and also divide. My project also outputs the correct answer. I know this because I used the calculator to do many sums and it always gave the correct answer to the sums multiple times. The program also allows the user to enter the digits that they want and the correct operation that is wanted by them.

A453

Task 2 Recipe

1: Requirements:

Create a program that will store the ingredients for a recipe.

- The program should ask the user to input:
- The number of people the recipe will serve
- A list of ingredients: item, quantity and units for example *flour, 150, grams*
- The program should store the recipe name, number of people and the list of ingredients with their quantities and units.
- The user should be able to retrieve the recipe and have the ingredients recalculated for a different number of people.

The program should ask the user to input the number of people. The program should output:

- The recipe name
- The new number of people
- The revised quantities with units for this number of people.

2: Design:

The program should display a screen, which allows the user to retrieve a recipe or create a new recipe for them. Also it should help the user with finding out how many ingredients are needed for a specific recipe. If they are creating a new recipe the program should allow them to add a new recipe and allow them to add new ingredients to their chosen recipe and it should allow them to add in how much is needed of each.

3: Approaching the task:

I scanned the internet to check whether there are any templates on the task, however unfortunately I couldn't find any but I found ways in how to start off most of the programming. Some of the websites I used were:

<http://stackoverflow.com/questions/16146075/python-recipe-program>

This website helped me during the start and middle of my code, however I used the OCR computing book to finish off the code.

4: Variables:

The variables I used were:

- Integer: This was used for the amount of something, for e.g 'flour 150g'
- String: This will help the user to enter words and letters when they're appending to the list (adding more recipes to the list).


```

IF Userinput 'no'THEN
Output "Do you want to 1, enter a new recipe, 2, search for a new recipe, or 3, leave?"
END WHILE
END ELSE
END

```

7: Loops and statements I plan to use:

The loop I have used in my code are the while loop. The while loop will allow me repeat some of the sections of my code. If the condition is true the code within the block is executed until the condition becomes false. The statements I have used are If, ELSE and the ELIF statements. The IF statement is used to specify a block of code to be executed, if a specified condition is true. The ELSE statement is used to specify a block of code to be executed, if the same condition is false. The ELIF statement is used to specify a new condition to test, if the first condition is false.

Loops

- A **loop** is a repetition of all or part of the commands in a program.
- A loop often has a counter (a variable) and continues to repeat a specified number of times.
- A loop may also continue until a certain condition is met (e.g., until the end of a file or until a number reaches a set limit)

```

#!/usr/bin/env python
import time

```

```

FirstDecision=""
Time.sleep(1)

```

```

While FirstDecision != "3":
    FirstDecision=input("Do you wish to 1: enter a recipe, 2: search a recipe, or 3: exit? Write 1, 2 or 3")
    If FirstDecision=="1":
        RecipeData=input("Please enter your recipe name ")
        RecipeFile=open(RecipeData,"w")
        RecipeData=input("How many people is this recipe for? ")
        RecipeFile.write(RecipeData+"\n")
        IngredientAmount=input("How many ingredients are needed for this recipe? ")
        IngredientAmount=int(IngredientAmount)
        IngredientRange=range(1, IngredientAmount)
        For count in IngredientRange:
            RecipeIngredient="Enter the ingredient"
            AmountRequested=("How much is needed for this ingredient? ")
            RecipeFile.write(RecipeIngredient+"\n")
            RecipeFile.write(RequestedAmount+"\n")
            RecipeFile.close()

```

```

elif FirstDecision=="2":
    RecipeData=input("Please enter the recipe name ")
    RecipeFile=open(RecipeData,"\n")
    RecipeData=input("How many people is the recipe for?")
    IncrementFactor=RecipeData/RecipeFile.readline(0)
    CurrentLine=1
    While RecipeFile.readline() != "":
        Print(RecipeFile.readline(CurrentLine))
        CurrentLine=CurrentLine+1
        Print(RecipeFile.readline(CurrentLine)*IncrementFactor)
        CurrentLine=CurrentLine+1
    RecipeFile.close()
else:
    FirstDecision="3"

```

8: Test table

Test Number	Success Criteria	Input	Expected Outcome
1	Creating the IF statement	<pre>if FirstDecision=="1":</pre>	The IF statement is used to specify a block of code to be executed if the condition is true.
2	Creating the while loop	<pre>while FirstDecision != "3":</pre>	The while loop should allow the code to be executed repeatedly based on given data.
3	Creating the ELSE statement	<pre>else: FirstDecision="3"</pre>	The ELSE statement is used to specify a block of code to be executed, if the same condition is false.
4	Creating the ELIF statement	<pre>elif FirstDecision=="2": RecipeData=input("Please</pre>	The ELIF statement is used to specify a new condition to test, if the first condition is false.

Creating all these were easy however getting them to programme was the difficult bit. The next test table should explain how I overcame these difficulties.

9: Test table 2:

Test Number	Date	Purpose	Input	Expected Outcome	Actual Outcome	Did it work? – If not how was it fixed?
1	25/11/13	Creating a while loop	While firstdecision != "3":	The loop should allow me to repeat the data.	Error	It didn't work because I didn't put the 'F' and the 'D' is capital letter and the 'w' should have been a smaller letter. It should have said while FirstDecision != "3":.
2	2/12/13	Creating the IF statement	if FirstDecission=="1":	The statement should specify a block of code to be executed if the condition is true.	The program used the IF statement to do exactly explained before in the previous block.	Yes it worked

10: Evaluation:

My program allows me to search a recipe and it allows me to enter the correct amounts of ingredients and the amounts I need. It also lets me exit out of the code and it also allows me to enter a new recipe to the list of old recipes. The program also asks the user the amount of people the recipe will serve so it can calculate how much of each ingredient is needed. It also stores the name of each of the recipes and allows the user to append to the list. The code also asks for the name of the recipe, so the code can save it in a text file.

A453

Task 3 Hangman 2015

1: Requirements

Design, code test and evaluate it to let the user play hangman online.

- Contains six lives
- If they guess a letter right it should place it in the correct position
- If they get it wrong, they lose a life
- Record number of lives lost
- Allow end user to enter a word into the array
- Select words randomly from an array
- Allows player to guess a letter
- Record the number of lives left after the word has been guessed and uses these as a score.

2: Design

The program should display a screen, which allows the user to enter a random letter and guess the word. It should take away lives, if the user gets a letter wrong. Also it should not allow the user to enter numbers and it will say, "please enter letters."

3: Approaching the task

To make sure the program works I will research all the data that I will need and use it in a way that would be helpful to design my final hangman code. Also I would use text wr

- **Methods**
- The `append()` list method
- The `lower()` and `upper()` string methods
- The `reverse()` list method
- The `split()` string method
- The `range()` function
- The `list()` function
- `for` loops
- `elif` statements
- The `startswith()` and `endswith()` string methods.
- The dictionary data type.
- key-value pairs
- The `keys()` and `values()` dictionary methods
- Multiple variable assignment, such as `a, b, c = [1, 2, 3]`

<http://inventwithpython.com/chapter9.html>

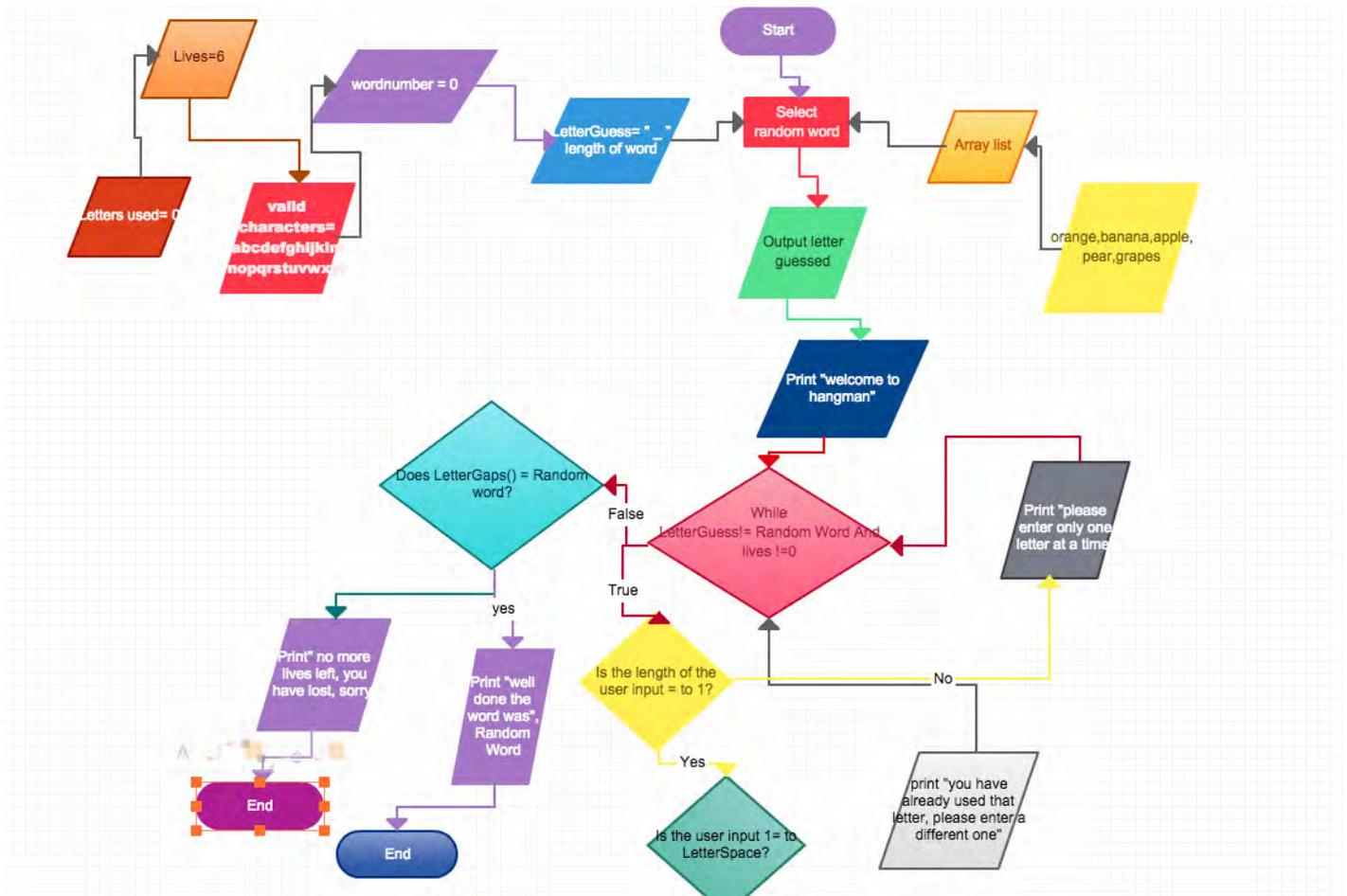
4: Variables:

The variables will be:

- String so it will help the user to enter words and also letters.
- List so it helps to keep track of the words in the array.
- Boolean which helps to check whether they are right or wrong, then they guess the word.

5: Flow Chart:

This flowchart will help my coding be easier because it helps to keep track of what I have done and what I haven't done.

**6: Pseudocode:**

```

Import random
List = (orange, banana, apple, pear, grapes)
Word=random.choice (WordList)
Lives = 6
WordGaps="_"* length (Word)
LettersUsed=0
LetterSpace=(
////////////////////////////////////
////////////////////////////////////
)
OUTPUT "welcome to hangman"
OUTPUT WordGaps
OUTPUT "Guess a letter"
WHILE WordGaps!=Word
LettersUsed=LettersUsed+1
LetterSpace (LettersUsed)=INPUT
OUTPUT "letters used", LetterSpace
  IF INPUT NOT in word THEN
    LivesLeft = Lives-1
    OUTPUT "that letter is not in the word, guess again"
  ELSE THEN
    FROM i = 1 to LENGTH (Word)
      IF INPUT = letter
        WordGaps (i) = INPUT
        OUTPUT WordGaps

```

```

    ENDIF
    OUTPUT "Correct, enter your next guess"
    END IF
END WHILE
OUTPUT "congrats, you have won"

```

7: The loops I will use are:

- While loop these are used the same as the for loop, to help repeat sections of the code for example:
- While (LetterGuessed) != secretword and Lives>0:
- If statement used as a statement to process an instruction for example:
- If (LetterGuessed)! =1:
- Else statement- this is used as an opposite statement to an if statement for example:
- Else: Print ("you have got it wrong, guess again")

8: Test Table:

Test Number	Success Criteria	Input	Expected Outcome
1	Creating an array	<code>list=["apple", "orange", "banana", "pear", "grapes"]</code>	The code should allow the user to enter letters equals to the amount of letters there is in the specific word.
2	Creating the while loop	<code>while (LetterGuessed) != secretword and Lives>0:</code>	The while loop should allow the code to be executed repeatedly based on given data.
3	Creating the IF statement	If LetterGuessed in secretword:	This is used to process an instruction. In this case it is telling the computer if the letter inputted by the user is in the secretword then carry out the next instruction.
4	Creating the ELSE statement.	Else: Print ("you have got it wrong, guess again")	The ELSE statement is the opposite of the IF statement. Also it specifies a block of code to be executed, if the same condition is false.

9: New test table:

Test Number	Date	Purpose	Input	Expected Outcome	Actual Outcome	Did it work? – If not how was it fixed?
1	03/10/14	The while loop	While (LetterGuessed) != secretword and lives>0	It will repeat the code based on the given data.	The code kept repeating until the user lost a life and eventually died.	Yes
2	25/11/13	The array	List=["apple","orange","grapes","pear","banana"]	The list will contain the words the user tries to guess at the start of the code	When the user got the word right, the word was the word in the list (array)	Yes

10: Evaluation:

The program allows the user to guess a letter and the words will always be the ones in the array. The program also tells the user if he/she has guessed a letter, it will point that out and correct them. Also if they have used a letter twice the program also will tell them this. It also provides the user with six lives and if they guess a correct letter, the program places it in the correct place. It also will take away a life from the six lives the user has if they have got a letter wrong. It will show how many lives the user has left before he/she dies.



We'd like to know your view on the resources we produce. By clicking on the 'Like' or 'Dislike' button 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

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. We update our resources on a regular basis, so please check the OCR website to ensure you have the most up to date version.

© OCR 2015 – 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:

Square down and Square up: alexwhite/Shutterstock.com

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

We will inform centres about any changes to the specification. We will also publish changes on our website. The latest version of our specification will always be the one on our website (www.ocr.org.uk) and this may differ from printed versions.

Copyright © OCR 2015. All rights reserved.

Copyright

OCR retains the copyright on all its publications, including the specifications. However, registered centres for OCR are permitted to copy material from this specification booklet for their own internal use.

ocr.org.uk/alevelreform

OCR customer contact centre

General qualifications

Telephone 01223 553998

Facsimile 01223 552627

Email general.qualifications@ocr.org.uk

OCR is part of Cambridge Assessment, a department of the University of Cambridge. For staff training purposes and as part of our quality assurance programme your call may be recorded or monitored. © OCR 2015 Oxford Cambridge and RSA Examinations is a Company Limited by Guarantee. Registered in England.

Registered office 1 Hills Road, Cambridge CB1 2EU. Registered company number 3484466. OCR is an exempt charity.

