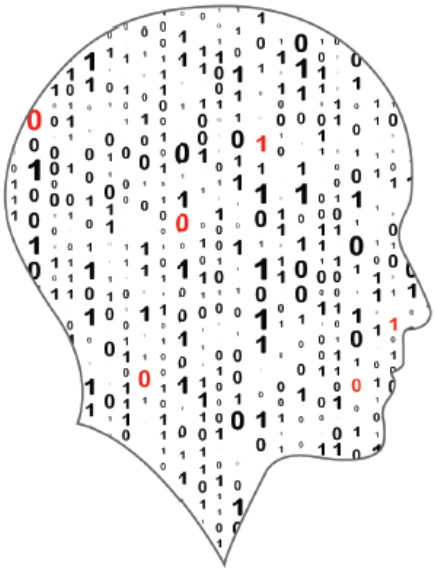
LC CS Python

Student Exercise Book



LEAVING CERTIFICATE
COMPUTER SCIENCE

Section 5

Programming Logic 2

While and *For* loops

Name: _____

Task 1

Modify the previous version of the Guess Game to the code below. Run the code and test it fully. Log any questions you still have in relation to this code.

```
1.  # Simple while loop
2.
3.  # Initialise a loop counter
4.  counter = 1
5.
6.  # Loop 10 times
7.  while counter <= 10:
8.      print("Hello World")  # Display a message
9.      counter = counter + 1 # Increment the counter
10.
11. # This line is only executed once
12. print("Goodbye")
```

Simple `while` loop demo.

Task 2

Complete the 'trace diagram' shown below.

Iteration #	counter	Program Output	counter <= 10
0	1		True
1	2	Hello World	True
2	3		
3			

Task 3

Modify the previous version of the Guess Game to the code below. Run the code and test it fully. Log any questions you still have in relation to this code.

```
1.  # Guess Game - 3 guesses
2.  import random
3.
4.  number = random.randint(1, 10)
5.  print(number) # have a sneak peek!
6.
7.  # Initialise a loop counter
8.  counter = 0
9.
10. # Loop 3 times
11. while counter < 3:
12.
13.     guess = int(input("Enter a number between 1 and 10: "))
14.     if guess == number:
15.         print("Correct")
16.         break # exit the loop immediately!
17.     elif guess < number:
18.         print("Too low")
19.     else:
20.         print("Too high")
21.
22.     counter = counter + 1
23.
24. print("Goodbye")
```

Task 4

Complete the 'trace diagram' shown below for Guessing Game v4

The diagram has been complete up to, but not including, the point when the user is about to enter a guess for the first time (i.e. the first execution of line 13).

The computer has generated a random number of 5 which has been recorded as `number`. The value has been displayed and `counter` has been initialised to zero. The condition `counter < 3` has been evaluated to `False` and this has also been recorded. You take over from this point.

Proceed by making up a value (i.e. guessing a number) and recording it in the first box underneath `guess`. Now trace the execution of line 14. This requires you (instead of Python) to evaluate the condition `guess == number`. Record your answer in the second black box.

Continue in this manner until the program has run to completion.

Iteration #	number	counter	counter < 3	guess	guess == number	Output
0	5	0	True			5

Experiment! Make some changes to the code.
Try achieving the same logic using different conditions e.g. ***`counter <= 3`*** ***or*** ***`counter < 4`*****. How would these conditions affect the initial value of** ***`counter`****?***

Task 5

Modify the previous version of the Guess Game to the code below. Run the code and test it fully. Log any questions you still have in relation to this code.

```
1.  # Guess Game v5 - while not found
2.  import random
3.
4.  number = random.randint(1, 10)
5.  print(number) # have a sneak peek!
6.
7.  correct = False # initialise the loop guard variable
8.
9.  # Loop until the variable correct becomes True
10. while not correct:
11.
12.     guess = int(input("Enter a number between 1 and 10: "))
13.     if guess == number:
14.         print("Correct")
15.         correct = True # this will cause the loop to exit
16.     elif guess < number:
17.         print("Too low")
18.     else:
19.         print("Too high")
20.
21. print("Goodbye")
```

Task 6

Complete the 'trace diagram' shown below for Guessing Game v5

The diagram has been complete up to but not including the first execution of line 12 i.e. the user is about to enter a guess for the first time.

The computer has generated a random number of 8 which has been recorded as `number`. The value has been displayed and the Boolean variable `correct` has been initialised to `False`. The condition `not correct` has been evaluated to `True` and this has also been recorded. You take over from this point.

Proceed by making up a value (i.e. guessing a number) and recording it in the first box underneath `guess`. Now trace the execution of line 13. This requires you (instead of Python) to evaluate the condition `guess == number`. Record your answer in the second black box.

Continue in this manner until the program has run to completion.

Iteration #	number	correct	not correct	guess	guess == number	Output
0	8	False	True			8

Take a moment to compare the loop guards used in versions 4 and 5. In what ways are they similar? How do they differ?

Task 7

Modify the previous version of the Guess Game to the code below. Run the code and test it fully. Log any questions you still have in relation to this code.

```
1.  # Guess Game v6 - while not correct - ask, go again?
2.  import random
3.
4.  number = random.randint(1, 10)
5.  print(number) # have a sneak peek!
6.
7.  # Initialise the loop guard variable
8.  keepGoing  = True
9.
10. # Loop as long as keepGoing is True
11. while keepGoing:
12.
13.     guess = int(input("Enter a number between 1 and 10: "))
14.
15.     if guess == number:
16.         print("Correct")
17.         goAgain = input("Play again? (Y/N): ")
18.         if goAgain == "N":
19.             keepGoing = False
20.         else:
21.             # Generate a new number
22.             number = random.randint(1, 10)
23.             print(number) # why not?
24.
25.     elif guess < number:
26.         print("Too low")
27.
28.     else:
29.         print("Too high")
30.
31. print("Goodbye")
```

Guessing Game v6

Task 8

Modify the previous version of the Guess Game to the code below. Run the code and test it fully. Log any questions you still have in relation to this code.

```
1.  # Guess Game v7 - while - go again? - data validation
2.  import random
3.
4.  number = random.randint(1, 10)
5.
6.  # Initialise the loop guard variable
7.  keepGoing  = True
8.
9.  # Loop as long as keepGoing is True
10. while keepGoing:
11.
12.     guess = input("Enter a number between 1 and 10: ")
13.     # Validate. Make sure the value is a number
14.     while not guess.isdigit():
15.        guess = input("Enter a number between 1 and 10: ")
16.
17.     # Conver the string to an integer
18.     guess = int(guess)
19.
20.     if guess == number:
21.         print("Correct")
22.         goAgain = input("Play again? (Y/N): ")
23.         if goAgain.upper() == "N":
24.             keepGoing = False
25.         else:
26.             # Generate a new number
27.             number = random.randint(1, 10)
28.
29.     elif guess < number:
30.         print("Too low")
31.
32.     else:
33.         print("Too high")
34.
35. print("Goodbye")
```

Guessing Game v7

Task 9

Experiment!
Try making the following changes to see what happens.

- Version 4 change the initial value of `counter` to 10
- Version 5 change the initial value of `correct` to be `True`
- Version 6 change the initial value of `keepGoing` to be `False`
- Version 7 remove (comment out line line 18