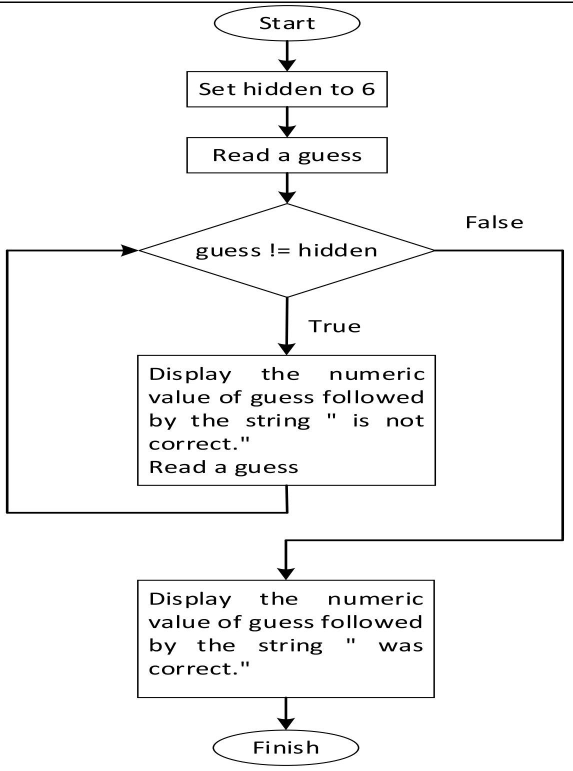
Week 4: While loops, for loops, nested loops

- Exercises 1 to 3: *while loops.*
- Exercises 4 to 7: *for loops*
- Exercise 8: *nested loops*
- Exercise 9 & 10: Additional/Challenge exercises

Exercise 1 - Guess the Number

Part A

Translate the following specification and flow chart into a Python program and test your solution works correctly for a range of integers.

Specification	**Flow Chart**
• Initialise variable *hidden* with a value of 6. • Ask user to enter a guess (number between 1 and 20). • Loop while the guess is not the hidden number. • Within the loop: ○ State guess is not correct, ○ ask user for next guess (number between 1 and 20). • If the guess is correct the loop will no longer run. • Let the user know their guess was correct	 Start Set hidden to 6 Read a guess guess != hidden False True Display the numeric value of guess followed by the string " is not correct." Read a guess Display the numeric value of guess followed by the string " was correct." Finish

Part B

- Amend the previous program so that the hidden number (between 1 and 20) is a random number generated by the program (check last week's tutorial for how to generate a random number).
- Hint: Remember to use *import random* at the top of your code.
- Hint: While testing your program it is useful to print the random number generated so you can test the program is working as specified.

Part C

- Amend the program to let the user know if the guess is **too low or too high**. Use an *if, else* statement.

Exercise 2

- Write a program that will allow a user to enter multiple scores until -9 is entered.
- Keep a running total of the scores entered and a count of the number of scores entered.
- When -9 is entered, leave the loop and print the average of the scores entered.
- **Most of the solution is shown below - just add the while loop where indicated**.

```
total = 0    # sum of scores
count = 0    # number of scores entered

# get one score & convert string to integer
score = int(input("Enter score, (Enter -9 to end): ") )

# Add while loop here. Loop while score is not -9
# Add score to total
# Keep a count of scores
# Get next score input

# print average of scores entered
average = float( total ) / count
print("Class average is", average)
```

- **Extra feature:**
  - What happens if you run the program and the first input is -9?
  - Add an if/else to ensure that at least one score has been entered before calculating the average.

Exercise 3 - Revision of exception handling with a while loop

- Most of the solution is shown. Just add the missing statement where indicated.
- Test the following example with different input (float, string, and integer).

```
while True:
     n = input("Please enter an integer: ")
     try:
         n = int(n)
         #  add missing statement here (break or continue)
     except ValueError:
         print("Requires a valid integer! Please try again.")

print("You successfully entered an integer.")
```

For loops (Exercises 4 to 7)

Exercise 4 - Write a program that uses a *for* loop to read in 5 numbers and output the total.

Exercise 5 - Write a *for* loop that will output **all odd numbers** between 0 and 20.

Exercise 6 - Displaying stars - Write a program which asks the user how many stars (*) are required and then make it display that number of stars across the screen.

Exercise 7 - Rolling two dice & counting doubles generated - Write a program that simulates the rolling of two six-sided dice and then counts the number of times that the dice generate a double (e.g. both dice show same value). Roll the dice 100 times and then print the number of doubles. Output: 'Out of 100 you rolled 20 doubles'.

Exercise 8 - Nested for loop (loop within a loop)

A) Try the following and check you understand the output of the nested loop code.

```
for number in range(3) :
      print("----------------------------------------")
      print("Outer loop iteration " + str(number))
      # Inner loop
      for another_number in range(4):
          print("****************************")
          print("In inner loop iteration " + str(another_number))
```

B) Try the following nested *for loop*.

```
for x in range(1,4):  # print 3 rows
    for y in range(1,4):  # 3 asterisks a row
        print('*', end='')
    print()
```

C) Then amend the program so that it prints this instead:

```
*
**
***
```

Additional/Challenge Exercises

Exercise 9 - Menu

Write a program that allows the user to select from four different menu options on a continuous loop. Control the loop by a Boolean (while True:)

- Choose menu options using numbers 1, 2, 3 and 4 and use option 4 to quit the program.
- Include a default for when an unrecognised menu option number is input by the user.
- Include exception handling for non-integer inputs.

Use the following input data to test your solution.

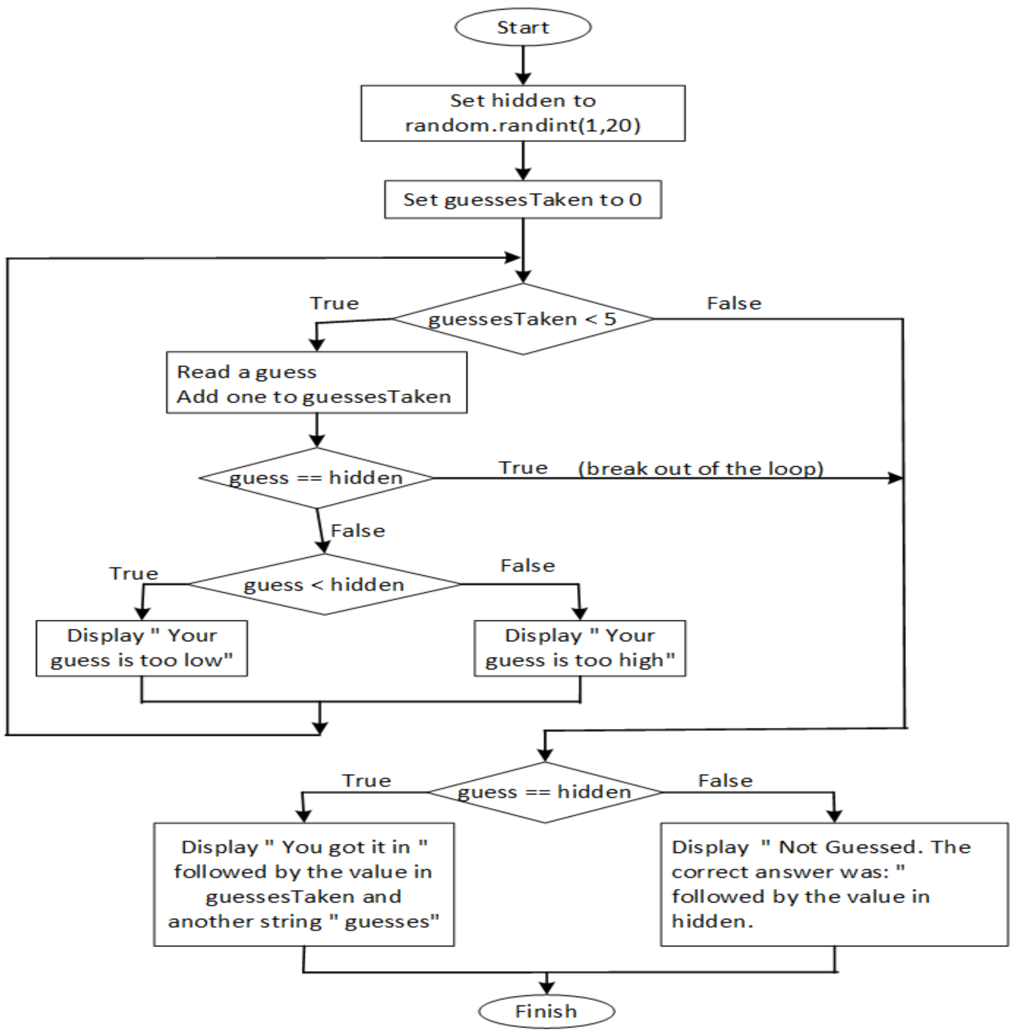
Input	Expected Result	Pass/Fail
1	Displays '1 selected' and program loops	
2	Displays '2 selected' and program loops	
3	Displays '3 selected' and program loops	
4	Displays 'Quit selected' and program ends	
5	Displays 'Option not recognised' and program loops	
A	Enter integer	

Exercise 10 - Guess the number (5 Guesses)

Create a new solution for the **guess the number** game. This solution will only allow a maximum of five guesses.

- The program picks a random number between 1 and 20.
- The program allows the user a maximum of five attempts to guess the number.
- For each incorrect guess, the program will let the user know if the correct answer is higher or lower than their guess.
- If the user doesn't guess the number in 5 attempts, end the loop.
- If the user guesses the number within 5 attempts, the program will **break** from loop.
- When the user has run out of guesses or guessed the correct number display an appropriate message. (Hint: check after the loop to decide which message):
    - let the user know they are correct and how many guesses they took, or
    - let the user know they did not guess the number and what the hidden number was.

Example Flow Chart:


Start
Set hidden to random.randint(1,20)
Set guessesTaken to 0
guessesTaken < 5
True
False
Read a guess
Add one to guessesTaken
guess == hidden
True (break out of the loop)
False
guess < hidden
True
False
Display " Your guess is too low"
Display " Your guess is too high"
guess == hidden
True
False
Display " You got it in " followed by the value in guessesTaken and another string " guesses"
Display " Not Guessed. The correct answer was: " followed by the value in hidden.
Finish