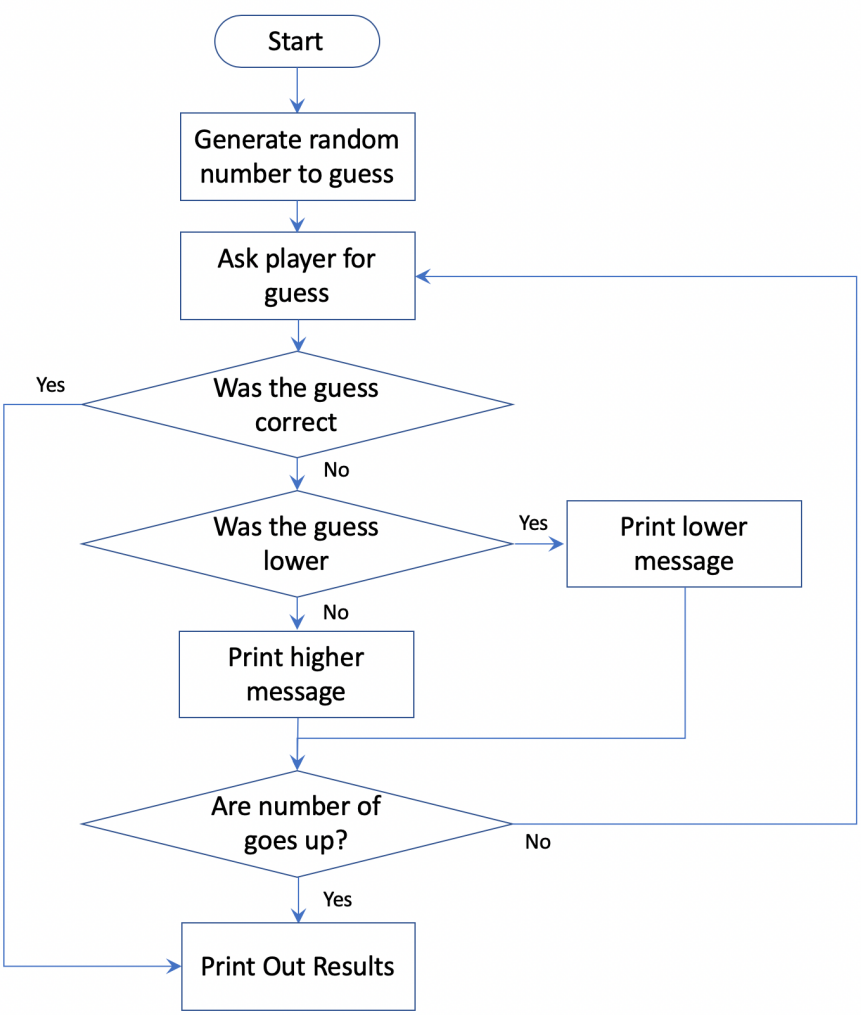
Lab: Flow of Control and iteration

In this lab you will create a simple number guessing game program, handling user input, using the if statement as well as using looping constructs.

The basic flow of the game is presented below in the form of a simple flow chart



Essentially the program logic is

- The program randomly selects a number between 1 and 10.
- It will then ask the player to enter their guess.
- It will then check to see if that number is the same as the one the computer randomly generated; if it is then the player has won.
- If the player's guess is not the same, then it will check to see if the number is higher or lower than the guess and tell the player.
- The player will have 4 goes to guess the number correctly; if they don't guess the number within this number of attempts, then they will be informed that they have lost the game and will be told what the actual number was.

Sample Output

A sample run of the program is shown here:

```
Welcome to the number guess game
Please guess a number between 1 and 10: 5
Sorry wrong number
Your guess was higher than the number
Please guess again: 3
Sorry wrong number
Your guess was lower than the number
Please guess again: 4
Well done you won!
You took 3 goes to complete the game
Game Over
```

Hints

Initialising Variables

In Python it is not necessary to declare a variable before you assign to it; however, it is necessary to give it an initial value before you reference it. Here *reference it* refers to obtaining the value held by a variable. For example

```
count = count + 1
```

what this says is obtain the value held by `count`, add 1 to it and then store the new value back into `count`.

If count does not have a value before you try to do this then you are trying to get hold of nothing and add 1 to it; which you can't do and thus an error will be generated in your program, such as:

```
NameError: name 'count_number_of_tries' is not defined
```

This is also true for

```
count += 1
```

Remember this is just a shorthand form of `count = count + 1` and thus still relies on `count` having a value before this statement.

This is why we needed to initialise the `count_number_of_tries` variable before we used it.

Generate the Random Number

You will need to use a random number generator.

To access the `random` module in Python you need to *import* it; this makes the module visible in the rest of the Python file (in our case to our program). This is done using

```
import random

number_to_guess = random.randint(1,10)
```

Once we have imported it, we can use the functions within this module, such as `randint`. This function returns a random integer between the first and second values parameters. In the above example it means that the random number generated will be between 1 and 10 inclusive.

The variable `number_to_guess` will now hold an integer which the player of the game must guess.

Obtain an input from the user

We now need to obtain input from the user representing their guess. We have already seen how to do this in previous chapters; we can use the `input()` function which returns a string and then the integer function that will convert that string into an integer (we will ignore error checking to make sure they have typed in a number at this point). We can therefore write:

```
guess = int(input('Please guess a number between 1 and 10: '))
```

This will store the number they guessed into the variable `guess`.

Extension Points

If you have time you can have a go at adding additional features to the game:

- Provide a cheat mode, for example if the user enters –1 print out the number they need to guess and then loop again. This does not count as one of their goes.
- If their guess is within 1 of the actual number tell the player this.
- At the end of the game, before printing 'Game Over', modify your program so that it asks the user if they want to play again; if they say yes then restart the whole thing.