# 823G5 Programming in Python

## Aim

To familiarise yourself with

- Functions
- Using function keyword arguments
- Building up a codebase using functions

## Exercises

Download the `Lab3Exercises` ZIP file for solutions. The code is a set of Python source code file (`.py`). You will need to open a new PyCharm project and import the `.py` files into it. Refer to the Lab 1 instructions if you cannot remember how to do this.

## Hints

There is a special reserved function name in Python `main()` that executes as soon as the Python interpreter is invoked. In general it is good practice to make obvious the point at which execution is intended to start. An example of a program with a `main()` function is shown below:

```
# main() is executed as soon as the Python interpreter is invoked for
# a .py file

def display_animal(name, age=0, species='unknown'):
    print('This animal is called', name, 'and is age: ', end='')
    if age == 0:
        print('unknown', end='\n')
    else:
        print(age, 'years', '')
    print('The species is:', species, end='\n')
    print()

# Starting point for this program
def main():
    display_animal('Amelia')
    display_animal('Freddy', age=5)
    display_animal('Baloo', age=8, species='bear')
    display_animal('Clive', species='snake', age=3)

# Just add this...
if __name__ == "__main__":
    main()
```

## Coding challenges 1

Here is some example code that can generate a random number in a specific range:

```
import random # A library that contains random number functions and methods


 x = random.randrange(98) # range 0 to 98 inclusive
 y = random.randrange(98) + 1 # range 1 to 99 inclusive
```

Try these challenges out:

- Create a random number generator function that takes two parameters e.g. `gen_random(min_val, max_val)`, and then returns an output value between those two values inclusive.
- Create a function to generate a lottery ticket. A lottery ticket can be conveniently represented using a list of 6 integer numbers with values between 1 and 50 inclusive. The function should return the list. Numeric values in a list can be sorted into ascending order using `list.sort(reverse=???)` where `list` is the list you want sorted a keyword argument reverse is used (value `True` or `False`) to specify whether you want ascending or descending order. There should be no duplicate values in your list.
- Create a function to check whether we have won the lottery. This function should take the lottery ticket list previously created as one parameter, and a list of the 6 winning numbers as the other argument. If 2 numbers match return a win of £1. If 3 numbers match, the function should return a win of £10, if 4 numbers match, return a win of £50, 5 numbers a win of £500 and 6 numbers £1,000,000.

Some suggested answers are provided in Canvas.

## Coding challenge 2

In this challenge, we recreate a simple number guessing game. The rules are simple:

- Python will generate a random answer between 1 and 99 inclusive.
- The user then has a maximum of 10 attempts to guess the number.
- If the user guesses correctly, the game is won.
- If the user guesses too low, a “too low” message is generated.

- If the user guesses too high, a “too high” message is generated.
- If the user enters a number that was previously input, a message “you tried that already” is generated and the turn count is not increased.
- The user can quit the game by entering a guess of -1.

All user guesses should be stored in a list for convenience.

To build this game, divide the functionality up into some appropriate functions:

- `generate_answer()`
- `analyse_guess()`: should take the user guess as a parameter and return -1 if it is too low, -2 is if it is too high and 0 if it is correct.
- `check_previous()`: should take a list of user guesses and the current user guess and return `True` is the user guess is already in that list, `False` otherwise.
- `main()`: to get the game working.
- See earlier in this document for how to generate random numbers.

An example solution is available on Canvas.

Dr. Benjamin Evans
B.D.Evans@sussex.ac.uk