

EMAT10007 – Introduction to Computer Programming

Exercises – Week 3. Data Structures

1. **Complete last week’s exercise sheet.** Ensure you took a look at last week’s brief exercises on conditional statements and loops.
2. **Simple self-test exercises on the last two weeks of material.**
Make sure you can do these two problems. You need to ask for help at this point if you find these too difficult.

- Write a program that assigns two variables, say A and B, integer values between 1 and 10 at random (you will need to use `random.randint()`). Your program should keep reassigning random values to A and B until `A == B`. Then, create another variable to count the number of times the program assigns A and B random values until both numbers are equal. When this happens, print out a success message and the number of assignments it took for `A == B`.
- If you haven’t already done so, implement the “Number Guessing Game” from last week, and then extend the program to provide additional information when the guess is incorrect e.g. “You guessed higher/lower. Guess again.”

What about when the user guesses a number outside of the range, say 1 – 10 (whatever range your program picks random integers from)? Add a check to your program to instruct the user to enter a guess within the accepted range.

3. Exploring data structures.

- Lists
 - Have a look at `help(list)`.
 - Make two list containing the values [1,2] and [3,4].
 - Change the value 1 to the value 5.
 - Sort the lists.
 - Make a nested list that contains both lists.
 - Use two loops to print out all the values in the nested list (2x2 matrix) one by one.
 - Use list comprehension (see slides) to make a list with all odd numbers from 0 to 100.
 - Write a program that asks the user to input a list of 10 words (strings) and which then creates a list containing the length of each word. Print out each word and word length, like so:
Word: Algorithm - Word length: 9
- Tuples
 - Have a look at `help(tuple)`.
 - Make a tuple containing the values (“gruyere”, “vacherin”).

- Print all the items in the tuple.
 - Change the value “gruyere” to the value “cheddar”. Does it work? Why? (remember fondue recipes are sacred).
 - Is there a function to remove the last item of the tuple? How else could you do it?
- Sets
 - Have a look at `help(set)`.
 - Make a set that contains {1,2,5,5,8} and one that contains {1,2,4,9,2}. Print out the sets, are there any duplicates?
 - Can you access an element of the set based on index?
 - Use the keyword `in` to check if 4 is in both sets.
 - Use the operators `&`, `|`, `-`, `^`. What do they do?
 - Remove the value 1 from the first set, and add the value 6.
 - Dictionaries
 - Have a look at `help(dict)`.
 - Make a dictionary that contains {“Jill”:21, “Sally”:20, “Bob”:20, “Harry”:21}.
 - Print out all the keys in the dictionary.
 - Add the item “Rachel”:19 to the dictionary.
 - Remove the item “Bob”.
 - Add the item “Jill”:22 to the dictionary. Are there two Jills now?
 - Check if Harry is in the dictionary.
4. **Next week’s class test.** Next week will be the class test on **elementary concepts** in programming. Refer to the other worksheet for example questions.