CSB110MC – Python Programming

Lauren Powell

The lab classes consist of tasks that are designed to be completed during the lab class. If you do not complete the tasks during the lab class then you should do them at home and have them ready to be checked in the next lab class.

Each task should be stored in its own `.py` file.

Do not use the interpreted mode for doing the tasks (only use the interpreted mode for experimentation). If you store your python code as `.py` files then you can go back and look at them at a later date.

CSB110MC Lab Class 2 — Thursday 16/02/2023

☐ Task 2.1

Write a program which prints the sequence 1, 2, 3, . . . , 9, 10 to the screen, each on a new line, using a while loop (and only one print statement).

Hint: Use a while loop with a counter variable.

Once you have done this, extend your program to print a second sequence of numbers:
10, 9, 8, . . . , 2, 1 to the screen, each on a new line, using a while loop (and only one print statement).

☐ Task 2.2

Write a program which asks for the name of your lecturer. If the name is "Lauren" then the program should display "Lauren teaches CSB110MC". If the name is not "Lauren" then it should display "<lecturer name> does not teach CSB110MC". Your program should be case insensitive.

☐ Task 2.3

Write a program to check if you are old enough to drive. The program should ask for the user's age in years, and output whether they are old enough to drive (in the UK).

Your program should include input validation. If the user enters a negative number, you should print an error message and prompt the user to input again (use a **while** loop).

☐ Task 2.4

Write a program that allows the user to enter a temperature in the form of an integer directly followed by a letter (in a single input). The integer is a temperature value and the letter is either 'C' for Celsius of 'F' for Fahrenheit.

The program should print whether water is liquid, solid, or gaseous at the given temperature.

An example run might be

```
Enter temperature: 95C
Water is liquid at this temperature.
```

Hint: This will involve you reading a string from input, extracting sub-strings and parsing one of those as an integer.

☐ Task 2.5

Write a program that takes three numbers as input and prints "increasing" if they are in increasing order, "decreasing" if they are in decreasing order, and "neither" otherwise. Here, "increasing" means "strictly increasing", with each value larger than its predecessor. The sequence 3 4 4 would not be considered increasing.

☐ Task 2.6

Write a program which asks the user for the size of a box between 3 and 10. The program should repeatedly prompt the user for a number until they give a valid input. The program should then ask the user if they want a filled box or an empty box.

The program should then print a box of this size (filled or not). E.g., if the user enters the number 5, and wants an unfilled box, the program should print:

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

whereas if they wanted a filled box it should print

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

Hint: This can be done with or without loops. You can do this using string replication.

☐ Task 2.7

Write a program which prompts and allows the user to enter a time in the 24 hour format hh:mm:ss (i.e., a single string). The program should pull out and store the hours, minutes and seconds as three separate integer variables. The program should then increment the seconds by 1. The program should properly deal with the seconds becoming 60, i.e., the seconds should be reset to zero and the minutes incremented by one. The program should similarly deal with the minutes becoming 60 and the hours becoming 24. Finally, the program should print the new time in the standard format hh:mm:ss.

For example, if the time is 09:59:59 and 1 second is added then the output will be 10:00:00.

Hint: A convenient way to print the hours, minutes and seconds is with the format specifier `%02d`. This will ensure your numbers are printed as two characters in length with a leading zero (if needed).

Note: You are not writing a stop watch or timer. You only need to add a single second.

☐ Task 2.8

A *palindrome* is a word that is spelt the same forwards and backwards. Example of palindromes are "noon", "madam", and "racecar". Write a program that allows the user to input a word and outputs whether it is a palindrome or not.

Hint: The are various ways to do this, but an easy way involves keeping two "pointers" referencing the positions in your string. The pointers start at the beginning and end of the string. You bring them in (towards the middle) each time that you find matching characters. If you find non-matching characters you know the string is not a palindrome.

☐ Task 2.9

This task extends your Nim program from Task 1.5.

Write a program that prompts the user for two integers, representing the number of tokens in two different piles. Print a representation of the two piles to the screen.

Next, the program should prompt the user to choose a pile (1 or 2) and a number of tokens to remove from that pile. Print the two piles with the desired tokens removed.

Finally, the computer should take a turn, randomly selecting one of the piles and a number of tokens to remove from that pile.

An example run of the program might be:

```
How many tokens would you like in the piles?  6,3

pile 1: ******
pile 2: ***

Which pile do you want to remove tokens from?  1
How many tokens would you like to remove?  5

pile 1: *
pile 2: ***

I remove 3 tokens from pile 2!

pile 1: *
pile 2:
```

Hint: If one of the piles is empty after the user has taken their turn, then the computer should be forced to remove tokens from the non-empty pile.