



COMP 132 Term Test
**Programming for the Natural
and Social Sciences**

Time Allowed: 50 minutes

OPEN BOOK

Permitted materials: All materials, calculators and dictionaries are permitted

Instructions: You can access the Term Test page by clicking the Term Test link from the green menu on the left hand side of our course home page. This Term Test page has links to the template file and the submission system.

Do not spend too much time on a question if you get stuck. For example, if you have to Google for a question, do it later after you work on other questions.

Write all your answers in the template file called `TermTest.ipynb`. Write your name and student ID at the top of your file as comments.

Save your file frequently to avoid data loss.

Submit your file using our online submission system immediately after the test.

The term test will be marked out of 50.

Questions

1. Using if [8]
2. Writing loops [8]
3. Strings [8]
4. Using lists [10]
5. Numpy Arrays [8]
6. Data visualization [8]

Question 1. Using if**[8 marks]**

A company wants to give a years-of-service bonus to their employees. If the employee has been working in the company for more than 5 years (including 5 years), the bonus is 5% of annual salary, otherwise 3%. Write a Python program that asks a user for the annual salary and years of service, and prints the bonus amount.

Question 2. Writing loops**[8 marks]**

Write a program to print the even numbers between 40 and 90, including 40 and 90. An even number is a number that is a multiple of 2. For example, 2,4,6,8 are even numbers because they are divisible by 2.

Question 3. Strings**[8 marks]**

Write a function `char_count(mystr)` that returns the total number of upper case letters and digits in the string `mystr`.

Conduct at least two tests.

- `char_count("COMP132testT2")` should return 9;
- `char_count("Hello World!")` should return 2.

Question 4. Using lists**[10 marks]**

The template file gives the following python list instance:

```
fruit_list = ["apple", "banana", "orange"]
```

Write code to do the following actions:

- (a) **[2 marks]** Print the third element of the `fruit_list`. (Your code should print `orange`)
- (b) **[2 marks]** Insert `"pear"` between `"banana"` and `"orange"` in `fruit_list`.
- (c) **[2 marks]** Insert `"grape"` to the beginning of `fruit_list`.
- (d) **[4 marks]** Write a function called `insert_to_list(word_to_add, words_list)` to insert a new word string `word_to_add` to the beginning of an existing list of words `words_list`, and to return the extended list.

The function should only add the word if it does not already exist in the list.

Test and print the returned results:

- `insert_to_list("apple", fruit_list)` should return the same `fruit_list` without any changes.
- `insert_to_list("plum", fruit_list)` should return an extended list that contains `"plum"` as its first element.

Question 5. Numpy Arrays**[8 marks]**

The template file starts this question by importing the Numpy library and gives five data arrays:

```
import numpy as np
temp_high = np.array([14, 15, 14, 14, 13, 12, 9])
temp_low = np.array([10, 12, 11, 10, 4, 7, 4])
rain = np.array([0, 0, 0, 0.6, 20, 0, 10])
wind = np.array([11, 9, 7, 35, 41, 20, 37])
day = np.array(["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"])
```

Suppose the data are the weather forecast data for next week from Monday to Sunday. For example, Monday's highest temperature is 14°C, lowest temperature is 10°C, rain is 0 mm, wind is 11 km/h. The days (Mon to Sun) are also represented as a numpy array for your convenience.

Write code to complete the tasks below:

(a) [4 marks] Print the diurnal range of temperature (the difference between the maximum and minimum temperature in the same day) for each day and the maximum diurnal range of temperature in the week.

(b) [4 marks] Suppose a good day is a day where there is no rain, wind is lower than 20 km/h and the highest temperature is greater than 12°C. Find out the good days and print the days (e.g. "Mon", "Fri") when the good days are.

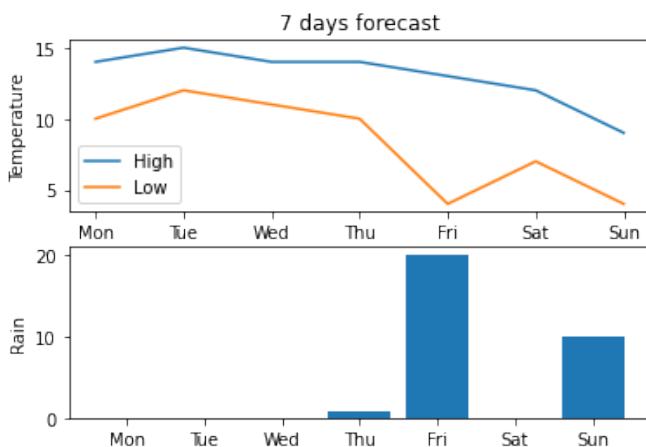
Question 6. Data visualisation

[8 marks]

The following is included in the template file

```
%matplotlib inline
import matplotlib.pyplot as plt
```

Using the same data as in Question 5, write code to create the following picture with two sub-figures: one for the plots of temp_high and temp_low and one for the plot of rain. Any color is fine.



Please note:

- The top figure has a legend to distinguish the two lines: one for high temperature and one for low temperature.
- The x axes show the days Mon to Sun in both figures.
- The y axes have labels.
- The title should be at the top.
