

CompSci 101 Assignment 3

**Some helpful information
about Question 7**

A3 Q7 - `is_a_valid_date()`

parameter – a string

returns – a boolean indicating whether the parameter string denotes a valid date or not.

The first two lines of the function are:

```
month_names = ["January", "February", "March", "April", "May",  
"June", "July", "August", "September", "October", "November",  
"December"]  
days_in_month = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
```

```
print("1.", is_a_valid_date("January 21"))  
print("2.", is_a_valid_date("Auust 3"))  
print("3.", is_a_valid_date(' June 15B '))  
print("4.", is_a_valid_date('February 0'))  
print("5.", is_a_valid_date(' December 3K1 '))  
print("6.", is_a_valid_date(' May 31 '))
```

1. True
2. False
3. False
4. False
5. False
6. True

A3 Q7 - `is_a_valid_date()`

How to approach this question

As soon as you know the result is `False`, return `False` and continue with the checking.

```
def some_function(...):  
    ...  
    if condition1:  
        return False #stop executing the  
                       #function and return False  
#continue the function code knowing that  
#condition1 is not true  
    ...  
    if condition2:  
        return False #stop executing the  
                       #function and return False  
#continue the function code knowing that  
#condition2 is not true
```

A3 Q7 - `is_a_valid_date()`

Strip off leading and trailing spaces

The `strip()` method can be applied to a string object.

```
def some_function(...):  
    word = "    August    5 "  
    word = word.strip()    #word is now "August    5"
```

Often you would like to return `False` if the string is the empty string

```
def some_function(...):  
    word = "    "  
    word = word.strip()  
    if the word string is the empty string,  
        #do not continue with the function,  
        #can now return False
```

Now can continue knowing word contains
some characters

A3 Q7 - `is_a_valid_date()`

Split a string into separate words

The `split()` method can be applied to a string object. The result is a list of the individual words in the list

```
def some_function(...):  
    info = "happy busy programmer"  
    info_list = info.split()  
  
    #At this point, you can check if the list of  
    individual words is of a certain length and  
    return False if it isn't  
  
    #Now can continue knowing the list of  
    individual words is the correct length
```

A3 Q7 - is_a_valid_date()

The 'in' operator

The 'in' operator can be used to check if a string is an element of a list.

```
def some_function(...):
    valid_list = [..., ..., ..., ...] #list of valid words
    word_to_check = ...

    if word_to_check not in valid_list:
        #can now return False if it isn't in the list

    #Now can continue knowing that the word_to_check
    #is in the list of valid words
```

A3 Q7 - `is_a_valid_date()`

Get the legal maximum number of days.

Find the index of an element which exists in the list using the `index()` method.

```
month_names = ["January", "February", "March",  
               "April", "May", "June", "July", "August",  
               "September", "October", "November", "December"]  
days_in_month = [31, 28, 31, 30, 31, 30, 31, 31,  
                  30, 31, 30, 31]
```

For example, if the month is `"April"`, the index of this in the `month_names` is 3. This means that the maximum number of days in the month of `"April"` is the number at index 3 in the `days_in_month` list (30).

IMPORTANT: A maximum number of days of 30 means that the day part of the `date_str` can be any number between 1 and 30 (both inclusive).