DAT154 – Assignment 0

Part 1 – C++:

Task 1

We will start by getting a bit familiar with C++.
Start by creating a new 'Console App' in Visual studio (Make sure it is the C++ one, and not a .net one).

Replace the code in main() with the following:

```
    char buffer[25];

    std::cout << "Enter a word: ";
    std::cin >> buffer;




    std::cout << buffer << "\n";
```

Now run the program.
What happens if you write a word longer than the buffer (25 characters)?

Task 2:

Explain the following code. What is *p and what is happening to it during the run?

```
int Myfunction(const char* p) {
    int i = 0;
    for (; *p; p++) {
        i++;
    }

    return i;
}
```

Task 3:

Implement the code from task 2 in your code from task 1, and use the result in the output in a meaningful way.

Task 4:

Write a function that takes in an integer as a parameter and changes it. This should be implemented in such a way that the change will propagate back to the calling code and not just be local to the function. Show both the function, and how to call it correctly.
Note that you should NOT use the return keyword to return a value from the function, you should change the value being passed as a parameter directly, your function should have return type void.

Task 5:

Write a small C++ program that asks the user to input a list of words (one word on each line), stores these in a vector, then sorts the vector alphabetically and then uses an iterator to write the words back out to screen one by one.

Task 6:

Write a small object oriented program for a parking garage and cars. The program should be simple and menu driven, with a menu similar to this:

1. Drive a red car into the garage
2. Drive a blue car into the garage
3. Drive a green car into the garage
4. Drive a car out of the garage
5. Quit

Option 1-3 should simply create a new car object with the appropriate properties, and put it into the garage. However, the garage should have limited capacity, so if it is full, an error message should be printed instead and no car should be put in. Option 4 should just remove the car that has been the longest in the garage, and print out which color car drove out.

Part 2 – SDK:

Task 7:

Start a new Windows Desktop Application (C++) in Visual Studio.

This will create an SDK program. Build the program and see if it works.
Try to find the message loop. WinMain and other parts. Experiment on your own. Try to change the standard cursor (Replace IDC_ARROW with IDC_CROSS !).
Also change the background color.

Task 8:

Place the cursor on all function calls and read about these. Use www.google.com to find documentation about these.

Also get to know the help system.

Now, implement a handler for left mouse button clicks (WM_LBUTTONDOWN) that triggers a simple MessageBox() with some text in it whenever the user click a spot.

Task 9:

Add

```
WCHAR dialogText[50];
```

As a global variable to your program. Then add

```
TextOut(hdc, 200, 200, dialogText, lstrlen(dialogText)+1);
```

After the HDC…… line in WM_PAINT

Design a simple dialog box with a simple text entry field.
Show this dialog whenever the user clicks the mouse button inside the window.
When the user clicks ok in the message box, the value from the text field should be retrieved and stored in the dialogText global variable.

You'll also want to throw in

```
InvalidateRect(hWnd, 0, 0);
```

Right after the call to show the dialog, as this ensures the screen gets updated (WM_PAINT gets called).

Try the dialog to make sure it works.