



# Introduction to Programming

## Tutorial Task 9.1: C Name Tester

### Overview

In this task you will convert your Pascal NameTester program (SillyName) to a C program.

- Purpose:** Learn to use the control flow statements within a C program.
- Task:** Create a program that tests a user's name and echoes a custom message.
- Time:** This task should be completed before the start of week 10.
- Resources:**

- Programming Arcana
- Google
- Swinburne CodeCasts ([YouTube Channel](#), [iTunesU](#))
  - [Learning a new language](#)
  - [Introducing Objects](#)

### *Submission Details*

You must submit the following files to Doubtfire:

- NameTester program C source code.

Make sure that your task has the following in your submission:

- Demonstrates use of C programming convention, including naming, indentation within selection and repetition statements.
- Demonstrates use of an **if** statement to perform selection, and a **while** or **for** loop to perform repetition.

## Instructions

Remember in Tutorial Task 3.1 we wrote a program that would print out 60 “Silly”s after a person’s name. This time you are given some (faulty) starter code which you must fix and add to so that it meets the following requirements:

Create a small program that will check the user’s name and respond with different messages for different people.

**Hint:** Use the notes from Lecture 7 to find example code to help you implement this.

1. Download and extract the resources for this task.
2. Open **name\_tester.c** using Sublime Text.
3. Implement a **main()** procedure with the following logic:
  - It reads a name from the user.
  - Check if the name is the same as your tutor’s name or your name.
  - If the name is the same as the tutor’s name or your name, then print out “Awesome name!”).
- Otherwise output the silly name message.

The pseudocode for this procedure follows:

Procedure: **main()**

Includes: `terminal_user_input.h`

-----  
Local Variables:

- name (the silly name String)
- index (an integer, used to count the number of loops)

-----

Steps:

- 1: Assign name, the result of calling `read_string` with the prompt: 'Please enter your name: '
- 2: if name is '-add-your-name-here-' or '-add-your-tutor's-name-here' then
3.   Output 'Awesome name!'
4. else
- 5:   Output (staying on the same line) name, 'is a '
- 6:   Make i equal 0
- 5:   While i is less than 60
- 6:   Output (on the same line) ' silly'
7.   Increment i
- 7:   Output ' name!' (moving to a new line)

**Hint:** Use the `strcmp()` function from the C library `<string.h>` (you will need to include this). See [Tutorials Point for an example](#).

4. Use the MinGW command line to compile and run your program.

Open a **Terminal** window and compile and run your program:

- Change into the directory with your code using the `cd` command
- Compile your program using `gcc -o name_tester name_tester.c terminal_user_input.c`
- Run your program using `./name_tester.exe`

5. Get a screenshot of the terminal showing the output of running this program with your name, and with someone else's name. Submit your code and your screenshot to Doubtfire for tutor feedback.