WARWICK

SCHOOL OF ENGINEERING

Module Code:

ES2C4

Module Title:

Computer Architecture and Systems

Learning Activity:

Lab. 2 C Programming (Week 18)

Learning Objectives:

- **Write C programs with pointers, structures and functions**
- **Design programs with shared declarations using headers**

Instructor:

Dr Sam Agbroko sam.agbroko@warwick.ac.uk

1. Introduction

You should complete Lab.1 before starting Lab. 2.

This lab was designed to guide you through writing C programs with pointers, structures and functions. These concepts have been covered in the lectures and practical demo videos.

After completing the guided tutorials in Section 2, try out the exercise in Section 3 to test your skills.

2. Guided tutorials

2.1. Structures

(i) Declare a structure for a rectangle as shown below:

```
struct rectangle
{
   int length;
   int width;
};
```

(ii) Declare an instance of the rectangle:

$$\underbrace{\texttt{struct rectangle}}_{\text{type}}\ \underbrace{\texttt{myRectangle;}}_{\text{variable name}}$$

(iii) Initialise the member variables of the new rectangle:

```
myRectangle.length = 9;
myRectangle.width = 5;
```

(iv) Print out the values stored in the member variables of the rectangle.

2.2. Functions

(i) Create a function to calculate the area of the rectangle:

$$\overbrace{\texttt{int}}^{\text{Return type}}\ \overbrace{\texttt{calcArea}}^{\text{Function name}}\ \overbrace{\texttt{(int length, int width)}}^{\text{Parameter types and names}}$$

```
{
   int _area;
   _area = length * width;
   return _area;
}
```

(ii) Use this function to calculate the area of the rectangle you created in Section 2.1 and print the result to the console.

(iii) Rewrite the function such that:

- it calculates the area,
- does not return any value,
- arguments to the function are passed by address,
- the result of the calculation is also saved to an address.

3. Exercises

(i) Write a program to keep a record of dogs at a vet. Each dog has the following information: Name, Height, Age and Breed.

(ii) There are four dogs on the record shown in the table below:

ID	NAME	HEIGHT (CM)	AGE	BREED
dog1	Ben	59	9	Golden retriever
dog2	Koko	40	8	Kelpie
dog3	Terry	25	10	Terrier
dog4	Skyes	32	15	Terrier

(iii) Write functions to do the following (all functions must have a void return type, therefor arguments should be pointers):

- Print out the name, height, age and breed of a dog when the dog struct is provided as an argument.
- Change the age of a dog when a new age and the dog struct is provided.
- Change the name of a dog when a new name and the dog struct is provided.
- Change the breed of a dog when a new name and the dog struct is provided.

(iv) Using the functions described in Section 3(iii), carry out the following tasks:

- Change Dog3's name to "Lando".
- Change Dog2's age to 10.
- Change Dog1's breed to "Husky".
- Print out the name, height, age and breed of a dog when someone types in the dog ID into the console window.

(v) Finally, create a separate source file for the functions you created. Use the '`#include`' directive to ensure the compiler can find the functions in the main source file.

(vi) Analyse the design and correctness of your program in terms of:

- The requirements specified in exercises i – v.
- Use of memory resources.
- Eliminating repetitive tasks using functions.
- General performance.