

# COMP 30400 – Programming I & II

## Lab 3 – 4 hours

### Practical Overview

This practical covers some topics introduced in Lecture 2, such as to assigning values to variables and arrays. This practical covers the new topics from Lecture 3: if conditions and while loops.

For each of the programs in this lab, add comments to your code to help explain what each statement and declaration is doing. Recall that the comment characters are:

```
/* like this */
```

where the order is reversed at the end.

### 1) Portfolio Program 8

Write a program that asks the user to enter 2 numbers.

Add the 2 numbers and display the result to the user.

Subtract the 2 numbers and display the result to the user.

Hints: `int number;`  
`scanf("%9d", &number);`

### 2) Portfolio Program 9

Use program 8 as a starting point, and modify it so that:

Instead of storing the user's input in 2 int variables, store the values in an int array.

When calculating the result of the sum (+) and subtraction (-), store the results in an array as well.

Hints: `scanf("%9d", &numbers[1]);`

### 3) Portfolio Program 10

Write a program that asks the user to input three numbers, and then display the largest of the three to the user.

**More tasks overleaf...**

#### 4) Portfolio Program 11

Copy the following code into a .c file and compile it.

```
#include <stdio.h>

int main(void)
{
    char word[10];
    char c;

    printf("enter a word: ");
    scanf("%9s", word);
    printf("enter any character: ");
    c = getchar();

    if (c == '\n') printf("warning: char c contains a newline\n");
    printf("you entered: word: %s, char: %c\n", word, c);

    return 0;
}
```

When you run this program, you'll notice that it asks you to enter a word and a character, but it won't wait for you to enter a character. This is because `scanf()` leaves the '\n' character in the input buffer.

When you run the program, it will output the line: `warning: char c contains a newline`

One solution to this is to call `getchar()` twice, as the first will return the newline, and the second character will be the user's actual input. A better solution is to use a `while()` loop, because if there is more than one newline character in the input buffer, the while loop can keep calling `getchar()` until the character, `c`, is not a newline.

Modify this program to use a while loop, where the condition should be *while c is equal to newline*, and the statement should call `getchar()`;

Verify that your solution works by running it and seeing if it outputs the warning.

#### 5) Portfolio Program 12

Based on program 9, make a simple calculator.

Ask the user which operation they wish to perform on the numbers (+, -, \*, /).

After the user specifies an operation, ask them to enter the 2 numbers (as done in program 9).

Calculate the result and display it.

Hints:    `char operator;`  
          `operator = getchar();`  
          `if (operator == '+') ...`

**recall that the first time you call `getchar()` after `scanf()`, it will return a '\n'.  
also remember to use 2 equal signs in Boolean expressions.**

### **6) Portfolio Program 13**

Building on program 12, modify the program so that it loops until the user enters 'q' as the operation.

The program should ask the user for the operation (+, -, \*, /, q), followed by the 2 numbers. After the results are displayed, the user should be prompted to enter the operation and 2 numbers again.

If the user enters 'q' as the operation, the program should **not** ask the user to enter 2 numbers.

When the user enters 'q' the program should finish (i.e. exit the loop).

### **7) Remember to backup your portfolio.**