Programming – DT211/1

Lab 11 – Wednesday, December 5th, 2012

Note: You are expected to finish all programmes in your own time if you do not get these done during the lab session. This is your own responsibility.

Pointers/Arrays

Remember: Use Symbollic names in your programs. Do not hard-code.

Write separate programs to:

1.   Write a program that uses a pointer (which points to a variable called *radius* to store the radius of a circle) to read the radius of a circle (i.e. use the *indirection operator* in a scanf() statement to read and store the radius inside the floating-point radius variable). Your program should calculate and display the diameter (2 x radius) and area (Π x radius^2) of the circle using other pointers (these also need to point to other variables required such as *diameter* and *area*) . Assume Π has the value 3.14

2.   What is the output of the following program segment?

```
int  count = 10, *temp, sum = 0;

temp = &count;
*temp = 20;
temp = &sum;
*temp = count;

printf("count = %d, *temp = %d, sum = %d\n",
                        count, *temp, sum );
```

3.   Write a program that reads 3 integers into an integer array and copies these numbers to a different integer array. Display the contents of each array side by side. You must use **pointer notation only** to read the numbers into the array, copying the numbers, and displaying the contents of both arrays.

4.   Declare and initialise a floating-point array with 3 elements to contain 1.1, 2.2, and 3.3 respectively. Using **pointer notation only**, try to display the contents of the addresses outside the bounds of this array, e.g. display the contents of the memory address before the first element (1.1) and after the last element (3.3) in the array. What do you see?

5.   Write a program that uses a 1-D array with 5 elements to store float numbers. Using **pointer notation only**, your program should do the following:

   a) Enter in values for each element in the array.
   b) Find the lowest and highest value in the array and display these on the screen.
   c) Calculate the average value in the array.
   d) Define another 1-D with 5 elements to store float numbers. Copy the values from the 1st array *in reverse order* into the 2nd array.
   e) Display the contents of both arrays side-by-side.