

Data Types Part II

Objectives:

So far we have looked at local variable now we switch our attention to other types of variables supported by the C programming language:

- 1.Global Variables
- 2.Constant Data Types

Global variables:

Variables can be declared as either local variables which can be used inside the function it has been declared in (more on this in further sections) and global variables which are known throughout the entire program. Global variables are created by declaring them outside any function. For example:

```
int max;

main()
{
    .....
}
f1()
{
    .....
}
```

The int max can be used in both main and function f1 and any changes made to it will remain consistent for both functions. The understanding of this will become clearer when you have studied the section on functions but I felt I couldn't complete a section on data types without mentioning global and local variables.

Constant Data Types:

Constants refer to fixed values that may not be altered by the program. All the data types we have previously covered can be defined as constant data types if we so wish to do so. The constant data types must be defined before the main function. The format is as follows:

```
#define CONSTANTNAME value
```

for example:

```
#define SALESTAX 0.05
```

The constant name is normally written in capitals and does not have a semi-colon at the end. The use of constants is mainly for making your programs easier to be understood and modified by others and yourself in the future. An example program now follows:

```
#define SALESTAX 0.05
#include <stdio.h>
main()
{
    float amount, taxes, total;
    printf("Enter the amount purchased : ");
    scanf("%f",&amount);
    taxes = SALESTAX*amount;
    printf("The sales tax is £%4.2f",taxes);
    printf("\n The total bill is £%5.2f",total);
}
```

The float constant SALESTAX is defined with value 0.05. Three float variables are declared amount, taxes and total. Display message to the screen is achieved using printf and user input handled by scanf. Calculation is then performed and results sent to the screen. If the value of SALESTAX alters in the future

it is very easy to change the value where it is defined rather than go through the whole program changing the individual values separately, which would be very time consuming in a large program with several references. The program is also improved when using constants rather than values as it improves the clarity.