Storage of Data in a C Program

One of the most commonly heard expressions in the computer industry these days is Information Technology, or IT. We use it to refer to any thing that processes data or manipulates some type of information. For a computer program to be able to work with information, it needs a standard approach to being able to save information. All data is ultimately saved as combinations of 1's and 0's (bits), but the number of bits needed depends on the type of information being stored. Certain types of information require very little storage space, while other types require a great deal of space.

Variables

The most commonly used information in your programs will be variables. As its name implies, a variable is a storage location for a piece of data that has a value that can vary. The data type will determine the range of values allowable, and the program will assign or modify this value as appropriate. To refer to a particular piece of data, programs use an identifier, or name, unique for every different piece of data. Names are chosen by the programmer to be meaningful, so that it is easy to debug or modify.

Common Data Types

Type	Used For
char	One byte: integers -127 to 128, or 0 to 255 if *unsigned* or a single letter
short	Two bytes: integers from –32767 to 32768; or 0 to 65535 if *unsigned*
int	Four bytes: integers from –2147483648to 2147483648 **
long	Same as int
float	decimal numbers
double	large decimal numbers

** In some compilers, these limits are different. A **short** may be a byte, an **int** two bytes and a **long** four bytes. Regardless, when using any compiler, an **unsigned** data type uses only the positive range for the values. Note that an unsigned 64 bit (8 byte) integer has a maximum value of 18446744073709551615 !

Declaring Variables

All variables in a C program should be declared at the very beginning of the main function. (Or the function they are to be used in) It is good form to comment variable declarations to both indicate their intended use, as well as their expected values.

Example

```
int main()
{
        int     age;                    //an integer used to store the age of a student
        float   averageMark;            //the student's average mark as a percent
```

Assigning Values to Variables
A variable may have its value specified when it is declared, or it may be assigned anywhere else in the program.

Example
```
int main()
{
        int     age = 15;                   //Gives the variable age a value of 15
        float   averageMark;                //the student's average mark as a percent

        averageMark = 85.6;                 //the student's mark for ICS3U
```

Output of Variables using printf

One of the easiest ways to display the value of a variable is to output it to the console using a printf statement. For each variable being printed, a format specifier must be given. Multiple variables can be printed in a single printf call, but each one must have its own format specifier given. The order of the both the variables and the specifiers is important, as shown in the example below.

Example
```
int main()
{
        int     age = 15;                   //Gives the variable age a value of 15
        float   averageMark;                //the student's average mark as a percent

        printf("My age is %i and my average is %f %%", age, averageMark);
```

Other format specifiers. %c for char, %d for digit. %3.2f will force a float to 2 decimals

Exercise
Create a new program file called myInfo.c that will display several pieces of information about you. You should have at least one variable of type char, short, int and float. You should include comment lines in your program, and use meaningful names for your variables. (Show your program to Mr. Ferguson by the end of class for marking.)