Tutorial Sheet 2   
ESC101 – Fundamentals of Computing

**Revision (ask for doubts)**

1. Distinction between declaration and initialization of variables.
2. Commenting code is useful and important too. Comment styles.
3. scanf for int, long, float. Multiple inputs via scanf and Mr C’s habit of skipping multiple whitespaces while reading ints/floats/longs.
4. Legal identifier names and best practices while giving names.
5. Mixed type operations (int, long, float). int a; long b; float c;  
   Mr C automatically typecasts upwards when operands mixed.
   1. b = a + a will cause integer addition as both operands int
   2. b = long(a) + a will cause long addition as one is long
   3. b = a + 50000000000 will cause long addition as 5e10 is long
   4. c = 2 / 3 will cause integer division and result will be c = 0.0
   5. c = 2 / 3.0 or c = 2 / (float) 3 will cause float division since at least one operand is float

**Sample Question to discuss**

#include<stdio.h>

int main(){

int a ,b;

long prod;

scanf(“\\\\%d||%d//”, &a, &b);

prod = (long)a \* (long)b;

printf(“%ld”, prod);

return 0;

}

Given: two integers between -1,000,000 and +1,000,000. Read them into two integer variables and print their product. Be careful about exceeding the range of integers. The integers will be presented in the input as

\\integer1||integer2//

**Some Pitfalls and recognizing compiler error messages**

1. Multiple declarations, simultaneous declarations allowed but mixed type declarations forbidden e.g. int a, float b; Mixed type initializations allowed though e.g. int a; float b; a = 1, b = 2.2;
2. Printf can handle without any need for escape commands,
   1. Latin letters: upper and lower case
   2. Arabic numerals: 0 – 9
   3. Arithmetic operators: + - / \* = < >
   4. Parentheses: ()[]{}
   5. Punctuation marks: space ~`!@#$^&\_|?/,.

However, “ \ % newline, tab require escape commands. Reason: tab, newline are whitespace characters requiring special handling, whereas “ \ % are themselves have special roles inside the format string so they themselves have to be escaped.

1. /\* \*/ comments do not nest i.e. /\* /\* \*/ \*/ will cause error
2. count = count + 1 will take count, add one to its value, and put the result back into count itself, overwriting the old value.
3. Importance of bracketing (a+b)/2 vs a+b/2