A290/A590 – Meeting 4 Guide
C – More of the Basics

Goal: We want to get some of the real basic things out "on the table." Most of you know something about all of these, but we still need to be sure we all understand them in more or less the same way.

I. `scanf` basics.

`scanf` is crucial to accepting user input, but there are a few things you need to keep in mind, based on what we have done so far:

1. Be sure you use the addressing operator (&) on all variable arguments.
2. Don't use a precision specifier like you do with `printf`.
3. [Others that apply to "strings", which will come later.]

II. Loops.

With Hoover's "sum of squares" example, we have used several while loops. What other kinds are there? How are they different?

There are three basic loop constructs: **for**, **while**, and **do-while**. Each is just a bit different. Hoover offers examples.

The "for loop" is designed for use when you know or set the number of iterations in advance and it will always be fixed. This means we set both a **starting condition** and an **ending condition**.

The "while loop" has only an **ending condition**, which means the number of iterations is not set or known in advance. It will run until the ending condition is met. Or, it will run forever.

The "do-while loop" is the while loop but the **ending condition** is moved to the end, meaning it will run at least once, where the while loop may not.

```
 /* A review of the loop types in C. (pg 38) */

#include <stdio.h>

int main(void)

{

int i,x;
x=0;

/*for loop: runs for values of i = 0, 1, 2, and 3*/

printf ("This is a 'for' loop\n");

for (i=0; i<4; i++)
  {
  x=x+i;
  printf("%d\n",x);
  }

/*while loop runs until i is less than 7, but what is the starting value of i for this
loop?*/

printf ("This is a 'while' loop\n");

while (i<7)
  {
```

```
  x=x+i;
  i++;
  printf("%d\n",x);
  }

/*do-while loop runs until i is less than 9, but runs at least once. What is the starting
value of i for this loop?*/

printf ("This is a 'do-while' loop\n");

do
  {
  x=x+i;
  i++;
  printf("%d\n",x);
  }
while (i<9);

return(0);

}
```

I have created a file called loops.c and we can test this code.

OUTPUT:

```
[jwhitmer@silo.luddy.indiana.edu] looptest
This is a "for" loop
0
1
3
6
This is a "while" loop
10
15
21
This is a "do-while" loop
28
36
```

III. Conditionals.

We have also used a conditional. What is the one we have used? What are the various "kinds" of conditionals?

Hoover again gives us an example.

```
/* A review of conditionals and blocks in C.  (pg 39) */

#include <stdio.h>

int main(void)

{

int i,x;
x=0;

for (i=0; i<5; i++)
  {
  if (i%2 == 0 || i == 1)
    x=x+i;
```

```
  else
    x=x-i;
  printf("%d\n",x);
  }

return(0);
}
```

What is the output of this example? Is this what you expected? Let's walk through it if we need to. What is the difference between `if-else` and just `if`?

What about `switch-case`?

```
/*An example of switch-case. Notice the cases are "grades" but they are looking for
values that are ints.*/

printf ("This is a \"switch-case\" conditional\n");

 int grade;

 printf ("Input Grade: \n");
 scanf ("%d", &grade);

 switch (grade) {
               case 1:
                        printf ("Fail (F)\n");break;
               case 2:
                        printf ("Not Good (D)\n");break;
               case 3:
                        printf ("Good (C)\n");break;
               case 4:
                        printf ("Very Good (B)\n");break;
               case 5:
                        printf ("Excellent (A)\n");break;
               default:
                        printf ("You have not input a valid grade value.\n");
                        break;
}
```

I have created a file `conditionals.c` where we can test these.

OUTPUT:

```
[jwhitmer@silo.luddy.indiana.edu] condtest
0
1
3
0
4
This is a "switch-case" conditional
Input Grade:
3
Good (C)
```

IV. Coming Next: Functions, Calling Functions, and Arrays.

Who has heard of these and/or used them?

NOTES: