CONTROL STATEMENTS:LOOPING–REPETITION

- Iteration and repetitive execution (for, while, do-while), nested loops.

Nested Loops

➲ Using one loop inside another loop is called Nested Looping. You can use one or more loops inside any other while, for, or do-while loop.

Syntax:

The syntax for a nested for loop statement in C is as follows:

```
for ( init; condition; increment )
{
      for ( init; condition; increment )
            {
            statement(s);
            }
      statement(s);
}
```

Example:

/* The following program uses a nested for loop to find the prime numbers from 2 to 100 */

```
#include <stdio.h>
void main ()
{
      int i, j;

      for(i=2; i<100; i++)
      {
            for(j=2; j <= (i/j); j++)
            {
            if(!(i%j))
            break;
            if(j > (i/j))
            printf("%d is prime\n", i);
            }
      }
}
```

Infinite loops

➲ A loop becomes an infinite loop if a condition never becomes false.

➲ The for loop is traditionally used for this purpose. Since none of the three expressions that form the 'for' loop are required, you can make an endless loop by leaving the conditional expression empty.

Syntax:

```
for( ; ; )
{
     Statement to execute for ever;
}

Or

While(1)
{
     Statement to execute for ever;
}
```

Example:

```
#include <stdio.h>
     void main ()
     {
          for( ; ; )
          {
          printf("This loop will run forever.\n");
          }
     getch();
     }
```

- When the conditional expression is absent, it is assumed to be true. You may have an initialization and increment expression, but C programmers more commonly use the for(;;) construct to signify an infinite loop.
- It can also be implemented with while and do-while loops also.

Note: *You can terminate an infinite loop by pressing Ctrl + C keys.*