**What are the Iterative statements?**

A looping construct lets your program run the same code repeatedly .looping is deciding how many times to take a certain action.

**For Loops:**

1. **for**(initialize ; condition; increment) {
2. }

Loops as long as condition is true after executing initialize. At the end of each lopp the increment is executed. Simple example:

1. **int** x;
2. **for**(x=0;x < 10; x++)
3. printf(“%d\n”,x);

These can compound:

1. **int** x,y;
2. **for**(x=0,y=100; x < y - x; x++, y—)
3. printf(“%d %d\n”,x,y);

This can be used to ierate through linked lists:

1. **for**(llp=list;\*llp;llp=llp->**next**)
2. process\_node(llp);

**Continue / Beak**

Starts the next iteration of a loop. Often used as a way to start the next list element

1. **int** x;
2. **for**(x=0;x<10;x++) {
3. **if**(x==5) // Don’t print 5
4. **continue**;
5. printf(“%d\n”);
6. }

The *break* statement edges a loop early

1. **int** x;
2. **for**(x=0;x<10;x++) {
3. **if**(x==5)
4. **break**; // Stop at five
5. printf("%d\n);
6. }

**While loops:**

Can take two forms:

1. **while**(condition) {
2. }

or

1. **do** {
2. } **while**(condition);

The difference being that with the do..while loop the code executes at least once whereas the while loop will onyl execute if the condition is true to begin with.

1. **int** x;
2. x=0;
3. **do** {
4. printf(“%d\n”,x);
5. } **while**(x++);
7. **while**(x++<10) {
8. printf(“%d\n”,x);
9. }
11. x=0;
12. **while**(x<10) {
13. printf(“%d\n”,x++);
14. }

**Switch Statement**

1. **switch**(variable) {
2. **case** :
3. **break**;
4. **case** :
5. **default**:
6. }

The switch statement evaluates the variable which must be an integer and executes the code under the proper case. If the *break* keyword is found, it exits the switch statement and executes the first statement after the switch otherwise code is executed until athe end of the switch statement. If no condition is found as a case, then the default executes. If not default is provided and none of the cases match then nothing is ran.

1. **int** i=6;
2. **switch**(i) {
3. **case** 6:
4. printf(“It **is** six\n”);
5. i++;
6. // Fall to next
7. **case** 7:
8. printf(“It **is** seven\n”);
9. **break**;
10. **case** 8:
11. printf("It is eight\n");
12. **break**;
13. **default**:
14. printf("IT is something else\n");
15. }

Some will say never to let one case fall through the next. They haven’t programmed anything too complicated in C. Ther e are many times this is exactly what you want, just make sure you put in a comment explaining why it is necessary.

**if else statement**

if condition is true the following code is executed ir if there is an *else* statement execute that in stead

1. i = 4;
2. **if**(i==4) {
3. printf("It is four\n");
4. }
5. **else** {
6. printf("It is not four\n");
7. }