COMP1917: Computing 1

4. Loops

Reading: Moffat, Chapter 4.

Loops

One thing you often need to do in procedural programs is to perform a series of statements repeatedly, for as long as certain conditions are satisfied.

C has two different `while` loop constructs:

```
// while loop
while( expression ) {
   statements;
}
```

```
// do .. while loop
do {
   statements;
} while( expression );
```

(The `do`.. `while` loop ensures the statements will be executed at least once.)

Printing Squares of Numbers from 1 to 10

```
x = 1;
while( x <= 10 ) {
   printf( "%d\n", x * x );
   x = x + 1;  //  or  x++;
}
```

```
1
4
9
16
25
36
49
64
81
100
```


Increment and Decrement Operators

- The operators `++` and `--` can be used to increment a variable (add 1) or decrement a variable (subtract 1)
- It is recommended to put the increment or decrement operator after the variable:

```
                // suppose k=7 initially
 n = k--;       // first assign k to n, then decrement k by 1
                // afterwards, k=6 but n=7
```

- It is also possible (but NOT recommended) to put the operator before the variable:

```
                // again, suppose k=7 initially
 n = --k;       // first decrement k by 1, then assign k to n
                // afterwards, k=6 and n=6
```

The `for` loop

There is also a construct called the `for` Loop:

```
for( expr1 ; expr2 ; expr3 ) {
  statements ;
}
```

- *expr1* is evaluated before the loop starts.
- *expr2* is evaluated at the beginning of each loop; if it is non-zero, the loop is repeated.
- *expr3* is evaluated at the end of each loop.

Example of `for` loop

```
for( x = 1; x <= 10; x++ ) {
   printf( "%d\n", x * x );
}
```

Questions:

1. what value will x have after the loop finishes?
2. can a `for` loop always be converted into a `while` loop?

`for` loops and `while` loops

These two are equivalent:

```
for( expr1 ; expr2 ; expr3 ) {
    statements ;
}
```

```
expr1 ;
while( expr2 ) {
    statements ;
    expr3 ;
}
```


Counting Down to Zero

Any of the 3 expressions in the `for` loop may be omitted, but the `';'` must still be present. For example:

```
printf("Enter starting number for Countdown: ");
scanf("%d", &n ); // initial value entered by user
for( ; n >= 0; n-- ) {
   printf("%d\n", n );
}
printf("Blast Off!\n");
```

