Operator Shorthand

C# Programming

Programming Tricks

- We are now moving into the parts of C# that are there just to make life easier for the programmer
- One thing that might be useful would be to make the program text simpler and shorter

Operator Shorthand

```
window_count = window_count + 1;
```

- There are certain things that programs do a lot
  - For example adding one to a variable to increment it
- C# provides “shorthand” ways to write these statements

++ and -- operators

```
window_count++;
```

- The ++ operator adds 1 to the contents of a variable
- There is also a – operator which will reduce a variable by 1
- Note that this is a “monadic” operator

More Operator Shorthand

```
house_cost = house_cost + window_cost
```

- Quite often we also want to add one value to another
- The C# language provides a quick way to do this as well

More Operator Shorthand

```
house_cost += window_cost;
```

- The += operator adds a value to another varaible
- There are -=, *= and /= operators too
- These make the program shorter, but I reckon they are a bit harder to understand

Statements and Values

```
i = (j=0);
```

- Every C# statement actually returns a value
- You can use this value in another statement
- The above code sets both `i` and `j` to `0`

The ++ operator and values

```
result = i++; // value of i before increment
result = ++i; // value of i after increment
```

- The value returned by a ++ statement depends on where the ++ is placed in relation to the variable is working on

Good Shorthand

- Shorthand is good because it makes your program smaller
- It can also make it execute more quickly as the compiler might be able to generate machine code that better matches the action to be performed
  - Many processors have an “increment memory” instruction

Shorthand Etiquette

- I don’t use shorthand much in my code
- This is because I place more emphasis on making sure the code is easy to understand than I do about anything else
- In my opinion the computer’s time is much less expensive than mine

Summary

- We now have the three fundamental loop constructions
- The trick with programming is to use the construction which is appropriate to the task in hand
- You can make the code work with any loop design