Lecture 3 (12/09/2014)

Operators and Expressions

Readings: Chapter 2 (less 2.10)

Arithmetic Operators

```
int main(void)
{
   int a = 13, b = 5;
   int c;

   c = a + b;
   c = a / b;
   c = a % b;
   c = -a;
   c = a++;
   c = ++a;

   return(0);
}
```

Further points

- Compound assignment operators

```
a += 2;
x *= y+1;
```

- Precedence

```
x = 5 + 2 * 7;
x = (5 + 2) * 7;
```

- Associativity

```
3 * 5 / 2
5 / 2 * 3
```

Problem

- Problem: Given the declarations of `m` and `k` as

```
float k = 10.0, m = 5.0;
int p = 3, q = 9;
```

What are the values of `k` and `q` after the following statements are executed?

```
k /= m + 1;
k += 1/m;
q -= ++p;
```

Type conversion and type casting

- What happens when an operator has operands of different types?

```
int x = 10;
double y = 6.5, z;

z = x + y;
```

- Operand of ‘lower’ type is ‘promoted’ to the ‘higher’ type
- Ranking of types (highest to lowest): `long double`, `double`, `float`, `long int`, `int`, `short int`, `char`.

Type conversion and type casting

- **Note:** It is possible to make type conversions *explicit* – this is done using the *type cast operator* (round brackets around type name)

```
int months;
double rent, rent_per_month;
rent = (double) months * rent_per_month;
```

- Type `char` is always promoted to `int` in expressions
- Conversion is via the ASCII character codes (check out http://www.ascii-code.com/)
- ‘a’ – ‘z’ correspond to 97-122 respectively
- ‘A’ – ‘Z’ correspond to 65-90 respectively
- ‘0’ – ‘9’ correspond to 48-57 respectively