Arithmetic Expressions

Math and Computer Science

SOUTH DAKOTA
SCHOOL OF MINES
& TECHNOLOGY

A Legacy of Excellence

Arithmetic Expressions

- (), ++, --
- ++, --
- *, /, %
- +, -
- =, *=, /=, %=, +=, -=

- Why are there two ++ and two --?
- Which one is the highest precedence?

Simple Expression

- X = 3 + 4 * 5;
  - X = 23?
  - X = 35?

- Use Parenthesis to override precedence
- X = (3 + 4) * 5;

Quadratic formula

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```
Root1 = (-b + sqrt( b * b – 4 * a * c ) )/( 2 * a );
Root2 = (-b - sqrt( b * b – 4 * a * c ) )/( 2 * a );
```

- Split it into separate variables

```
radical = sqrt( b * b – 4 * a * c);
denom = 2 * a;

Root1 = ( -b + radical ) / denom;
Root2 = ( -b – radical ) / denom;
```

Integer division

- No fractional portion to the answer
- int x, y = 9, z = 4;
- x = y / z;   // x has the value of 2, not 2.25
- Must typecast if you want the fractional part
- X = float(y) / z;

Typecasting

- Implicit – compiler does type casting to do the operations
  - Can only do operation on same types
  - Always promotes the smaller data type to the larger data type
    - Short int -> long int -> long long int -> float -> double -> long double
  - Only demotes on an assignment operator
    - int x, double z;
    - x = z / 4.6;   // z / 4.6 is a double and is demoted to a integer for the =

Typecasting

- The programmer typecasts it in the code.
- A = float (b) / double (z);
- Watch the warning the compiler gives you. They usually point out an errors.

```
int main()
{
    int x;
    float y = 8.3, z = 9.1;
    x = y * z;

    return 0;
}
```

-
  ```
  1>c:\users\arro_000\documents\visual studio 2012\projects\demo\source.cpp(14): warning
  C4244: '=' : conversion from 'float' to 'int', possible loss of data
  ```

Typecasting

- Fixed, no warnings

```
int main()
{
    int x;
    float y = 8.3, z = 9.1;
    x = int (y * z);

    return 0;
}
```