Introduction to Programming EC-105

Lecture 3

Operators

Operators

- “Operators are words or symbols that cause a program to do something to variables.”

OPERATOR TYPES:

Type	Operators	Usage
Arithmetic	‘+’ ‘-’ ‘*’ ‘/’ ‘%’	a+b a-b a*b a/b a%b
Arithmetic assignment	‘+=’ ‘-=’ ‘*=’ ‘/=’ ‘%=’	a+=b is the same as a=a+b a-=b a*=b a/=b a%=b
Increment and decrement	‘++’ ‘- -’	a++ is the same as a=a+1 a-- is the same as a=a-1
Relational	‘<’ ‘>’ ‘<=’ ‘>=’ ‘==’ ‘!=’	
Logical	‘&&’ ‘||’	

Manipulating Values

- Mathematical Operators

  - Common mathematical operators are available in C++ for manipulating values e.g. addition(+), subtraction(-), multiplication(*), division(/), and modulus (%).

- C++ has many other operators also which we will study in due course.

Arithmetic Expression Evaluation

- To evaluate an arithmetic expression two concepts needs to be understood
  - Operator Precedence
    - Operator precedence controls the order in which operations are performed
  - Operator Associativity
    - The associativity of an operator specifies the order in which operations of the same precedence are performed

Operator Precedence and Associativity

- Operators Precedence and Associativity for C++ is following

1. *, /, % → Do all multiplications, divisions and remainders from left to right.
2. +, - → Do additions and subtractions from left to right.

Evaluating an Expression

6 + 2 * 3 / 6

- Three operators are in this expression.
- However, * and / both have the same precedence and + has lower precedence then these two.
- * and / will be evaluated first but both have the same precedence level.
- Therefore, operator associatively will be used here to determine the first to get evaluated i.e. left to right.
- The left most sub expression will be evaluated followed by the next right one and so on.

- * will be evaluated first then /

Arithemetic Operator

Type	Operators	Usage
Arithmetic	‘+’ ‘-’ ‘*’ ‘/’ ‘%’	a+b a-b a*b a/b a%b

- The Modulus Operator
  - ***%*** is known as the Modulus Operator or the Remainder Operator.
  - It calculates the remainder of two variables
  - It can only be used with two ***ints..***

  - ***3%2 =1***
  - ***5%2=1***
  - ***6%3=0***
  - ***8%5=3***

Arithmetic Operator Precedence and associativity

Operator(s)	Operation(s)	Order of evaluation (precedence)
( )	Parentheses	Evaluated first. If the parentheses are nested, the expression in the innermost pair is evaluated first. If there are several pairs of parentheses "on the same level" (i.e., not nested), they are evaluated left to right.
* / %	Multiplication Division Modulus	Evaluated second. If there are several, they are evaluated left to right.
+ -	Addition Subtraction	Evaluated last. If there are several, they are evaluated left to right.

Polynomials in C++

- In algebra

$$m = \frac{a + b + c + d + e}{5}$$

- In c++

$$m = (a + b + c + d + e)/5$$

- What happens if I don't put the parenthesis ()?

Evaluate

In what order will the expression be evaluated?

Algebra: $z = pr\%q + w/x - y$

C++:

```
z = p * r % q + w / x - y;
```

Evaluate

In what order will the expression be evaluated?

Algebra: $z = pr\%q + w/x - y$

C++:

```
z = p * r % q + w / x - y;
```

6 1 2 4 3 5

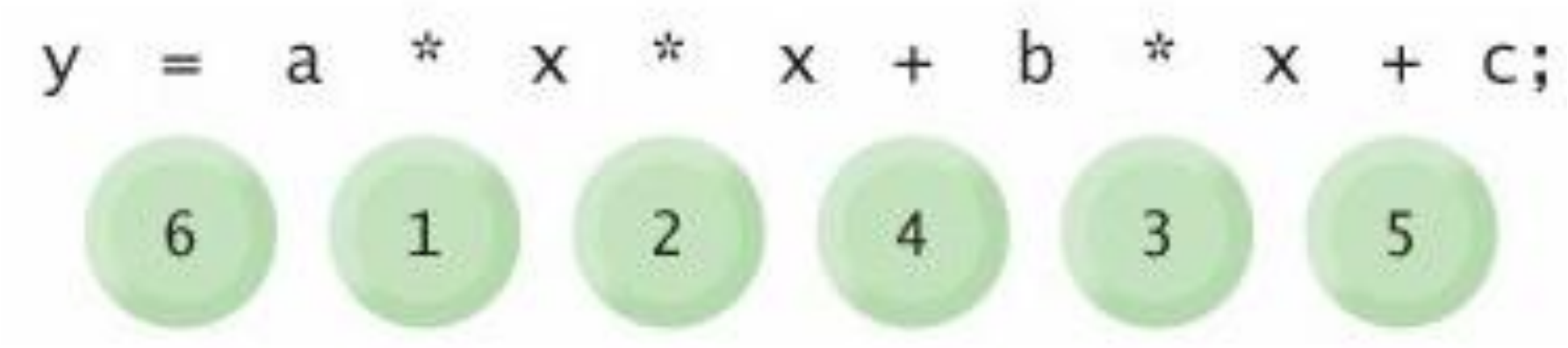
2nd Degree Polynomial

$$y = ax^2 + bx + c$$

```
y = a * x * x + b * x + c;
```

2^{nd} Degree Polynomial

$$y = ax^2 + bx + c$$



Arithmetic Assignment Operators

Type	Operators	Usage
Arithmetic assignment	‘+=’ ‘-=’ ‘*=’ ‘/=’ ‘%=’	a+=b is the same as a=a+b a-=b a*=b a/=b a%=b

Relational and Equality Operators

Standard algebraic equality or relational operator	C++ equality or relational operator	Sample C++ condition	Meaning of C++ condition
Relational operators			
>	>	x > y	x is greater than y
<	<	x < y	x is less than y
≥	>=	x >= y	x is greater than or equal to y
≤	<=	x <= y	x is less than or equal to y
Equality operators			
=	==	x == y	x is equal to y
≠	!=	x != y	x is not equal to y

(Deitel and Deitel (5th Ed), fig 2.12)

Logical Operators

Type	Operators	Usage
Logical	‘&&’ ‘||’	

- Logical operators are carried out on statements, e.g. statement1 && statement 2, etc.
- Logical AND (&&)
  - false && false= false
  - false && true = false
  - true && false= false
  - true && true = true
- Logical OR (||)

Logical Operators

- Logical OR (||)
  - false || false = false
  - false || true = true
  - true || false = true
  - true || true = true
- Logical NOT (!)
  - !false = true
  - !true = false

Unary Increment and Decrement Operators

Type	Operators	Usage
Increment and decrement	‘++’ ‘- -’	a++ is the same as a=a+1 a-- is the same as a=a-1

- Prefix
  - ++a;
  - --a;
- Postfix
  - a++;
  - a--;

Precedence of Operators

OPERATORS	**TYPE**
++ --	unary
* / %	multiplicative
+ -	additive
< > <= >=	relational
== !=	equality
&& ||	logical
= *= /= %= += -=	assignment

Home Task
Precedence of Operators

- a += b-- + ++d * c % e / f

- Consider all variables to be equal to 2 and calculate the answer
- a = ?