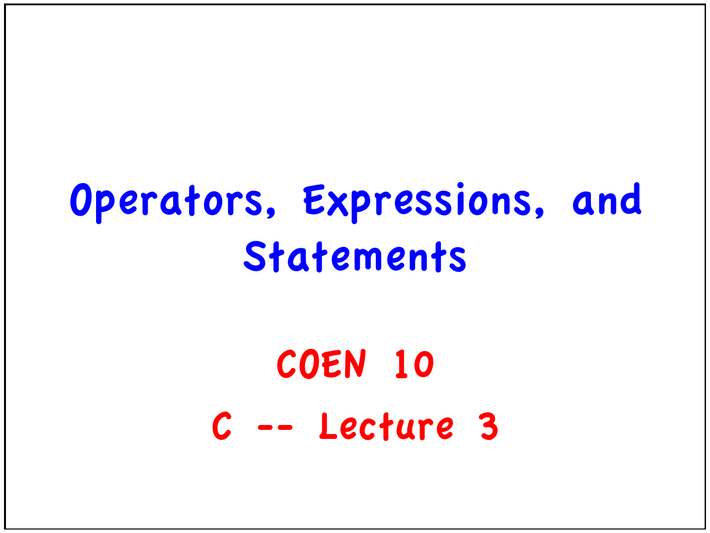
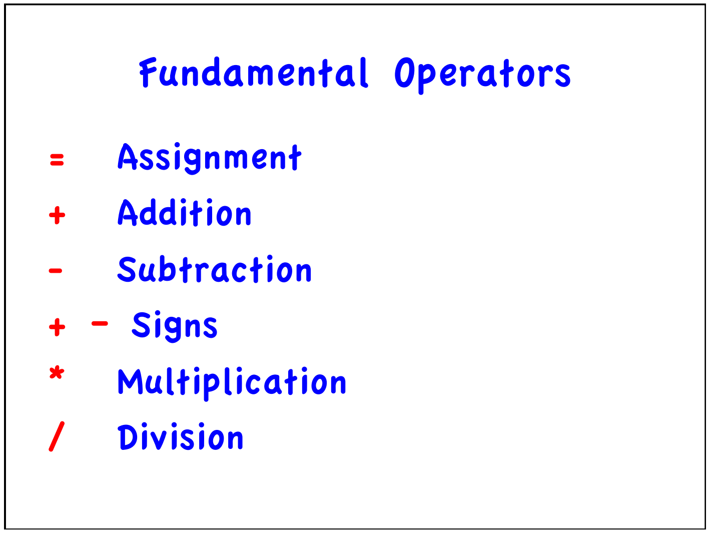


Operators, Expressions, and Statements

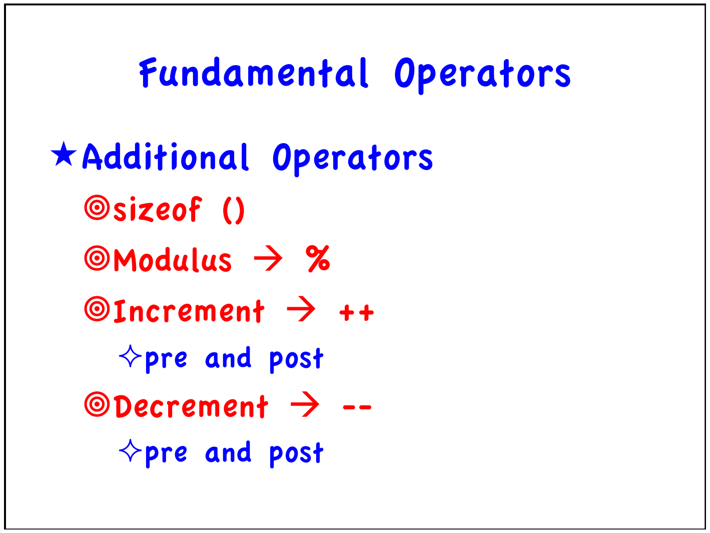
COEN 10

C -- Lecture 3



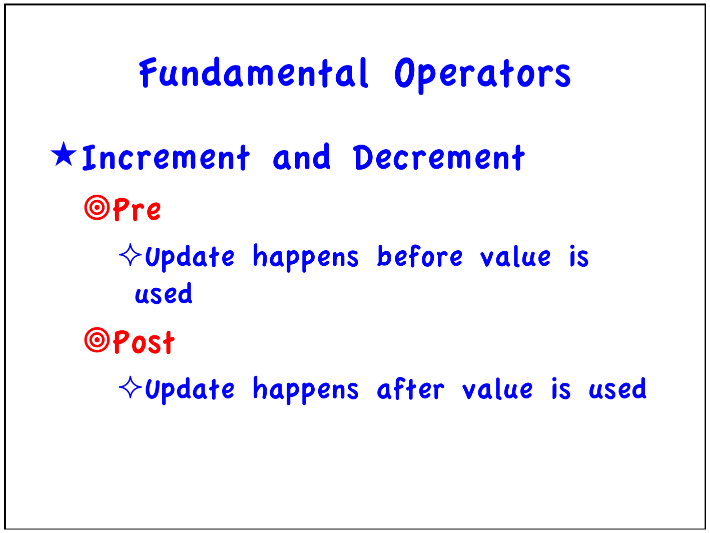
Fundamental Operators

=	Assignment
+	Addition
-	Subtraction
+ -	Signs
*	Multiplication
/	Division



Fundamental Operators

- ★Additional Operators
  - ◎sizeof ()
  - ◎Modulus → %
  - ◎Increment → ++
    - ✧pre and post
  - ◎Decrement → --
    - ✧pre and post



Fundamental Operators

- ★Increment and Decrement
  - ◎Pre
    - ✧Update happens before value is used
  - ◎Post
    - ✧Update happens after value is used

Fundamental Operators

★Precedence Rules

1.	++,-- postfix ( )	→ left to right
2.	++,-- prefix + - sizeof	→ right to left
3.	* / %	→ left to right
4.	+ - (binary)	→ left to right
5.	=	→ right to left

Fundamental Operators

★Increment and decrement very useful in loops

- ◎Example: for loops

```
for (i = 0; i < value; i++)
{
     …
}
```

Fundamental Operators

★Increment and decrement very useful in loops

- ◎Example: while loops

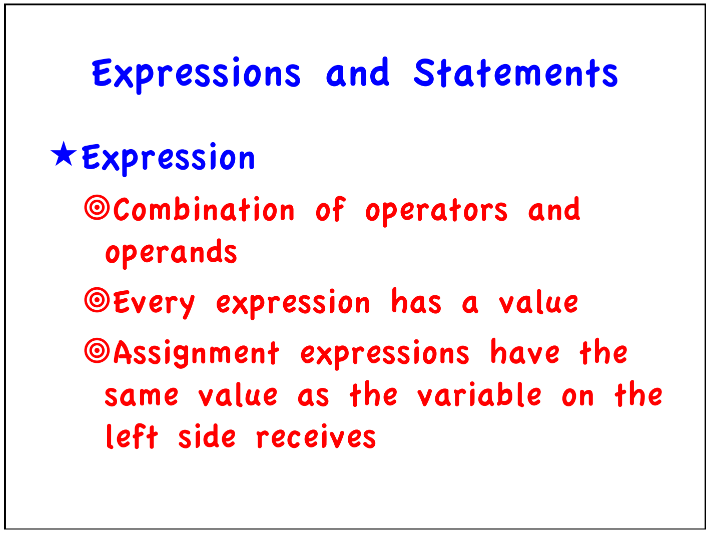
```
i = 0;
while (++i < value)
{
     …
}
```

Fundamental Operators

★Increment and Decrement

- ◎Avoid them in expressions!!
- ◎Example: while loops

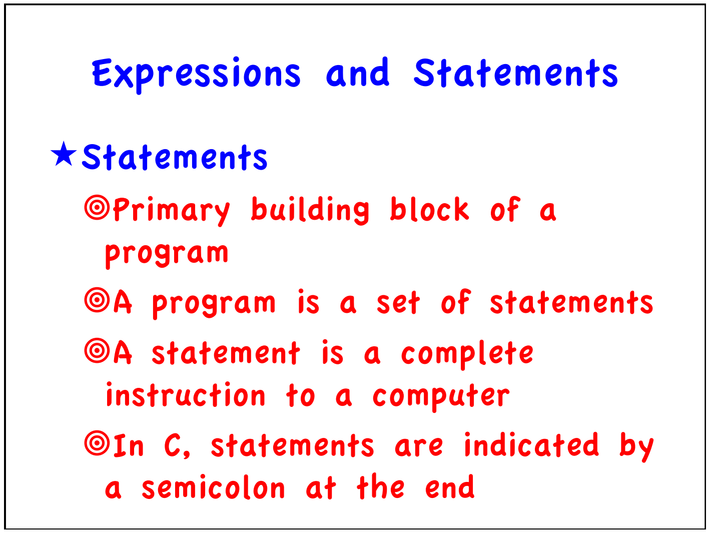
```
i = 0;
while (i < value)
{
     …
     i++;
}
```



Expressions and Statements

★Expression

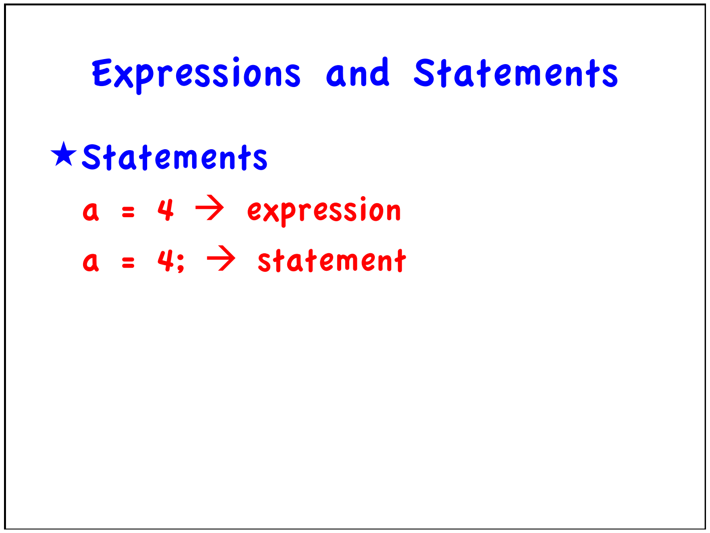
- ◎Combination of operators and operands
- ◎Every expression has a value
- ◎Assignment expressions have the same value as the variable on the left side receives



Expressions and Statements

★Statements

- ◎Primary building block of a program
- ◎A program is a set of statements
- ◎A statement is a complete instruction to a computer
- ◎In C, statements are indicated by a semicolon at the end

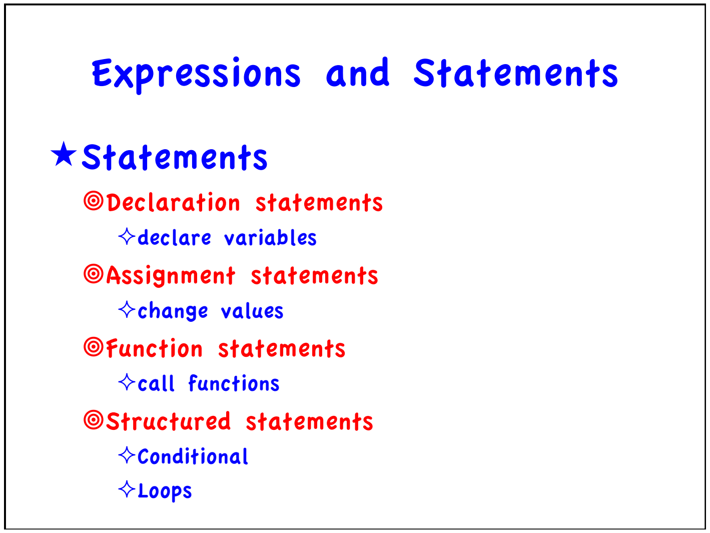


Expressions and Statements

★Statements

a = 4 → expression

a = 4; → statement



Expressions and Statements

★Statements

- ◎Declaration statements
  - ✧declare variables
- ◎Assignment statements
  - ✧change values
- ◎Function statements
  - ✧call functions
- ◎Structured statements
  - ✧Conditional
  - ✧Loops

Expressions and Statements

- Side Effects and Sequence Points
  - Side Effect
    - Modification of a data object or file

Expressions and Statements

- Side Effects and Sequence Points
  - Sequence Point
    - Point in a program execution at which all side effects are evaluated before going to the next step
    - Example: semicolon

Expressions and Statements

- Side Effects and Sequence Points
  - Full Expression
    - One that is not part of another one
    - Examples:
      - Expression in an assignment statement
      - Expression in a test condition

Expressions and Statements

- Compound Statements
  - Two or more statements grouped together and enclosed with braces
  - Also called a block

Type Conversions

★Types in expressions should match

★When they don't

◎Compiler will use some rules to adapt

Type Conversions

★Rules

1. Covert to the largest type, promotion
2. Convert to the higher ranking
3. Ranking: H→L
   - ✧ Long double, double, float, unsigned long long, long long, unsigned long, long, unsigned int, int
4. In an assignment, type of calculation is converted to the type of the variable being assigned, promotion or demotion
5. When passed as a function argument, char and short to int, and float to double, promotion

Type Conversions

★Rules

◎Promotion – simple process

◎Demotion – complicated because data may be lost

Type Conversions

★The Cast Operator

◎When necessary, demand the precise type conversion with a cast

◎Operator: (type)

◎Example

```
int x = 1.6 + 1.7;              → 3
int x = (int)1.6 + (int)1.7;    → 2
```