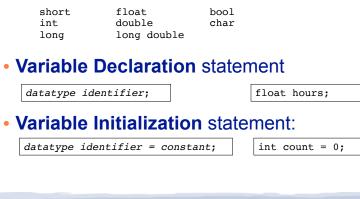
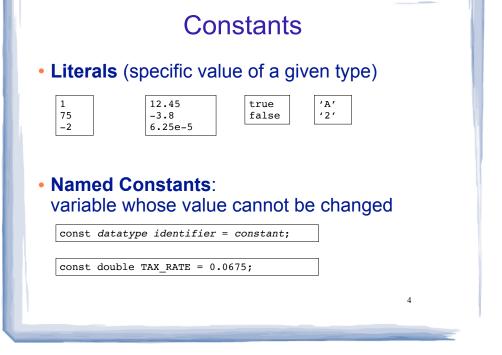
#### Structure of a C++ Program Basic C++ • Hello world: (What you should already know) //This program outputs a message to the screen #include <iostream> Chapters 1-5 using namespace std; int main() { cout << "Hello world!" << endl;</pre> CS 2308 In general: Fall 2018 //This is a comment #include <includefile> ... **Jill Seaman** using namespace std; int main() { statements ... 1

3

# Variables, Data Types

- Variable: portion of memory that stores a value
- Identifier: name of a program element
- Fundamental data types





2

### Assignment statement, expressions

• To change the value of a variable:

variable = expression;

count = 10;

- \* The lefthand side must be a variable
- \* The righthand side is an *expression* of the right type
- What is an expression?
  - \* an expression has a type and evaluates to a value
    - literal
    - named constant
    - variable
    - arithmetic expression
    - etc.

# Arithmetic and Relational Operations

#### • arithmetic operators:

- + addition
- subtractionmultiplication
- / division

==

!=

>

<

% modulo

		2			
/	6	Z			
8	+	5	*	10	

x + 10

#### Watchout: Integer division!!

• relational operators (result is bool):

Equal to	
Not equal to	
Greater than	
Less than	
Greater than or e	0

- >= Greater than or equal to
- <= Less than or equal to

7 < 25 89 == xx % 2 != 0 8 + 5 \* 10 <=100 \* n

## Logical Operations, precedence

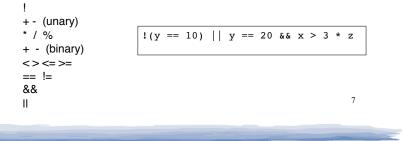
logical operators (values and results are bool):

! not && and II or

[	x < 10 && x > 0
	x < 10 && x > 0 y == 10    y == 20 !(a == b)
	! (a == b)

5

#### • operator precedence (which happens first?):



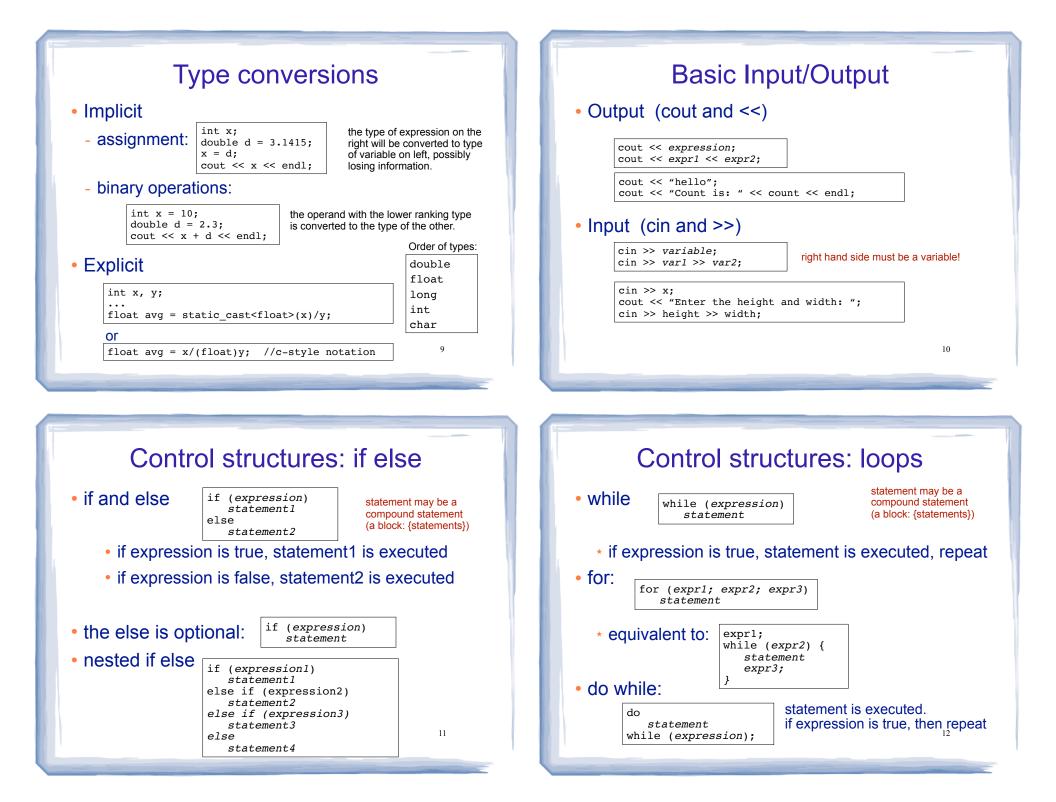
### More assignment statements

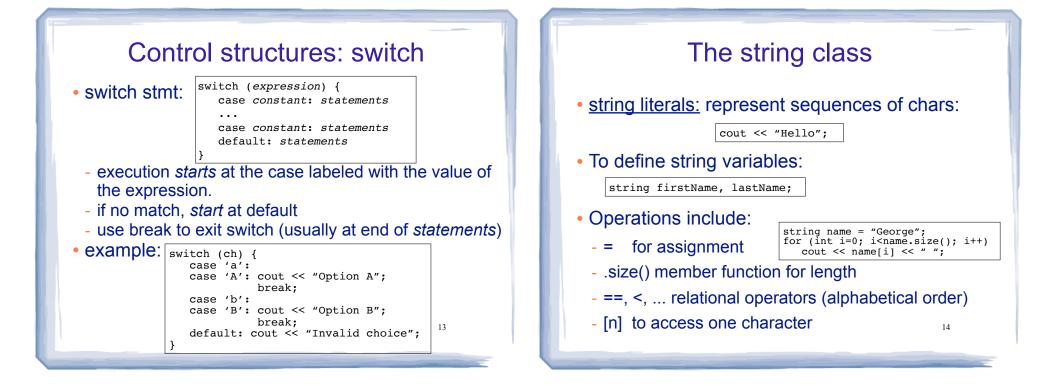
#### Compound assignment

operator	usage	equivalent syntax:
+=	x += e;	x = x + e;
-=	x -= e;	x = x - e;
*=	x *= e;	x = x * e;
/=	x /= e;	x = x / e;

#### increment, decrement

operator	usage		equivalent syntax:
++	x++;	++x;	x = x + 1;
	x;	x;	x = x - 1;





# File Input/Output

- #include <fstream>
- Output (ofstream)

```
ofstream fout;
fout.open("filename.txt");
fout << "hello";
fout << "Count is: " << count << endl;
fout.close();
```

```
    Input (ifstream)
```

```
ifstream fin;
fin.open("data.txt");
if (!fin) { Check for file open errors
    cout << "error opening file" << endl;
    return (0);
}
int x;
fin >> x; right hand side must be a variable!
cout << "x is " << x << endl;
fin.close();
```