Foundation of Computer Science
Relational Operators and if

Dr Kafi Rahman, PhD
Truman State University

Date: 5 Feb 2020

Agendas

- Relational Operators
- Relational Data Type
- Assertions
- Boolean Data Type
- Comparing Double Values
- if condition structure
- Group Discussion

Relational operators

Operator	Meaning
>	greater than
<	less then
>=	greater than or equal to
<=	less than or equal to
==	equal to
!=	not equal to

- there is another whole family of operators to understand relation between two values
- they allow us to compare one value to another (please see table 4-1 on page 152)

Relational operators (cont.)

- We add two integer numbers
  - foo = 10 + 5; the assigned value would be of integer type.

- When we compare two integer numbers
  - bar = 5 < 15; we get a boolean value
    - above statement 5 < 15 is an assertion
    - this assertion can either be true or false

Relational operators: Assertions

- the relational operators are used to make assertions that evaluate to either true or false
  - $5 < 10$ is a true assertion
  - $10 < 5$ is a false assertion

Relational operators (cont.)

- x < y, example 5<10
  - Testing whether the value of x is less than y
- x > y, example 10>5
  - Testing whether the value of x is greater than y
- x >= y, example 10>=10
  - Testing whether the value of x is greater than equal to y
- x <=y, example 2<5
  - Testing whether the value of x is less than equal to y
- x == y, example 5==5
  - Testing whether the value of x is equal to y
- x != y, example 5!=7
  - Testing whether the value of x is NOT equal to y

Relational operators (cont.)

- 2 < 5 is True
- 10 > 7 is True
- 50 == 55 is False
- 50 != 55 is True
- 10 >= 20 is False
- 10 <= 20 is True
- (5+2) > 3 is True

Relational operators: Boolean Data Type

- We can store the result in a Boolean variable
  - In C++ a Boolean Data Type is called: bool

```
int x = 10;
int y = 7;

bool foo = x < 5;
// foo is now false

bool bar = x >= y;
// bar is now true
```

Displaying Boolean Values

- cout displays Boolean values as 0 or 1 for printing true or false respectively

- We can use the boolalpha io manipulator (sticky) to print the actual Boolean values if needed

Comparing Floating Values

```
int main() {
    double a = 1.5;
    double b = a + 0.0000000000000001;

    bool equal = (a == b);
    cout << boolalpha << equal << endl;
}
```

- We must take caution when comparing between two float/double values

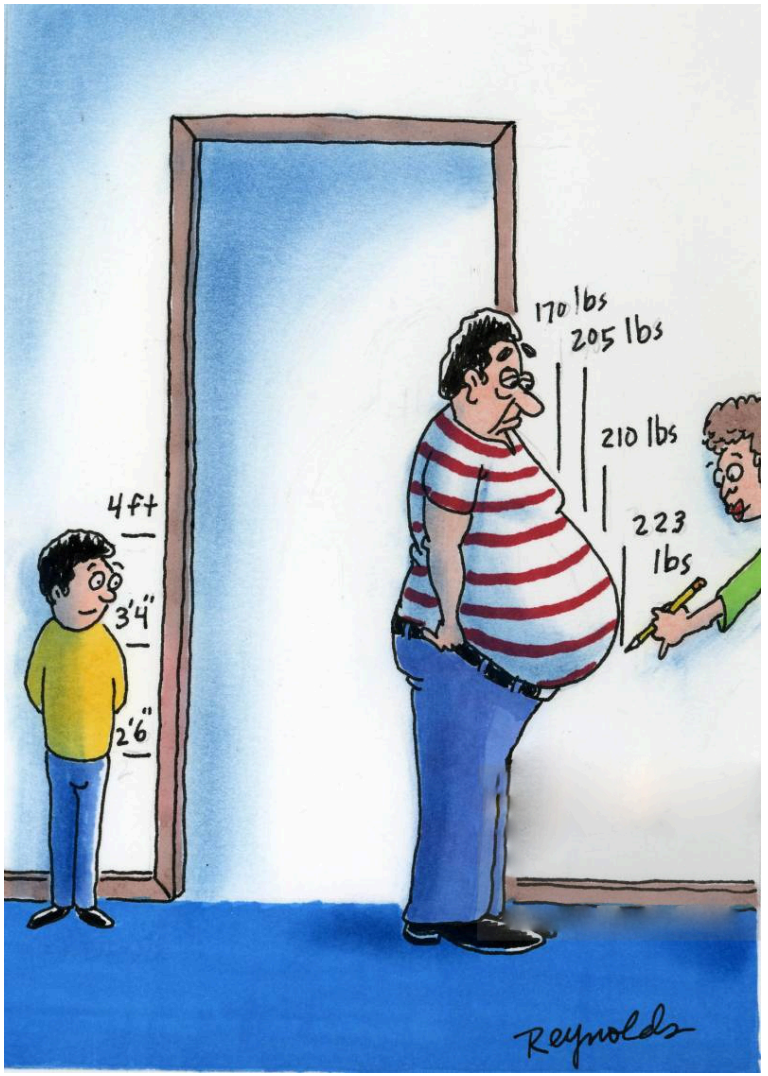
Comparing Floating Values (cont.)

```
int main() {
    const double EPSILON = 0.001;
    double foo = 3.9535;
    double bar = 3.9532;
    bool equal = abs(foo - bar) <= EPSILON;

    cout << boolalpha << equal << endl;
}
```

- for each situation, we decide an appropriate EPSILON

What is abs()?

```
// The abs() function returns
// the absolute value of x
// i.e. |x|

int a = abs(-50);
// a is 50

a = abs(100-50);
// a is 50

a = abs(50-100);
// a is 50
```



Program Flow

- all of the programs we have seen so far are sequential
- they start with the first statement after the opening curly brace of main
- and end with the return 0; just before the closing curly brace of main
- one statement follows another, without exception

Conditional Execution

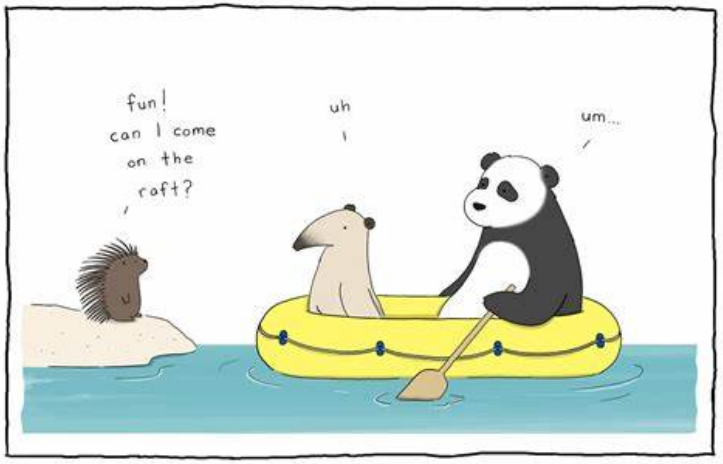
- We can use relational operators to create conditional execution of statements

- We can choose whether a set of statements are executed or not depending on the result of some relational operations.

Decision Making

- If its raining outside then:
  - Watch a movie

- To make some decisions in programming, we use if statements



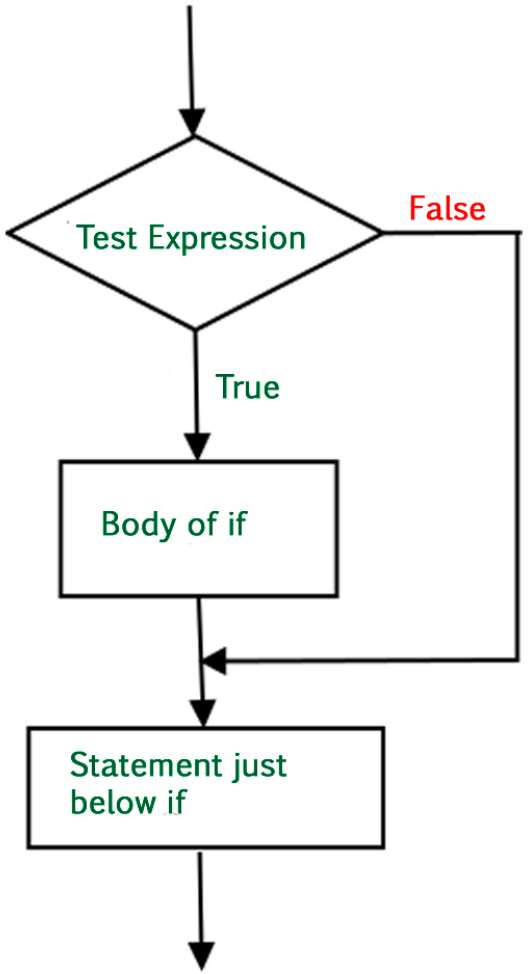
If Statement Structure

- if is a reserved word
- expression is a Boolean condition using relational operators
- multiple statements (one or more) form the body of the if statement;
- there is no semicolon after the closing brace — the closing brace itself ends the if statement

```
if (expression) {

    statement;

    statement;

    …
}
```

Decision Making (cont.)

```
int main() {
int foo = 10;
cout<<"Before the if”<<endl;

if(foo == 10){
  cout<<"foo equal to 10"<<endl;
  cout<<"inside the if”<<endl;
}

cout<< "After the if"<<endl;
return 0;
}
```



Decision Making (cont.)

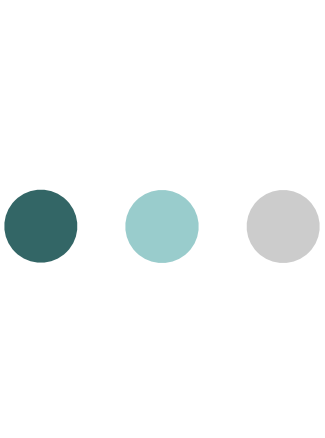
- Let us examine another program:

```
int main()
{     float weight = 0.0f;
      cout<<"Enter the weight in pounds: ";
      cin>>weight;

      if (weight > 50){
            cout<<"Luggage is more than 50 lbs"<<endl;
            cout<<"Please pay $25 extra"<<endl;
      }
      cout<<"Thank you for your business.";
      return 0;
}
```

Common Errors in if

```
if (a = b);
  foo=10 + 5;
  bar = 100 - 25;
```

- Semicolon after the if condition will end the if-statement
- Its always recommenced to use the opening and closing braces to enclose the body of the if-statement



Questions?