

CSC 142

Conditionals [Reading: chapter 7]

CSC 142 G 1

Syntax

```

if (condition) ← must be a boolean. It is an
                error to omit ( and )
{
    statement1;
    statement2; } ← executed if
                    condition is true
}
else /* can be omitted if there is no need for an
     else case */
{
    statement3;
    statement4; } ← executed if
                    condition is false
}

```

CSC 142 G 2

Example

- Braces are optional if there is only one statement in the block, e.g.

```

if (temperature <= 98.6)
    System.out.println("I am feeling great");
else
    System.out.println("I am not feeling good");
    callDoctor();
}

```
- A common pitfall

```

if (snowHeight >= 10);
    System.out.println("Go skiing");
// Go skiing is always printed
// System.out.println is not part of the if
// the if statement ends at the ;

```

don't write a ; here

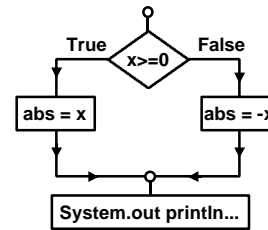
CSC 142 G 3

Flowchart

```

if (x >= 0)
    abs = x;
else
    abs = -x;
System.out.println("absolute value="+abs);

```



CSC 142 G 4

Writing conditions

- Relational operators: <=, >=, >, <, == and !=
- Boolean operators
 - && for AND
 - || for OR
 - ! for NOT
- Recall the logical table

P	Q	!P	P && Q	P Q
T	T	F	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	F

P, Q are conditions
T and F stand for True and False

CSC 142 G 5

Examples of conditions

- age (an **int** variable) is greater than 10 and less than 20

```

age > 10 && age < 20 // Don't write 10 < age < 20

```
- answer (a **char** variable) is not equal to 'y' and not equal to 'n'

```

answer != 'y' && answer != 'n'
!(answer == 'y' || answer == 'n') // also OK

```
- How would you test that year is a leap year? (a year is a leap year if it can be divided by 4 and not by 100 or if it can be divided by 400)

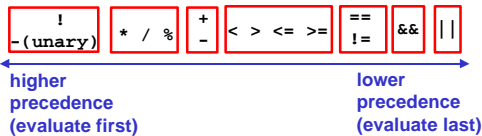
```

if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0)
    System.out.println(year + " is a leap year");

```

CSC 142 G 6

Precedence order



```
int a = 3;
int b = 5;
char c = 'n';
boolean z;
z = a+b!=6 && !(c!='y') || b%a>b/a+1;
// What is z?
```

- z is false

CSC 142 G 7

Order of evaluation

- && and || associate left to right
- condition1 && condition2
If condition1 is false, condition2 is not evaluated. No need to since condition1 && condition2 is false whether condition2 is true or false.
- condition1 || condition2
If condition1 is true, condition2 is not evaluated since condition1 || condition2 is true whether condition2 is true or false.

```
//city is a String variable. The 2 conditions
// below never give an error when city is null
if (city!=null && city.length(>)>3)...
if (city==null || city.length(>)>3)...
// But (city.length(>)>3 && city!=null) gives
// an error if city is null
```

CSC 142 G 8

De Morgan's laws

- $!(P \ \&\& \ Q)$ is equivalent to $!P \ || \ !Q$
 - $!(P \ || \ Q)$ is equivalent to $!P \ \&\& \ !Q$
- ```
!(NumberOfChildren>3 && income<=25000.)
// same as
!(NumberOfChildren>3) || !(income<=25000.)
// which can written
NumberOfChildren<=3 || income>25000.
```

- Proof

| P | Q | P&&Q | !(P&&Q) | !P | !Q | !P  !Q |
|---|---|------|---------|----|----|--------|
| T | T | T    | F       | F  | F  | F      |
| T | F | F    | T       | F  | T  | T      |
| F | T | F    | T       | T  | F  | T      |
| F | F | F    | T       | T  | T  | T      |

P, Q are conditions  
T and F stand for True and False

CSC 142 G 9

## Nested ifs

- Given an age, print the age category

```
// One way
if (age < 13)
 System.out.println("Child");
if (age >=13 && age <20)
 System.out.println("Teenager");
// etc...
```

conditions are mutually exclusive

- Better: nested ifs

```
if (age < 13)
 System.out.println("Child");
else if (age<20)
 System.out.println("Teenager");
else if (age<35)
 System.out.println("Young adult");
else if (age<60)
 System.out.println("Middle aged");
else
 System.out.println("Senior");
```

conditions are mutually inclusive. Order counts!

CSC 142 G 10

## switch statement

- Simulate an ATM

```
switch(userChoice){ // userChoice is a char variable
 case 'c': // make the choice case insensitive
 case 'C':
 System.out.println("Checking account"); if userChoice is 's',
 break; // execution in the switch block begins here and ends here
 case 's': // make the choice case insensitive
 case 'S':
 System.out.println("Savings account");
 break;
 default:
 System.out.println("Unavailable choice");
 break;
}
```

CSC 142 G 11

## switch syntax

```
switch(expression) {
 case constant1:
 case constant2:
 statement;
 break;
 case constant3:
 statement;
 break;
 default:
 statement;
 break;
}
```

must be of type int or char  
must be a constant of type int or char  
Don't forget the break statement. If we omitted the break statement at this location, the default case would be executed when expression is equal to constant3

CSC 142 G 12

## A shortcut: ?:

- Code to compute an absolute value

```
if (x>=0)
 abs=x;
else
 abs=-x;
```

- It can be rewritten as

```
abs = (x>=0)?x:-x;
```

- ?: is a ternary operator (it takes 3 operands)

```
(condition)?expression1:expression2
is expression1 if condition is true
is expression2 if condition is false
```

CSC 142 G 13

## Comparing references to objects

- Consider

```
Integer i1 = new Integer(4);
Integer i2 = new Integer(4);
if (i1==i2) System.out.println("i1==i2");
// Is i1==i2 printed?
```

- No! `i1==i2` tests if `i1` and `i2` are references to the **same** object

- To test equality of two objects, the class must provide a method to do so.

```
if (i1.equals(i2))
 System.out.println("i1 is equal to i2");
```

- Don't use `==`, `!=` to compare objects (except when comparing to `null`). Check the class documentation.

CSC 142 G 14