CSE 142

Conditional Statements & Boolean Expressions

Outline for Today

- Conditional statements – if
- Boolean expressions
  - Comparisons (<, <=, >, >=, !=, ==)
  - Boolean operators (and, or, not - &&, ||, !)
- Class invariants

Withdraw Method for BankAccount

- In the version demonstrated in lecture, we added a withdraw method and used it to implement transferFrom

```
/** Withdraw the requested amount from this BankAccount */
public void withdraw(double amount) {
    balance = balance + amount;
}
```

- Critique: is this good/bad/incomplete?

Class Invariants

- In many class definitions, the state must obey some rules to have a sensible value
  - For a BankAccount, one rule might be that balance >= 0.0 always, or that the account must have a non-empty name
- These rules are called ***class invariants***
  - Things that must always be true
- Write these down in a comment
- When you implement methods, double check that you never violate these rules
  - Very powerful bug prevention technique

A Better Withdraw Method

- Specification

```
/** Withdraw requested amount from this BankAccount provided that the
 *  balance is at least as large as the amount requested.  Otherwise do nothing */
public void withdraw(double amount) {
}
```

- How do we implement this?
  - We want to say something like

    "if the amount is less than or equal to the balance, withdraw the amount"
- Java solution: if statement

Withdraw Method Implementation

```
/** Withdraw requested amount from this BankAccount provided that the
 *  balance is at least as large as the amount requested.  Otherwise do nothing */
public void withdraw(double amount) {
    if (amount <= balance) {
      balance = balance – amount;
    }
}
```

If Statement

- **Syntax**

```
if ( condition ) {
    list of statements
}
```

or

```
if ( condition ) {
    list1 of statements
} else {
    list2 of statements
}
```

- ***condition* must be a Boolean expression – one that is either true or false**
- ***list of statements* may contain any Java statements, including if(!)**

- **Meaning of first form**
  - **Evaluate condition**
  - **If the condition is true, execute the list of statements**
  - **If it is false, do nothing (skip statements)**
- **Meaning of second form**
  - **Evaluate condition**
  - **If the condition is true, execute the first list of statements and skip the second one**
  - **If the condition is false, skip the first list of statements and execute the second one**

Better Withdraw Method

- **Instead of silently doing nothing if amount is too large, return a Boolean result to indicate if the withdraw succeeded**

```
/** Withdraw requested amount from this BankAccount and return true, provided
  * that the balance is at least as large as the amount requested.  Otherwise
  * return false */
public boolean withdraw(double amount) {
    if (amount <= balance) {
      balance = balance – amount;
      return true;
    } else {
      return false;
    }
}
```


Boolean Expressions

- **Constants**

  true
  false

- **Simple relations (comparisons)**

  > >= < <= != ==

  - **Note use of == for equality comparison (not!!! single =)**
  - **Examples**

    x > y
    x*2.5 - 17.0 <= 0.0
    balance >= amount

Boolean Operators

- **Combine simpler Boolean expressions**
- **And – true if both expressions are true, false otherwise**

  x > 10 && x <= 100

  - Can only compare two things at a time; can't do 10 < x <= 100
- **Or – true if either expression is true, false if both are false**

  x > y || x <= 0

- **Not – true if expression is false**

  ! (x < y) // means same thing as x >= y

Exercise

- **Recall that the statement**

  System.out.println("Hi there!");

  will write a message (in this case, "Hi there!")
- **Exercise 1: assume that we have a double variable called temperature holding the outside temperature. Write the message "Too Hot!" if the temperature is above 80.**
- **Exercise 2: use the variable temperature as above, but this time write "Too Hot!" if the temperature is above 80, "Too Cold!" if it is below 60, and "Just Right" if it is in between.**

Solution to Exercise 1

Solution to Exercise 2

Summary

- Conditional execution – if statement
- Boolean expressions
  - Comparisons
  - Operators – and, or, not