Lecture 5
Fourth cup of JAVA: Problem Solving

Winston & Narasimhan: Chapt 21 - 26

JAVA Cup 4: Problem Solving

• Boolean operators and methods
• Conditional statements
• Combining Boolean expressions
• Iteration Statements
• Recursion
• Multi-way Conditions

Booleans

EXAMPLE:

```java
public class Demonstrate {
    public static void main (String argv[]) {
        System.out.println(28==2);
        Gringotts js = new Gringotts();
        MuggleBank js2 = new MuggleBank();
        System.out.println(js instanceof Gringotts);
        System.out.println(js instanceof Account);
        System.out.println(js2.equals(js));
    }
}
```
Predicates Example 2

```java
public class Demonstrate1 {
    public static void main (String argv[]) {
        TTM ttm = new TTM();
        Gringotts hp = new Gringotts();
        System.out.println(ttm instanceof Account);
        System.out.println(ttm instanceof Gringotts);
        System.out.println(ttm instanceof TTM);
        System.out.println(hp instanceof Account);
        System.out.println(hp instanceof Gringotts);
        System.out.println(hp instanceof TTM);
    }
}
```

Predicates Example 3

```java
public class Demonstrate {
    public static void main (String argv[]) {
        Gringotts hp = new Gringotts(3,4,5);
        Gringotts rw = new Gringotts(3,4,5);
        Gringotts hg = new Gringotts(4,5,6);
        System.out.println(hp.equals(rw));
        System.out.println(rw.equals(hg));
        hg=rw=hp;
        System.out.println(hp.equals(rw));
        System.out.println(rw.equals(hg));
    }
}
```

Conditional Statements

- The usual if-else statements
- Each "else" belongs to nearest unmatched "if"
- Embedding allowed

```java
if (Boolean expression)
    { ... }
else
    { ... }
```
Conditional Statement Example 1

```java
class Demonstrate {
    public static void main (String argv[]) {
        int s = 5;
        if (s == 1) {
            System.out.println("Harry has " + s + " sickle");
        } else {
            System.out.println("Harry has " + s + " sickles");
        }
    }
}
```

Conditional Statement Exercise

Quidditch is the premier sport of the wizarding world. It is played on broomsticks. Everyone follows Quidditch. The World Cup matches attract hundreds of thousands of fans from all over the world.

Design a class called Quidditch, where each instance contains information about the competing teams and their scores. Write a method to display the state of the game (i.e. who is winning).

The Conditional Operator

- Boolean expression ? if-true expression : if-false expression

```java
class Demonstrate {
    public static void main (String argv[]) {
        int s = 5;
        System.out.println("HP has " + s);
        System.out.println(s == 1 ? "sickle." : "sickles.");
    }
}
```
Combining Boolean Expressions

- $$\&\&$$ = the AND operator
- $$||$$ = the OR operator
- Evaluations from Left to Right

Iteration Statements

- ```
  while ( Boolean expression ) {
    ...
  }
```  
- ```
  for ( entry expression; Boolean expression;
      continuation expression ) {
    ...
  }
```  
- ```
  for ( counter initialization expression;
      counter testing expression; counter
      reassignment expression ) {
    ...
  }
```  

Examples

```java
public static int powerOf2 (int n) {
    int result = 1;
    while (n != 0) {
        result = 2 * result;
        n = n - 1;
    }
    return result;
}
```

```java
public static int powerOf2 (int n) {
    int result = 1;
    while (n != 0) {
        result = 2 * result;
        n = n - 1;
    }
    return result;
}
```
Augmented Assignment Operator

• Short-cut
  – `<variable> = <variable> <operator> <expr>
  – `<variable> <operator> = <expr> e.g. `x*=2

• Increment / Decrement short-cuts
  – Prefix (`-x, `++x): hands over new value
  – Suffix (`x-, `x++): hands over original value

QUIZ: what is the difference between `++x + x and (x++) + x?

More For-Loop Examples

```java
public static int powerOf2 (int n) {
  int result = 1;
  for (int i = n; i != 0; --i) {
    result *= 2;
  }
  return result;
}
```

Breaking the Loop!

```java
public static int powerOf2 (int n) {
  int result = 1;
  for (int i = n; i != 0; i = i-1) {
    if (i==0) {
      break;
    }
    result = 2 * result;
  }
  return result;
}
```
Recursion

• Method occurs in its own definition
• New storage is allocated with each call

```java
public static int recursivePowerOf2 (int n) {
    if (n == 0) {return 1;}— BASE part
    else { return 2*recursivePowerOf2(n-1); }
} RECURSIVE part
```

DIRECT RECURSION
```java
public static int rabbits (int n) {
    if (n == 0 || n == 1) (return 1;)
    else { return rabbits(n-1) + rabbits(n-2); }
}
```

INDIRECT RECURSION
```java
public static int rabbits (int n) {
    if (n == 0 || n == 1) (return 1;)
    else { return previous(n) + penultimate(n); }
    public static int previous (int n) { return rabbits(n-1); }
    public static int penultimate (int n) { return rabbits(n-2); }
```

Some like it Recursive

• Some prefer recursions
  – Inherent elegance
• Some prefer iterative definitions
  – Can you guess why?
Multiway Conditional Statements

- Keywords: switch, case, default
- Statements terminated by break or return, otherwise statement falls through to next set of statements.

```java
public static int rabbits (int n) {
    switch (n) {
    case 0: return 1;
    case 1: return 1;
    default: return rabbits(n-1) + rabbits(n-2);
    }
}
```

EXERCISE

- You have been hired by Gringotts to write a program that counts the number of possible ways to make change (in coins) for a given amount of money. Ultimately, you want a method that takes two arguments:
  - number of sickles, # of types of coins allowed.
  - E.g. countChange(175, 3) would ask how many ways there are make change for a 175 knuts, using sickles, knuts and galleons.
- Design an algorithm and Implement it.