The Coin Class

- Let's consider a class that represents a coin that can be flipped
- Instance data is used to indicate which face (heads or tails) is currently showing
- Methods are for constructing, flipping, getting information of a coin
- Ref: CoinFlip.java (driver), Coin.java

Coin.java

```java
public class Coin {
    private final int HEADS = 0;
    private final int TAILS = 1;
    private int face;

    public Coin () {   flip();
    }

    public void flip () {  face = (int) (Math.random() * 2);
    }
}
```
Coin.java (cont.)

// Returns true if the current face of the coin is heads.
public boolean isHeads ()
{
    return (face == HEADS);
}

// Returns the current face of the coin as a string.
public String toString()
{
    String faceName;
    if (face == HEADS)
        faceName = "Heads";
    else
        faceName = "Tails";
    return faceName;
}

CoinFlip.java

// Demonstrates the use of an if-else statement.
public class CoinFlip
{
    // Creates a Coin object, flips it, and prints the results.
    public static void main (String[] args)
    {
        Coin myCoin = new Coin();
        myCoin.flip();
        System.out.println (myCoin);
        if (myCoin.isHeads())
            System.out.println ("You win.");
        else
            System.out.println ("Better luck next time.");
    }
}
The this reference

- The this reference allows an object to refer to itself

- One usage of the this reference is to
  - allow instance variables and local variables to have same names and distinguish
    - the instance variables of a class from
    - corresponding method parameters or other local variables with the same names
  - For example, the constructor of the Account class (from Chapter 4) deliberately uses different names for same information

```java
public class Account
{
  private final double RATE = 0.035;  // interest rate of 3.5%
  private long acctNumber;
  private double balance;
  private String name;
  //-----------------------------------------------------------------
  //  Sets up the account by defining its owner, account number,
  //  and initial balance.
  //-----------------------------------------------------------------
  public Account (String owner, long account, double initial)
  {
    name = owner;
    acctNumber = account;
    balance = initial;
  }
...```

© 2007 Pearson Addison-Wesley. All rights reserved
The this reference

- But it could have been written as follows:

```java
public class Account
{
    private final double RATE = 0.035; // interest rate of 3.5%

    private long acctNumber;
    private double balance;
    private String name;

    public Account(String name, long acctNumber, double balance)
    {
        this.name = name;
        this.acctNumber = acctNumber;
        this.balance = balance;
    }
}
```

The Flight Class (PP 4.5)

- “Design and implement a class called Flight that represents an airline flight. It should contain instance data that represents the airline name, flight number, and the flight’s origin and destination cities. Define the flight constructor to accept and initialize all instance data. Include getter and setter methods for all instance data. Include a toString method that returns a one-line description of the flight. Create a driver class called FlightTest, whose main method instantiates and updates several flight objects.”
public class Flight
{
    private String airline, origin, destination;
    private int flightNumber;

    // Sets up this Flight object with the specified data.
    public Flight (String airline, String origin, String destination, int flightNumber)
    {
        this.airline = airline;
        this.origin = origin;
        this.destination = destination;
        this.flightNumber = flightNumber;
    }

    // Accessors (getters)
    public String getAirline ()
    {
        return airline;
    }

    public String getOrigin ()
    {
        return origin;
    }

    public String getDestination ()
    {
        return destination;
    }

    public int getFlightNumber ()
    {
        return flightNumber;
    }
}
Flight.java (cont.)

// Mutators (setters)

public void setAirline (String airline)
{
    this.airline = airline;
}

public void setOrigin (String origin)
{
    this.origin = origin;
}

public void setDestination (String destination)
{
    this.destination = destination;
}

public void setFlightNumber (int flightNumber)
{
    this.flightNumber = flightNumber;
}

Flight.java (cont.)

// Returns a string representation of this flight.

public String toString ()
{
    return airline + " " + flightNumber + " -- From: " + origin + 
    " To: " + destination;
}

```java
package FlightTest;

public class FlightTest {

    public static void main(String[] args) {
        Flight f1 = new Flight("US Air", "Boston", "Los Angeles", 347);
        Flight f2 = new Flight("Delta", "Philadelphia", "London", 212);
        Flight f3 = new Flight("Continental", "Atlanta", "Chicago", 822);

        System.out.println(f1);
        System.out.println(f2);
        System.out.println(f3);

        System.out.print("flight 3 \t old flight number: "+f3.getFlightNumber());
        f3.setFlightNumber(101);
        System.out.println("\t new flight number: "+f3.getFlightNumber());
    }
}
```