

If Statements The syntax of a simple if statement is as follows: if (booleanExpression) trueBody else falseBody booleanExpression is a boolean expression and trueBody and falseBody are each either a single statement or a block of statements enclosed in braces ("{" and "}").

Compound if Statements

 There is also a way to group a number of boolean tests, as follows:

if (firstBooleanExpression)
 firstBody
else if (secondBooleanExpression)
 secondBody
else
 thirdBody

© 2014 Goodrich, Tamassia, Goldwasser

Java Primer 2

3

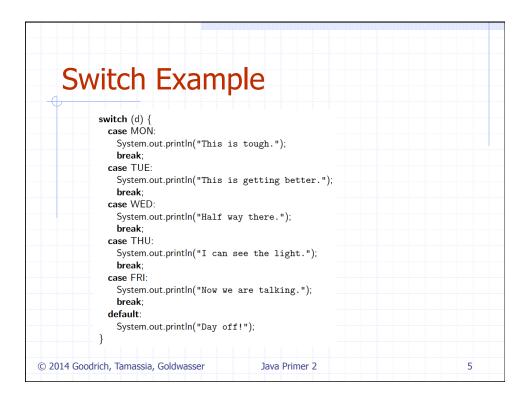
Switch Statements

- Java provides for multiple-value control flow using the switch statement.
- The switch statement evaluates an integer, string, or enum expression and causes control flow to jump to the code location labeled with the value of this expression.
- If there is no matching label, then control flow jumps to the location labeled "default."
- This is the only explicit jump performed by the switch statement, however, so flow of control "falls through" to the next case if the code for a case is not ended with a **break** statement

© 2014 Goodrich, Tamassia, Goldwasser

Java Primer 2

4



Break and Continue

- Java supports a **break** statement that immediately terminate a while or for loop when executed within its body.
- Java also supports a continue statement that causes the current iteration of a loop body to stop, but with subsequent passes of the loop proceeding as expected.

© 2014 Goodrich, Tamassia, Goldwasser

Java Primer 2

While Loops

 The simplest kind of loop in Java is a while loop.

- Such a loop tests that a certain condition is satisfied and will perform the body of the loop each time this condition is evaluated to be true.
- The syntax for such a conditional test before a loop body is executed is as follows:

while (booleanExpression) loopBody

© 2014 Goodrich, Tamassia, Goldwasser

Java Primer 2

7

Do-While Loops

- Java has another form of the while loop that allows the boolean condition to be checked at the end of each pass of the loop rather than before each pass.
- This form is known as a do-while loop, and has syntax shown below:

do

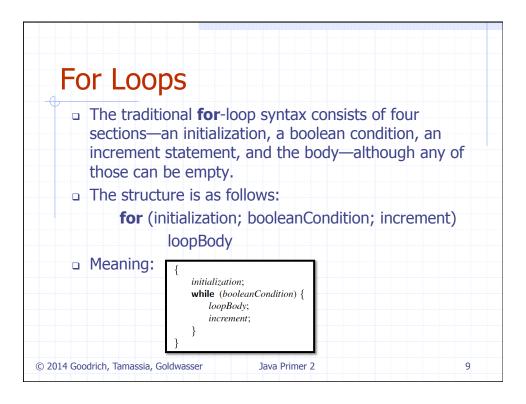
loopBody

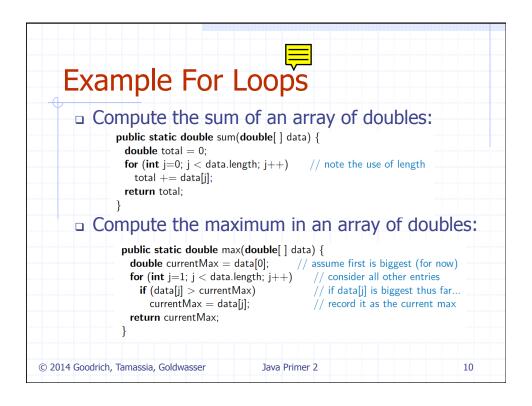
while (booleanExpression)

© 2014 Goodrich, Tamassia, Goldwasser

Java Primer

8





For-Each Loops

- Since looping through elements of a collection is such a common construct,
 Java provides a shorthand notation for such loops, called the **for-each** loop.
- The syntax for such a loop is as follows:
 for (elementType name : container)
 loopBody

© 2014 Goodrich, Tamassia, Goldwasser

Java Primer 2

11

For-Each Loop Example

Computing a sum of an array of doubles:

- When using a for-each loop, there is no explicit use of array indices.
- The loop variable represents one particular element of the array.

© 2014 Goodrich, Tamassia, Goldwasser

Java Primer 2

12

© 2014 Goodrich, Tamassia, Goldwasser

Java provides a built-in static object, called
 System.out, that performs output to the "standard output" device, with the following methods:

```
print(String s): Print the string s.

print(Object o): Print the object o using its toString method.

print(baseType b): Print the base type value b.

println(String s): Print the string s, followed by the newline character.

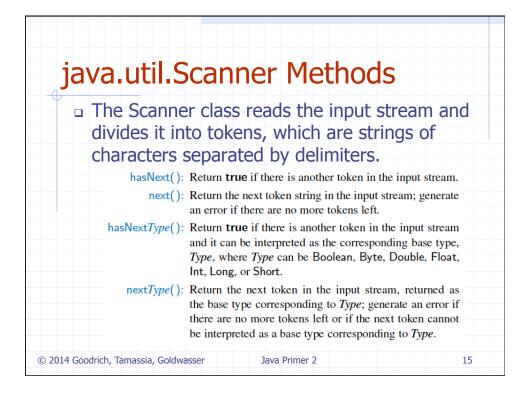
println(Object o): Similar to print(o), followed by the newline character.

println(baseType b): Similar to print(b), followed by the newline character.
```

Java Primer 2

13

Simple Input There is also a special object, System.in, for performing input from the Java console window. A simple way of reading input with this object is to use it to create a **Scanner** object, using the expression **new** Scanner(System.in) Example: import java.util.Scanner; // loads Scanner definition for our use public class InputExample { public static void main(String[] args) { ${\sf Scanner input} = {\sf new Scanner}({\sf System.in});$ System.out.print("Enter your age in years: "); double age = input.nextDouble(); System.out.print("Enter your maximum heart rate: "); double rate = input.nextDouble(); double fb = (rate - age) * 0.65; System.out.println("Your ideal fat-burning heart rate is " + fb); © 2014 Goodrich, Tamassia, Goldwasser Java Primer 2



```
Sample Program
            public class CreditCard {
               // Instance variables:
                                                  // name of the customer (e.g., "John Bowman")
// name of the bank (e.g., "California Savings")
// account identifier (e.g., "5391 0375 9387 5309")
               private String customer;
               private String bank;
               private String account;
              private int limit;
                                                  // credit limit (measured in dollars)
               protected double balance;
                                                  // current balance (measured in dollars)
               // Constructors:
               public CreditCard(String cust, String bk, String acnt, int lim, double initialBal) {
                 customer = cust;
                 bank = bk;
        12
                 account = acnt;
                 limit = lim:
       13
                 balance = initialBal;
       15
               public CreditCard(String cust, String bk, String acnt, int lim) {
       16
       17
                 this(cust, bk, acnt, lim, 0.0);
                                                                     // use a balance of zero as default
       18
© 2014 Goodrich, Tamassia, Goldwasser
                                                      Java Primer 2
                                                                                                        16
```

```
Sample Program
                 public String getCustomer() { return customer; }
                 public String getBank() { return bank; }
                 public String getAccount() { return account; }
                 public int getLimit() { return limit; }
          24
                 public double getBalance() { return balance; }
                     Update methods:
                 public boolean charge(double price) {
                                                                         // make a charge
                    if (price + balance > limit)
                                                                            if charge would surpass limit
                      return false;
                                                                           refuse the charge
          29
                       at this point, the charge is successful
          30
                    balance += price;
                                                                           update the balance
          31
                                                                         // announce the good news
                    return true:
          32
          33
                 public void makePayment(double amount) {
                                                                        // make a payment
          34
                    balance -= amount;
          35
          36
                  // Utility method to print a card's information
          37
                 public static void printSummary(CreditCard card) {
                    System.out.println("Customer = " + card.customer);

System.out.println("Bank = " + card.bank);

System.out.println("Account = " + card.account);

System.out.println("Balance = " + card.balance); /

System.out.println("Limit = " + card.limit); /
                                                                               // implicit cast
          42
                                                                                // implicit cast
          43
                     main method shown on next page...
© 2014 Goodrich, Tamassia, Goldwasser
                                                                     Java Primer 2
                                                                                                                                     17
```

```
Sample Program
               \begin{array}{ll} \textbf{public static void } main(String[\ ]\ args)\ \{ \\ CreditCard[\ ]\ wallet = \textbf{new } CreditCard[3]; \end{array} 
                wallet[0] = new CreditCard("John Bowman", "California Savings",
                                              "5391 0375 9387 5309", 5000);
                wallet[1] = new CreditCard("John Bowman", "California Federal",
        5
                                              "3485 0399 3395 1954", 3500);
                wallet[2] = new CreditCard("John Bowman", "California Finance",
                                              "5391 0375 9387 5309", 2500, 300);
        9
                for (int val = 1; val \neq 16; val++) {
       10
       11
                  wallet[0].charge(3*val);
       12
                  wallet[1].charge(2*val);
       13
                  wallet[2].charge(val);
       14
       15
                for (CreditCard card : wallet) {
       16
       17
                  CreditCard.printSummary(card);
                                                            // calling static method
       18
                  while (card.getBalance() > 200.0) {
       19
                     card.makePayment(200);
       20
                     System.out.println("New balance = " + card.getBalance());
       21
       22
       23
© 2014 Goodrich, Tamassia, Goldwasser
                                                       Java Primer 2
                                                                                                        18
```