Chapter 2
Console Input and Output

Key Terms
console I/O
System.out.println
print versus println
System.out.printf
format specifier
field width
conversion character
e and g
right justified-left justifie
more arguments (for printf)
format string
new lines
%n
legacy code
NumberFormat
package
java.text
import statement,
java.util
java.lang
import
patterns
percentages
e-notation
mantissa
import
nextInt,
whitespace
nextDouble,
word
empty string
echoing input

Brief Outline
2.1 Screen Output
   System.out.println
   Formatting Output with printf
   Money Formats Using NumberFormat
   Importing Packages and Classes

Teaching Suggestions
This chapter discusses both the topics of input and output for Java programs. The Chapter features the `Scanner` class, which is the new input option available in Java 5.0. Other options for input like the original Java class `BufferedReader` and the `JOptionPane` that were covered in previous versions of this text are no longer covered.

Outputting information is something the students have already seen and done. However, this section deals with outputting in a way so that the information is formatted. This is especially important when dealing with money or decimal numbers. Students may have already seen results that did not look nice when printed to the screen. Several of the built-in formatters are introduced to help students get better looking results when they output. Also introduced is using `System.out.printf` for outputting information to the screen.

The programs that the students have been able to write so far have not been able to require input from the user. They have simply been computations that the computer executes and gives results for. Now it is time to introduce a way for the students to get user input and use that input in their computations.

The method that is introduced makes use of the `java.util.Scanner` class, which is new to Java 5.0. Upon creating an instance of the `Scanner` object, the programmer can use the `nextInt`, `nextDouble`, `next`, `nextLine` and other methods to get input from the keyboard. The `Scanner` delineates different input values using whitespace.

Key Points
println Output. We have seen using `System.out.println` already, but it is important to once again note that we can output Strings as well as the values of variables of primitive type or a combination of both.

println versus print. While `println` gives output and then positions the cursor to produce output on the next line, `print` will ensure that multiple outputs appear on the same line. Which one you use depends on how you want the output to look on the screen.

System.out.printf. This new feature of Java 5.0 can be used to help format output to the screen. `System.out.printf` takes arguments along with the data to be outputted to format the output as specified by the user. This feature is similar to the `printf` function that is available in the C language.

Outputting Amounts of Money. When we want to output currency values, we would like the appropriate currency markers to appear as well as our output to have the correct number of decimal places (if dollars and cents). Java has built in the facilities for giving currency output for many countries. This section introduces the idea that there are packages in Java, and that these packages may contain classes that can be useful. In order to gain access to classes in another package, we can use an import statement. We can then create an instance of the `NumberFormat` class to help us output currency in the proper format.

DecimalFormat Class. We might also like to format decimal numbers that are not related to currency. To do this, we can use the `DecimalFormat` class. When we use this class, we need to give a pattern for how we want the output to be formatted. This pattern can include the number of digits before and after a decimal point, as well as helping us to format percentages.

Keyboard Input Using the Scanner Class. To do keyboard input in a program, you need to first create an instance of the `Scanner` class. This may be the first time the students are required to create an instance of an object using the keyword *new*. The methods for the `Scanner` class, `nextInt`, `nextDouble`, and `next` are used to get the user's input. The `nextLine` method reads an entire line of text up to the newline character, but does not return the newline character.

Tips
Different Approaches to Formatting Output. With the addition of `System.out.printf` in Java 5, there are now two distinct ways to format output in Java programs. This chapter of the book will discuss both methods.

Formatting Money Amounts with printf. When formatting output that is of currency, the most common way that the programmer wants the user to see the ouput is with only two decimal places. If the amount of money is stored as a double, you can use `%.2f` to format the output to two decimal places.

Legacy Code. Code that is in an older style or an older language, commonly called legacy code can often be difficult and expensive to replace. Many times, legacy code is translated to a more modern programming language. Such is the legacy of `printf` being incorporated into Java.

Prompt for Input. It is always a good idea to create a meaningful prompt when asking the user to provide input to the program. This way the user knows that the program is waiting for a response and what type of response he/she should provide.

Echo Input. It is a good idea to echo the user's input back to them after they have entered it so that they can ensure its accuracy. However, at this point in time, we have no way to allow for them to answer if it is correct and change it if it is not. Later on, we will introduce language constructs (if-statements) that will help us do this type of operation. For now, this tip can serve as a good software engineering practices tip.

Pitfalls

Dealing with the Line Terminator '\n'. This pitfall illustrates the important point that some of the methods of the `Scanner` class read the new line character and some methods do not. In the example shown, `nextInt` is used to show a method that does not read the line terminator. The `nextInt` method actually leaves the new line character on the input stream. Therefore, the next read of the stream would begin with the new line character. The method `readLine` does read the line terminator character and removes it from the input stream. This pitfall is one that should be discussed when reading input of different types from the same input stream.

Programming Example
Self-Service Check Out. This example program is an interactive check-out program for a store, where the user inputs the information about the items that are being purchased and the program outputs the total amount owed for the purchase. It illustrates the use of the `Scanner` class as well as `System.out.printf`.

Programming Projects Answers

1.
```
/**
 * Question1.java
 *
 *  This program uses the Babylonian algorithm, using five
 *  iterations, to estimate the square root of a number n.
 * Created: Sat Mar 05, 2005
 *
 * @author Kenrick Mock
 * @version 1
 */

import java.util.Scanner;

public class Question1
{
  public static void main(String[] args)
        {
        // Variable declarations
        Scanner scan = new Scanner(System.in);
        double guess;
        int n;
        double r;

        System.out.println("This program makes a rough estimate for square roots.");
        System.out.println("Enter an integer to estimate the square root of: ");
        n = scan.nextInt();

        // Initial guess
```

```
        guess = (double) n/2;

        // First guess
        r = (double) n/ guess;
        guess = (guess+r)/2;

        // Second guess
        r = (double) n/ guess;
        guess = (guess+r)/2;

        // Third guess
        r = (double) n/ guess;
        guess = (guess+r)/2;

        // Fourth guess
        r = (double) n/ guess;
        guess = (guess+r)/2;

        // Fifth guess
        r = (double) n/ guess;
        guess = (guess+r)/2;

        // Output the fifth guess
        System.out.printf("The estimated square root of %d is %4.2f\n", n, guess);
        }
} // Question1
```

2.

```
/**
 * Question2.java
 *
 * This program outputs a name in lowercase to a name in Pig Latin
 * with the first letter of each name capitalized.  It inputs the
 * names from the console using the Scanner class.
 * Created: Sat Mar 05, 2005
 *
 * @author Kenrick Mock
 * @version 1
 */
import java.util.Scanner;

public class Question2
{
    public static void main(String[] args)
        {
```

```
            // Variable declarations
            Scanner scan = new Scanner(System.in);
            String first;
            String last;

            System.out.println("Enter your first name:");
            first = scan.nextLine();
            System.out.println("Enter your last name:");
            last = scan.nextLine();

            System.out.println(first + " " + last + " turned to Pig Latin is:");

            // First convert first name to pig latin
            String pigFirstName = first.substring(1,first.length()) + first.substring(0,1) + "ay";
            // Then capitalize first letter
            pigFirstName = pigFirstName.substring(0,1).toUpperCase() +
                    pigFirstName.substring(1,pigFirstName.length());

            // Repeat for the last name
            String pigLastName = last.substring(1,last.length()) + last.substring(0,1) + "ay";
            // Then capitalize first letter
            pigLastName = pigLastName.substring(0,1).toUpperCase() +
                    pigLastName.substring(1,pigLastName.length());

            System.out.println(pigFirstName + " " + pigLastName);
        }
} // Question2
```

3.

```
    /**
     * Question3.java
     *
     * Created: Sat Nov 08 16:11:48 2003
     * Modified: Sat Mar 05 2005, Kenrick Mock
     *
     * @author Adrienne Decker
     * @version 2
     */
    import java.util.Scanner;

    public class Question3
    {

      public static void main(String[] args)
      {
```

```java
      Scanner keyboard = new Scanner(System.in);

      System.out.println("Enter first number to add:");
      int first = keyboard.nextInt();

      System.out.println("Enter second number to add");
      int second = keyboard.nextInt();

      int result = first + second;
      System.out.println("Adding " + first + " + " + second +
                   " equals " + result);

      System.out.println("Subtracting " + first + " - " + second +
                   " equals " + (first - second));

      System.out.println("Multiplying " + first + " * " + second +
                   " equals " + (first * second));
   }
} // Question3
```

4.

```java
   /**
    * Question4.java
    *
    * Created: Sat Nov 08 16:11:48 2003
    * Modified: Sat Mar 05 2005, Kenrick Mock
    *
    * @author Adrienne Decker
    * @version 2
    */
   public class Question4
   {
     public static void main(String[] args)
     {
        Scanner keyboard = new Scanner(System.in);

      System.out.println("Enter the distance of the commute in miles:");
      int distanceOfCommute = keyboard.nextInt();

      System.out.println("Enter the number of miles your car gets to "
                   + "the gallon:");
      int milesPerGallon = keyboard.nextInt();;

      System.out.println("Enter the price of a gallon of gas as a "
                   +"decimal number:");
```

```java
        double costOfGallonGas = keyboard.nextDouble();


        double gallonsNeeded = (double) distanceOfCommute / milesPerGallon;
        double result = gallonsNeeded * costOfGallonGas;

        NumberFormat moneyFormater = NumberFormat.getCurrencyInstance();

        System.out.println("For a trip of " + distanceOfCommute +
                    " miles, with a consumption rate of "
                    + milesPerGallon + " miles per gallon, and a"
                    + " cost of " +
                    moneyFormater.format(costOfGallonGas)
                    + " per gallon of gas, your trip will cost you "
                    + moneyFormater.format(result));
    }
} // Question4
```

5.

```java
/**
 * Question5.java
 *
 * Created: Sat Nov 08 16:11:48 2003
 * Modified: Sat Mar 05 2005, Kenrick Mock
 *
 * @author Adrienne Decker
 * @version 2
 */

import java.text.NumberFormat;
import java.util.Scanner;

public class Question5
{

  public static void main(String[] args)
  {
        Scanner keyboard = new Scanner(System.in);

    System.out.println("Enter the purchase price of the"
                    + " item as a decimal number:");
    double purchasePrice = keyboard.nextDouble();


    System.out.println("Enter the expected number of "
```

```
                          + "years of service for the item:");
    int yearsOfService = keyboard.nextInt();


    System.out.println("Enter the salvage price of the "
                         + "item as a decimal number: ");
    double salvagePrice = keyboard.nextDouble();


    double yearlyDepreciation =
        (purchasePrice - salvagePrice) / yearsOfService;

    NumberFormat moneyFormater = NumberFormat.getCurrencyInstance();

    System.out.println
        ("For an item whose initial purchse price was " +
         moneyFormater.format(purchasePrice) + "\nand whose expected "
         + "number of years of service is " + yearsOfService
         + "\nwhere at the end of those years of service the salvage "
         + "price will be " + moneyFormater.format(salvagePrice) +
         ",\nthe yearly depreciation of the item will be " +
         moneyFormater.format(yearlyDepreciation) + " per year.");
  }
} // Question5
```

6.


```
/**
 * Question6.java
 *
 * Created: Sat Nov 08 15:41:53 2003
 * Modified: Sat Mar 05 2005, Kenrick Mock
 *
 * @author Adrienne Decker
 * @version 2
 */
import java.util.Scanner;

public class Question6
{

  public static final int WEIGHT_OF_CAN_SODA_GRAMS = 30;
  public static final double AMT_SWEETNR_IN_SODA = 0.001;

  public static void main(String[] args)
```

```java
   {
          Scanner keyboard = new Scanner(System.in);

      System.out.println("Enter the amount of grams of " +
                                    "sweetener needed to kill the " +
                                    "mouse: ");
      int amountNeededToKillMouse = keyboard.nextInt();


      System.out.println("Enter the weight of the mouse "
                                    + "in pounds.");
      int weightOfMouse = keyboard.nextInt();


      System.out.println("Enter the desired weight of the"
                          + " dieter in pounds.");
      double desiredWeight = keyboard.nextDouble();


      double amountPerCanGrams = (double)WEIGHT_OF_CAN_SODA_GRAMS *
          AMT_SWEETNR_IN_SODA;

      double proportionSwtnrBodyWeight =
          (double) (amountNeededToKillMouse / weightOfMouse );

      double amtNeededToKillFriend = proportionSwtnrBodyWeight *
          desiredWeight;

      double cansOfSoda = amtNeededToKillFriend * amountPerCanGrams;

      System.out.println("You should not drink more than " +
                          cansOfSoda + " cans of soda.");

   }
} // Question6
```

7.

```java
/**
 * Question7.java
 *
 * Created: Sat Nov 08 15:41:53 2003
 * Modified: Sat Mar 05 2005, Kenrick Mock
 *
 * @author Adrienne Decker
 * @version 2
```

```java
 */
import java.util.Scanner;

public class Question7
{
  public static void main(String[] args)
  {
        Scanner scan = new Scanner(System.in);

     System.out.println("Enter the price of the item " +
                        "from 25 cents to one dollar " +
                        "in five cent increments:");

     int priceOfItem = scan.nextInt();

     int change = 100 - priceOfItem;

     int numQuarters = change / 25;
     change = change - (numQuarters * 25);

     int numDimes = change / 10;
     change = change - (numDimes * 10);

     int numNickels = change / 5;

     System.out.println("You bought an item for " + priceOfItem +
          " cents and gave me one dollar, so your change is\n" +
          numQuarters + " quarters,\n" + numDimes +
          " dimes, and \n" + numNickels + " nickels.");

  }
} // Question7
```

8.

```java
/**
 * Question8.java
 *
 * Created: Sat Nov 08 15:41:53 2003
 * Modified: Sat Mar 05 2005, Kenrick Mock
 *
 * @author Adrienne Decker
 * @version 2
 */
import java.util.Scanner;
```

```java
public class Question8
{

  public static void main(String[] args)
  {
         Scanner keyboard = new Scanner(System.in);
    System.out.println("Enter the text: " );
    String firstString = keyboard.nextLine();

    System.out.println("The text in all upper case is: \n" +
         firstString.toUpperCase());

    System.out.println("The text in all lower case is: \n" +
         firstString.toLowerCase());

  }
} // Question8
```

9.

```java
/**
 * Question9.java
 *
 * Created: Sat Nov 08 15:41:53 2003
 * Modified: Sat Mar 05 2005, Kenrick Mock
 *
 * @author Adrienne Decker
 * @version 2
 */

import java.util.Scanner;
public class Question9
{

  public static void main(String[] args)
  {
    Scanner keyboard = new Scanner(System.in);
    System.out.println("Enter a line of text: " );
    String firstString = keyboard.nextLine();

    int position = firstString.indexOf("hate");

    String firstPart = firstString.substring(0, position);
    String afterHate = firstString.substring(position + 4);

    String newString = firstPart + "love" + afterHate;
```

```
    System.out.println("I have rephrased that line to read:\n" +
        newString);

  }
} // Question9
```

10.

```
/**
 * Question10.java
 *
 *  This program outputs a bill for three items.
 *  The bill should be formatted in columns with 30
 *  characters for the name, 10 characters for the
 *  quantity, 10 characters for the price, and
 *  10 characters for the total.
 *
 * Created: Sat Mar 15 2009
 *
 * @author Kenrick Mock
 * @version 1
 */
import java.util.Scanner;

public class Question10
{
        public static double SALESTAX = 0.0625;

        public static void main(String[] args)
        {
                String name1, name2, name3;
                int quantity1, quantity2, quantity3;
                double price1, price2, price3;
                double total1, total2, total3;
                double subtotal;
                double tax;
                double total;
                Scanner kbd = new Scanner(System.in);

                System.out.println("Name of item 1:");
                name1 = kbd.nextLine();
                System.out.println("Quantity of item 1:");
                quantity1 = kbd.nextInt();
```

```java
            System.out.println("Price of item 1:");
            price1 = kbd.nextDouble();
            kbd.nextLine();

            System.out.println("Name of item 2:");
            name2 = kbd.nextLine();
            System.out.println("Quantity of item 2:");
            quantity2 = kbd.nextInt();
            System.out.println("Price of item 2:");
            price2 = kbd.nextDouble();
            kbd.nextLine();

            System.out.println("Name of item 3:");
            name3 = kbd.nextLine();
            System.out.println("Quantity of item 3:");
            quantity3 = kbd.nextInt();
            System.out.println("Price of item 3:");
            price3 = kbd.nextDouble();
            kbd.nextLine();

            total1 = quantity1 * price1;
            total2 = quantity2 * price2;
            total3 = quantity3 * price3;
            subtotal = total1 + total2 + total3;
            tax = SALESTAX * subtotal;
            total = tax + subtotal;

            // Allot 30 characters for the name, then 10
            // for the quantity, price, and total. The hyphen after the %
            // makes the field left-justified
            System.out.printf("%-30s %-10s %-10s %-10s\n", "Item", "Quantity",
                                    "Price", "Total");
            System.out.printf("%-30s %-10d %-10.2f %-10.2f\n", name1,
                    quantity1, price1, total1);
            System.out.printf("%-30s %-10d %-10.2f %-10.2f\n", name2,
                    quantity2, price2, total2);
            System.out.printf("%-30s %-10d %-10.2f %-10.2f\n", name3,
                    quantity3, price3, total3);
            System.out.println();
            System.out.printf("%-52s %-10.2f\n", "SubTotal", subtotal);
            System.out.printf("%-52s %-10.2f\n", SALESTAX * 100 + " Sales Tax",
                    tax);
            System.out.printf("%-52s %-10.2f\n", "Total", total);
        }
} // Question 10
```

11.

```
/**
 * Question11.java
 *
 *  This program calculates the total grade for three
 * classroom exercises as a percentage.  It uses the DecimalFormat
 * class to output the value as a percent.
 * The scores are summarized in a table.
 *
 * Created: Sat Mar 15 2009
 *
 * @author Kenrick Mock
 * @version 1
 */

import java.util.Scanner;
import java.text.DecimalFormat;

public class Question11
{
        public static void main(String[] args)
        {
                String name1, name2, name3;
                int points1, points2, points3;
                int total1, total2, total3;
                int totalPossible, sum;
                double percent;
                Scanner kbd = new Scanner(System.in);

                System.out.println("Name of exercise 1:");
                name1 = kbd.nextLine();
                System.out.println("Score received for exercise 1:");
                points1 = kbd.nextInt();
                System.out.println("Total points possible for exercise 1:");
                total1 = kbd.nextInt();
                kbd.nextLine();

                System.out.println("Name of exercise 2:");
                name2 = kbd.nextLine();
                System.out.println("Score received for exercise 2:");
                points2 = kbd.nextInt();
                System.out.println("Total points possible for exercise 2:");
                total2 = kbd.nextInt();
                kbd.nextLine();
```

```
                System.out.println("Name of exercise 3:");
                name3 = kbd.nextLine();
                System.out.println("Score received for exercise 3:");
                points3 = kbd.nextInt();
                System.out.println("Total points possible for exercise 3:");
                total3 = kbd.nextInt();
                kbd.nextLine();

                totalPossible = total1 + total2 + total3;
                sum = points1 + points2 + points3;
                percent = (double) sum / totalPossible;

                // Allot 30 characters for the exercise name, then 6
                // for the score and total. The hyphen after the %
                // makes the field left-justified
                System.out.printf("%-30s %-6s %-6s \n", "Exercise", "Score",
                                            "Total Possible");
                System.out.printf("%-30s %-6d %-6d \n", name1, points1, total1);
                System.out.printf("%-30s %-6d %-6d \n", name2, points2, total2);
                System.out.printf("%-30s %-6d %-6d \n", name3, points3, total3);
                System.out.printf("%-30s %-6d %-6d \n", "Total", sum, totalPossible);
                DecimalFormat formatPercent = new DecimalFormat("0.00%");
                System.out.println("\nYour total is " + sum + " out of " +
                        totalPossible + ", or " + formatPercent.format(percent) +
                        " percent.");

        }
} // Question 11


/**
 * Question12.java
 *
 * This program changes "hate" to "love" in a file.  It is a bit awkward because
 * if statements haven't been covered yet.
 *
 * Created: Fri Apr 27 2012
 *
 * @author Kenrick Mock
 * @version 1
 */

import java.util.Scanner;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
```

```java
public class Question12
{
        public static void main(String[] args)
        {
                Scanner fileIn = null;  // Initializes fileIn to an empty object
                try
                {
                        // Attempt to open the file
                        fileIn = new Scanner(
                                new FileInputStream("file.txt"));
                }
                catch (FileNotFoundException e)
                {
                        // If the file could not be found, this code is executed
                        // and then the program exits
                        System.out.println("File not found.");
                        System.exit(0);
                }

                // If the program gets here then
                // the file was opened successfully
                String line;
                line = fileIn.nextLine();

                int position = line.indexOf("hate");
                System.out.println(
                        line.substring(0,position) + "love" +
                        line.substring(position+4));

                fileIn.close();
        }
} // Question12
```

Question 13: No solution provided