Chapter 5.

5.1.  
(a) 6  
(b) 0  
(c) 1  
(d) 0.8  
(e) 1.3333333333333333  
(f) 2  
(g) 9  
(h) 9  
(i) 2  
(j) 0  
(k) 19.0  
(l) 5.0  
(m) -3  
(n) 18  
(o) 0  
(p) 2  
(q) -1.6666666666666667  
(r) 4

5.2.  
a. 2*i  
b. return 2*i;  
c. i = 2*i;

5.3.  
(a) i: 4; j: 5  
(b) i: 4; j: 4  
(c) i: 4; j: 2  
(d) i: 4; j: 2

5.4.  
ones = i%10:  
tens = i/10;

5.5.  
dollars = dollars + pennies/100;  
cents = cents + pennies%100;

5.6.  
dollars = dollars + (nickles*5)/100;  
cents = cents + (nickles*5)%100;

5.7.  
dollars = dollars + (nickles*5+pennies)/100;  
cents = cents + (nickles*5+pennies)%100;

5.8.  

\texttt{public void doubleCount () \{}  
\hspace{1em} \texttt{tally = 2*tally;  
\}  

5.9.  

\texttt{public Counter () \{}  
\hspace{1em} \texttt{tally = 0;  
\hspace{1em} \texttt{savedTally = 0;  
\}  

\texttt{public void reset (0 \{}  
\hspace{1em} \texttt{savedTally = tally;  
\hspace{1em} \texttt{tally = 0;  
\}  

/\* *\*/
public void unReset () {
    tally = savedTally;
}

private int savedTally; // The value of tally prior to the last reset.

public class Date {

    /**
     * Create a new Date with the specified day, month, and year.
     * day must a value in the range 1 to maximum number of days
     * in the month.
     * month must a value in the range 1 to 12, with 1 indicating
     * January, etc.
     * year must be a value greater than 0.
     */
    public Date (int day, int month, int year) {
        thisDay = day;
        thisMonth = month;
        thisYear = year;
    }

    /**
     * The Day of this Date.
     */
    public int getDay () {
        return thisDay;
    }

    /**
     * The month of this Date. 1 is January, etc.
     */
    public int getMonth () {
        return thisMonth;
    }

    /**
     * The year of this Date.
     */
    public int getYear () {
        return thisYear;
    }
}
// Components:

private int thisDay; // the day of the month
private int thisMonth; // the month: 1 is January, etc.
private int thisYear; // the year (AD)

} // end of class Date

5.11.

public class Lamp {

   // Constants:

   public final static int OFF = 0; // the off state
   public final static int LOW = 1; // the low state
   public final static int MED = 2; // the medium state
   public final static int HIGH = 3; // the high state

   // Constructors:

   /**
   * Create a new Lamp. Lamp is initially off.
   */
   public Lamp () {
      state = OFF;
   }

   //Queries:

   /**
   * The state of this Lamp. Lamp.OFF == off, Lamp.LOW == low,
   * Lamp.MED == medium, Lamp.HIGH == high.
   */
   public int getState () {
      return state;
   }

   // Commands:

   /**
   * Advance the Lamp to the next state. If off, advance
   * to low; if low, advance to medium; if medium,
   * advance to high; if high, advance to off.
   */
   public void advance () {
      state = (state+1) % NUMBER_OF_STATES;
   }
}
// Private constants:

private final static int NUMBER_OF_STATES = 4; // the number of distinct states.

// Components:

private int state; // the current state of this Lamp

} // end of class Lamp

5.12.

public class Rectangle {

    // Components // the width of this Rectangle // the length of this Rectangle

    private double thisLength; private double thisWidth; /**
    * Create a new Rectangle with the specified length and width.
    * length and width must be non-negative.
    */
    public Rectangle (double length, double width) {
        thisLength = length;
        thisWidth = width;
    }

    // Queries:

    /**
     * The length of this Rectangle.
     */
    public double getLength () {
        return thisLength;
    }

    /**
     * The width of this Rectangle.
     */
    public double getWidth () {
        return thisWidth;
    }

    /**
     * The area of this Rectangle.
     */
    public double getArea () {
        return thisLength*thisWidth;
    }
// Commands:
/**
 * Set the length of this Rectangle. newLength must be non-negative.
 */
public void setLength (double newLength) {
    thisLength = newLength;
}

/**
 * Set the width of this Rectangle. newWidth must be non-negative.
 */
public void setWidth (double newWidth) {
    thisWidth = newWidth;
}

} // end of class Rectangle

5.13.
/**
 * Move to the specified Room.
 */
public void move (rooms.Room newRoom) {
    room.exit();
    room = newRoom;
    room.enter();
}
5.14. a.

```java
private String thisName;
private String thisAddress;
private String thisSsn;
private int creditHours;
private int feesPaid;
private courses.CourseList schedule;
```

b.

```java
/**
 * The social security number of this Student.
 */
public String getSsn () {
    return thisSsn;
}

/**
 * Change this Student’s social security number to the
 * specified value.
 */
public void changeSsn (String newSsn) {
    thisSsn = newSsn;
}

c.

```java
/**
 * This Student’s current fees (dollars).
 */
public int getFees () {
    return 100*creditHours; // better to use a named constant.
}

d.

```java
/**
 * Fees paid by this Student (dollars).
 */
public int getFeesPaid () {
    return feesPaid;
}

/**
 * This Student has paid the specified amount (dollars).
 */
public void payFees (int amount) {
    feesPaid = feesPaid + amount;
}

c.

```java
/**
 * Create a new Student with the specified name, address, and social
 * security number.
 */
```