APPENDIX M

Answers to Odd-Numbered Review Questions

Chapter 1

Multiple Choice and True/False

1. b
3. a
5. b
7. c
9. a
11. a
13. b

Find the Error

1. The algorithm performs the math operation at the wrong time. It multiplies width by length before getting values for those variables.

Algorithm Workbench

1. *Display "What is the customer's maximum amount of credit?"*
   Input maxCredit.
   Display "What is the amount of credit used by the customer?"
   Input creditUsed.
   availableCredit = maxCredit − creditUsed.
   Display availableCredit.

3. *Display "What is the account's starting balance?"*
   Input startingBalance.
   Display "What is the total amount of the deposits made?"
   Input deposits.
   Display "What is the total amount of the withdrawals made?"
   Input withdrawals.

Display "What is the monthly interest rate?"
Input interestRate.
balance = startingBalance + deposits − withdrawals.
*interest = balance * interestRate.*
balance = balance + interest.
Display balance.

Predict the Result

1. 7

Short Answer

1. Main memory, or RAM, holds the sequences of instructions in the programs that are running and the data those programs are using. Main memory, or RAM, is usually volatile. Secondary storage is a type of memory that can hold data for long periods of time—even when there is no power to the computer.

3. An operating system is a set of programs that manages the computer's hardware devices and controls their processes. Windows and UNIX are examples of operating systems. Application software refers to programs that make the computer useful to the user. These programs solve specific problems or perform general operations that satisfy the needs of the user. Word processing, spreadsheet, and database packages are all examples of application software.

5. Because the computer is only capable of directly processing machine language instructions.

7. A file that contains source code, which is the code written by the programmer.

9. An algorithm is a set of well-defined steps for performing a task or solving a problem.

11. An *application* is a stand-alone program that runs on your computer. An *applet* is designed to be transmitted over the Internet from a Web server, and then executed in a Web browser.

13. A Java Virtual Machine (JVM) program.

15. Because Java byte code is the same for all computers.

17. When an object's internal data is hidden from outside code and access to that data is restricted to the object's methods, the data is protected from accidental corruption. In addition, the programming code outside the object does not need to know about the format or internal structure of the object's data. The code only needs to interact with the object's methods. When a programmer changes the structure of an object's internal data, he or she also modifies the object's methods so they may properly operate on the data. The way in which outside code interacts with the methods, however, does not change.

19. The ability to reuse software components many times.

21. A class is a description of an object. An object is an instance of the class that exists in memory.

23. A text editor.

25. Byte code

27.

    a) `LabAssignment.class`

    b) The byte code generated by the compiler.

    c) `java LabAssignment`

Chapter 2

Multiple Choice and True/False

1. c
3. a
5. a, c, and d
7. c
9. a
11. b
13. a
15. a
17. True
19. True
21. False

Predict the Output

1.
```
0

100
```

3.
```
I am the incrediblecomputing
machine
and I will
amaze
you.
```

5.
```
23
1
```

Find the Error

- The comment symbols in the first line are reversed. They should be / * and * / .
- The word `class` is missing in the second line. It should read `public class MyProgram`.
- The `main` header should not be terminated with a semicolon.
- The fifth line should have a left brace, not a right brace.
- The first four lines inside the `main` method are missing their semicolons.
- The comment in the first line inside the `main` method should begin with forward slashes (//), not backward slashes.
- The last line inside the main method, a call to `println`, uses a string literal, but the literal is enclosed in single quotes. It should be enclosed in double quotes, like this: `"The value of c is"`.
- The last line inside the `main` method passes `C` to `println`, but it should pass `c` (lowercase).
- The closing brace for the class is missing.

Algorithm Workbench

1.
```
double temp, weight, age;
```

3.
a) `b = a + 2;`
b) `a = b * 4;`
c) `b = a / 3.14;`
d) `a = b – 8;`
e) `c = 'K';`
f) `c = 66;`

5.
a) `3.287E6`
b) `-978.65E12`
c) `7.65491E-3`

7.
```
int speed, time, distance;
speed = 20;
time = 10;
distance = speed * time;
System.out.println(distance);
```

9.
```
double income;
// Create a Scanner object for keyboard input.
Scanner keyboard = new Scanner(System.in);
// Ask the user to enter his or her desired income
System.out.print("Enter your desired annual income: ");
income = keyboard.nextDouble();
```

11.
```
total = (float)number;
```

Short Answer

1. Multiline style
3. A self-documenting program is written in such a way that you get an understanding of what the program is doing just by reading its code.
5. The `print` and `println` methods are members of the `out` object. The `out` object is a member of the `System` class. The `System` class is part of the Java API.
7. You should always choose names for your variables that give an indication of what they are used for. The rather nondescript name, `x`, gives no clue as to what the variable's purpose is.
9. In both cases you are storing a value in a variable. An assignment statement can appear anywhere in a program. An initialization, however, is part of a variable declaration.
11. Programming style refers to the way a programmer uses spaces, indentations, blank lines, and punctuation characters to visually arrange a program's source code. An inconsistent programming style can create confusion for a person reading the code.
13. `javadoc SalesAverage.java`

Chapter 3

Multiple Choice and True/False

1. a
3. d
5. b
7. a
9. b
11. a
13. d
15. True
17. True
19. False

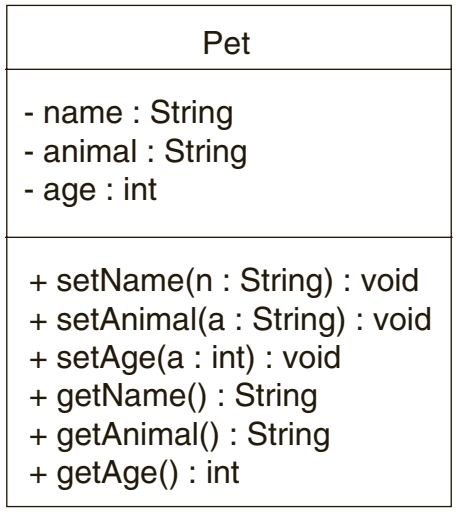
Find the Error

1. The constructor cannot have a return type, not even `void`.
3. The parentheses are missing. The statement should read:

```
Rectangle box = new Rectangle();
```

Algorithm Workbench

1. a) UML diagram:



b) Class code:

```
public class Pet
{
   private String name;     //  The pet's name
   private String animal;   //  The type of animal
   private int age;         //  The pet's age

   /**
    * setName method
    */

   public void setName(String n)
   {
      name = n;
   }

   /**
    * setAnimal method
    */

   public void setAnimal(String a)
   {
      animal = a;
   }

   /**
    * setAge method
    */
```

```
    public void setAge(int a)
    {
       age = a;
    }

    /**
     * getName method
     */

    public String getName()
    {
       return name;
    }

    /**
     * getAnimal method
     */

    public String getAnimal()
    {
       return animal;
    }

    /**
     * getAge method
     */

    public int getAge()
    {
       return age;
    }
}
```

3. a) After eliminating duplicates, objects, and primitive values, the potential classes are: *bank*, *account*, and *customer*

   b) The only class needed for this particular problem is *account*.

   c) The account class knows its balance and interest rate.
   The account can calculate interest earned.

Short Answer

1. A class is a collection of programming statements that specify the attributes and methods that a particular type of object may have. You should think of a class as a "blueprint" that describes an object. An instance of a class is an actual object that exists in memory.

3. An accessor method is a method that gets a value from a class's field but does not change it. A mutator method is a method that stores a value in a field or in some other way changes the value of a field.

5. Methods that are members of the class.

7. It looks in the current folder or directory for the file `Customer.class`. If that file does not exist, the compiler searches for the file `Customer.java` and compiles it. This creates the file `Customer.class`, which makes the `Customer` class available. The same procedure is followed when the compiler searches for the `Account` class.

9. Because they execute when an object is created.

11. An argument is a value that is passed into a method. A parameter variable is a variable in the method that holds a copy of the argument.

13. The default constructor.

Chapter 4

Multiple Choice and True/False

1. b
3. a
5. c
7. a
9. a
11. a
13. c
15. False
17. True
19. False

Find the Error

1. Each `if` clause is prematurely terminated by a semicolon.
3. The conditionally-executed blocks of code should be enclosed in braces.
5. The `!` operator is only applied to the variable `x`, not the expression. The code should read:

```
if (!(x > 20))
```

7. The statement should use the `||` operator instead of the `&&` operator.
9. The `equalsIgnoreCase` method should be used instead of the `equals` method.

Algorithm Workbench

1.
```
if (y == 0)
   x = 100;
```

3.
```
if (sales < 10000)
   commission = .10;
else if (sales <= 15000)
   commission = .15;
else
   commission = .20;
```

5.
```
if (amount1 > 10)
   if (amount2 < 100)
       System.out.println(amount1 > amount2 ? amount1 : amount2);
```

7.
```
if (temperature >= -50 && temperature <= 150)
   System.out.printn("The number is valid.");
```

9.
```
if (title1.compareTo(title2) < 0)
   System.out.println(title1 + " " + title2);
else
   System.out.println(title2 + " " + title1);
```

11. C, A, B

13. "0.00"

Short Answer

1. Conditionally executed code is executed only under a condition, such as an expression being `true`.

3. By indenting the conditionally executed statements, you are causing them to stand out visually. This is so you can tell at a glance what part of the program the `if` statement executes.

5. A flag is a `boolean` variable that signals when some condition exists in the program. When the flag variable is set to `false`, it indicates the condition does not yet exist. When the flag variable is set to `true`, it means the condition does exist.

7. It takes two `boolean` expressions as operands and creates a `boolean` expression that is `true` only when both subexpressions are `true`.

9. It determines whether a specific relationship exists between two values. The relationships are greater-than, less-than, equal-to, not equal-to, greater-than or equal-to, and less-than or equal-to.

Chapter 5

Multiple Choice and True/False

1. a
3. c
5. a
7. b
9. c
11. a
13. a
15. d
17. d
19. True
21. False
23. False
25. True

Find the Error

1. The conditionally-executed statements should be enclosed in a set of braces. Also, the `again` variable should be initialized with either 'y' or 'Y'.
3. The expression being tested by the `do-while` loop should be `choice == 1`. Also, the `do-while` loop must be terminated by a semicolon.

Algorithm Workbench

1.
```
Scanner keyboard = new Scanner(System.in);
int product = 0, num;
while (product < 100)
{
   num = keyboard.nextInt();
   product = num * 10;
}
```

3.
```
for (int x = 0; x <= 1000; x += 10)
{
   if (x < 1000)
      System.out.print(x + ", ");
   else
      System.out.println(x);
}
```

5.
```
int total = 0;
for (int num = 1, denom = 30; num <= 30; num++, denom--)
   total = num / denom;
```

7.
```
Scanner keyboard = new Scanner(System.in);
int x;
do
{
       System.out.print("Enter a number: ");
       x = keyboard.nextInt();
} while (x > 0);
```

9.
```
for (int count = 0; count < 50; count++)
   System.out.println("count is " + count);
```

11.
```
Scanner keyboard = new Scanner(System.in);
int number;
System.out.print("Enter a number in the range " +
                 "of 1 through 5: ");
number = keyboard.nextInt();
while (number < 1 || number > 5)
{
     System.out.print("Invalid number. Enter a " +
                      "number in the range " +
                      "of 1 through 5: ");
     number = keyboard.nextInt();
}
```

13.
```
PrintWriter outFile = new PrintWriter("NumberList.txt");
for (int i = 1; i <= 100; i++)
   outFile.println(i);
outFile.close();
```

15.
```
File file = new File("NumberList.txt");
Scanner inFile = new Scanner(file);
int input, total = 0;
while (inFile.hasNext())
{
   input = inFile.nextInt();
   total += input;
}
inFile.close();
System.out.println(total);
```

Short Answer

1. In postfix mode the operator is placed after the operand. In prefix mode the operator is placed before the variable operand. Postfix mode causes the increment or decrement operation to happen after the value of the variable is used in the expression. Prefix mode causes the increment or decrement to happen first.

3. A pretest loop tests its test expression before each iteration. A posttest loop tests its test expression after each iteration.

5. The `while` loop is a pretest loop and the `do-while` loop is a posttest loop.
7. The `do-while` loop.
9. An accumulator is used to keep a running total of numbers. In a loop, a value is usually added to the current value of the accumulator. If it is not properly initialized, it will not contain the correct total.
11. There are many possible examples. A program that asks the user to enter a business's daily sales for a number of days, and then displays the total sales is one example.
13. Sometimes the user has a list of input values that is very long, and doesn't know the number of items there are. When the sentinel value is entered, it signals the end of the list, and the user doesn't have to count the number of items in the list.
15. There are many possible examples. One example is a program that asks for the average temperature for each month, for a period of five years. The outer loop would iterate once for each year and the inner loop would iterate once for each month.
17. Closing a file writes any unsaved data remaining in the file buffer.
19. After the `println` method writes its data, it writes a newline character. The `print` method does not write the newline character.
21. The file does not exist.
23. You create an instance of the `FileWriter` class to open the file. You pass the name of the file (a string) as the constructor's first argument, and the `boolean` value `true` as the second argument. Then, when you create an instance of the `PrintWriter` class, you pass a reference to the `FileWriter` object as an argument to the `PrintWriter` constructor. The file will not be erased if it already exists and new data will be written to the end of the file.

Chapter 6

Multiple Choice and True/False

1. c
3. b
5. a
7. b
9. d
11. a
13. a
15. False
17. False
19. False

21. False

23. True

Find the Error

1. The static method `setValues` cannot refer to the non-static fields `x` and `y`.

3. The square methods must have different parameter lists. Both accept an `int`.

Algorithm Workbench

1.

a)
```
public Circle()
{
   radius = 0;
}
```

b)
```
public Circle(double r)
{
   radius = r;
}
```

c)
```
public String toString()
{
   String str;
   str = "Radius: " + radius;
   return str;
}
```

d)
```
public boolean equals(Circle c)
{
   boolean status;

   if (c.getRadius() == radius)
     status = true;
   else
     status = false;

   return status;
}
```

e)
```
public boolean greaterThan(Circle c)
{
   boolean status;

   if (c.getArea() > getArea())
     status = true;
   else
     status = false;

   return status;
}
```

Short Answer

1. Access an instance variable or instance method of the class without the context of an object.

3.
   a) 3
   b) 3
   c) 1
   d) 0
   e) `Thing.putThing(5);`

5. Several different versions of the same method can be created, each performing a different operation.

7. The default `equals` method returns true if the memory addresses of the two objects being compared are the same.

9. The program will crash.

11. The key word `this` is the name of a reference variable that an object can use to refer to itself. It is available to all non-static methods.

13. a)
```
POODLE
BOXER
TERRIER
```
    b)
```
0
1
2
```
    c)
```
BOXER is NOT greater than TERRIER
```

Chapter 7

Multiple Choice and True/False

1. b
3. b
5. c
7. b
9. d
11. c
13. a
15. True
17. True

19. True
21. True
23. True

Find the Error

1. The size declarator cannot be negative.
3. The loop uses the values 1 through 10 as subscripts. It should use 0 through 9.
5. A subscript should be used with words, such as `words[0].toUpperCase()`.

Algorithm Workbench

1.
```
for (int i = 0; i < 20; i++)
    System.out.println(names[i]);
```

3.

a)
```
String[] scientists = {"Einstein", "Newton",
                       "Copernicus", "Kepler"};
```

b)
```
for (int i = 0; i < scientists.length; i++)
    System.out.println(scientists[i]);
```

c)
```
int total = 0;
for (int i = 0; i < scientists.length; i++)
    total += scientists[i].length();
System.out.println("The total length is " + total);
```

5.
```
// Define the arrays.
int[] id = new int[10];
double[] weeklyPay = new double[10];
// Display each employee's gross weekly pay.
for (int i = 0; i < 10; i++)
{
    System.out.println("The pay for employee "
                       + id[i] + " is $"
                       + weeklyPay[i]);
}
```

7.
```
final int NUM_ROWS = 30;
final int NUM_COLS = 10;
int total = 0;
```

```
double average;
for (int row = 0; row < grades.length; row++)
{
    for (int col = 0; col < grades[row].length; col++)
    {
        total += grades[row][col];
   }
}
average = (double) total / (NUM_ROWS * NUM_COLS);
```

9.
```
double total = 0.0; // Accumulator
// Sum the values in the array.
for (int row = 0; row < 10; row++)
{
    for (int col = 0; col < 20; col++)
        total += values[row][col];
}
```

11.
```
// Create an ArrayList.
ArrayList<String> cars = new ArrayList<String>();
// Add three car names to the ArrayList.
cars.add("Porsche");
cars.add("BMW");
cars.add("Jaguar");
// Display the contents of cars.
for (String str : cars)
   System.out.println(str);
```

Short Answer

1. The size declarator is used in a definition of an array to indicate the number of elements the array will have. A subscript is used to access a specific element in an array.

3. a) 2

   b) 14

   c) 8

5. Because this statement merely makes `array1` reference the same array that `array2` references. Both variables will reference the same array. To copy the contents of `array2` to `array1`, the contents of `array2's` individual elements will have to be assigned to the elements of `array1`.

7. It will have to read all 10,000 elements to find the value stored in the last element.

Chapter 8

Multiple Choice and True/False

1. c
3. a
5. a
7. b
9. d
11. a
13. d
15. a
17. True
19. True
21. False
23. False
25. True

Find the Error

1. The `valueOf` method is static. It must be called like this:

```
str = String.valueOf(number);
```

3. The statement should read:

```
number = Integer.parseInt(str);
```

5. If anything other than whitespace is to be used as a delimiter, then the delimiter must be passed as an argument to the `StringTokenizer` constructor, such as:

```
StringTokenizer strTokenizer =
       new StringTokenizer("One;Two;Three", ";");
```

Algorithm Workbench

1.
```
if (Character.toUpperCase(choice) == 'Y')
```
Or
```
if (Character.toLowerCase(choice) == 'y')
```

3.
```
int total = 0;
for (int i = 0; i < str.length(); i++)
{
    if (Character.isDigit(str.charAt(i)))
      total++;
}
```

5.
```
public static boolean dotCom(String str)
{
   boolean status;
   if (str.endsWith(".com"))
      status = true;
   else
      status = false;
   return status;
}
```

7.
```
public static void upperT(StringBuilder str)
{
   for (int i = 0; i < str.length(); i++)
   {
       if (str.charAt(i) == 't')
           str.setCharAt(i, 'T');
   }
}
```

9.
```
if (d <= Integer.MAX_VALUE)
   i = (int) d;
```

11.
```
value = Double.parseDouble(str);
```

Short Answer

1. This will improve the program's efficiency by reducing the number of `String` objects that must be created and then removed by the garbage collector.

3. They accept a `String` argument and return the contents of the argument converted to a numeric data type.

5. Each of the numeric wrapper classes has `final` static fields named `MAX_VALUE` and `MIN_VALUE`. These fields hold the maximum and minimum values for the data type.

Chapter 9

Multiple Choice and True/False

1. b
3. d
5. a
7. c
9. a
11. a
13. b
15. a
17. True
19. False
21. True
23. False
25. False
27. False

Find the Error

1. The `Car` class header should use the word `extends` instead of `expands`.
3. Because the `Vehicle` class does not have a default or no-arg constructor, the `Car` class constructor must call the `Vehicle` class constructor.

Algorithm Workbench

1. `public class Poodle extends Dog`
3.

```
public abstract class B
{
   private int m;
   protected int n;
   public void setM(int value)
   {
      m = value;
   }
   public void setN(int value)
   {
      n = value;
   }
```

```
   public int getM()
   {
      return m;
   }
   public int getN()
   {
      return n;
   }
   public abstract double calc();
}
public class D extends B
{
   private double q;
   protected double r;
   public void setQ(double value)
   {
      q = value;
   }
   public void setR(double value)
   {
      r = value;
   }
   public double getQ()
   {
      return q;
   }
   public double getR()
   {
      return r;
   }
      public double calc()
   {
      return q * r;
   }
}
```

5. `setValue(10);`

   Or

   `super.setValue(10);`

7.
```
public class Stereo extends SoundSystem
                    implements CDPlayable,
                               TunerPlayable,
                               CassettePlayable
```

Short Answer

1. When an "is a" relationship exists between objects, it means that the specialized object has all of the characteristics of the general object, plus additional characteristics that make it special.

3. `Dog` is the superclass and `Pet` is the subclass.

5. No.

7. Overloading is when a method has the same name as one or more other methods, but a different parameter list. Although overloaded methods have the same name, they have different signatures. When a method overrides another method, however, they both have the same signature.

9. At runtime.

11. An abstract class is not instantiated itself, but serves as a superclass for other classes. The abstract class represents the generic or abstract form of all the classes that inherit from it.

Chapter 10

Multiple Choice and True/False

1. b

3. a

5. b

7. d

9. c

11. c

13. d

15. c

17. True

19. False

21. True

23. False

Find the Error

1. The `try` block must appear first.

3. The `catch (Exception e)` statement and its block should appear after the other catch blocks, because this is a more general exception than the others.

Algorithm Workbench

1.

```
B
D
```

3.

```
public static int arraySearch(int[] array, int value)
                                throws Exception
{
   int i;  // Loop control variable
   int element;      // Element the value is found at
   boolean found;    // Flag indicating search results
   // Element 0 is the starting point of the search.
   i = 0;
   // Store the default values element and found.
   element = -1;
   found = false;
   // Search the array.
   while (!found && i < array.length)
   {
      if (array[i] == value)
      {
         found = true;
         element = i;
      }

      i++;
   }
   if (element == -1)
      throw new Exception("Element not found.");
   else
      return element;
}
```

5.

```
public class NegativeNumber extends Exception
{
   /**
    * No-Arg constructor
    */

   public NegativeNumber()
   {
      super("Error: Negative number");
   }
```

```
    /**
     * The following constructor accepts the number that
     * caused the exception.
     */

    public NegativeNumber(int n)
    {
        super("Error: Negative number: " + n);
    }
}
```

7.
```
public int getValueFromFile() throws IOException,
                                     FileNotFoundException
```

9.
```
FileOutputStream fstream =
     new FileOutputStream("Configuration.dat");
```

Short Answer

1. An exception object has been created in response to an error that has occurred.
3. Control of the program is passed to the previous method in the call stack (that is, the method that called the offending method). If that method cannot handle the exception, then control is passed again, up the call stack, to the previous method. This continues until control reaches the `main` method. If the `main` method does not handle the exception, then the program is halted and the default exception handler handles the exception.
5. The first statement after that `try/catch` construct.
7. Any object derived from the `Throwable` class.
9. Unchecked exceptions are those that are derived from the `Error` class or the `RuntimeException` class. You should not handle these exceptions because the conditions that cause them can rarely be dealt within the program. All of the remaining exceptions (that is, those that are not derived from `Error` or `RuntimeException`) are *checked* exceptions. These are the exceptions that you should handle in your program.
11. A binary file contains raw binary data. A text file contains data that is formatted as Unicode characters.
13. Object serialization is the process of converting an object to a series of bytes and saving them to a file. Deserialization is the process of reconstructing a serialized object.

Chapter 11

Multiple Choice and True/False

1. c
3. d
5. b
7. c
9. a
11. a
13. a
15. c
17. d
19. a
21. False
23. False
25. True
27. False
29. False

Find the Error

1. The x is missing in the package name. The statement should read:

   ```
   import javax.swing.*;
   ```

3. The arguments passed to the `GridLayout` constructor are reversed. This statement creates 10 rows and 5 columns.

5. You do not create an instance of `BorderFactory`. Instead you call one of its static methods to create a `Border` object. The statement should read:

   ```
   panel.setBorder(BorderFactory.createTitledBorder("Choices"));
   ```

Algorithm Workbench

1. ```
   JOptionPane.showMessageDialog(null, "Have a nice day.");
   ```

3. ```
   String str;
   double temp;
   str = JOptionPane.showInputDialog("Enter the temperature");
   temp = Double.parseDouble(str);
   if (temp <= 50)
      JOptionPane.showMessageDialog(null, "A bit cold!");
   ```

```
else if (temp > 50 && temp < 80)
   JOptionPane.showMessageDialog(null, "Nice day!");
else
   JOptionPane.showMessageDialog(null, "A bit warm!");
```

5. `myWindow.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);`

7.
```
myButton.setBackground(Color.WHITE);
myButton.setForeground(Color.RED);
```

9. `setLayout(new GridLayout(5, 10));`

11.
```
// Create three radio buttons.
JRadioButton radio1 = new JRadioButton("Option 1", true);
JRadioButton radio2 = new JRadioButton("Option 2");
JRadioButton radio3 = new JRadioButton("Option 3");
// Create a ButtonGroup object.
ButtonGroup group = new ButtonGroup();
// Add the radio buttons to the ButtonGroup object.
group.add(radio1);
group.add(radio2);
group.add(radio3);
```

Short Answer

1. The `JOptionPane.showInputDialog` returns the value entered by the user.

3. Because Swing generates a thread, which is a process running in the computer. If the `System.exit` method is not called, this thread will continue to execute, even after the end of the `main` method has been reached.

5. This prevents the component from being resized. The `BorderLayout` manager resizes components to fill up any extra space in a region. When you place a component inside a panel, and then place the panel in a `BorderLayout` region, the panel is resized instead of the component it contains.

7. By creating a class that extends `JPanel`.

Chapter 12

Multiple Choice and True/False

1. d

3. b

5. a

7. c

9. b

11. c

13. a
15. b
17. a
19. b
21. a
23. False
25. False
27. True
29. True
31. True
33. False
35. True

Find the Error

1. The argument `false` should have been passed to the `setEditable` method.
3. You should pass `list` as an argument to the `JScrollPane` constructor:

```
JScrollPane scrollPane = new JScrollPane(list);
```

5. The second statement should read:

```
label.setIcon(image);
```

7. The statement should read:

```
JTextArea textArea = new JTextArea (5, 20);
```

Algorithm Workbench

1.
```
JTextField textField = new JTextField(20);
textField.setEditable(false);
```

3.
```
dayList.setVisibleRowCount(4);
JScrollPane scrollPane = new JScrollPane(dayList);
```

5.
```
selectionIndex = myComboBox.getSelectedIndex();
```

7.
```
ImageIcon image = new ImageIcon("picture.gif");
label.setIcon(image);
```

9.
```
JFileChooser fileChooser = new JFileChooser();
 int status = fileChooser.showOpenDialog(null);
 if (status == JFileChooser.APPROVE_OPTION)
 {
```

```
        File selectedFile = fileChooser.getSelectedFile();
        String filename = selectedFile.getPath();
      }
```

11.

```
// Create an Open menu item.
JMenuItem openItem = new JMenuItem("Open");
openItem.setMnemonic(KeyEvent.VK_O);
openItem.addActionListener(new OpenListener());

// Create a Print menu item.
JMenuItem exitItem = new JMenuItem("Print");
exitItem.setMnemonic(KeyEvent.VK_P);
exitItem.addActionListener(new PrintListener());

// Create an Exit menu item.
JMenuItem exitItem = new JMenuItem("Exit");
exitItem.setMnemonic(KeyEvent.VK_X);
exitItem.addActionListener(new ExitListener());

// Create a JMenu object for the File menu.
JMenu fileMenu = new JMenu("File");
fileMenu.setMnemonic(KeyEvent.VK_F);

// Add the menu items to the File menu.
fileMenu.add(openItem);
fileMenu.add(printItem);
fileMenu.add(exitItem);

// Create the menu bar.
JMenuBar menuBar = new JMenuBar();

// Add the file menu to the menu bar.
menuBar.add(fileMenu);

// Set the window's menu bar.
setJMenuBar(menuBar);
```

Short Answer

1. Single selection mode
3. An uneditable combo box combines a button with a list, and allows the user to only select items from its list. An editable combo box combines a text field and a list. In addition to selecting items from the list, the user may also type input into the text field. The default type of combo box is uneditable.
5. A mnemonic is a key on the keyboard that you press in combination with the Alt key to quickly access a component such as a button. When you assign a mnemonic to a button, the user can click the button by holding down the Alt key and pressing the mnemonic key.

7. A tool tip is text that is displayed in a small box when the user holds the mouse cursor over a component. The box usually gives a short description of what the component does.

9. The item is deselected, which causes the check mark to disappear. The checked menu item component also generates an action event.

11. Because, as the user moves the `JSlider` component's knob, it will only take on values within its established range.

Chapter 13

Multiple Choice and True/False

1. c
3. b
5. b
7. d
9. c
11. a
13. d
15. c
17. c
19. c
21. b
23. True
25. True
27. False
29. True
31. False
33. False
35. False

Find the Error

1. The tag should specify the file `MyApplet.class` instead of `MyApplet.java`.
3. Call `repaint` instead of `paint`.
5. The class must provide all of the methods specified by the `MouseListener` interface.

Algorithm Workbench

1. `<CENTER><H1>My Home Page</H1></CENTER>`

3. Line 1: Change `JFrame` to `JApplet`
   Line 3: Change to `public void init()`
   Line 5: Delete
   Line 6: Delete
   Line 8: Delete
   Line 9: Delete
   Line 15: Delete
   Line 16: Delete
   Line 17: Delete

5.
```
private class MyMouseMotionListener extends MouseAdapter
{
   public void mouseMoved(MouseEvent e)
   {
      mouseMovments += 1;
   }
}
```

Short Answer

1. It is executed by the user's system.

3. Applets are important because they can be used to extend the capabilities of a Web page. Web pages are normally written in Hypertext Markup Language (HTML). HTML is limited, however, because it merely describes the content and layout of a Web page, and creates links to other files and Web pages. HTML does not have sophisticated abilities such as performing math calculations and interacting with the user. A programmer can write a Java applet to perform these types of operations and associate it with a Web page.

5. Some browsers, such as Microsoft Internet Explorer and older versions of Netscape Navigator, do not directly support the Swing classes in applets. These browsers require a plug-in in order to run applets that use Swing components. If you are writing an applet for other people to run on their computers, there is no guarantee that they will have the required plug-in. If this is the case, you should use the AWT classes instead of the Swing classes for the components in your applet.

7. When the component is first displayed and is called again any time the component needs to be redisplayed

9. If you want to load the sound file and keep it in memory so it can be played more than once, or if you want to play the sound file repeatedly.

Chapter 14

Multiple Choice and True/False

1. b
3. d
5. d
7. True
9. False

Find the Error

1. The recursive method, `myMethod`, has no base case. So, it has no way of stopping.

Algorithm Workbench

1.
```
public static void main(String [] args)
{
   String str = "test string";
   display(str, 0);
}

public static void display(String str, int pos)
{
   if (pos < str.length())
   {
      System.out.print(str.charAt(pos));
      display(str, pos + 1);
   }
}
```
3. `10`
5. `55`
7.
```
public static int factorial(int num)
{
   int fac = 1;
   for (int i = 1; i <=num; i++)
   {
       fac = fac * i;
   }
       return fac;
}
```

Short Answer

1. An iterative algorithm uses a loop to solve the problem, while a recursive algorithm uses a method that calls itself.

3. For question 3 the base case is reached when `arg` is equal to 10. For question 4 the base case is also reached when `arg` is equal to 10. For question 5 the base case is reached when `num` is less-than or equal to 0.

5. Recursive algorithms are usually less efficient than iterative algorithms. This is because a method call requires several actions to be performed by the JVM, such as allocating memory for parameters and local variables, and storing the address of the program location where control returns after the method terminates.