APPENDIX M

Answers to Odd-Numbered Review Questions

Chapter 1

Multiple Choice

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 machine language programs are streams of binary numbers, and high-level language programs are made up of words.

7. *Syntax errors* are mistakes that the programmer has made that violate the rules of the programming language. *Logical errors* are mistakes that cause the program to produce erroneous results.

9. A program that translates source code into executable code.

11. Machine language code is executed directly by the CPU. Byte code is executed by the JVM.

13. Object-oriented programming

15. The object's methods.

17. No

19. `javac LabAssignment.java`

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 class is missing its closing brace.

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. `-9.7865E12`

   c. `7.65491E-3`

7. `10 20 1`

9. `a`

11.
```
int speed, time, distance;
speed = 20;
time = 10;
distance = speed * time;
System.out.println(distance);
```

13.
```
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();
```

15. `total = (float)number;`

Short Answer

1. Multi-line 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. b
3. a
5. c
7. a
9. a
11. a
13. c
15. c
17. True
19. True
21. True

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)
      {
         if (amount1 > amount2)
   ```
   ```

```
            {
               System.out.println(amount1);
            }
            else
            {
               System.out.println(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. `System.out.printf("%,.2f", number);`

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. They determine 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 4

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. The following code simply prints the numbers, separated by spaces.

```
for (int x = 0; x <= 1000; x += 10)
   System.out.print(x + " ");
```

The following code prints the numbers separated by commas.

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

5.
```
double 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.
```
int number;
Scanner keyboard = new Scanner(System.in);
System.out.print("Enter a number in the range " +
                 "of 1 through 4: ");
number = keyboard.nextInt();
while (number < 1 || number > 4)
{
   System.out.print("Invalid number. Enter a " +
                    "number in the range " +
                    "of 1 through 4: ");
   number = keyboard.nextInt();
}
```

13.
```
for (int r = 7; r > 0; r--)
{
   for (int c = 0; c < r; c++)
   {
      System.out.print("*");
   }
   System.out.println();
}
```

15.
```
import java.util.Random;
public class ReviewQuestion15
{
   public static void main(String[] args)
   {
      Random rand = new Random();
      System.out.println(rand.nextInt(10) + 1);
   }
}
```

17.
```
PrintWriter outFile = new PrintWriter("NumberList.txt");
for (int i = 1; i <= 100; i++)
   outFile.println(i);
outFile.close();
```

19.
```
File file = new File("NumberList.txt");
Scanner inFile = new Scanner(file);
String input;
int number, total = 0;
```

```
while (inFile.hasNext())
{
   input = inFile.nextLine();
   number = Integer.parseInt(input);
   total += number;
}
System.out.println("The total is " + total);
inFile.close();
```

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 5

Multiple Choice and True/False

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

Find the Error

1. The header should not be terminated with a semicolon.
3. The method should have a `return` statement that returns a `double` value.

Algorithm Workbench

1. `doSomething(25);`
3. The value 3 will be stored in `a`, 2 will be stored in `b`, and 1 will be stored in `c`.
5. `result = cube(4);`
7.
```
public static double timesTen(double num)
{
   return num * 10.0;
}
```
9.
```
// Assume java.util.Scanner has been imported.
public static String getName()
{
   String name;
   Scanner keyboard = new Scanner(System.in);
   System.out.print("Enter your first name: ");
   name = keyboard.nextLine();
   return name;
}
```

Short Answer

1. A large complex problem is broken down into smaller manageable pieces. Each smaller piece of the problem is then solved.

3. An argument is a value that is passed into a method when the method is called. A parameter variable is a variable that is declared in the method header, and receives the value of an argument when the method is called.

5. When an argument is passed to a method, only a copy of the argument is passed. The method cannot access the actual argument.

Chapter 6

Multiple Choice and True/False

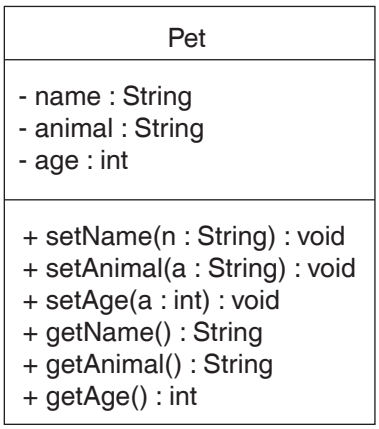
1. a
3. d
5. b
7. d
9. b
11. c
13. b
15. True
17. False
19. False

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();
   ```
5. The `square` methods must have different parameter lists. Both accept an `int`.

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
      @param n The pet's name.
   */
   public void setName(String n)
   {
      name = n;
   }
   /**
      setAnimal method
      @param a The type of animal.
   */
   public void setAnimal(String a)
   {
      animal = a;
   }
   /**
      setAge method
      @param a The pet's age.
   */
   public void setAge(int a)
   {
      age = a;
   }
   /**
      getName method
      @return The pet's name.
   */
   public String getName()
   {
      return name;
   }
   /**
      getAnimal method
```

```
         @return The type of animal.
      */
      public String getAnimal()
      {
         return animal;
      }
      /**
         getAge method
         @return The pet's age.
      */
      public int getAge()
      {
         return age;
      }
   }
```

3. a.
```
   public Square()
   {
      sideLength = 0.0;
   }
```
   b.
```
   public Square(double s)
   {
      sideLength = s;
   }
```

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 same 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. If you do not write a constructor for a class, Java automatically provides one.

11. By their signatures, which include the method name and the data types of the method parameters, in the order that they appear.

Chapter 7

Multiple Choice and True/False

1. b
3. b
5. c
7. b
9. d
11. c
13. a
15. False
17. True
19. True
21. False
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. a.
```
// Define the arrays.
int[] id = new int[10];
double[] weeklyPay = new double[10];
```

   b.
```
// 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. b
7. d
9. c
11. b
13. False
15. False
17. True

Find the Error

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

Algorithm Workbench

1. a.
```
public String toString()
{
   String str;
   str = "Radius: " + radius + " Area: " + getArea();
   return str;
}
```
b.
```
public boolean equals(Circle c)
{
   boolean status;
   if (c.getRadius() == radius)
      status = true;
   else
      status = false;
   return status;
}
```
c.
```
public boolean greaterThan(Circle c)
{
   boolean status;
   if (c.getArea() > getArea())
      status = true;
   else
      status = false;
   return status;
}
```

3.
```
enum Pet { DOG, CAT, BIRD, HAMSTER }
```

Short Answer

1. Access a non-static member.
3. When a variable is passed as an argument, a copy of the variable's contents is passed. The receiving method does not have access to the variable itself. When an object is passed as an argument, a reference to the object (which is the object's address) is passed. This allows the receiving method to have access to the object.
5. It means that an aggregate relationship exists. When an object of class B is a member of class A, it can be said that class A "has a" class B object.
7. It is not advisable because it will allow access to the private fields. The exception to this is when the field is a `String` object. This is because `String` objects are immutable, meaning that they cannot be changed.
9. a) `Color`
   b) `Color.RED, Color.ORANGE, Color.GREEN, Color.BLUE`
   c) `Color myColor = Color.BLUE;`
11. When there are no references to it.

Chapter 9

Multiple Choice and True/False

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

Find the Error

1. The `valueOf` method is static. It should be called like this:
   `str = String.valueOf(number);`
3. The very first character is at position 0, so the statement should read:
   `str.setCharAt(0, 'Z');`

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;
```

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. Converts a number to a string.

Chapter 10

Multiple Choice and True/False

1. b

3. d

5. a

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

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 constructor or a 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
```

9. `Computable half = x -> x / 2;`

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.
13. The class must implement an interface, or extend a superclass.
15. An expression that creates an object that implements a functional interface.

Chapter 11

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.
      @param n The number.
   */
   public NegativeNumber(int n)
   {
      super("Error: Negative number: " + n);
   }
}
```

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

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

11.
```
FileOutputStream outStream =
                     new FileOutputStream("ObjectData.dat");
ObjectOutputStream objectOutputFile =
                     new ObjectOutputStream(outStream);
objectOutputFile.writeObject(r);
```

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 that inherits from the `Throwable` class.

9. Unchecked exceptions are those that inherit from the `Error` class or the `RuntimeException` class. You should not handle these exceptions because the conditions that cause them can rarely be dealt with within the program. All of the remaining exceptions (that is, those that do not inherit from `Error` or `RuntimeException`) are *checked* exceptions. These are the exceptions that you should handle in your program.

11. All of the data stored in a text file is formatted as text. Even numeric data is converted to text when it is stored in a text file. You can view the data stored in a text file by opening it with a text editor, such as Notepad. In a binary file, data is not formatted as text. Subsequently, you cannot view a binary file's contents with a text editor.

13. When an object is serialized it is converted to a series of bytes, which are written to a file. When the object is deserialized, the series of bytes are converted back into an object.

Chapter 12

Multiple Choice and True/False

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

Find the Error

1. The parameter for the overridden `start` method must be a `Stage` object, not a `Scene` object.
3. The argument that is passed to the `setAlignment` method should be `Pos.CENTER`.
5. The correct code is:

```
myImageView.setFitWidth(100);
myImageView.setFitHeight(100);
```

Algorithm Workbench

1. `HBox hbox = new HBox(label1, label2, label3);`
3. `primaryStage.setScene(scene);`
5.
```
Image image = new Image("file:Cat.png");
ImageView imageView = new ImageView(image);
```
7. `HBox hbox = new HBox(10, label1, label2, label3);`
9. `gridPane.add(button, 2, 5);`
11.
```
class ButtonClickHandler implements EventHandler<ActionEvent>
{
    @Override
    void handle(ActionEvent event)
    {
        outputLabel.setText("Hello World");
    }
}
```

13.
```
myButton.setOnAction(event ->
{
   outputLabel.setText("Hello World");
};
```

Short Answer

1. A program that responds to the user's actions.
3. What is the purpose of the `Application` class's abstract `start` method? It is the entry point for a JavaFX application.
5. `javafx.scene.control`
7. The aspect ratio is the ratio of the image's width to the image's height. If you change an image's aspect ratio, the image will appear stretched (either horizontally or vertically). The `ImageView` class has a method named `setPreserveRatio` that you can call to make sure that an image's aspect ratio is preserved.
9. You call the control's `getText` method. The method returns the value that has been entered into the TextField as a `String`.
11. BorderPane

Chapter 13

Multiple Choice and True/False

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

Find the Error

1. The = operator should be a colon.
3. The last two statements should be:

```
radio1.setToggleGroup(radioGroup);
radio2.setToggleGroup(radioGroup);
```

Algorithm Workbench

1.
```
.label {
   -fx-font-size: 24pt;
   -fx-font-style: italic;
   -fx-font-weight: bold;
}
```

3.
```
.label, .button, .text-field {
   -fx-font-size: 24pt;
   -fx-font-style: italic;
   -fx-font-weight: bold;
}
```

5.
```
rgb(100, 20, 255);
```

7.
```
calcButton.setId("calc-button");
```

9.
```
radio1.setToggleGroup(radioGroup);
radio2.setToggleGroup(radioGroup);
radio3.setToggleGroup(radioGroup);
```

11.
```
ListView<String> snackListView = new ListView<>();
snackListView.getItems().addAll("cheese", "olives", "crackers");
```

13.
```
Slider slider = new Slider(0.0, 1000.0, 500.0);
slider.setOrientation(Orientation.HORIZONTAL);
slider.setShowTickMarks(true);
slider.setMajorTickUnit(100);
slider.setMinorTickCount(25);
slider.setShowTickLabels(true);
```

Short Answer

1. A type selector is used to apply a style definition to all nodes of a specific type. An ID selector can be used to apply a style definition to a node with a specific ID.

3. `RadioButtons`

5. You group `RadioButtons` together with a `ToggleGroup` so only one of the `RadioButtons` can be selected at a time.

7. An editable `ComboBox` allows the user to select an item from a drop-down list, or type input into a field that is similar to a `TextField`. By default, `ComboBox` controls are uneditable, which means that the user cannot type input into the control; he or she can only select items from a dropdown list.

9. `ComboBox`

Chapter 14

Multiple Choice and True/False

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

Algorithm Workbench

1. `Line myLine = new Line(50, 25, 150, 125);`
3. `Rectangle myRectangle = new Rectangle(100, 200, 175, 150);`
5.
```
Polygon poly = new Polygon(100.0, 100.0,
                           200.0, 100.0,
                           200.0, 200.0,
                           100.0, 200.0);
```
7. `myBox.setRotate(45.0);`
9.
```
Circle myCircle = new Circle(0, 100, 30);
TranslateTransition ttrans =
         new TranslateTransition(new Duration(5000), myCircle);
ttrans.setFromX(0);
ttrans.setFromY(100);
ttrans.setToX(200);
ttrans.setToY(150);
ttrans.play();
```

Short Answer

1. (639, 0)
3. `setFromX` and `setFromY`
5. `setFromAngle`
7. `setByAngle`

9. `setToX` and `setToY`
11. `BoxBlur`, `GuassianBlur`, and `MotionBlur`

Chapter 15

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.
7. The value of an argument is usually reduced.

Chapter 16

Multiple Choice and True/False

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

Find the Error

1. The numbers in the array are not sorted.

Algorithm Workbench

1.

Array Size	50 Elements	500 Elements	10,000 Elements	100,000 Elements	10,000,000 Elements
Sequential Search (Average Comparisons)	25	250	5,000	50,000	5,000,000
Sequential Search (Maximum Comparisons)	50	500	10,000	100,000	10,000,000
Binary Search (Maximum Comparisons)	6	9	14	17	24

Short Answer

1. It will have to read all 10,000 elements of the array.
3. One time.

5. Because it moves the items in the array only by one element at a time.
7.
   - Move 1 to element 0. The array now looks like this:

     1, 4, 3, 2
   - Move 2 to element 1. The array now looks like this:

     1, 2, 3, 4
   - The values are now in order. No further swaps are made.

Chapter 17

Multiple Choice and True/False

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

Find the Error

1. The type argument must also be provided in the variable declaration. The statement should read:
   ```
   ArrayList<String> myList = new ArrayList<String>();
   ```
3. The static method cannot refer to the class's type parameter `T`. It must define its own type parameter.

Algorithm Workbench

1. ```
   ArrayList<Customer> customers = new ArrayList<Customer>();
   ```
3. ```
   public class MyType<T extends String>
   ```
5. ```
   public static <T extends Comparable> T max(T a, T b)
   {
     if (a.compareTo(b) > 0)
        return a;
     else
        return b;
   }
   ```
7. ```
   public class MyType<T extends Number, S extends String>
   ```

Short Answer

1. At compile time.
3. When the Java compiler encounters generic code, it uses a process known as erasure to compile the code. The process is called erasure because the compiler erases the generic notation and substitutes an actual type for each type parameter.
5. `Object`.

Chapter 18

Multiple Choice and True/False

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

Find the Error

1. The `LinkedHashSet` class does not implement the `List` interface. Replace with `LinkedList` or `ArrayList`.
3. You cannot use a `ListIterator` from an object that implements the `Set` interface.

Algorithm Workbench

1.
```
List<String> sList = new ArrayList<>();
```
3.
```
while (it.hasNext())
{
   System.out.println(it.next());
}
```

Short Answer

1. A source
3. You can change the class that implements the interface without changing the rest of the code.
5. By calling the `iterator()` method. Or if you want to, you can call the `listIterator()` method.
7. By internally generating code that uses an iterator.
9. Because `IntStream` allows streams of the primitive type `int`. You cannot get a stream of primitive types from `Stream<Integer>`.
11. Yes.
13. A map associates keys with values. A `HashMap` object will use the hash codes of the keys to find the keys and then from there, to find the values.

Chapter 19

Multiple Choice and True/False

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

Find the Error

1. Replace `ref++` with `ref = ref.next;`
3. You need to set the `prev` reference of the second node. Add the code:

```
if (myList.next != null)
   myList.next.prev = myList;
```

5. You need to check to see if the list is now empty, and you need to set the `next` reference of the last node to null.

```
if (last == null)
   first = null;
else
   last.next = null;
```

Algorithm Workbench

1.
```
static void print(Node ref)
{
   if (ref != null)
   {
      // Print the first element
      System.out.print(ref.value +  " ");
      // Print elements in the tail
      print(ref.next);
   }
}
```

3.
```
Node reverse(Node list)
{
   // Check base case
   if (list == null || list.next == null)
      return null;
   // reverse the tail
   Node revTail = reverse(list.next);
   // Put head at the end of the reversed tail
   list.next = null;
   Node last = revTail;
   while (last.next != null)
      last = last.next;
   last.next = list;

   // revTail is now head of the reversed list
   return revTail;
}
```

Short Answer

1. They need to be passed a `Node` reference as parameter. `Node` is an implementation detail that should not be made public.

3. It is the zero-based position of the element in the list.

5. Successive elements of the list are not necessarily stored in consecutive memory locations. Instead, the list stores with each element a reference to its successor.

Chapter 20

Multiple Choice and True/False

1. b

3. b

5. c

7. False
9. False

Find the Error

1. The flow of execution never reaches the statement `top --;`
3. `pop` forgets to remove the value from the stack.
5. `dequeue` should remove from the front, not the rear. Also, the index should wrap to the beginning of the array when it gets to the end.

Algorithm Workbench

1.
```
static boolean palindrome(String s)
{
   Stack<Character> stack = new Stack<Character>();
   // Push a copy of the string onto the stack
   for(int k = 0; k < s.length(); k++)
      stack.push(s.charAt(k));
   // Compare the two copies using the stack
   for (int k = 0; k < s.length(); k++)
   {
     if (s.charAt(k) != stack.pop())
        return false;
   }
   // No discrepancy found, so s is  a palindrome
   return true;
}
```

3. To add *x* to the queue, push *x* onto the first stack. To remove an item from the queue, pop everything from the first stack and store it into the second stack, then pop the item at the top of the second stack and save it to be returned. Pop everything from the second stack back onto the first stack. Return the saved item.

Short Answer

1. A stack
3. A stack
5. To facilitate garbage collection.

Chapter 21

Multiple Choice and True/False

1. d
3. c

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

Find the Error

1. The function does not use the values returned from the recursive calls. Should be:

```
if (tree!= null)
   return NodeCount(tree.left)+NodeCount(tree.right)+1;
else
   return 0;
```

3. The code will fail if `tree` is null.

Short Answer

1. It keeps the height of the tree shallow, making algorithms that traverse a path of the tree from the root to a leaf more efficient.
3. The length of the longest path from the root to a leaf.
5. A collection that stores elements from an ordered set, and from which elements are removed according to that order, with the least element in the order being removed first.
7. The AVL tree must satisfy a balance condition: the difference in the heights of the two subtrees of any node must be at most 1.

Algorithm Workbench

1.
```
Node preorder(Node bTree)
{
   if (tree != null)
   {
       System.out.print(bTree.value + " ");
       preorder(bTree.left);
       preorder(bTree.right);
   }
}
```

3.
```
int treeSize(Node bTree)
{
   if (bTree == null)
      return 0;
   else
      return treeSize(bTree.left) + treeSize(bTree.right) + 1;
}
```

5.
```
boolean contains(Node btree, int x)
{
   if (btree == null)
      return false;
   if (btree.value == x)
      return true;
   if (contains(btree.left, x))
      return true;
   if (contains(btree.right, x))
      return true;
   else
      return false;
}
```

7.
```
static class DecOrder implements Comparator<String>
{
   public int compare(String x, String y)
   {
      if (x.length() == y.length())
         return x.compareTo(y);
      else
         return x.length()-y.length();
   }
}
```

Chapter 22 (Available on the Book's Companion Website)

Multiple Choice and True/False

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

17. c
19. True
21. False
23. False

Find the Error

1. The string "French Roast Dark" should be enclosed in single quotes instead of double quotes.
3. The last statement should call the `executeQuery` method instead of the `execute` method.

Algorithm Workbench

1. - `INTEGER or INT`
   - `REAL`
   - `CHARACTER(`*n*`) or CHAR(`*n*`) or VARCHAR(`*n*`)`
   - `DOUBLE`
3. ```
   SELECT * FROM Stock
   ```
5. ```
   SELECT TradingSymbol, NumShares FROM Stock
   ```
7. ```
   SELECT * FROM Stock
   WHERE TradingSymbol LIKE 'SU%'
   ```
9. ```
   SELECT TradingSymbol, NumShares FROM Stock
   WHERE SellingPrice > PurchasePrice AND
   NumShares > 100
   ORDER BY NumShares
   ```
11. ```
    UPDATE Stock
    SET TradingSymbol = 'ABC'
    WHERE TradingSymbol = 'XYZ'
    ```
13. ```
    Connection conn = DriverManager.getConnection(DB_URL);
    ```
15. ```
    ResultSet result = stmt.executeQuery(sqlStatement);
    ```
17. ```
    CREATE TABLE Car
    ( Manufacturer CHAR(25),
      Year INTEGER,
      VehicleID CHAR(20) )
    ```

Short Answer

1. Traditional text and binary files are not practical when a large amount of data must be stored and manipulated. Many businesses keep hundreds of thousands, or even millions, of data items in files. When a text or binary file contains this much data, simple operations such as searching, inserting, and deleting become cumbersome and inefficient.

3. The data that is stored in a database is organized into one or more tables. Each table holds a collection of related data. The data that is stored in a table is then organized into rows and columns. A row is a complete set of information about a single item. The data that is stored in a row is divided into columns. Each column is an individual piece of information about the item.
5. A result set is an object that is somewhat similar to a collection, and contains the results of an SQL statement.
7. The first row is row 1. The first column is column 1.
9. A foreign key is a column in one table that references a primary key in another table.

Chapter 23 (Available on the Book's Companion Website)

Multiple Choice and True/False

1. c
3. c
5. d
7. b
9. b
11. a
13. False
15. True
17. False
19. 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. `myWindow.setSize(500, 250);`
3. `myWindow.setVisible(true);`

5. `setLayout(new FlowLayout(FlowLayout.LEFT));`

7. `panel.add(button, BorderLayout.WEST);`

9. `panel.setBorder(BorderFactory.createLineBorder(Color.blue, 2));`

Short Answer

1. The window is hidden from view, but the application does not end.

3. Radio buttons are normally used to select one of several possible items. Because a mutually exclusive relationship usually exists between radio buttons, only one of the items may be selected. Check boxes, which may appear alone or in groups, allow the user to make yes/no or on/off selections. Because there is not usually a mutually exclusive relationship between check boxes, the user can select any number of them when they appear in a group.

Chapter 24 (Available on the Book's Companion Website)

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 printItem = new JMenuItem("Print");
printItem.setMnemonic(KeyEvent.VK_P);
printItem.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 select only 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 a 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 25 (Available on the Book's Companion Website)

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. True
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)
   {
      mouseMovements += 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 older versions of Microsoft Internet Explorer, 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.