

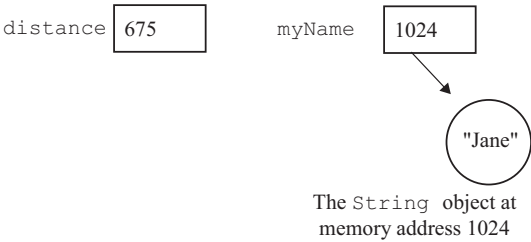
SOLUTIONS TO SELECTED ODD KNOWLEDGE EXERCISES

CHAPTER 1

1. (b) The number of computers grew from 200 to 800 – a factor of 4
3. Operating systems (such as: Windows, Linux or Apple OS X) are the instructions used by the computer system to schedule tasks, to allocate memory and other system resources, to detect errors and to perform other computer system functions. Application software is commonly used by the human user, while system software is used by the computer system. Examples of application software include word processors, spreadsheets, mail readers, Web browsers, and game programs.
5. Both a and c are characteristics of secondary storage which is nonvolatile, has a very large capacity and is slower and cheaper than RAM.
7. (a) A device that is only used for output is a printer or a speaker.
(c) Devices used for both input and output include touch screens, flash drives, floppy disks, and writable CDs, DVDs.
9. The computer as we know it today was the work of many people over hundreds of years, beginning with the development of early calculating machines: the abacus, the slide rule, Napier's bones and the Pascaline. The modern computer was based on the designs of Babbage, von Neumann, Mauchly, and Eckert. Lady Ada Lovelace and Grace Hopper were pioneers in the field of programming languages. Metcalfe and Boggs, Cerf and Kahn and Berners-Lee connected computers together into networks, the Internet and the World Wide Web, respectively.
11. (b) loses its contents if power is interrupted.
13. (c) chips, replacing the larger transistor circuits.
15. (a) First programmer - Lady Ada Augusta Byron, the Countess of Lovelace
(b) Inventor of the Java programming language – James Gosling
17. Platform independence is the ability of software to run on any computer system or platform. Every manufacturer's chipset has its own unique machine language and therefore usually requires its own translating program to translate from source code instructions to its machine language. Java achieves platform independence by compiling the source code instructions into byte code, which is later translated on the end user's computer into its own specific machine code.
19. A class is a group of related data members and member methods. It is the template used to create an object. An object is a particular instance of a class. From one class we can create an unlimited number of objects or instances of the class, just as with a blueprint we can create many houses, or with a cookie cutter, we can create many batches of cookies.
21. (a) CPU – central processing unit (c) I/O – input/output
(e) JVM – Java virtual machine (g) GUI- graphical user interface
23. (d) (5, 30) since it is 5 pixels to the left of the left boundary and 30 pixels below the top.
25. (1) character data, (2) translated instructions, and (3) numeric data.
27. (a) 01010011 = 83 in decimal (b) 00101111 = 47 in decimal

CHAPTER 2

1. (a) False, the contents of the variable may change but the data type does not
(c) False (e) False
3. A variable is a named memory cell that stores one data item that can change during program execution. Primitive variables can store a single numeric data value, one character, or one Boolean truth value. Reference variables store (RAM) memory addresses.

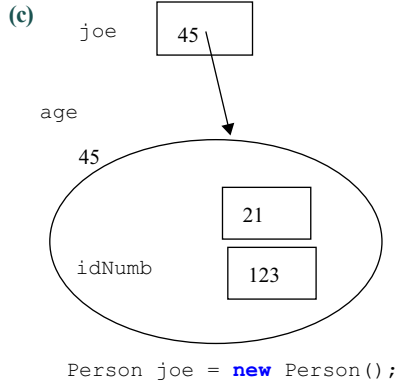
5. (a) **boolean** false (c) **double** 0.0
7. Numeric literals, containing decimals such as 19.5, are assumed to be type double. If a numeric literal is to be assigned to a **float** variable, the letter **f** for float, must be appended to the literal to inform the translator that a loss of precision is acceptable, otherwise an error results. (This is a correct declaration **float** weight = 19.5f;)
9. (a) `System.out.println("Sara Larson");`
`System.out.println("Smalltown, USA");`
 (b) `System.out.println("Sara Larson \nSmalltown, USA \n");`
 (a) `System.out.printf("Sara Larson \n");`
`System.out.printf("Smalltown, USA \n");`
 (b) `System.out.printf("Sara Larson \nSmalltown, USA \n\n");`
11. 
`int distance = 675; String myName = "Jane";`
13. (a) $17 - 5 * 2 + 12 = 19$ (c) $(48 + 12) / 12 + 18 * 2 = 41$
 (d) $21 - 9 + 18 + 4 * 3.7 = 44.8$
15. **double** average = ((**double**)(55 + 57 + 60)) / 3;
17. (a) True (b) False, it is used for output
 (c) True (d) False, it would return the empty string ("")
 (e) True
19. `sBalance = JOptionPane.showInputDialog("Type your current " + "checking account balance");`
21. **double** deposit;
`deposit = Double.parseDouble(sDeposit);`

CHAPTER 3

1. (a) True (b) False, it is the method signature
 (c) False (d) False, this method returns a value
3. (a) The signature of a method that does not operate on an object must contain the keyword **static**.
 (c) When we invoke a static method, we begin the invocation statement with the name of the class followed by a dot.
5. (a) True
 (c) False, the client method sends an argument into the worker method's parameter
 (e) False, a method can only return a single value
 (g) False
 (h) False, value parameters
7. The statement following the statement that invoked the method executes next.
9. **static double** checkAmount;
11. (a) `drawRect` (c) `drawOval`
 (e) `fillOval`, using the same value for the height and width
13. (a) House is to object as blueprint is to class.

- (c) The name of the graphic used to specify a class is a UML diagram.
 (e) Member methods of a class are usually designated to have private access.

15. (a) The address of the object `joe`



17. (a) `public CoffeeCup(int size, double price)`

```
{
    this.size = size;
    this.price = price;
}
```

(b) `CoffeeCup cup1 = new CoffeeCup(8, 3.85);`

(c) `System.out.println(cup1);`

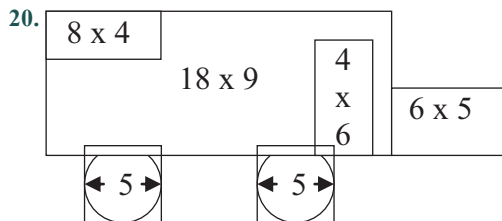
(d) `System.out.println(cup1.toString());`

(e) `CoffeeCup@456af2` (the address, 456af2, will probably be different)

(f) `g.drawString(cup1.toString(), 200, 250);`

19. `public void show(Graphics g)`

```
{
    g.drawString("size is: " + size, 250, 250);
    g.drawString("price is: " + price, 250, 280);
}
```



21. This might be a typical response based on the model given in Exercise 20

Component	Shape	Shape's X or Line's X ₁ coordinate	Shape's Y Line's Y ₁ coordinate	Width Line's X ₂ coordinate	Height Line's Y ₂ coordinate
window	rectangle	x	y	8	4
body	rectangle	x	y	18	9
door	rectangle	x + 14	y + 2	4	6
cab	rectangle	x + 18	y + 4	6	5
rear tire	circle	x + 3	y + 8	5	5
front tire	circle	x + 11	y + 8	5	5

23. (a) `this.total = total * 2;`

(c) `public void setTotal(int total)`

- ```

 {
 this.total = total;
 }
(e) int currentTotal = myAccount.getTotal();
 myAccount.setTotal(currentTotal * 2);
(g) public void toString()
 {
 System.out.println("The total is: " + total);
 }
(i) private
25. (a) static Starship largest(Starship ship1, Starship ship2);
 (b) ship1 = largest(ship1, ship2);
 (c) The new color.
 (d) public boolean sameModel(Starship ship1, Starship ship2);
 (e) isSame = sameModel(ship1, ship2);

```

## CHAPTER 4

1. (a) True    (c) False    (e) True
3. Method invocations and control-of-flow (or control) statements, such as decision and loops, alter the execution path.
5. **if** (myBalance ==10.0)
 

```

 {
 System.out.println(myBalance);
 }
 else
 {
 System.out.println("my balance is not 10.0");
 }

```
7. (a) True
  - (c) True, although it can be empty
  - (e) True
  - (g) True
9. (a) False, but it is good programming style to include a default statement
  - (c) True
  - (e) True
  - (g) False, it can only be written as a switch statement if the selection statements are of the appropriate type
11. **if**(item.equals("Hamburger"))
 

```

 {
 System.out.println("You ordered a Hamburger.");
 }
 else if(item.equals("Taco"))
 {
 System.out.println("You ordered a Taco.");
 }
 else if(item.equals("BLT"))
 {
 System.out.println("You ordered a BLT sandwich.");
 }
 else

```



```

int sum = 0;
String instr;
instr = JOptionPane.showInputDialog("Type a number: ");
n = Integer.parseInt(instr);
while(count <= n)
{
 if (count % 2 == 0) //number is even
 {
 sum = sum + count;
 }
 count++;
} //end while
JOptionPane.showMessageDialog(null, "The sum of even integers " +
 "from 1 to " + n + " is " + sum);

```

5. (a) The value of *i* is never equal to 20, so the loop never terminates.  
 (b) Because the loop does not terminate the output statement after the loop is never reached and is not executed.

7. (a) Output: 8, 5, 2, -1

(b) 

```
for (int x = 8; x >= -1; x = x - 3)
{
 System.out.println(x);
}
```

9. 

```
int trys = 0;
int input;
String sInput;
do
{
 trys++;
 sInput = JOptionPane.showInputDialog("Enter a number from 0 to 5");
 input = Integer.parseInt(sInput);
 if(input >= 0 && input <= 5)
 {
 JOptionPane.showInputDialog("Thanks for the valid input");
 break;
 }
 else
 {
 JOptionPane.showInputDialog("invalid input");
 }
} while(trys < 3);
```

11. (a) 

```
int randomNumber;
Random randomObject1 = new Random(); // uses time of day
for(int i=1; i<=20; i++)
{
 randomNumber = randomObject1.nextInt(501);
 System.out.print(randomNumber + " ");
}

SecureRandom randomObject2 = new SecureRandom();
System.out.println();
for(int i=1; i<=20; i++)
{
 randomNumber = randomObject2.nextInt(501);
 System.out.print(randomNumber + " ");
}
```

```
(c) int randomNumber;
 int min = 7;
 int max = 500;
 Random randomObject = new Random(2468); // uses seed
 for(int i=1; i<=20; i++)
 {
 randomNumber = min + randomObject.nextInt(500 - min + 1);
 System.out.print(randomNumber);
 }
```

## CHAPTER 6

1. (a) True (c) True  
(e) True (g) False, arrays can be multi-dimensional  
(i) True
3. An array element is a reference variable, while a non-array element may be a primitive or a reference variable. An array variable is able to store many elements, while a primitive variable only stores one. An array variable uses square brackets ( [ ] ) and an index to indicate the position of an element in the array, while a non-array variable does not.

5. (a) True (c) False, gameScores[99]  
(e) False, 100 (g) System.out.println(gameScores[99]);

```
(i) int total = 0;
 for(int i = 0; i < gameScores.length; i++)
 {
 total = total + gameScores[i];
 }
 System.out.println(total / gameScores.length);
```

7. (a) 45 (c) 4  
(e) y[4] = y[4] + 20.5; (f) z = y[0] + y[1] + y[2];

9. (a) String[] names = new String[50];  
double[] weights = new double[50];  
double[] targetWeights = new double[50];  
(c) for(int i = 0; i < names.length; i++)  
{  
 if(names[i].equalsIgnoreCase("joe smith")  
{  
 System.out.println(weight[i] + " " + targetWeight[i]);  
 }  
}

## CHAPTER 7

1. (a) True (c) True  
(e) False, a deep copy (g) True
3. A shallow comparison compares reference variables or the addresses of two objects to determine if they refer to the same object or two different objects. A deep comparison compares the contents of the data members of two objects to determine if they are the same.
5. Explain the difference between a deep copy and a clone. A deep copy copies the values of the data members of one object into the data members of another object, using the set method. When an object is cloned, a new instance of the object's class is created, and the values of all of an existing object's data members are copied into the corresponding data members of the new object. There are now two objects instead of one.





## CHAPTER 9

1. (a) True  
 (c) False, usually the most difficult part is the discovery of the reduced problem  
 (e) True, if the base case is not realized  
 (g) False; typically they are slower than their loop base (iterative) counterparts because of the time required to transfer execution to the recursive invocations they make
3. The symbol with the number 5 to its left
5. Iterative:  $f_1 = 1$ ;  $f_2 = 1$ ;  $f_3 = 1 + 1 = 2$ ;  $f_4 = 1 + 2 = 3$ ;  $f_5 = 2 + 3 = 5$ ;  $f_6 = 3 + 5 = 8$ ;  
 $f_7 = 5 + 8 = 13$ ;  $f_8 = 8 + 13 = 21$ ;  
 Non-iterative:  $f_8 = f_7 + f_6 = (f_6 + f_5) + (f_5 + f_4) =$   
 $(f_5 + f_4) + (f_4 + f_3) + (f_4 + f_3) + (f_3 + 1) =$   
 $(f_4 + f_3) + (f_3 + 1) + (f_3 + 1) + (1 + 1) + (f_3 + 1) + (1 + 1) + (1 + 1) + 1 =$   
 $(f_3 + 1) + (1 + 1) + (1 + 1) + 1 + (1 + 1) + 1 + (1 + 1) + (1 + 1) + 1 +$   
 $(1 + 1) + (1 + 1) + 1 =$   
 $(1 + 1) + 1 + (1 + 1) + (1 + 1) + 1 + (1 + 1) + 1 + (1 + 1) + (1 + 1) + 1 +$   
 $(1 + 1) + (1 + 1) + 1 =$   
 21
7. Dynamic programming
9. Base case: `if(m == n) return n;`  
 Reduced problem: sum of the even integers from  $m-2$  to  $n$   
 General Solution:  $m$  + the reduced problem
11. Because that is the base case, which halts the recursive invocations.
13. Combine the base case, reduced problem and general solution into a recursive algorithm, using a flow chart similar to the one shown in Figure 9.6
15. To move six rings:  $2^6 - 1$  For ten rings:  $2^{10} - 1$  For  $n$  rings:  $2^n - 1$

## CHAPTER 10

1. (a) True (c) True  
 (e) True (h) False  
 (j) False, but if the exception is a checked exception the method's signature must contain a `throws` clause  
 (l) True (n) False
3. When the error that caused the problem is a serious error, because the translator will then warn the programmer that a catch block to deal with the problem was not included in the program that invoked the method.
5. Exception
7. (a) Checked (c) Checked  
 (e) Unchecked (g) Unchecked
9. Invoke the `getMessage` method on the exception object passed to the catch clause:  

```
String error = e.getMessage();
```

## CHAPTER 11

1. (a) False, it stands for Graphical User Interface  
 (c) True  
 (e) True  
 (g) True  
 (i) False, a `Pane` container should be used  
 (k) True

3. Buttons are used to initiate processing; Radio buttons are used to select one item from a set of mutually exclusive items; Check boxes are used select to one or more items from a set of items.
5. North, west, center, east, and south
7. 

```
Pane root = new Pane();
 root.setStyle("-fx-background-color: red");
 Scene scene = new Scene(root, 600, 650);
```
9. 

```
compute.setOnAction(e -> clickHandler(e));
```
11. 

```
scene.setOnMouseClicked(e -> clickHandler(e));
```
13. 

```
scene.setOnKeyReleased(e -> keyHandler(e));
```

## CHAPTER 12

1. (a) False, they are normally used to select one input from a set of mutually exclusive inputs  
 (c) False, only one selection can be made (e) True  
 (g) True (i) False, they are defined in an `ObservableList`  
 (k) True (m) False
3. A combo box is used to select one item from a set of items; lists are used to select one or more items from a set of items.
5. Only one item can be selected from a combo box, one or more values can be selected from a list. The items in a combo box are displayed when the arrow in its drop-down button is clicked. A list is displayed with a scroll bar by default when the size of the list box is too small to display all of its values.
7. Menus take up very little space in a window, because their items are only displayed when the user indicates that she wants to use them by clicking them.
9. The path to the selected file.
11. 

```
salad.setOnAction(e -> saladClickHandler());
```
13. 

```
if(cbl.isSelected() == true)
```

## CHAPTER 13

1. (a) True (c) False  
 (e) False (g) True  
 (i) True (k) True
3. 

```
T1[] copy;
 copy = Arrays.copyOf(values, values.length);
```
5. (a) True (c) True  
 (f) True
7. 

```
public class G7Class <T> implements Comparable<GClass>
```
9. 

```
ArrayList <String> s2;
```
11. `HashMap, or TreeMap, or LinkedHashMap`
13. 

```
PriorityQueue <PhoneListing> pl = new PriorityQueue <PhoneListing>();
```

## CHAPTER 14

1. (a) True, until the program it is part of ends  
 (c) True  
 (e) True, or they can extend the class `Thread` (which implements the interface `Runnable`).  
 (g) False

- 3. New, runnable, waiting, timed waiting, blocked, and terminated
- 5. The wait method
- 7. The method invokes the `wait` method, or the method invokes the `sleep` method.
- 9. (a) True (c) False  
(e) False (g) False
- 11. The consumer task is using data generated by another task, the producer task. Two problems can occur. The producer generates a data item and overwrites a previously generated data item not yet processed by the consumer task, or the consumer reprocess a previously processed data item (or a data item containing a default value) because the producer has not generated a new data item.
- 13. Use synchronized methods or synchronized statements.

