

Data Structures with Java

Text Errata

Chapter 2:

- Page 39: Section "Wrapper Classes) paragraph 3, replace *boolean* with **Boolean**.
In Figure 2.1 description, uppercase **Integer**, **Character**, and **Double**.
- Page 54: Section 2.5, declaration:
SalaryEmployee sEmp = **new** SalaryEmployee("Morris, Mike", "569-34-0382", 1250.00);
- Page 55: Output: replace *Social Security Number:* with **SS#:**

Chapter 3:

- Page 92: In method `resize()` comment lines 7 and 15, replace *resized* with **resize**
- Page 99: Class header: **public** class Dice
- Page 102: Section "Import Statement", replace *java.swing* with **java.swing.***

Chapter 4:

- Page 117: In method `selectionSort()`, fix format
- ```
for (j = pass+1; j < n; j++)
 // if smaller element found, assign smallIndex to that index
 if (arr[j] < arr[smallIndex])
 smallIndex = j;
```
- Page 128: For function  $T(n)$  in (4.11),  $T(n) = \text{int}(\log_2 n) + 1$

### Chapter 5:

- Page 162: In method `binSearch()`, delete statement/comment *int orgList = last; // save original value of last*

### Chapter 6:

- Page 174: Section "Multibase Representation", last tem in representations for 95 base 2, 5, and 8 is  $1(2^0)$ ,  $0(5^0)$ , and  $7(8^0)$  respectively.
- Page 182: In method `hanoi()`, replace string *move* with **Move** in the `System.out.println()` statement.

### Chapter 7:

- Page 200- 202: In each index range, use the variable name **midpt**. (E.g. `[first, midpt)` and `[midpt, last)` )
- Page 210: Generic tag for method `pivotIndex()` ends with ">>": `<T extends Comparable<? super T>>`.

### Chapter 8:

- Page 240: In method `resize()` second last line, replace *check is m* with **check if m**

### Chapter 9:

- Page 260: Fourth line from the bottom, the cast **(String)** in `(String)diplomaList.get(i)` is not necessary.
- Page 264: Comment in first line of method `main()`, replace *a file* with **arrays**
- Page 273: In class `Time24` header, put blank space between *Time24* and *implements*
- In the catch block for method `clone()`, statement `throw new InternalError();` is not a comment.

### Chapter 10:

Page 291: Section "Scanning a Linked List", line 4: method toString(front) is in the **Nodes** class.

Page 293: Use generic version of code to insert a new node in a list  
`Node<T> curr, prev, newNode;`  
`newNode = new Node<T>(item);`

Page 293: Use generic version of code to delete a node in a list  
`Node<T> curr, prev;`

Page 299: Line 2: replace prevNode (curr.previous) with prevNode (curr.**prev**).

### Chapter 11:

Page 332: In Program 11.1, line 13: Scanner keyIn = **new Scanner**("System.in");

Page 333: In method jumbleLetters() first line, use inequality <: for (int i = 0; i < word.length(); i++)

### Chapter 12:

Page 353: In the figure, the sequence is **8 12 3 2** after calling iter.set(12). Replace *17* with **12**

Page 356: In method insertList() line 9: replace *aList* with **alter**.

Sample code, replace *insert()* with **insertList()**: insertList(stateListA, stateListB, 2).

Page 359: In figure 12.4, the sequence in part(c) is **5 7 8 8**. Replace 5 with 7.

### Chapter 13:

Page 379: In class header, *ListIteratorImpl* should be **ListIteratorImpl**.  
Section subheader remove space between *List* and *Iterator*. Should be **ListIterator Constructor**

### Chapter 14:

Page 409: In method compute(), delete the colon (:) at the end of the header

### Chapter 15:

Page 429: Line 1, replace word *operator* with **algorithm**

### Chapter 16:

Page 474: In method buildTree(), the header is public static TNode<**Character**> buildTree(int n)

Page 482: In method levelOrderDisplay(), class type for q is **LinkedList<TNode<T>>** (missing >)

### Chapter 17:

Page 522: In the code for buildShadowTree() line 6, replace *buildShadowTreeD(. . .)* with **buildShadowTree(...)**

In figure 17.6, the comment to the right should be Cell(level, col) = **(1,4)**

### Chapter 18:

Page 542: In method add(): in first comment line, replace *it* with **if**

Page 546: In method removeNode() line 27, delete comment *// complete the link to the parent node*

## Chapter 20:

- Page 606: Section 20.2, line 7: replace MapEntry with **Map.Entry**  
line 10: replace This interface with **The OrderedMap interface**

## Chapter 21:

- Page 635: In line 2: replace *LinkedList collections* with **nodes**  
  
In section "Evaluating Chaining with Separate List", line 6: replace *collections* with **linked lists**
- Page 648: In the code for put() line 15, make the comment  
`// scan the linked list; update a match and return previous value`

## Chapter 22:


- Page 666: In declaration of class Less, place a blank between *Less<T>* and *implements*
- Page 670: Title for figure 22.3 is "Heap insertion and the path of parents"
- Page 674: In method adjustHeap(), replace modifier private with **public** in the method header

## Chapter 24:

- Page 749: The description for Figure 24.9 should exchange variable names v and w. The description is

Figure 24.9 Discovering a cycle assuming v is  
a GRAY neighbor of w (Edge(w,v) is a back edge.

## Chapter 25:

- Page 775: In Step 5, (Figure [e]) is missing. the figure is   
(e)

- Page 781: Steps 5-6 (last line). replace *MI(E,6)* with **MI(E,11)**

## Chapter 27:

- Page 837: In method add(): in first comment line, replace *it* with **if**
- Page 843: Edit label for figure (a) to be Split 4-node(4, 12, 25)

## Chapter 28:

- Page 868: In the example,  $50 = 5 \pmod{15}$
- Page 875: First line, delete the word **prime** from the expression "a ~~prime~~ number"

## Appendix A:

- Page 930: Line 24 (add spaces): `inch + " in metric is " + centimeter + " cm";`
- Page 939: In Table A.2, start Level 2 with the multiplication operator \*.
- Page 941: In Table A.3, example for "greater than or equal" is `ch >= 'A'`
- Page 948: Ten lines from the bottom, correct spelling of boolean variable is "haveInsufficientFunds"

## Appendix C:

- Page 964: The character with ASCII value 62 is `>`

## **Data Structures with Java**

### **Exercises Errata**

#### Chapter 10:

Page 310: Exercise 10.2. Remove the statement `newNode = new Node<Character>('T');` "

Page 311: Exercise 10.4(b). Statement #3 is: `newNode.next = front` (replace `front` with `next`)

Page 318: Exercise 10.24(b). in declaration for `icon`, replace `IconImage` with `ImageIcon`  
`Icon icon = new ImageIcon("imageFile");`

#### Chapter 16:

Page 500: Exercise 16.9. add a space between the 1 and `edges`; statement is "it must have  $n - 1$  edges"