Chapter 14
Exercise solutions of
Java TM How to Program
Early Objects
TENTH edition

Paul Deitel • Harvey Deitel

BLANK PAGE

Yet an another sample XD

14.1 State whether each of the following is true or false. If false, explain why.

a) When String objects are compared using ==, the result is true if the Strings contain the same values.

Ans: False. '==' operator will check whether they share the same memory or not.

b) A String can be modified after it's created.

Ans: False. A string is an immutable object and thus can't be modified.

14.2 For each of the following, write a single statement that performs the indicated task:

a) Compare the string in s1 to the string in s2 for equality of contents.

```
1  s1.equals(s2)
```

b) Append the string s2 to the string s1, using +=.

```
1  s1.equals(s2)
```

c) Determine the length of the string in s1 .

```
1  s1.equals(s2)
```

14.3 (Palindromes)

```
1  public class Palindrome {
2    public static void main(String[] args) {
3      String s = "madam";
4      System.out.println(isPalindrome(s));
5    }
6
7    static boolean isPalindrome(String s) {
8      int n = s.length();
9      for (int i = 0; i < n/2; i++) {
10       if (s.charAt(i) != s.charAt(n-i-1)) {
11         return false;
12       }
13     }
14     return true;
15   }
16 }
```

14.4 (Comparing Portions of Strings)

```java
import java.util.Scanner;

public class Compare {
  public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    System.out.println("Enter first string: ");
    String s1 = input.nextLine();
    System.out.println("Enter second string: ");
    String s2 = input.nextLine();
     System.out.println("Enter number of characters to be compared: ");
     int n = input.nextInt();
     System.out.println("Enter starting index of the comparison: ");
     int i = input.nextInt();
     input.close();

     if (s1.regionMatches(true, i, s2, i, n)) {
       System.out.println("The strings are equal.");
     } else {
       System.out.println("The strings are not equal.");
     }
   }
}
```

14.5 (Random Sentences)

```java
public class SentenceGeneration {
  String[] article = { "the", "a", "one", "some", "any" };
  String[] noun = { "boy", "girl", "dog", "town", "car" };
  String[] verb = { "drove", "jumped", "ran", "walked", "skipped" };
  String[] preposition = { "to", "from", "over", "under", "on" };

  int randomNum(int min, int max) {
    return (int) (Math.random() * (max - min + 1) + min);
  }

   String randomArticle() {
```

```
12        return article[randomNum(0, article.length - 1)];
13    }
14
15    String randomNoun() {
16        return noun[randomNum(0, noun.length - 1)];
17    }
18
19    String randomVerb() {
20        return verb[randomNum(0, verb.length - 1)];
21    }
22
23    String randomPreposition() {
24        return preposition[randomNum(0, preposition.length - 1)];
25    }
26
27    String randomSentence() {
28        String sentence = randomArticle() + " " + randomNoun() + " " +
randomVerb() + " " + randomPreposition() + " "
29                + randomArticle() + " " + randomNoun() + ".";
30        return sentence.substring(0, 1).toUpperCase() + sentence.substring(1);
31    }
32
33    public static void main(String[] args) {
34        SentenceGeneration sentenceGeneration = new
SentenceGeneration();
35        for (int i = 0; i < 20; i++) {
36            System.out.println(sentenceGeneration.randomSentence());
37        }
38    }
39 }
```

14.6 (Project: Limericks)

```
1  public class Limericks {
2     String[] threeRhymer = { "There was a young lady of station\n", "I love
man was her sole exclamation\n",
3           "Isle of Man is the true explanation\n" };
```

```
    String[] twoRhymer = { "But when men cried, \"You flatter\"\n", "She replied, \"Oh! no matter!\n" };

    int randomNum(int min, int max) {
        return (int) (Math.random() * (max - min + 1) + min);
    }

    String threeRimeGen() {
        return threeRhymer[randomNum(0, threeRhymer.length - 1)];
    }

    String twoRimeGen() {
        return twoRhymer[randomNum(0, twoRhymer.length - 1)];
    }

    String randomSentence() {
        String sentence = threeRimeGen() + threeRimeGen() + twoRimeGen() + twoRimeGen() + threeRimeGen();
        return sentence.substring(0, 1).toUpperCase() + sentence.substring(1);
    }

    public static void main(String[] args) {
        Limericks sentenceGeneration = new Limericks();
        for (int i = 0; i < 20; i++) {
            System.out.println(sentenceGeneration.randomSentence());
        }
    }
}
```

14.7 (Pig Latin)

```
import java.util.Scanner;

public class PigLatin {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter a sentence: ");
```

```
7       String sentence = input.nextLine();
8       input.close();
9
10       String[] words = sentence.split(" ");
11       for (String word : words) {
12          System.out.print(word.substring(1) + word.charAt(0) + "ay ");
13       }
14    }
15 }
```

14.8 (Tokenizing Telephone Numbers)

```
1  import java.util.Scanner;
2
3  public class TokeinizingTelephone {
4     public static void main(String[] args) {
5        Scanner input = new Scanner(System.in);
6        System.out.println("Enter a telephone number: ");
7        String telephoneNumber = input.nextLine();
8        input.close();
9
10       String[] tokens = telephoneNumber.split("[()\\- ]");
11       String areaCode = tokens[1];
12       String firstThreeDigits = tokens[3];
13       String lastFourDigits = tokens[4];
14       String phoneNumber = firstThreeDigits + lastFourDigits;
15
16       System.out.println("Area code: " + areaCode);
17       System.out.println("Phone number: " + phoneNumber);
18    }
19 }
```

14.9 (Displaying a Sentence with Its Words Reversed)

```
1  import java.util.Scanner;
2
3  public class ReverseSentence {
4     public static void main(String[] args) {
5        Scanner input = new Scanner(System.in);
6        System.out.println("Enter a sentence: ");
7        String sentence = input.nextLine();
8        input.close();
```

```
 9
10        String[] words = sentence.split(" ");
11        for (int i = words.length - 1; i >= 0; i--) {
12           System.out.print(words[i] + " ");
13        }
14     }
15  }
```

14.10 (Longest Word in a Sentence)

```
 1  import java.util.Scanner;
 2
 3  public class LongestWord {
 4     public static void main(String[] args) {
 5        Scanner input = new Scanner(System.in);
 6        System.out.println("Enter a sentence: ");
 7        String sentence = input.nextLine();
 8        input.close();
 9
10         String[] words = sentence.split(" ");
11         int maxLength = 0;
12         String longest_word = "";
13         for (String word : words) {
14            if (word.length() > maxLength) {
15               longest_word = word;
16               maxLength = word.length();
17            }
18         }
19
20         System.out.println("The longest word is: " + longest_word);
21     }
22  }
```

Knowledge should be free and open source 🙃