

```
1  /*
2  @author mjelks
3
4  Sample run of the cashier program: (bold-face is user input)
5
6  Scan an item: Chocolate Chip
7
8  Enter price: $3.99
9
10 Scan an item: Sharp Microwave
11
12 Enter price: $75.40
13
14 Scan an item: Logitech Wireless Mouse
15
16 Enter price: $35.99
17
18 Customer pays:  $150
19
20
21
22 =====
23
24 Change
25
26 Dollars:   34
27
28 Quarters:  2
29
30 Dimes:      1
31
32 Nickels:    0
33
34 Pennies:    2
35
36 *****
37
38 PURCHASE RECEIPT
39
40 Chocolate Chip           $    3.99
41
42 Sharp Microwave         $   75.40
43
44 Logitech Wireless Mouse $   35.99
45
```

```
46 Total:                $ 115.38
47
48 Amount paid:          $ 150.00
49
50 Change:                $ 34.62
51
52 *****
53
54 */
55 package CashRegister;
56
57 import java.io.*;      // include Java I/O library package
58 import java.util.ArrayList; // want to store value input into array
59 import java.util.HashMap;
60 import java.util.Iterator;
61 import java.util.List;
62 import java.util.Map;
63
64 public class CalculateChange {
65
66     HashMap<String, InventoryItem> productMap = new HashMap<String, InventoryItem>();
67
68     double customerAmount = 0.0;
69     int counter = 0;
70
71     public static void main(String[] args) {
72         HashMap<String, InventoryItem> productMap = new CalculateChange().getInput();
73         double customerAmount = new CalculateChange().getCustomerAmount();
74         double total = new CalculateChange().calculateTotal(productMap);
75         new CalculateChange().printChange(total, customerAmount);
76         new CalculateChange().printReceipt(productMap, customerAmount);
77     } // end main method
78
79     public HashMap<String, InventoryItem> getInput() {
80         try {
81             String description;
82             double price;
83
84             String buffer;
85             BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
86
87             // read a string
88             System.out.print("Scan an item: ");
89             buffer = br.readLine();
90             description = buffer;
```

```
91
92     // read double
93     System.out.print("Enter price: $");
94     buffer = br.readLine();
95     price = Double.parseDouble(buffer);
96
97     InventoryItem priceItem = new InventoryItem(description, price);
98
99     productMap.put(new Integer(this.counter).toString(), priceItem);
100    this.counter++;
101
102    System.out.print("Scan another item? ");
103    buffer = br.readLine();
104
105    if (buffer.equals("y") || buffer.equals("yes")) {
106        this.getInput();
107    }
108
109    } catch (IOException e) {
110        e.printStackTrace();
111    }
112
113    return productMap;
114 }
115
116 public double getCustomerAmount() {
117     try {
118         String buffer;
119         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
120         System.out.print("Customer amount: $");
121         buffer = br.readLine();
122         this.customerAmount = Double.parseDouble(buffer);
123     } catch (IOException e) {
124         e.printStackTrace();
125     }
126     return this.customerAmount;
127 }
128
129 public double calculateTotal(HashMap<String, InventoryItem> productMap) {
130     double total = 0.00;
131     HashMap<String, InventoryItem> items = productMap;
132
133     for (String key : items.keySet()) {
134         //System.out.println("key: " + key + " value: " + items.get(key).productDescr
135         total += items.get(key).productPrice;
```

```
136     }
137
138     return total;
139 }
140
141 public void printChange(double total, double customerAmount) {
142     if (total > customerAmount) {
143         System.out.println("Please enter an amount that is greater than the total of ");
144     } else {
145         HashMap<String, Integer> changeMap = this.calculateChange(customerAmount - total);
146
147         System.out.println("=====");
148         System.out.println();
149         System.out.println("Change");
150         System.out.println();
151         System.out.println("Hundreds: " + changeMap.get("hundreds"));
152         System.out.println();
153         System.out.println("Fifties: " + changeMap.get("fifties"));
154         System.out.println();
155         System.out.println("Twenties: " + changeMap.get("twenties"));
156         System.out.println();
157         System.out.println("Tens: " + changeMap.get("tens"));
158         System.out.println();
159         System.out.println("Fives: " + changeMap.get("fives"));
160         System.out.println();
161         System.out.println("Ones: " + changeMap.get("ones"));
162         System.out.println();
163         System.out.println("Quarters: " + changeMap.get("quarters"));
164         System.out.println();
165         System.out.println("Dimes: " + changeMap.get("dimes"));
166         System.out.println();
167         System.out.println("Nickels: " + changeMap.get("nickels"));
168         System.out.println();
169         System.out.println("Pennies: " + changeMap.get("pennies"));
170         System.out.println();
171         System.out.println("=====");
172         System.out.println();
173     }
174 }
175
176
177 public HashMap<String, Integer> calculateChange(double customerAmount) {
178     int dollarAmount = 0;
179     double changeAmount = 0;
180     int convertedAmount = 0;
```

```
181     HashMap<String, Integer> changeMap = new HashMap<String, Integer>();
182
183     dollarAmount = (int) customerAmount; // grab the int value (left of decimal)
184     // get the stuff after the decimal place (WE SHOULD REALLY BE USING BigDecimal)
185     changeAmount = (customerAmount % 1) * 100;
186     convertedAmount = (int) Math.round(changeAmount);
187
188     int hundreds = dollarAmount / 100;
189     int fifties = ((dollarAmount - (100 * hundreds)) / 50);
190     int twenties = ((dollarAmount - (100 * hundreds) - (50 * fifties)) / 20);
191     int tens = ((dollarAmount - (100 * hundreds) - (50 * fifties) - (20 * twenties));
192     int fives = ((dollarAmount - (100 * hundreds) - (50 * fifties) - (20 * twenties));
193     int ones = ((dollarAmount - (100 * hundreds) - (50 * fifties) - (20 * twenties));
194
195     int quarters = convertedAmount / 25;
196     int dimes = ((convertedAmount - (25 * quarters)) / 10);
197     int nickels = ((convertedAmount - (25 * quarters) - (10 * dimes)) / 5);
198     int pennies = (convertedAmount - (25 * quarters) - (10 * dimes) - (5 * nickels));
199
200     changeMap.put("hundreds", hundreds);
201     changeMap.put("fifties", fifties);
202     changeMap.put("twenties", twenties);
203     changeMap.put("tens", tens);
204     changeMap.put("fives", fives);
205     changeMap.put("ones", ones);
206
207     changeMap.put("quarters", quarters);
208     changeMap.put("dimes", dimes);
209     changeMap.put("nickels", nickels);
210     changeMap.put("pennies", pennies);
211
212     return changeMap;
213 }
214
215 //public void output
216 // output with some formatting -- the receipt
217 public void printReceipt(HashMap<String, InventoryItem> productMap, double custom
218     HashMap<String, InventoryItem> items = productMap;
219     double total = this.calculateTotal(productMap);
220
221     // print top section
222     System.out.println("*****");
223     System.out.println();
224     System.out.println("PURCHASE RECEIPT");
225     System.out.println();
```

```
226
227 // loop through the product hash map and output the description and price
228 for (String key : items.keySet()) {
229     System.out.format("%-16s\t\t$8.2f\n", items.get(key).productDescription, items.get(key).price);
230     System.out.println();
231 }
232
233 // formatted print total
234 System.out.format("%-16s\t\t$8.2f\n", "Total:", total);
235 System.out.println();
236
237 // formatted print amount paid
238 System.out.format("%-16s\t\t$8.2f\n", "Amount Paid:", customerAmount);
239 System.out.println();
240
241 // formatted print change
242 System.out.format("%-16s\t\t$8.2f\n", "Change:", customerAmount - total);
243 System.out.println();
244
245 // print bottom border
246 System.out.println("*****");
247
248 }
249 } // end class
250
```