```
47          int numRows = resultSet.getRow(); // Get row number
48          resultSet.first();                // Move to first row
49
50          // Create an array for the coffee names.
51          String[] listData = new String[numRows];
52
53          // Populate the array with coffee names.
54          for (int index = 0; index < numRows; index++)
55          {
56             // Store the coffee name in the array.
57             listData[index] = resultSet.getString(1);
58
59             // Go to the next row in the result set.
60             resultSet.next();
61          }
62
63          // Close the connection and statement objects.
64          conn.close();
65          stmt.close();
66
67          // Return the listData array.
68          return listData;
69       }
70
71       /**
72          The getProdNum method returns a specific
73          coffee's product number.
74          @param coffeeName The specified coffee.
75       */
76
77       public String getProdNum(String coffeeName)
78                                throws SQLException
79       {
80          String prodNum = ""; // Product number
81
82          // Create a connection to the database.
83          conn = DriverManager.getConnection(DB_URL);
84
85          // Create a Statement object for the query.
86          Statement stmt = conn.createStatement();
87
88          // Execute the query.
89          ResultSet resultSet = stmt.executeQuery(
90                                "SELECT ProdNum " +
91                                "FROM Coffee " +
92                                "WHERE Description = '" +
93                                coffeeName + "'");
94
```

```
95          // If the result set has a row, go to it
96          // and retrieve the product number.
97          if (resultSet.next())
98             prodNum = resultSet.getString(1);
99
100         // Close the Connection and Statement objects.
101         conn.close();
102         stmt.close();
103
104         // Return the product number.
105         return prodNum;
106      }
107
108      /**
109         The getCoffeePrice method returns the price
110         of a coffee.
111         @param prodNum The specified product number.
112      */
113
114      public double getCoffeePrice(String prodNum)
115                                   throws SQLException
116      {
117         double price = 0.0;  // Coffee price
118
119         // Create a connection to the database.
120         conn = DriverManager.getConnection(DB_URL);
121
122         // Create a Statement object for the query.
123         Statement stmt = conn.createStatement();
124
125         // Execute the query.
126         ResultSet resultSet = stmt.executeQuery(
127                               "SELECT Price " +
128                               "FROM Coffee " +
129                               "WHERE ProdNum = '" +
130                               prodNum + "'");
131
132         // If the result set has a row, go to it
133         // and retrieve the price.
134         if (resultSet.next())
135            price = resultSet.getDouble(1);
136
137         // Close the connection and statement objects.
138         conn.close();
139         stmt.close();
140
141         // Return the price.
142         return price;
```