|  |
| --- |
| **Skill 31.2 Exercise 1** |
| 1. Declare an ArrayList of String elements called *songs* which represents the songs a user downloads from a music site. |
| 1. Write code that will add two songs to the list |
| 1. Write code that will add a song to index 1 of the list. |
| 1. Write code that will change the song at index 0 to a different song. |
| 1. Write code that will remove the song at index 0. |
| 1. Indicate what is printed,   System.out.println(song.get(1)); |
| 1. Indicate what is printed,   String mySong = song.get(0);  System.out.println(mySong); |

|  |
| --- |
| **Skill 31.3 Exercise 1** |
| (a) Declare an ArrayList of int objects called iObjects |
| (b) Add three elements to the list iObject |
| (c) Declare a new int called iPrimitive and assign its value to the Integer object at index 2 of iObjects. |

|  |  |
| --- | --- |
| **Skill 29.3 Exercise 1** | |
| Consider the following binarySearch method. The method correctly performs a binary search.  /\*\* Precondition: data is sorted in increasing order. \*/  public static int binarySearch(int[] data, int target) {    int start = 0;  int end = data.length - 1;  while (start <= end) {  int mid = (start + end) / 2; /\* Calculate midpoint \*/    if (target < data[mid]) {  end = mid - 1;  } else if (target > data[mid]) {  start = mid + 1;  } else {  return mid;  }  }  return -1;  } | |
| Consider the following code segment.    int[] values = {1, 2, 3, 4, 5, 8, 8, 8};  int target = 8;    What value is returned by the call binarySearch(values, target) ? | Suppose the binarySearch method is called with an array containing 2,000 elements sorted in increasing order.  What is the maximum number of times that the statement indicated by /\* Calculate midpoint \*/ could execute? |