Rudd Fawcett
Litvin AP-CS

BJP3 Self-Check 10.2: ArrayListSyntax

Status: ✅ You have solved this problem; good work!
Added by: Marty Stepp on 2013/04/01
Keywords: ArrayList
Popularity: **52** likes Like

Which of the following is the correct syntax to construct an <code>ArrayList</code> to store integers?

(The order of the answer choices is randomly shuffled each time.)

a. `ArrayList list<integer> = new ArrayList<integer>();`
b. `ArrayList list = new ArrayList();`
c. `ArrayList<Integer> list = new ArrayList<Integer>();`
d. `ArrayList<Integer> list = new ArrayList();`
e. `ArrayList[int] list = new ArrayList[int]();`

Submit

#	question	
1	Which of the following is the correct syntax to construct an `ArrayList` to store integers?	`ArrayList&lt;Inte`

You passed 1 of 1 tests. Congratulations!
(submitted Fri Jan 22 06:34:11 PST 2016; took 15 ms)

You have now solved **24** total problems.

Did you like this problem? If so, click **"Like"** above and let the author know! :-)

BJP3 Self-Check 10.8: maxLength

Status: ✅ You have solved this problem; good work!
Added by: Marty Stepp on 2013/04/01
Language: Java
Keywords: ArrayList, collections
Popularity: **98** likes Like

Write a method `maxLength` that takes an `ArrayList` of `String`s as a parameter and that returns the length of the longest string in the list. If your method is passed an empty list, it should return 0.

> ***Method problem:*** *For this problem, you are supposed to write a Java method as described. You should not write a complete Java class; just write the method(s) described in the problem statement.*

```
1  public int maxLength(ArrayList<String> list) {
2      if (list.size() == 0) {
3          return 0;
4      }
5
6      int max = list.get(0).length();
7
8      for (String item : list) {
9          if (item.length() > max) {
10             max = item.length();
11         }
12     }
13
14     return max;
15 }
```

Submit

#	name	expected output	your output	result
1	["to", "be", "or", "not", "to", "be", "hamlet"]	6	6	pass
2	["to", "be", "or", "not", "to", "be"]	3	3	pass
3	["biggest", "next", "not"]	7	7	pass
4	["Only one really long string"]	27	27	pass
5	[]	0	0	pass

You passed 5 of 5 tests. Congratulations!
(submitted Fri Jan 22 06:36:49 PST 2016; took 928 ms)

You have now solved **25** total problems.

BJP3 Self-Check 10.15: ArrayListMystery1

Status: ✅ You have solved this problem; good work!
Added by: Roy McElmurry on 2013/04/01
Keywords: ArrayList, ArrayList mystery, collections
Popularity: **87** likes Like

Consider the following method:

```
public static void mystery1(ArrayList<Integer> list) {
    for (int i = list.size() - 1; i > 0; i--) {
        if (list.get(i) < list.get(i - 1)) {
            int element = list.get(i);
            list.remove(i);
            list.add(0, element);
        }
    }
    System.out.println(list);
}
```

Write the output produced by the method when passed each of the following `ArrayList`s:

`[2, 6, 1, 8]`	`[1, 2, 6, 8]`
`[30, 20, 10, 60, 50, 40]`	`[10, 30, 40, 20, 60, 50]`
`[-4, 16, 9, 1, 64, 25, 36, 4, 49]`	`[-4, 1, 25, 4, 16, 9, 64, 36, 49]`

Submit

#	question	your answer	result
1	`[2, 6, 1, 8]`	`[1, 2, 6, 8]`	✅ pass
2	`[30, 20, 10, 60, 50, 40]`	`[10, 30, 40, 20, 60, 50]`	✅ pass
3	`[-4, 16, 9, 1, 64, 25, 36, 4, 49]`	`[-4, 1, 25, 4, 16, 9, 64, 36, 49]`	✅ pass

You passed 3 of 3 tests. Congratulations!
(submitted Fri Jan 22 06:42:36 PST 2016; took 15 ms)

You have now solved **26** total problems.

BJP3 Self-Check 10.16: ArrayListMystery2

Status: ✅ You have solved this problem; good work!
Added by: Marty Stepp on 2013/04/01
Keywords: ArrayList, ArrayList mystery, collections
Popularity: **42** likes Like

Consider the following method:

```
public static void mystery2(ArrayList<Integer> list) {
    for (int i = list.size() - 1; i >= 0; i--) {
        if (i % 2 == 0) {
            list.add(list.get(i));
        } else {
            list.add(0, list.get(i));
        }
    }
    System.out.println(list);
}
```

Write the output produced by the method when passed each of the following `ArrayList`s:

`[10, 20, 30]`	`[20, 10, 20, 30, 30, 20]`
`[8, 2, 9, 7, 4]`	`[8, 7, 8, 2, 9, 7, 4, 4, 2, 8]`
`[-1, 3, 28, 17, 9, 33]`	`[33, 28, 33, -1, 3, 28, 17, 9, 33, 17, -1, 33]`

Submit

#	question	your answer	result
1	`[10, 20, 30]`	`[20, 10, 20, 30, 30, 20]`	✅ pass
2	`[8, 2, 9, 7, 4]`	`[8, 7, 8, 2, 9, 7, 4, 4, 2, 8]`	✅ pass
3	`[-1, 3, 28, 17, 9, 33]`	`[33, 28, 33, -1, 3, 28, 17, 9, 33, 17, -1, 33]`	✅ pass

You passed 3 of 3 tests. Congratulations!
(submitted Fri Jan 22 06:44:04 PST 2016; took 15 ms)

You have now solved **27** total problems.

BJP3 Self-Check 10.18: ArrayListMystery4

Status: ✅ You have solved this problem; good work!
Added by: Eric Spishak on 2013/04/01
Keywords: ArrayList, ArrayList mystery, collections
Popularity: **19** likes Like

Consider the following method:

```
public static void mystery4(ArrayList<Integer> list) {
    for (int i = 0; i < list.size(); i++) {
        int element = list.get(i);
        list.remove(i);
        list.add(0, element + 1);
    }
    System.out.println(list);
}
```

Write the output produced by the method when passed each of the following `ArrayList`s:

`[10, 20, 30]`	`[31, 21, 11]`
`[8, 2, 9, 7, 4]`	`[5, 8, 10, 3, 9]`
`[-1, 3, 28, 17, 9, 33]`	`[34, 10, 18, 29, 4, 0]`

Submit

#	question	your answer	result
1	`[10, 20, 30]`	`[31, 21, 11]`	✅ pass
2	`[8, 2, 9, 7, 4]`	`[5, 8, 10, 3, 9]`	✅ pass
3	`[-1, 3, 28, 17, 9, 33]`	`[34, 10, 18, 29, 4, 0]`	✅ pass

You passed 3 of 3 tests. Congratulations!
(submitted Fri Jan 22 06:44:39 PST 2016; took 14 ms)

You have now solved **28** total problems.

BJP3 Self-Check 10.17: ArrayListMystery3

Status: ✅ You have solved this problem; good work!
Added by: Roy McElmurry on 2013/04/01
Keywords: ArrayList, ArrayList mystery, collections
Popularity: **27** likes Like

Consider the following method:

```
public static void mystery3(ArrayList<Integer> list) {
    for (int i = list.size() - 2; i > 0; i--) {
        int a = list.get(i);
        int b = list.get(i + 1);
        list.set(i, a + b);
    }
    System.out.println(list);
}
```

Write the output produced by the method when passed each of the following `ArrayList`s:

`[72, 20]`	`[72, 20]`
`[1, 2, 3, 4, 5, 6]`	`[1, 20, 18, 15, 11, 6]`
`[10, 20, 30, 40]`	`[10, 90, 70, 40]`

Submit

#	question	your answer	result
1	`[72, 20]`	`[72, 20]`	✅ pass
2	`[1, 2, 3, 4, 5, 6]`	`[1, 20, 18, 15, 11, 6]`	✅ pass
3	`[10, 20, 30, 40]`	`[10, 90, 70, 40]`	✅ pass

You passed 3 of 3 tests. Congratulations!
(submitted Fri Jan 22 06:45:41 PST 2016; took 15 ms)

You have now solved **29** total problems.

BJP3 Exercise 10.3: removeEvenLength

Status: ✅ You have solved this problem; good work!
Added by: Marty Stepp on 2013/04/01
Language: Java
Keywords: ArrayList, collections
Popularity: **59** likes Like

Write a method `removeEvenLength` that takes an `ArrayList` of `String`s as a parameter and that removes all of the strings of even length from the list.

> ***Method problem:*** *For this problem, you are supposed to write a Java method as described. You should not write a complete Java class; just write the method(s) described in the problem statement.*

```
1 public void removeEvenLength(ArrayList<String> list) {
2         for (int i=list.size() - 1; i >= 0; i--) {
3             if (list.get(i).length() % 2 == 0) {
4                 list.remove(i);
5             }
6         }
7 }
```

Submit

#	name	expected output	your output	result
1	["This", "is", "a", "test"]	[a]	[a]	✅ pass
2	["even", "odd", "ev", "o"]	[odd, o]	[odd, o]	✅ pass
3	["Did", "you", "solve", "it", "or", "what?"]	[Did, you, solve, what?]	[Did, you, solve, what?]	✅ pass
4	[]	[]	[]	✅ pass

You passed 4 of 4 tests. Congratulations!
(submitted Fri Jan 22 06:49:51 PST 2016; took 1046 ms)

You have now solved **30** total problems.

BJP3 Exercise 10.4: doubleList

Status: ✅ You have solved this problem; good work!
Added by: Marty Stepp on 2013/04/01
Language: Java
Keywords: ArrayList, collections, doubleList
Popularity: 35 likes Like

Write a method `doubleList` that takes an `ArrayList` of `String`s as a parameter and that replaces every string with two of that string. For example, if the list stores the values `{"how", "are", "you?"}` before the method is called, it should store the values `{"how", "how", "are", "are", "you?", "you?"}` after the method finishes executing.

> ***Method problem:*** *For this problem, you are supposed to write a Java method as described. You should not write a complete Java class; just write the method(s) described in the problem statement.*

```
1  public void doubleList(ArrayList<String> list) {
2      int original = list.size();
3
4      for (int i=0; i < original; i++) {
5          list.add(2*i + 1, list.get(2*i));
6      }
7  }
```

Submit

#	name	expected output	your output
1	["how", "are", "you?"]	[how, how, are, are, you?, you?]	[how, how, are, are, you?, you?]
2	["I", "am", "great,", "thanks!"]	[I, I, am, am, great,, great,, thanks!, thanks!]	[I, I, am, am, great,, great,, thanks!, than
3	["One string only"]	[One string only, One string only]	[One string only, One string only]
4	["1", "4", "3"]	[1, 1, 4, 4, 3, 3]	[1, 1, 4, 4, 3, 3]
5	[]	[]	[]

You passed 5 of 5 tests. Congratulations!
(submitted Fri Jan 22 06:54:08 PST 2016; took 1111 ms)

You have now solved **31** total problems.

BJP3 Exercise 10.7: removeDuplicates

Status: ✅ You have solved this problem; good work!
Added by: Marty Stepp on 2013/04/01
Language: Java
Keywords: ArrayList, collections
Popularity: **38** likes Like

Write a method `removeDuplicates` that takes as a parameter a sorted `ArrayList` of `String`s and that eliminates any duplicates from the list. For example, suppose that a variable called `list` contains the following values: `{"be", "be", "is", "not", "or", "question", "that", "the", "to", "to"}` After calling `removeDuplicates(list);` the list should store the following values: `{"be", "is", "not", "or", "question", "that", "the", "to"}`

Because the values will be sorted, all of the duplicates will be grouped together.

> ***Method problem:*** *For this problem, you are supposed to write a Java method as described. You should not write a complete Java class; just write the method(s) described in the problem statement.*

```
1 public void removeDuplicates(ArrayList<String> list) {
2     for (int i=0; i < list.size() - 1; i++) {
3         if (list.get(i).equals(list.get(i + 1))) {
4             list.remove(i + 1);
5             i--;
6         }
7     }
8 }
```

Submit

#	name	expected output	
1	["be", "be", "is", "not", "or", "question", "that", "the", "to", "to"]	[be, is, not, or, question, that, the, to]	[be, is,
2	["duplicate", "duplicate", "duplicate", "duplicate", "duplicate"]	[duplicate]	[duplica
3	["unique"]	[unique]	[unique]
4	["be", "is", "not", "or", "question", "that", "the", "to"]	[be, is, not, or, question, that, the, to]	[be, is,
5	[]	[]	[]

You passed 5 of 5 tests. Congratulations!
(submitted Fri Jan 22 06:58:15 PST 2016; took 595 ms)

You have now solved **32** total problems.

BJP3 Exercise 10.6: minToFront

Status: ✅ You have solved this problem; good work!
Added by: Marty Stepp on 2013/04/01
Language: Java
Keywords: ArrayList, collections
Popularity: **46** likes 👍 Like

Write a method `minToFront` that takes an `ArrayList` of integers as a parameter and that moves the minimum value in the list to the front, otherwise preserving the order of the elements. For example, if a variable called list stores the following values: `{3, 8, 92, 4, 2, 17, 9}` and you make this call: `minToFront(list);` it should store the following values after the call: `{2, 3, 8, 92, 4, 17, 9}` You may assume that the list stores at least one value.

> ***Method problem:*** *For this problem, you are supposed to write a Java method as described. You should not write a complete Java class; just write the method(s) described in the problem statement.*

```
1  public void minToFront(ArrayList<Integer> list) {
2      int idx = 0;
3
4      for (int i = 1; i < list.size(); i++) {
5          if (list.get(idx) > list.get(i)) {
6              idx = i;
7          }
8      }
9
10     int min = list.remove(idx);
11     list.add(0, min);
12 }
```

Submit

#	name	expected output	your output	result
1	[3, 8, 92, 4, 2, 17, 9]	[2, 3, 8, 92, 4, 17, 9]	[2, 3, 8, 92, 4, 17, 9]	✅ pass
2	[1]	[1]	[1]	✅ pass
3	[6, 1, 4, -2]	[-2, 6, 1, 4]	[-2, 6, 1, 4]	✅ pass
4	[0, 1, 2, 3]	[0, 1, 2, 3]	[0, 1, 2, 3]	✅ pass
5	[3, 2, 1, 0]	[0, 3, 2, 1]	[0, 3, 2, 1]	✅ pass

You passed 5 of 5 tests. Congratulations!
(submitted Fri Jan 22 07:00:36 PST 2016; took 565 ms)

You have now solved **33** total problems.