1
2
3
4
5
6
7

☆ Classes and Objects: Distance

You will be given an abstract class with its fields and methods declared. The object will contain a distance in feet and inches. Implement the methods as described to itialize the class instances and return which of the distances is greater, if there is one.

The following code is provided in the locked portion of the editor:

- The declaration for an abstract class named *Distance* with the following fields and methods:

Fields		
Name	**Type**	**Description**
`feet`	`integer`	A private instance variable to store the number of *feet* in the *Distance* object.
`inches`	`floating-point`	A private instance variable to store the number of *inches* in the *Distance* object.
The total value represented by a *Distance* object is a combination of its *feet* and *inches*. For example, if *feet = 1* and *inches = 6*, the total length represented by the *Distance* object is *18in* or *1.5* feet.		

Methods				
Return Type	**Method Name**	**Param. Type**	**Param. Name**	**Description**
`void`	`setFeetAndInches`	`integer`	`feet`	Assign a value to the *feet* instance variable.
		`floating-point`	`inches`	Assign a value to the *inches* instance variable.
`integer`	`getFeet`	`no parameters`		Returns the value of the *feet* instance variable.
`integer`	`getInches`	`no parameters`		Returns the value of the *inches* instance variable.
`string`	`getDistanceComparison`	`Distance`	`dist2`	Compares the distance between the object it's called on and the *Distance* object passed as an argument (i.e., *dist2*) and returns one of the following strings: ○ If the object's distance is greater than *dist2*'s, return *First distance is greater.* ○ If *dist2*'s distance is greater than the object's, return *Second distance is greater.* ○ If both distances are equal, return *Both distances are equal.*

- A *main* method that does the following:
  - Creates a *DistanceImplementation* object named *dist1*.
  - Calls the *setFeetAndInches* method on *dist1*.
  - Creates a *DistanceImplementation* object named *dist2*.
  - Calls the *setFeetAndInches* method on *dist2*.
  - Calls *getDistanceComparison* on *dist1* and passes *dist2* to the function as an argument.

Complete the partially implemented code in the editor according to the specifications. The *getDistanceComparison* method must return a string denoting the result of the distance comparison as described above.

Constraints

- *1 ≤ feet ≤ 100*
- *1 ≤ inches ≤ 100*

Vipps Java Developer Hiring Test Senior 01h : 06m to test end 5/7 Attempted Aakash

1
2
3
4
5
6
7

Sample Case 0

Sample Input 0

```
1
2.0
3
4.1
```

Sample Output 0

```
Second distance is greater.
```

Explanation 0

The first distance is *1ft + 2.0in = 12.0in + 2.0in = 14.0in* and the second distance is *3ft + 4.1in = 36.0in + 4.1in = 40.1in*.

Sample Case 1

Sample Case 2

YOUR ANSWER

We recommend you take a quick tour of our editor before you proceed. The timer will pause up to 90 seconds for the tour. Start tour ✕

For help on how to read input and write output in Java 7, click here. ✕

Original Code Java 7

```
import java.util.Scanner; ⋯

// Enter your code here.



public class DistanceCalculator { ⋯
```

Line: 13 Col: 1

Vipps Java Developer Hiring Test Senior

01h : 06m
to test end

5/7 Attempted

Aakash

Download sample test cases *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*

About Privacy Policy Terms of Service

1

2

3

4

5

6

7