

In-Class Worksheet #11

CS62 - Fall 2024

Alexandra Papoutsaki

1. A programmer discovers that they frequently need only the odd numbers in an `ArrayList` of `Integers`. As a result, they decided to write a class `OddIterator` that implements the `Iterator` interface. Please help them implement the constructor and the `hasNext()` and `next()` methods so that they can retrieve the odd values, one at a time. For example, if the `ArrayList` contains the elements `[7, 4, 1, 3, 0]`, the iterator should return the values 7, 1, and 3.

```
import java.util.*;

public class OddIterator implements Iterator<Integer> {

    // The array whose odd values are to be enumerated
    private ArrayList<Integer> myArrayList;

    //any other instance variables you might need

    //An iterator over the odd values of myArrayList
    public OddIterator(ArrayList<Integer> myArrayList){

    }
    //should run in O(n) time
    public boolean hasNext(){

    }
    //should run in O(1) time
    public Integer next(){

    }

}
```

2. Given the `Employee` class and based on what we discussed about `Collections.sort`, fill in what you think will be the result of each of the following print statements:

```
public static void main(String[] args) {  
  
    Employee e1 = new Employee(5, "Yash", 100000);  
    Employee e2 = new Employee(8, "Tharun", 25000);  
    Employee e3 = new Employee(4, "Yush", 10000);  
  
    List<Employee> list = new ArrayList<Employee>();  
    list.add(e1);  
    list.add(e2);  
    list.add(e3);  
  
    System.out.println(list);  
  
    Collections.sort(list);  
    System.out.println(list);  
  
    Collections.sort(list, Employee.nameComparator);  
    System.out.println(list);  
  
    Collections.sort(list, Employee.salaryComparator());  
    System.out.println(list);  
  
}
```