CS62: Spring 2025 | Lecture #11 (Iterators & Comparators) worksheet | Jingyi Li

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){




    }

    //runs in O(n) time
    public boolean hasNext(){







    }

    //runs in O(1) time
    public Integer next(){









    }
}
```

2. What is printed?

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;

public class Employee implements Comparable<Employee> {

    private int id;
    private String name;
    private int salary;

    public Employee(int id, String name, int salary) {
        this.id = id;
        this.name = name;
        this.salary = salary;
    }

    public int compareTo(Employee e) {
        if (this.id < e.id) {
            return -1;
        } else if (this.id > e.id) {
            return 1;
        } else
            return 0;
        // return Integer.valueOf(this.id).compareTo(Integer.valueOf(e.id)); OR
        // return Integer.compare(this.id, e.id);

    }

    public static Comparator<Employee> nameComparator = new Comparator<Employee>() {
        public int compare(Employee e1, Employee e2) {
            return e1.name.compareTo(e2.name);
        }
    };

    public static Comparator<Employee> salaryComparator() {
        return (Employee e1, Employee e2) -> Integer.compare(e1.salary, e2.salary);
    }

    public String toString() {
        return "Name: " + name + " ID: " + id + " Salary: " + salary;
    }

    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);
    }
}
```