W2D1 Homework

1 What’s character of these Collections, Collection, List, Set and Map? ArrayList and LinkedList? HashMap and TreeMap?

* **Collections** – refers to a class that consists static methods that operates for collections.
* **Collection** - refers to a group of objects, which is its elements. Mainly used in storing, retrieving, manipulating, communicating, and aggregating data.
* **List** – refers to a collection wherein elements are stored in an ordered sequence. Can duplicate
* **Set** – refers to a collection wherein elements are unordered and unique. Don’t duplicate
* **Map** – does not implement collection interface. It refers to a set of keys that helps the elements to be found easily.
* **ArrayList** – refers to a collection of objects. Its capacity increases when elements are added, and it is not certain to be sorted.
* **LinkedList** – refers to an ordered set of elements. The memory of LinkedList is relocatable.
* **HashMap** – refers to the interface of Map which is equivalent to HashTable. It is not certain that the order when using HashMap is constant.
* **TreeMap** - also an interface of Map. Compared to HashMap, it is certain that the order is ascending.

2. （List）Read the codes

**import java.util.\*;**

**public class Test {**

**public static void main(String args[]) {**

**List list = new ArrayList();**

**list.add("Hello");**

**list.add("World");**

**list.add(1, "Learn");**

**list.add(1, "Java");**

**printList(list);**

**}**

**public static void printList(List list) {**

**// 1**

**for (int i = 0; i < list.size(); i++) {**

**System.out.println(list.get(i));**

**}**

**for (Object o : list) {**

**System.out.println(o);**

**}**

**Iterator itor = list.iterator();**

**while (itor.hasNext()) {**

**System.out.println(itor.next());**

**}**

**}**

**}**

Requirement:

1. Complete the codes at //1, and need to print out all the elements of the list.
2. Write the output of the code.

Hello

Java

Learn

World

Hello

Java

Learn

World

Hello

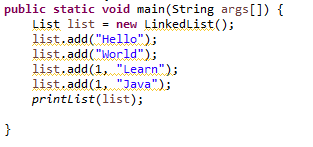
Java

Learn

World

1. Where and how to modify if change Arraylist with LinkedList? What’s the difference between ArrayList and LinkedList?

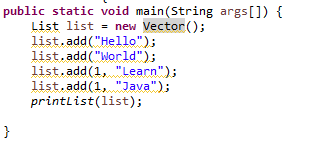
* *Can modify in the declaration of list. Change the new ArrayList() into new LinkedList.*



* *ArrayList means capacity increases when elements are added, and it is not certain to be sorted.*
* *While the memory of LinkedList is relocatable.*

1. Where and how to modify if change Arraylist with Vector? What’s the difference between ArrayList and Vector?

* *Can modify in the declaration of list. Change the new ArrayList() into new Vector.*



* *ArrayList means capacity increases when elements are added, but not certain to be sorted.*
* *VectorList is similar to ArrayList, size can grow when adding elements. But, Vector is synchronized*

3. （List）Write the output of the program.

**import** java.util.\*;

**public** **class** TestList {

**public** **static** **void** main(String args[]) {

List list = **new** ArrayList();

list.add("Hello");

list.add("World");

list.add("Hello");

list.add("Learn");

list.remove("Hello");

list.remove(0);

**for**(**int** i = 0; i < list.size(); i++) {

System.***out***.println(list.get(i));

}

}

}

***Output:***

***Hello***

***Learn***

4. Select the right one?

**import** java.util.\*;

**public** **class** TestListSet {

**public** **static** **void** main(String args[]) {

List list = **new** ArrayList();

list.add("Hello");

list.add("Learn");

list.add("Hello");

list.add("Welcome");

Set set = **new** HashSet();

set.addAll(list);

System.***out***.println(set.size());

}

}

1. Compile with error
2. Compile correctly, but throw exception when running.
3. **Compile and run well, and output 3**
4. Compile and run well, and output 4

* *Answer: 3*

5 (List, Map)

**public** **class** Worker {

**private** **int** age;

**private** String name;

**private** **double** salary;

**public** Worker() {

}

**public** Worker (String name, **int** age, **double** salary) {

**this**.name = name;

**this**.age = age;

**this**.salary = salary;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **double** getSalary() {

**return** salary;

}

**public** **void** setSalary(**double** salary) {

**this**.salary = salary;

}

**public** **void** work() {

System.***out***.println(name + "is working");

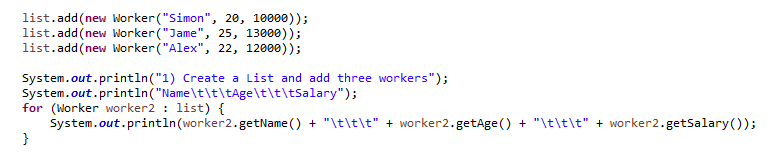
}

}

Please finish the requirement:

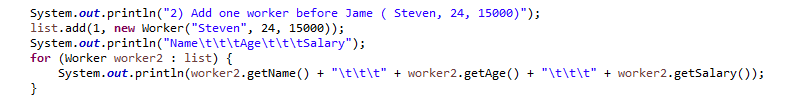
1. To create a List and add three workers, and their information shown like this:

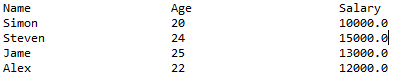
|  |  |  |
| --- | --- | --- |
| Name | Age | Salary |
| Simon | 20 | 10000 |
| Jame | 25 | 13000 |
| Alex | 22 | 12000 |



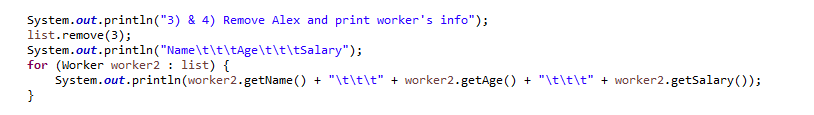


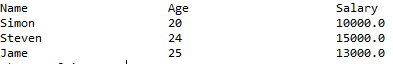
1. Add one worker before Jame ( Steven, 24, 15000)



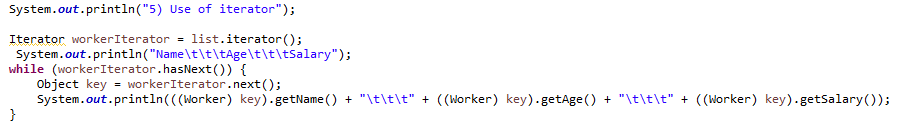


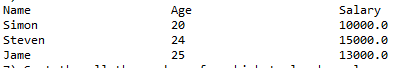
1. Remove the worker Alex’s information
2. Go through the list using for statement and print out all the worker’s information.



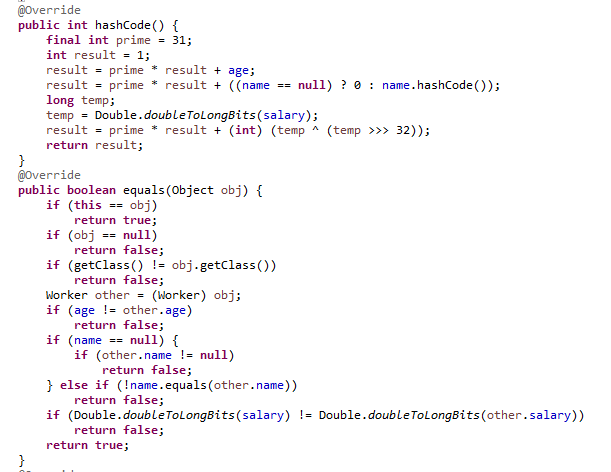


1. Go through the list using Iterator statement to call all the worker’s method work.

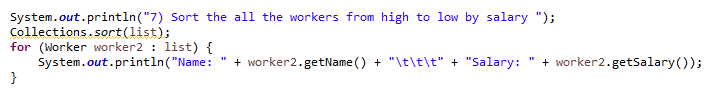




1. Over write the equals method for the class Worker. New equals method return true only if the workers’ name, age and salary are the same at the same time.



1. **Sort the all the workers from high to low by salary** and print out the all the workers information with the format “Name: “ + name + “ Salary: “ + salary.





1. Add a id to Worker class, and save the above data to workMap. Map<String, Worker > ( Worker ID, Worker) . **At least three ways t**o go through the workMap, to print out all the workder’s information with Worker id and all other information like “Worker Id: “ + “Name: “ + name + “Age: “ + age + “ Salary: “ + salary.



