W2D1 Homework

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

**ANSWER**

**Collections** is the utility class containing methods that do certain operations related to Collection.

**Collection** interface is the root interface in the collection hierarchy which represents a group of object.

**List** interface allows duplicate elements and allows inserting elements at specified index.

**Set** interface does not allow duplicate elements and can be used to compare elements.

**Map** shows elements with their key and value.

**ArrayList** – is an implementation of the list interface that uses the array’s characteristics. Allows quick adding of elements at the end but slow when in the front. This also allows the array to be resizeable.

**LinkedList** – is also an implementation of the list interface but is slower when adding elements at its last part.

**HashMap** – hashmap is an implementation of map that does not require ordering on keys or values.

**TreeMap** – Usually ordered by key unlike hashmap.

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.

**ANSWER**

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?

**ANSWER**

Change line number 5, from new ArrayList(); to  


new LinkedList():

**ArrayList** is an implementation of the list interface that uses the array’s characteristics. Allows quick adding of elements at the end but slow when in the front. This also allows the array to be resizeable. On the other hand, a **linkedlist** allows insertion or removals using iterators but only sequential access of elements. This makes the linkedlist slow when finding a specific position in the list depending on the size of the list.

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

**ANSWER**

Modify line number 5, change from new ArrayList(); to



**Arraylist** allows non synchronization unlike **vectors** which needs to be sychronized

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));

}

}

}

**ANSWER**

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");

}

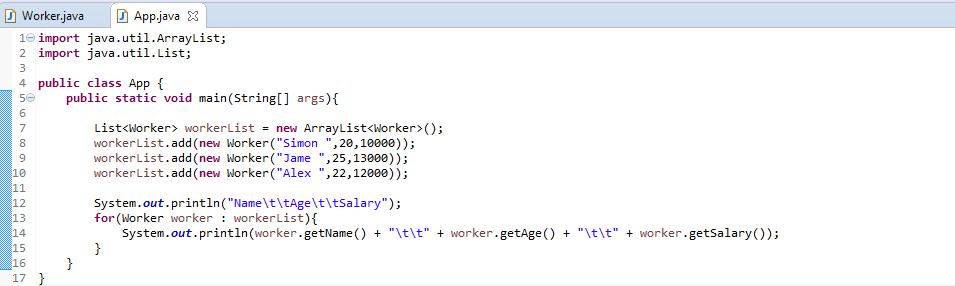
}

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 |

**ANSWER**



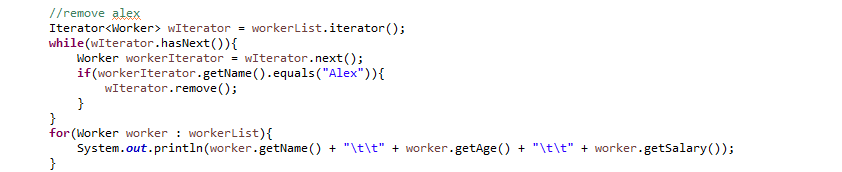
1. Add one worker before Jame ( Steven, 24, 15000)

**ANSWER**



1. Remove the worker Alex’s information

**ANSWER**



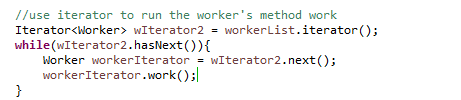
1. Go through the list using for statement and print out all the worker’s information.

**ANSWER**

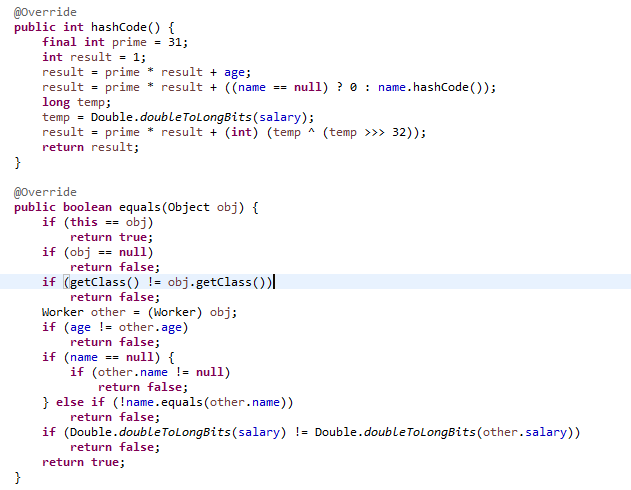


1. Go through the list using Iterator statement to call all the worker’s method work.

**ANSWER**

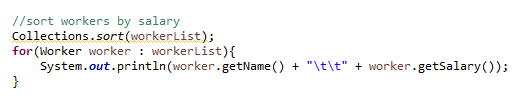


1. Over write the equals method for the class Worker. New equals method return true only if the workers’ name, age and salary are the same at the same time.

**ANSWER**

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.

**ANSWER**



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.

**ANSWER**

