LAB – 8 – HOMEWORK

**Problem 1:** Practice functions as first-class citizens and identify the suitable functional Interface and its implementation for the given tasks. Refer to the Problem1.java file for the startup code.

**Problem 2:** Get practice on Sorting.

**class** Product {

**final** String title;

**final** **double** price;

**final** **int** model;

**public** String getTitle() {

**return** title;

}

**public** **double** getPrice() { **return** price;

}

**public** **int** getModel() {

**return** model;

}

**public** Product(String title, Double price, **int** model) { **this**.title = title;

**this**.price = price;

**this**.model = model;

}

@Override

**public** String toString() {

**return** String.*format*("\n %s : %s : %s", title, price, model);

}

}

1. Sort by implementing a comparator for price attributes and printing the product list.
2. Sort by implementing a comparator for title attribute and print product list.
3. Implement the sort method so that only one type of Comparator is used for tasks a & b in a Java 7 Way using closure.
4. If the title is the same, use the model as another attribute to sort it. Do this by using lambdas. (Java 8 Way)

Task a & b – Using separate comparators – not closure (refer: comparator2 package )

Task c: Refer comparator3 package.

Task d: Refer closures.java8 package.

**Get practice to use method references**

3. In the lecture, one of the examples of a method reference of type *object::instanceMethod* was this::equals. Since every lambda expression must be converted to a functional interface, find a functional interface in the java.util.function package that would be used for this lambda expression.

Hint #1: The implicit reference `this' refers to the active object. So, to answer this question, Create a class MyClass with two attributes x and y of any type. Override the equals() method.

In which you have referenced this::equals with an appropriate type(Suitable functional Interface), add a method myMethod(MyClass cl) [testing method to check the equality] which uses this method expression to return true if cl is equal to 'this'.

Hint #2: Take a look at the api docs here:

<http://docs.oracle.com/javase/8/docs/api/java/util/function/package-summary.html>

Code Template

public class MyClass {

int x ; String y;

public MyClass(int x, String y) {

this.x = x; this.y = y;

}

// testing method to check the equality

public void myMethod(MyClass cl) {

//Implement

}

@Override

public boolean equals(Object ob) {

if(ob == null) return false; if(ob.getClass() != getClass()) return false;

MyClass mc = (MyClass)ob;

return mc.x == x && mc.y.equals(y);

}

public static void main(String[] args) {

MyClass myclass = new MyClass(1, "A"); MyClass myclass1 = new MyClass(1,"B"); myclass.myMethod(myclass); //

myclass.myMethod(myclass1);

}

}

Problem 4:

String[] names = {"Alexis", "Tim", "Kyleen", "Kristy"};

1. Use Arrays.sort() to sort the names by ignore case using Method reference.
2. Convert the sorted names array into List.
3. Print the sorted list using method reference.