MPP Final Exam

(Time allowed 2Hr:30Min, Total Questions=7, Total pages=4)

21st Dec, 2017

Problem 1. (10 points) Your package Problem1.exam contains three subpackages, **partA** and **partB**. Each contains a designated class file (respectively **PartA, PartB**), along with (possibly) other classes.

At the top of each of the designated class files, you will see a lambda expression. There are several things you will need to do with this expression.

a. Assign an appropriate type (some functional interface)
b. Express it as a method expression
c. State the type of method expression you have used
d. Express it as static inner class
e. Evaluate the lambda, the method expression and the inner class inside an **evaluator()** method.

There is a main method in each of the designated class files that attempts to run the evaluator method. In the body of the evaluator method, you should test your typed lambda expression, your method reference, and your inner class operation.

Each designated class provides a template for your work. You must follow this template. A sample solution is provided in the package Problem1.sample. Follow the format of this sample very closely.

The lambdas provided in the three parts are:

PartA: *(Double x, Double y) -> x * y < x + y*

PartB: *(CheckoutRecord record) -> record.getCheckoutEntries()*

You can not use BiFucntion in this question, using BiFunction will decrease 50% of your marks.

Problem 2. (10 Points) In your Problem2 Package you have **Employees** and **DefinedEmployees** classes with data members in it, **Main** class has test data and Static function ***getEmpList***. ***getEmpList*** is unimplemented and supposed to return **DefinedEmployees** in map from a List of **Employees** provided to it who have Gender defined as "*Male*" or "*Female*".

You can not define any local variable in getEmpList and must use single stream line to return the result without changing method signature and test data provided. Your final output should be without any Compile time and run time error.

Function signature:

```
public static Map<String,DefinedEmployees> getEmpList(List<Employees> emplist)
```

Function Call:

```
Map<String,DefinedEmployees> empmap=Main.getEmpList(emplist);
```

Expected output:

```
Key:28-12335Ayla Maersk   Value:Id:28-12335 Name:Ayla Maersk Age:21 Status:Gender Verified
Key:99-22258Daniyal Mahd   Value:Id:99-22258 Name:Daniyal Mahd Age:19 Status:Gender Verified
Key:99-22223David Lynch   Value:Id:99-22223 Name:David Lynch Age:23 Status:Gender Verified
Key:28-12331Tong Ren   Value:Id:28-12331 Name:Tong Ren Age:21 Status:Gender Verified
Key:44-85331Joyce Mac   Value:Id:44-85331 Name:Joyce Mac Age:21 Status:Gender Verified
Key:99-22255Ayman Bayed   Value:Id:99-22255 Name:Ayman Bayed Age:33 Status:Gender Verified
Key:78-12331Nicolase Bryen   Value:Id:78-12331 Name:Nicolase Bryen Age:21 Status:Gender Verified
```

Problem 3. (10 Points) In your Problem3 Package you have Students class with data members in it, Main class has test data and Static function getStdList. getStdList is unimplemented and supposed to return Students sorted by their city name and than sorted by their last name in reverse order.

You can not define any local variable in getStdList and must use single stream line to return the result.

Your final output should be as without any Compile time and run time error:

```
Student:Last name:Mahd First name:Daniyal City:Cedar Rapid
Student:Last name:Lam First name:Joyce City:Cedar Rapid
Student:Last name:Bayed First name:Ayman City:Des Moines
Student:Last name:Adam First name:Yunatan City:Des Moines
……

……

Student:Last name:Bryen First name:Nicolase City:Kirkwood
```

You can not define any local variable in getStdList and must use single stream line to return the result without changing method signature and test data provided. Your final output should be without any Compile time and run time error.

Function call:

```
Main.getStdList(stdlist).forEach((v)->System.out.println("Student:"+ v));
```

Expected output:

```
String of student after sorting is:
            Student:Last name:Mahd First name:Daniyal City:Cedar Rapid
            Student:Last name:Lam First name:Joyce City:Cedar Rapid
            Student:Last name:Bayed First name:Ayman City:Des Moines
            Student:Last name:Adam First name:Yunatan City:Des Moines
            Student:Last name:Ren First name:Tong City:Fairfield
            Student:Last name:Creek First name:Jordan City:Indianola
            Student:Last name:Maersk First name:Ayla City:Iowa City
            Student:Last name:Lynch First name:David City:Iowa City
            Student:Last name:Bryen First name:Nicolase City:Kirkwood
```

Problem 4. (10 Points) In your package Problem4.exam there are two classes: **Person** and **TestCode**. The **TestCode** class contains an unimplemented static method ***int getNamesLen(List<Person> testData)***. For this problem implement this method, which must return the sum of length of all Person object's name occurring in the input list. Your implementation must use the reduce method for Streams with single stream pipeline. (If you do not use the reduce method, you will receive less than 50% credit for this problem.)

Function call:

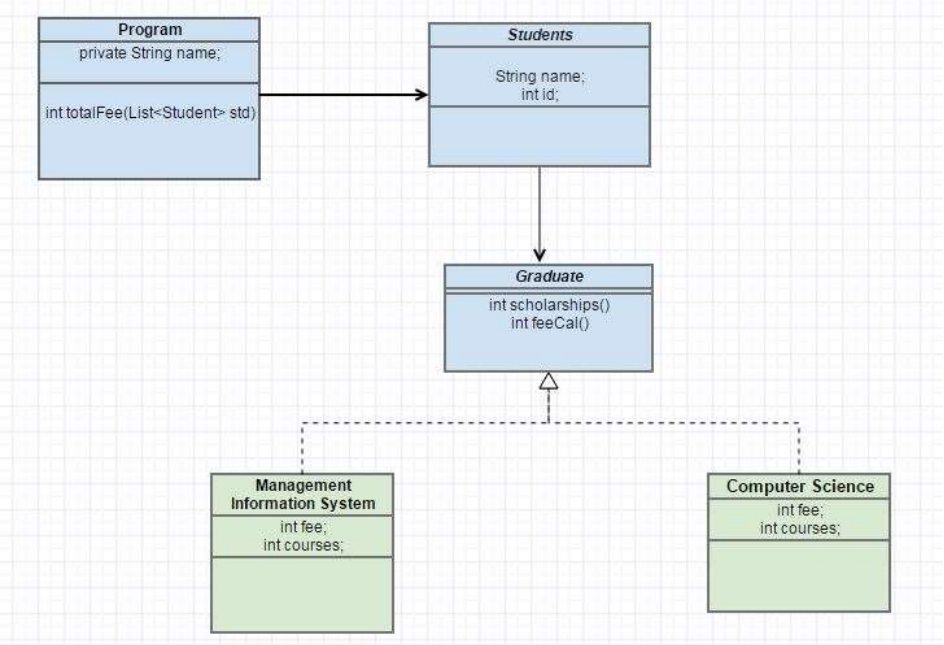
```
System.out.println("Total length of Names is:"+ getNamesLen(testData));
```

Expected output:

```
Total length of Names is:64
```

You can not define any local variable in getStdList and must use single stream line to return the result without changing method signature and test data provided. Your final output should be without any Compile time and run time error.

Problem 5. (10 Points)



UML diagram shows relationship among classes, shell of these classes is given in Problem5.Exam folder. **Garduate** interface has two functions ***int scholarships()*** and ***int fecal(),*** where ***int scholarship()*** returns scholarship amount for both **CS** and **MIS** @ fixed 2500 USD which is non changeable. While ***int feecal()*** by default calculates it by formula *fee*courses+1000*, otherwise for **CS** it is *fee*courses+1500* and for **MIS** it is *fee*courses+1700*.

Uncomment code given in the *main()* available in **Program** class, you cannot change code in *main()* as it creates type of Graduate student i.e. Computer Science or Management information Science. You are not supposed to add any further classes beside given in UML, you can add functions in shell as per requirement or can change type of functions as per requirement (instance/static/Default).

Expected output: `Total fee:4731`

Problem 6. (10 points) In Problem6.Exam you have one file **FilterItems** class, it has only one method i.e. filter which returns collection based on two parameters i) criteria provided with expression & ii) Collection. You need to implement this method using java generic so that it can work for all values commented out in main() method.

Method calls in main():

```
Collection<Integer> onlyOdds = filter(n -> n % 2 != 0, myInts);
System.out.println(onlyOdds);
Collection<Integer> onlyEvens = filter(n -> n % 2 == 0, myInts);
System.out.println(onlyEvens);
```

```
Collection<String> onlyLong = filter(n ->n.toString().length()>3, myString);
System.out.println(onlyLong);
Collection<String> onlyM = filter(n ->n.toString().startsWith("M"), myString);
System.out.println(onlyM);
```

Expected out:

```
//                      [1, 3, 5, 7, 9]
//                      [0, 2, 4, 6, 8]
//                      [Micheal, Mahd, Kristy]
//                      [Micheal, Mahd]
```

*** You can not change the test data and code in main() and can not add new user defined class or interface beside the provided one.**

Problem7. (5 Points) Write a paragraph that link any of the topic we studied during MPP course with SCI. (Type in text file and paste it in Problem7 folder).

******Best of luck******