**CS390 - Fundamental Programming Practices**

## Programming Test

## Date : 10.13.17 Duration : 1.30 Hours

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Student ID \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Problem 1 : [ Polymorphism and Sorting – 50 Points ] -** In the prob1 package of your workspace, you will find fully implemented classes Faculty and Staff representing the employees of an organization. There is Enum Gender has two values MALE, FEMALE. There is an Enum SortType has two values NAME, SALARY. Also a Main class with all the necessary statements ready to test your application. You have an interface Payable has no methods. There is a Utility class, which consists of four unimplemented Methods. Your task is to implement those methods.

**Note :** You should not change the signature in Utility.java, some arguments and return type shown with raw type. You have to modify those with Parameterized type.

**Requirements for this Problem**

**Task 1 :** In your prob1 package there is an Interface Payable has no methods. You have to identify the methods need to be included to achieve polymorphism and to perform the following

1. Calculate the total pay of all Faculty and Staff by calling calculatePay()
2. Perform the sorting using name field by calling getName()
3. Make a count on Gender by calling getGender()
4. Change Faculty and Staff class to make a relationship to achieve polymorphism. Don’t remove the keyword final and don’t change the implementation.

**Perform Task 2 – Task 5 in Utility.java class.**

**Task 2 :** The method MakePayableList() must combine the two input list of List<Faculty> and List<Staff> into a single List, appropriately typed return list. You must use the Parameterized type for the return value.

public static List /\*<implement>\*/ MakePayableList(List<Faculty> faculty, List<Staff> staff) {

/\* implement \*/

}

**Task 3 :** The method computeTotalSalary() polymorphically determines the total salary of all the employees combined in the list. It does this by calling calculatePay() on each object. If you obtain the total by first adding total salary of Faculty list and then adding the totals obtained from Staff list, you will receive no credits.

public static double computeTotalSalary(List/\*<implement>\*/ list) {

/\*implement \*/

}

**Task 4 :** This method print the Payable objects in the sorted order based on the SortType using Comparator for the type of Payable. if SortType is NAME then sort the list using getName() in alphabetical order.(A-Z) if SortType is SALARY then sort the list using calculatePay() in descending order. (Highest to lowest)

**Hint : Comparator signature**

public interface Comparator{

int compare(Object o1, Object o2);

}

public static void sort(List/\*<implement>\*/ list, SortType type){

/\*implement \*/

}

**Task 5 :** This method return the count of MALE or FEMALE based the input argument. if Gender type is MALE then return the MALE count using getGender(). if Gender type is FEMALE then return the FEMALE count using getGender()

public static int GenderCount(List/\*<implement>\*/ list, Gender type){

/\*implement \*/

}

-------------------------------------------------------------------