Department of Electrical and Computer Engineering
University of Puerto Rico
Mayagüez Campus

ICOM 4035 – Data Structures
Fall 2013
Midterm Exam # 1

Name: _____

Student Number: _____

Section: _____

Instructions:

1. Write your name on all pages of this exam now!

2. You have two hours to complete this exam. Use your time wisely. Do not spend too much time on a problem, when you can work on others.

3. There are 3 problems for a maximum score of 75 points. Complete as many problems as you can, and earn as many points as possible

GOOD LUCK!

Scores

1	/25
2	/25
3	/25
Total	/75

Problem 1. (25 points) General questions on ADTs.
Consider the following code fragment:

```
Set<String> BaseballPlayers = new DynamicSet(100);
Set<String> BasketballPlayers = new DynamicSet(100);
Set<String> FootballPlayers = new DynamicSet(100);
... // other work done later
```

a) **(5 pts)** In only one line, write a Java statement to find all players that practice every sport.

b) **(5 pts)** In only one line, write a Java statement to determine if the set of students playing both football and basketball is empty.

c) **(5 pts)** In only one line, write a Java statement to determine if student "Tom" is playing both football and basketball, but not baseball.

Problem 1 (Continuation)
Consider the following code fragment:
```
List<String> L = new ArrayList<String>(10);
```

d) (**5 pts**) In only one line, write a Java expression to determine if an element X is duplicated.

e) (**5 pts**) Consider the following code:
```
Bag<Integer> B1 = new StaticBag<Integer>(10);
Bag<Integer> B2  = new DynamicBag<Integer>(10);
B1 = (StaticBag)B2;
```

Is there any error in the 3rd line of this code? If there is one, explain.

Problem 2. (25 pts) Understanding and Using ADTs
Use the material discussed in class about ADTs to answer the following questions:

a) **(10 pts)** Write a **non-member** method named `Replacer`, which replaces from a List every copy of an element X, with a copy of element Y. For example, if L = {Bob, Joe, Bob, Ned, Bob, Ned}, then calling `Replacer(L, "Ned", "Jil")` makes L = {Bob, Joe, Bob, Jil, Bob, Jil}. The method returns the number of replacements made.

```
public int Replacer (List<String> L, String X, String Y){
```

Problem 2 (Continuation)
 b) (**15 pts**) Write a **non-member** method named `isPartition`, which determines if a collection of sets $S_1, S_2, S_3, ..., S_n$, forms a partition to another set B. All the sets are sets of String. The function receives as parameters:
   a. `theSets` – array with the sets that **might** form a partition
   b. `B` – the target set

The function returns true if the parameter `theSets` forms a partition on `B`, or false otherwise. The definition of partition is as follows. A collection of sets $S_1, S_2, S_3, ..., S_n$ forms a partition on B if and only if:

  1. No set S_i is empty
  2. For any two sets S_i and S_j, it holds that $S_i \cap S_j = \varnothing$ if $i \neq j$
  3. $B = S_1 \cup S_2 \cup S_3 \cup ... \cup S_n$

For example if the input is: theSets = [{Joe}, {Ned}, {Amy, Pol}] and the set B = {Joe, Ned, Amy, Pol}, then the method returns true.
But if theSets = [{Joe, Ned}, {Ned}, {Amy, Pol}] and the set B = {Joe, Ned, Amy, Pol}, the method returns false since {Joe, Ned} ∩ {Ned} ≠ ∅

PROVIDE YOUR ANSWER ON THE NEXT PAGE.

Problem 2 (Continuation)

```java
public boolean isPartition (Set<String>[] theSets, Set<String> B){
```

Problem 3. (25 pts) Understanding and Implementing ADTs

a) (**10 pts**) Extend the functionality of the `ArrayList` by implementing a **member** method named `removeDuplicates` which removes duplicates from the List. The relative order of the elements must be kept. For example, if L = {1, 0, 4, 3, 1, 1, 2, 4}, then `L.removeDuplicates()` make L= {1, 0, 4, 3, 2}.

```java
public void removeDuplicates(){
```

Problem 3. (Continuation)

b) (**15 pts**) Extend the functionality of the `DynamicBag` by writing a **member** method named `bagAnalizer`. This method receives as parameter a Bag B. The method **counts** the number of copies that **this** Bag holds for each element x in Bag B. The method returns a new List of `Integer`, representing the count found for each element x in bag B. For example, if the target bag is M = {1, 4, 5, 4, 11, 4, 0, 1} and B = {1, 8, 1, 4}, then the result of `M.bagAnalizer(B)` will be [2, 0, 2, 3]. As another example, if M = {Joe, Ned, Pol} and B = {Ron, Joe}, then M `M.bagAnalizer(B)` = [0, 1].

```
public List<Integer> bagAnalizer(Bag<E> B){
```