Data Structures Name -
Maps Worksheet #1

1.
What is the output of the following code segment?

```
String key = "";
Map<String, String> map = new HashMap<String, String>();
for (int k = 0; k < 3; k++)
{
      key += k;
      String value = "A";
      map.put(key, value);
      value += "B";
      map.put(key, value);
}
System.out.println(map.size());
```

A. 1
B. 2
C. 3
D. 4
E. 6

2.
A `Map` named `thingsToDo` associates a `Resort` object with a `Set` of activities available at that resort. The following code segment is intended to remove "`Golf`" from the activity sets in all resorts:

```
Iterator<String> iter = thingsToDo.keySet().iterator();
while (iter.hasNext())
      < Missing statement >
```

Which of the following should replace < *Missing statement* > ?

A. `(Set) iter.next().remove("Golf");`
B. `((Set) iter.next()).remove("Golf");`
C. `thingsToDo.remove((Set) iter.next(), "Golf");`
D. `thingsToDo.remove((Resort) iter.next(), "Golf");`
E. `((Set) thingsToDo.get(iter.next())).remove("Golf");`

3.
What is the output of the following code segment?

```
Map<String, String> m = new HashMap<String, String>();
m.put("La", "La");
m.put("La-La", "La");
m.put("La-La-La", "Ye-Ye");
Iterator<String> it = m.keySet().iterator();
while (it.hasNext())
      System.out.println(m.get(it.next()) + " ");
```

A. `La Ye-Ye`
B. `La La Ye-Ye`
C. `La La-La-La`
D. `La La La-La-La Ye-Ye`
E. `La La La-La La La-La-La Ye-Ye`

4.

```
public class PetDog implements Comparable
{
      private String myName, myBreed;
      public PetDog(String name, String breed)
      {     myName = name; myBreed = breed;     }
      public String getName() { return myName; }
      public String getBreed()      { return myBreed; }
      public boolean equals(Object other)
      {     return other != null && getName().equals(other.toString()); }
      public int compareTo(Object other)
            {return getName().compareTo(other.toString());  }
      public int hashCode() { return getName().hashCode();  }
      public String toString() { return getName() + " - " + getBreed(); }
}
```

Suppose the following object variables are declared, initialized, & instantiated:

```
PetDog honey = new PetDog("Honey", "Cocker Spaniel");
PetDog lucie = new PetDog("Lucie", "Springer Spaniel");
PetDog murray = new PetDog("Murray", "Golden Retriever");
```

The following code segment should display `Honey is a Cocker Spaniel`

```
Map<String, PetDog> map = new HashMap<String, PetDog>();
map.put(honey.getName(), honey);
map.put(lucie.getName(), lucie);
map.put(murray.getName(), murray);
System.out.println(honey.getName() + " is a " + <missing expression> );
```

Which of the following can replace <*missing expression*> ?

I. `(PetDog) map.get(honey.getName())`
II. `((PetDog) map.get(honey)).getBreed()`
III. `((PetDog) map.get(honey.getName())).getBreed()`

A. I only
B. II only
C. I and II
D. II and III
E. I, II, and III

5. A course registration database holds information about students' sign-ups for courses in a given semester. It must be able to perform two tasks quickly: determining whether or not a particular student is registered for a given course, and generating a list of all the students in a given course. Which of the following data organizations methods is the most appropriate for these tasks?

A. A map that uses students as keys and a course as a value associated with a key.
B. A map that uses courses as keys and a student as a value associated with a key.
C. A two-dimensional array of `boolean` values with rows corresponding to courses and columns corresponding to students; the element at the intersection of course and student is set to true if the student is taking the course.
D. A map that uses courses as keys; a value associated with the key is a set of all students taking this course.
E. A map that uses students as keys; a value associated with the key is a list of all courses taken by the student.