

Victoria University of Wellington  
School of Mathematics, Statistics and Computer Science

## SWEN221: Software Development

### Lab Handout (worth $\approx 1.3\%$ of overall mark)

The purpose of this Tutorial is to sharpen your understanding of Java Generics. In addition to the lecture notes, you may find it helpful to consult the Java Generics Tutorial when doing this lab. By the end of the lab, the tutor should have given you a grade based on your performance. **Be sure that you have been given a grade before you leave, since attendance at labs is mandatory.**

**NOTE:** it is also recommended that you submit your final lab code via the online submission system, which can be found on the SWEN221 homepage. This is useful in case your lab grade is lost, or you believe you have been given the incorrect grade.

#### 1 Bidirectional HashMap

The aim of this activity is to implement a bidirectional HashMap. A classic problem with the HashMap implementation found in the Java standard library is that, while it is easy to get the value a particular key maps to, it is not so easy to get the keys that a particular value maps to! For example, consider the following code snippet:

```
HashMap<String,String> map = new HashMap<String,String>();
map.put("Dave","ENGR202");
map.put("Alex","COMP205");
map.put("James","ENGR202");
```

At this point, we can easily determine what course string the key "Dave" maps to by calling `map.get("Dave")`. However, how can we find the people involved with the course "ENGR202"? The only way is using something similar to the following:

```
ArrayList<String> people = new ArrayList<String>();
for(Map.Entry<String,String> e : map.entrySet()) {
    if(e.second().equals("ENGR202")) {
        people.add(e.first());
    }
}
```

A *bidirectional* HashMap provides a method `List getKeys(V value)`. This method could be used instead of the above code as follows:

```
BiHashMap<String,String> map = new BiHashMap<String,String>();
map.put("Dave","ENGR202");
```

```
map.put("Alex", "COMP205");  
map.put("James", "ENGR202");
```

```
List<String> people = map.getKeys("ENGR202");
```

Here, we see that `map.getKeys("ENGR202")` returns the set of keys whose value is "ENGR202" in `map`.

## 2 What to do

An initial implementation of the bidirectional `HashMap` has been provided for you to help, along with a small suite of unit tests. These can be downloaded from the course webpage, on the lecture schedule.

There are several major problems with the given implementation of `BiHashMap` which you must fix. These are:

1. The current implementation of `BiHashMap` does not pass all the unit tests supplied. You should begin by fixing all of the problems. (for now comment lines that does not compile)
2. The current implementation of `BiHashMap` does not use generics. This is because it was developed before Java Generics were introduced into the Java language. Therefore, your next job is to modify the implementation of `BiHashMap` to use Java generics appropriately.
3. The current implementation of `BiHashMap` does not implement the interface `Map`. Your job now is to extend `BiHashMap` so that it implements this interface correctly. To do this, you may need to add additional methods to `BiHashMap` as required by the interface. (Now, de-comment lines you commented in the tests in part 1)

**HINT:** To complete part (3), you will need to modify `BiHashMap.putAll()` appropriately. This will require the use of wild-cards (covered in the second lecture on generics). Therefore, follow the Eclipse hints, or ask the tutor.

## Marking Guide

Each lab is worth just under 1% of your overall mark for SWEN221. The lab should be marked during the lab sessions, according to the following grade scale:

- **0:** Student didn't attend lab.
- **E:** Student did not really participate in the lab.
- **D:** Student's participation was *poor*. For example, he/she made some attempt to work on the lab, but did not complete any activities.
- **C:** Student's participation was *satisfactory*. That is, he/she completed at least one activity (e.g. fixed all initial problems with `BiHashMap`).
- **B:** Student's participation was *good*. That is, he/she has completed activities 1 and 2 (i.e. fixed all initial problems, successfully genericised `BiHashMap`)
- **A:** Student's participation was *excellent*. That is, he/she completed all activities.