COMP 587
Spring 2020

Automated Testing with Loops

1.) Consider the code snippet below, taken from the Java library:

```
public final class Math {
  public static int min(int a, int b) { ... }
}
```

Use loops to issue at least 100 unique `assertEquals` calls involving `min`. You may not implement your own `min` as part of your testing.

2.) Consider the code snippet below, adapted from the Java library:

```
public final class String {
  // Returns true if this String contains str somewhere
  public boolean contains(String str) { ... }
}
```

How might you automatically test `contains`? As a hint, think of the properties of `contains` (i.e., when it will return true or false), and then think of ways to generate inputs which are guaranteed to illustrate those properties. You do not need to write the code that performs this generation.

3.) Consider the code snippet below, adapted from the Java library:

```
public class Map<K, V> {
  // Creates an empty map
  public Map() { ... }

  // Inserts a key/value pair in the map.  Overwrites the
  // existing key/value pair, if there is one.
  public void put(K key, V value) { ... }

  // Gets the value associated with the key, or null if the
  // given key is not in the map.
  public V get(K key) { ... }

  // Gets the number of key/value pairs in the map.
  public int size() { ... }
}
```

3.a.) How might you automatically test , specifically considering the case where the map does not already contain a given key? As before, first think of the properties that should hold of these methods, and then think of how to generate inputs which illustrate those properties. You do not need to write the code that performs this generation.

3.b.) Same as 3.a, but now focus specifically on the case when the map **already** contains a given key.