**1. Write a precondition or requires clause for the method removeDuplicates, so all duplicates from List lst are removed.**

public static void removeDuplicates (List lst)

requires: input of a list with duplicate elements

\*\* Reasoning: As the exact postcondition is worded as “so all duplicates from List lst are removed”, in order for all duplicates to be removed, there needs to be duplicates in the first place. If the post condition was worded differently the precondition could be none.\*\*\*\*

**2. Write a return statement in place of the line marked TODO to make hashCode() consistent with equals().**

import java.text.NumberFormat;

class Person {

    private String firstName;

    private String lastName;

    ...

    public boolean equals (Object obj) {

        if(!(obj instanceof Person))

            return false;

        Person that = (Person) obj;

        return this.lastName.toUpperCase().equals(that.lastName.toUpperCase());

    }

    public int hashCode() {

        //TODO

        return Integer.parseInt(lastName);

    }

}

**3.**

    public static void prettyPrintApple(List<Apple> inventory, ApplePredicate aplString){

        for (Apple apple: inventory) {

            String output = aplString.printApple(apple);

            System.out.println(output);

        }

    }

\*\*\* Attached to the email is the code project that I used to test to see if this works\*\*\*\*

**4. What is a functional interface? Why is a functional interface required for a lambda expression?**

A functional interface is an interface that contains only on abstract method. They can have only one functionality to exhibit. Lambdas can be used only in the context of a functional interface, Lambda expressions allow the user to call upon the functionality of the one method in the in functional interface inline, however it assumes the use of only one method and as such does not specify which one to use. This is why only functional interfaces work, it assumes only one method.

**5. What is a function descriptor? Give two examples for function descriptors.**

A function descriptor is the signature of the abstract method in a functional interface. Two examples for function descriptors are *run* in *Runnable* and *compareTo* in *Comparable*.