

# IMPLEMENTING SETS – COIN COLLECTION

**Lab Goal:** You will implement your own set class, and employ it when working with a Coin Collection class. The Set interface places additional stipulations, beyond those inherited from the `Collection` interface, on the contracts of all constructors and on the contracts of the `add`, `equals` and `hashCode` methods. You will need to include these in your implementation.

**Lab Description :** As a coin collector, you are interested in assembling as many coins as possible. When you come across a new coin, you will only add it to your collection if it is “different” from the ones you have. Specifically, coins are different if they are different denominations, so a 1951 penny is different from a 1951 dime. However, you do not consider coins different if they are the same denomination and were issued in the same decade. Therefore, you would consider a 1951 penny and a 1957 penny as “duplicates”. Read in the sample data, using your `equals` method to add coins to your set only as your `equals` method decides that they are not duplicates. Add other needed functionality to help manage your coin collection.

## Sample Data :

```
1941, penny
2010, dime
2009, quarter
1954, nickel
1987, dime
1949, penny
1982, dime
1955, penny
2002, quarter
2008, quarter
1952, nickel
1931, penny
1989, dime
```

## Sample Output :

```
COIN COLLECTION CONTENTS
1941, penny
2010, dime
2009, quarter
1954, nickel
1987, dime
1955, penny
1931, penny
```