AP Computer Science Name: _____
Processing Data Structures of Objects

Many of the challenging questions on the AP exam will test your understanding of object oriented programming concepts in the context of data structures. Consider the following classes:

```
public class Coin {

      private double value;
      private String id;

      //constructor
      public Coin(double v, String id) {
         this.value = v;
         this.id    = id;
      }

      //accessor
      public double getValue() {
         return this.value;
      }

      //accessor
      public String getID() {
         return this.id;
      }

      public boolean equals(Coin c){
         return this.value == c.value;

      }
}
```

```
public class Wallet {

      private ArrayList<Coin> coins;

      //constructor
      public Wallet() {
         coins = new ArrayList<Coin>();
      }

      //adds the Coin c to the ArrayList
      public void add(Coin c) {
         coins.add(c);
      }

      //returns total value in wallet
      public double getTotal() {
         //implementation not shown
      }
}
```

1. Complete the getTotal method of the Wallet class:

```
public double getTotal() {
	double sum = 0;
	for (int i = 0; i < coins.size(); i++) {
		Coin c = coins.get(i);
		sum += c.getValue();
	}
	return sum;
}
```

2. A boolean method find is added to the Wallet class, which returns true if the Wallet contains the coin object with the same id as the parameter Coin, false otherwise. Write this method:

```
public boolean contains(Coin c) {
	for (int i = 0; i < coins.size(); i++) {
		Coin myCoin = coins.get(i);
		if (myCoin.getID().equals(c.getID())) {
			return true;
		}
	}
	return false;
}
}
```

3. Now suppose we had a class called Bank (not shown) that has the following instance variable.

```
private ArrayList<Wallet> wallets;
```

Write a mutator method in the bank class called removeEmptyWallets that removes all Wallets that have a total value of zero from “wallets.”

```
public void removeEmptyWallets(){
	for(int i = wallets.size()-1; i >= 0; i--) {
		Wallet w = wallets.get(i);
		if (w.getTotal() == 0) wallets.remove(i);
	}
}
```

4. For the Bank class write a method called getTotalValue that calculates the sum of the coins contained in the all the Wallet objects in “wallets.”

```
public void getTotalValue() {
	double sum = 0;
	for (Wallet w: wallets) {
		sum += wallets.getTotal();
	}
	return sum;
}
```

5. Let’s assume that we redesign the Wallet and change the ArrayList of coins to an Array of coins instead. Write the new getTotal method(). Note that elements in the array before nextCoinIndex will be null.

```
public class Wallet {
	private Coin[] coins;
	private int nextCoinIndex; //keeps track of index to add the next coin
	private static in MAX_SIZE = 100; //max number of coins you can store in a wallet
	//constructor
	public Wallet() {
	   coins = new Coin[MAX_SIZE];
	   nextCoinIndex = 0;
	}
	//adds the Coin c to the Array
	public void add(Coin c) {
	   if (end < MAX_SIZE) {
	      coins[nextCoinIndex] = c;
	      nextCoinIndex++;
	   }
	}
	//returns total value in wallet
	public double getTotal() {
	   double sum = 0;
	   for (int i = 0; i < nextCoinIndex; i++) {
	      sum += coins[i].getValue();
	   }
	   return sum;
	}
}
```

6. How does this change affect the Bank class? It doesn’t change

7. Are Coin objects mutable or immutable? Immutable
   How about Wallet objects? Mutable (add method is the mutator)