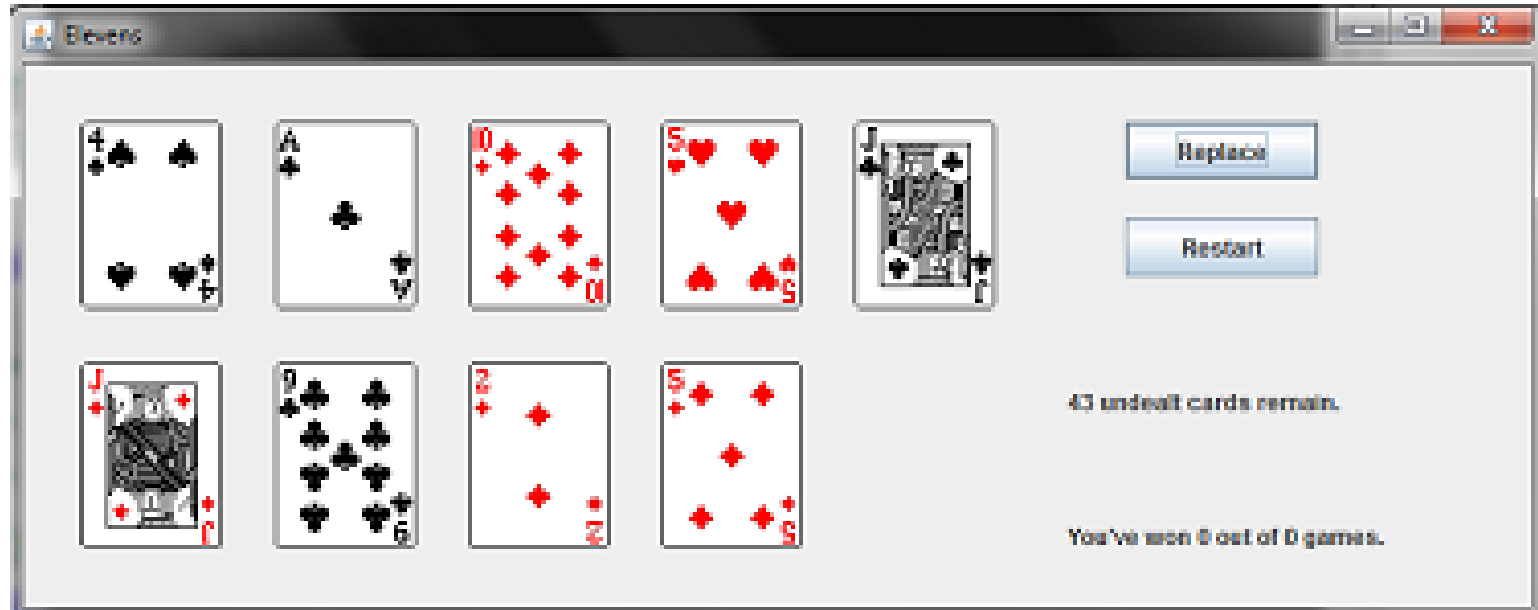


# Elevens



# What is Elevens?

**Elevens is a lab about classes and Lists.**

**List< SomeClass> is a major concept being tested by the Elevens lab.**

**Elevens is a multi-class project that uses a Card and Deck class to simulate the playing of cards.**

# What is Elevens?

**For Elevens, you must understand the basics of classes and Lists.**

**Building a class with instance variables, constructors, and methods is step 1.**

```
public class Dog
{
    private int age;
    private String name;

    public Dog( String n, int a ) {
        age = a;
        name = n;
    }

    public int getAge() {
        return age;
    }

    public String getName() {
        return name;
    }

    public String toString() {
        return "Dog - " + name + " " + age;
    }
}
```

# Basic Dog Class

**open**  
**Dog.java**  
**DogRunner.java**

# What is Elevens?

**For Elevens, you also understand Lists and how to store items in a list.**

**ArrayList is the class that will be used to store Card reference. ArrayList has many method that can be used to add and remove items.**

# **ArrayList**

## **frequently used methods**

<b>Name</b>	<b>Use</b>
<b>add(item)</b>	<b>adds item to the end of the list</b>
<b>add(spot,item)</b>	<b>adds item at spot – shifts items up-&gt;</b>
<b>set(spot,item)</b>	<b>put item at spot    <math>z[\text{spot}] = \text{item}</math></b>
<b>get(spot)</b>	<b>returns the item at spot    <math>\text{return } z[\text{spot}]</math></b>
<b>size()</b>	<b>returns the # of items in the list</b>
<b>remove()</b>	<b>removes an item from the list</b>
<b>clear()</b>	<b>removes all items from the list</b>

```
import java.util.ArrayList;
```

# Lists

```
List<String> ray;  
ray = new ArrayList<String>();  
ray.add("hello");  
ray.add("whoot");  
ray.add("contests");  
out.println(ray.get(0).substring(0, 1));  
out.println(ray.get(2).substring(0, 1));
```

OUTPUT

h

c

ray stores String references.



# open Generics.java

# What is Elevens?

**Taking a class and loading instances of that class into a List is required for this project and to be successful on the AP CS A exam.**

**The List < SomeClass > concept must be isolated and mastered before attempting to build any type of real Card game.**

```
public class Dog
{
    private int age;
    private String name;

    public Dog( String n, int a ) {
        age = a;
        name = n;
    }

    public int getAge() {
        return age;
    }

    public String getName() {
        return name;
    }

    public String toString() {
        return "Dog - " + name + " " + age;
    }
}
```

# Basic Dog Class

# List of References

```
List<Dog> ray;  
ray = new ArrayList<Dog>();
```

```
ray.add( new Dog( "fred", 11) );  
ray.add( new Dog( "ann", 21) );
```

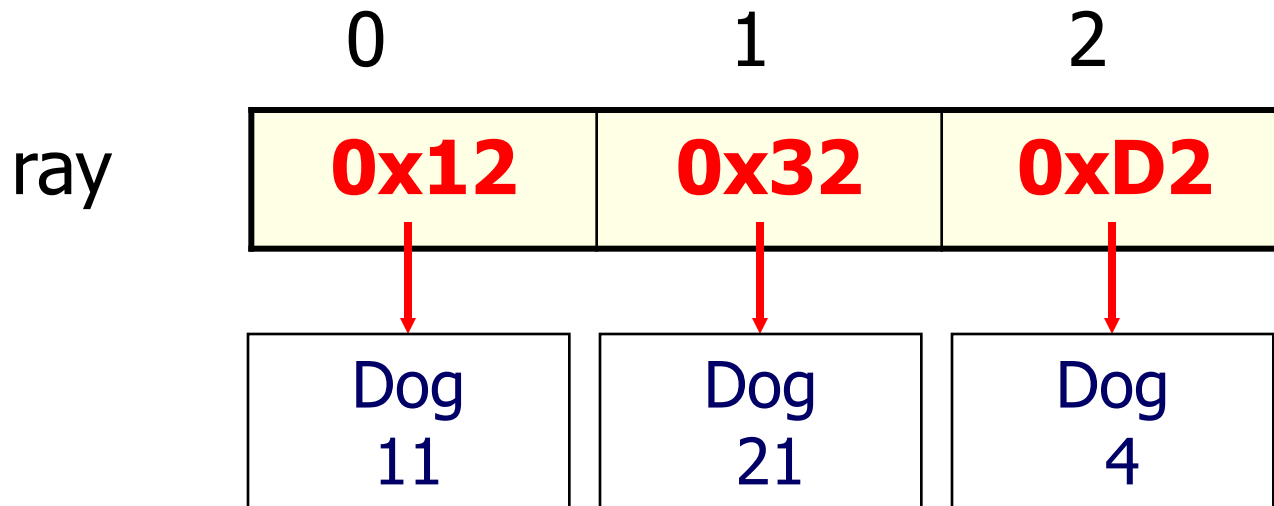
```
System.out.println( ray );
```

**OUTPUT**

[Dog - fred 11, Dog - ann 21]

# List of References

```
ray.add( new Dog( "fred", 11) );  
ray.add( new Dog( "ann", 21) );  
ray.add( new Dog( "bob", 4) );
```



**open**

**DoggiesRunner.java**

# What is Elevens?

**Elevens is a lab about classes and Lists.**

**Elevens involves making Cards, a Deck, Players, a Dealer, and a game.**

**Any card game involving multi-classes using Cards, a Deck, and Players would illustrate the concepts in Elevens.**

```

public class Card
{
    public static final String FACES[] = {"ZERO","ACE","TWO","THREE","FOUR",
        "FIVE","SIX","SEVEN","EIGHT","NINE","TEN","JACK","QUEEN","KING"};

    //instance variables
    private String suit;
    private int face;

    //constructors and modifiers

    //accessors methods

    //toString
    public String toString()
    {
        return FACES[face] + " of " + suit;
    }
}

```

# Basic Card Class

This is a simple Card class. Card is extended to make specific types of cards. `getValue( )` is implemented in each specific card.



```
public class Card
{
    private String face;
    private String suit;
    private int value;
```

```
//constructors and methods
//not shown
```

```
    public String toString() {
        return face + " " + suit + " " + value;
    }
}
```

# 11s Card Class

There are many ways to build a card class. The Card and other classes from Elevens will not be directly tested. Classes and Lists of Classes will be tested.

# **Open**

# **Activity1 Starter Code**

# **Card.java**

# **CardTester.java**

# Basic Deck Class

```
public class Deck  
{  
    private List<Card> cards;
```

```
    public Deck() {  
        cards = new ArrayList<Card>();  
        //use loops to add new Cards  
        //to the List of Cards
```

```
    }  
}
```

You must spend considerable time  
working with Lists of Classes /  
References.

**Open**  
**Activity2 Starter Code**  
**Card.java**  
**Deck.java**  
**DeckTester.java**

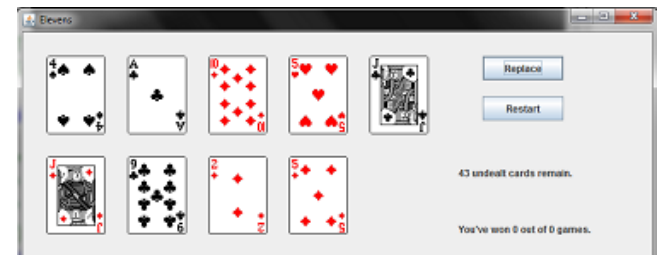
# Basic Deck Class

```
public void shuffle()  
{  
    // implement a  
    // shuffle method  
  
    // go online and research  
    // shuffle methods  
}
```

**Open**  
**Activity4 Starter Code**  
**Card.java**  
**Deck.java**  
**DeckTester.java**

# Building a Card Game

**Card games require Card classes and a Deck of Cards. Most Card games require Player classes to be created. Often, you need a List < Player > in order to simulate a real game. Creating a Card game with List < Card > and List < Player > is great prep for the AP CS A Exam.**



# Elevens Alternatives

**Any game that uses Cards and Deck of Cards is a perfectly good substitute for the Elevens project.**

**Blackjack – 21 is a great project as it has Cards, a Deck, Players, and a Dealer. Multi-player 21 is a very fun project.**



# 21 Game

**21 is a great card game that has relatively simple logic.**

**Objects needed : Card, Deck, Player, Dealer, Game**

**Deck - List of Cards**

**People - List of Players**

Start work on Elevens,  
Blackjack – 21,  
or another List<Class> lab