 

**Advanced Placement Computer Science**

[**Shenendehowa HS**](http://www.shenet.org/shen-high-school/)[**mr Hanley**](http://hanley.co.nr)

**Unit 6: Arrays/ArrayList**

**Lesson: ArrayListFILLED\_OUT**

***Last Updated:*** *1/21/2022*

The list interface in Java contains the following methods;(partial list)

**add, remove, size**

The ArrayList is a flexible class for organizing **items in a list**

The ArrayList will maintain an array and  **resize** when needed.

Consider the following class definition

public class Card {

 final static int HEARTS=3, DIAMONDS=4, CLUBS=5, SPADES=6;

 public int suit, faceValue /\*what it would count in blackjack\*/, rank /\*(who beats who)\*/;

 public String description; //ie “Queen” or “Jack” or “Two”

public Card(int st, int fv, int r, String desc){

 suit = st; faceValue=fv; rank = r; description = desc;

}

public String toString(){

 String tempSuit=””;

 switch(suit) {

   case HEARTS:

   tempSuit = “Hearts”;

 break;

  case SPADES:

   tempSuit = “Spades”;

 break;

 case DIAMONDS:

   tempSuit = “Diamonds”;

 break;

 case CLUBS:

   tempSuit = “Clubs”;

}

 return description + “ of “ + tempSuit;

}

}

To create an ArrayList of Cards use

**ArrayList hand = new ArrayList();**To add an element into an ArrayList, use

**hand.add(new Card(CLUBS,10, 12, “Queen”));**

You try adding two more Cards into the ArrayList

**hand.add(new Card(SPADES,11, 14, “Ace”));  //adds at end
hand.add(new Card(DIAMONDS,2, 2, “Two”));**

To retrieve an element from an ArrayList(typecasting required here)

**Card c = (Card) hand.get(1) ;  //got the 2nd second**

You try, retrieving your two Cards out

**Card c1 = (Card) hand.get(0) ;  //got the first**

**Card c2 = (Card) hand.get(2) ;  //got the third**

Changing an element in the ArrayList

**set replaces an object in place**

**hand.set(0,new Card(SPADES,10, 11,”Jack”));**

You try changing the 3rd card
**hand.set(2,new Card(DIAMONDS, 10, 10, “Ten”));**

Adding an element into the ArrayList, Sliding elements at position index and higher to the right

**hand.add(1,new Card(HEARTS,10,13,”King”));**

The ArrayList now has 4 elements with the former 2nd and 3rd shifted right

You try, adding a 5th element into the beginning of the ArrayList

**hand.add(0,new Card(CLUBS,10, 11,”Jack”));**

Removing an element from the ArrayList, sliding elements left if necessary

**hand.remove(0);**

Finding the size of the ArrayList

**int x = hand.size();**

You try, Looping through the cards and displaying them to the screen

**for(int i =0; i<hand.size(); i++)
    System.out.println(hand.get(i));**

Generics

Java SDK 1.5 added some really nifty features.

One badly needed feature was they added generics to the language

To declare an ArrayList of Strings, use

**ArrayList<String> movies = new ArrayList<String>();
movies.add(“Rocky I”);
String mov = movies.get(i);  //no typecast**

Now you can add and remove Strings with

You Try: Create an ArrayList of Students

**ArrayList<Student> roster = new ArrayList<Student>();**

Add a Student named Ian Fisher to the ArrayList

**roster.add(new Student(“Ian Fisher”));**

Add a second student to the ArrayList, with a name of your own choosing

**roster.add(new Student(“PJ the DJ”));**

Add a third student to the ArrayList, with a name of your own choosing

**roster.add(new Student(“Mork”));**

Add a quiz of 95 to the third student

**roster.get(2).addQuiz(95));**

Add a quiz of 99 to the first student

**roster.get(o).addQuiz(99);**

Cycle through the ArrayList and print out the students

**for(inti =0; i< roster.size(); i++){
  System.out.println(roster.get(i));
}**

FYI, there is also an enumerated type in 1.5

public class Card {

 public enum suit { HEARTS, DIAMONDS, CLUBS, SPADES};

 public suit st;

 public int faceValue /\*what it would count in blackjack\*/, rank /\*(who beats who)\*/;

 public String description; //ie “Queen” or “Jack” or “Two”

public Card(suit s, int fv, int r, String desc)

 {

 st = s; faceValue=fv; rank = r; Description = desc;

}

public String toString()

{

 String tempSuit=””;

 switch(suit) {

   case HEARTS:

   tempSuit = “Hearts”;

 break;

  case SPADES:

   tempSuit = “Spades”;

 break;

 case DIAMONDS:

   tempSuit = “Diamonds”;

 break;

 case CLUBS:

   tempSuit = “Clubs”;

}

 return description + “ of “ + tempSuit;

}

}

//Outside of the class, the type is known as Card.suit

//Outside of the class, the individual values are known as Card.suit.HEARTS and

Card.suit. SPADES,  Card.suit.DIAMONDS