

# ECAP615

## Programming in Java



Harjinder Kaur  
Assistant Professor

# Learning Outcomes



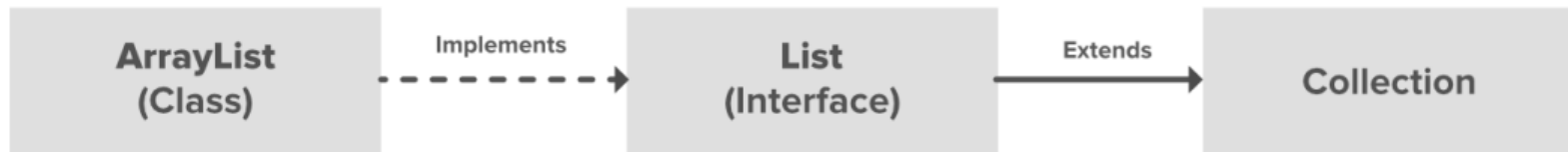
After this lecture, you will be able to

- learn the basic concept ArrayList Class,
- understand the different constructors of ArrayList Class,
- analyze the basic operations of ArrayList Class.

# ArrayList Class

- ArrayList class uses a *dynamic array* for storing the elements.
- It is like an array, but there is *no size limit*.
- We can add or remove elements anytime.
- So, it is much more flexible than the traditional array.
- It is found in the *java.util* package.

# ArrayList Class



- ArrayList is a part of collection framework and is present in java.util package.
- The ArrayList class extends AbstractList and implements the List interface.
- Array lists are created with an initial size. When this size is exceeded, the collection is automatically enlarged.
- When objects are removed, the array may be shrunk.

# Features of ArrayList

- ArrayList inherits AbstractList class and implements List interface.
- ArrayList is initialized by the size.
- However, the size is increased automatically if the collection grows or shrinks if the objects are removed from the collection.
- Java ArrayList allows us to randomly access the list.

# Features of ArrayList

- ArrayList can not be used for primitive types, like int, char, etc. We need a wrapper class for such cases.
- ArrayList in Java can be seen as a vector in C++.
- ArrayList is not Synchronized. Its equivalent synchronized class in Java is Vector.

# Creation of ArrayList

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

```
ArrayList<String> cars = new ArrayList<String>();
```

# Points to Remember

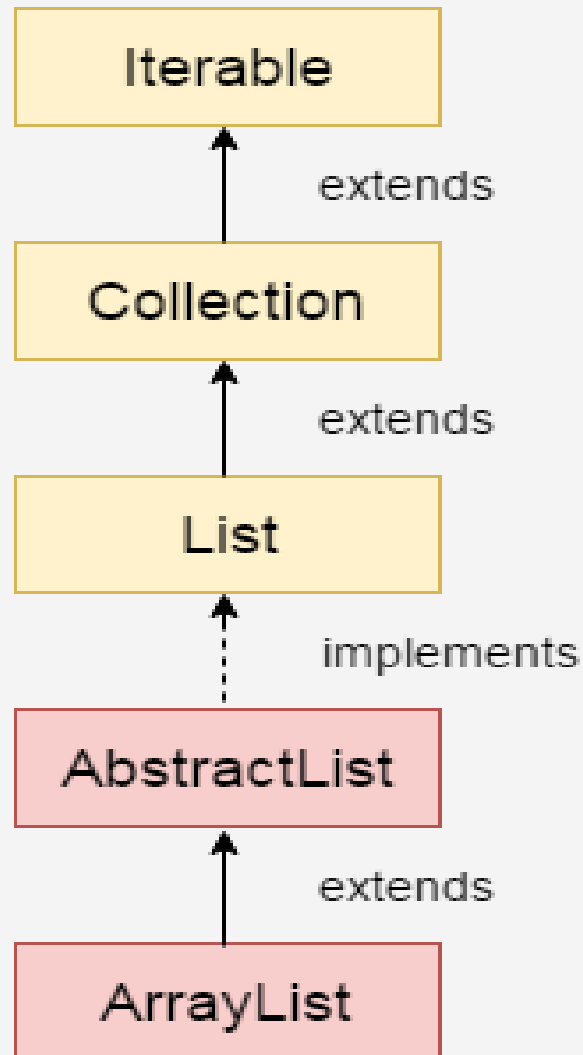
- ArrayList class can contain duplicate elements.
- ArrayList class maintains insertion order.
- ArrayList class is non synchronized.
- ArrayList allows random access because array works at the index basis.
- In ArrayList, manipulation is little bit slower because a lot of shifting needs to occur if any element is removed from the array list.



# ArrayList Constructors

Constructor	Description
<code>ArrayList( )</code>	This constructor builds an empty array list.
<code>ArrayList(Collection c)</code>	This constructor builds an array list that is initialized with the elements of the collection c.
<code>ArrayList(int capacity)</code>	This constructor builds an array list that has the specified initial capacity.

# Hierarchy of ArrayList class



# Basic operations on the ArrayList

- Adding Elements

- `add(Object)`
- `add(int index, Object)`

- Changing Elements

- Removing Elements

- `remove(Object)`
- `remove(int index)`

# Basic operations on the ArrayList

- Access an Item
- ArrayList Size
- Loop Through an ArrayList
- Sort an ArrayList

# Add() Example

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

```
public class Main {
```

```
    public static void
```

```
main(String[] args) {
```

```
    ArrayList<String> cars =  
new ArrayList<String>();
```

```
    cars.add("Volvo");
```

```
cars.add("BMW");
```

```
cars.add("Ford");
```

```
cars.add("Mazda");
```

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

```
    } }
```

# Changing Elements

```
import java.util.ArrayList;

public class Main {

    public static void
    main(String[] args) {

        ArrayList<String> cars =
        new ArrayList<String>();

        cars.add("Volvo");
```

```
        cars.add("BMW");

        cars.add("Ford");

        cars.add("Mazda");

        cars.set(0, "Opel");

        System.out.println(cars);

    } }
```

# Removing Elements

```
import java.util.ArrayList;

public class Main {

    public static void
    main(String[] args) {

        ArrayList<String> cars = new
        ArrayList<String>();

        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
```

```
        cars.add("Mazda");
        cars.remove(0);
        System.out.println(cars);
    } }
```

# Access an Item

```
import java.util.ArrayList;

public class Main {

    public static void
    main(String[] args) {

        ArrayList<String> cars =
        new ArrayList<String>();

        cars.add("Volvo");

        cars.add("BMW");
```

```
        cars.add("Ford");

        cars.add("Mazda");

        System.out.println(cars.get(
0));

    } }
```



# ArrayList Size

```
import java.util.ArrayList;

public class Main {

    public static void
main(String[] args) {

    ArrayList<String> cars =
new ArrayList<String>();

    cars.add("Volvo");

    cars.add("BMW");
```

```
        cars.add("Ford");

        cars.add("Mazda");

    System.out.println(cars.size());

    }

}
```



That's all for now...