



**The University of the West Indies, St. Augustine
COMP 2603 Object Oriented Programming 1
2021/2022 Semester 2**

ArrayLists

An `ArrayList` is a general purpose List implementation based on an array. As the list grows or shrinks, it is re-created as necessary. An `ArrayList` can be used to store a collection of Objects.

Creating an ArrayList

```
import java.util.ArrayList; //necessary import statement

//ArrayList of Account objects. Default capacity is 10
ArrayList<Account> accounts = new ArrayList<Account>();

//ArrayList with capacity of 20 Course objects
ArrayList<Course> courses = new ArrayList<Course>(20);
```

Adding Objects to an ArrayList

```
ArrayList<Account> accounts = new ArrayList<Account>();
Account a1 = new Account (10, 500.00);
Account a2 = new Account (20, 300.00);
accounts.add(a1);
accounts.add(a2);
accounts.add(new Account (30, 700.00));
```

Retrieving an Object from an ArrayList

The `get(..)` method is used to retrieve an Object from an `ArrayList`.

```
Account a = accounts.get(0); // retrieve the Account object in the first position
```

Removing an Object from an ArrayList

The `remove(..)` method is index-based and overloaded. It can accept an integer (index position) or an Object.

```
accounts.remove(2); // remove Account at position 2
accounts.remove(a1); // remove Account object referred to be a1
```

Traversing an ArrayList - Iterator

The Iterator class can be used to traverse an ArrayList. It has two methods that we use: `next()` and `hasNext()`. See the Java API for more information: [Google: ArrayList Java](https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html)

```
import java.util.Iterator; // import statement for Iterator

Iterator<Account> iter = accounts.iterator(); // Accounts ArrayList reused
while(iter.hasNext( )){           // while there are elements in the ArrayList
    Account a = iter.next( );      // get the next element in list
    a.withdraw(10.00);             //do something useful with Account object
}
```

Traversing an ArrayList - New For Loop

```
ArrayList<Account> accounts = new ArrayList<Account>();
accounts.add(new Account(10, 300.00));
accounts.add(new Account(20, 500.00));

for(Account a : accounts){ // Account object variable 'a' stores the next element in
    // the ArrayList. Loop advances automatically
    a.withdraw(15.00);      // do something useful with Account object
}
```

ArrayList Methods (Sourced from <https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>)

Modifier and Type	Method and Description
boolean	add(E e) Appends the specified element to the end of this list.
void	add(int index, E element) Inserts the specified element at the specified position in this list.
E	get(int index) Returns the element at the specified position in this list.
int	indexOf(Object o) Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element.
boolean	remove(Object o) Removes the first occurrence of the specified element from this list, if it is present.

Modifier and Type	Method and Description
E	set(int index, E element) Replaces the element at the specified position in this list with the specified element.
int	size() Returns the number of elements in this list.