

Lists and Iterators

```
ArrayList<Integer> lst = new ArrayList<Integer>();
```

```
for (int i = 0; i < 3; i ++ ) {  
    lst.add(i*i);  
}
```

```
int j = 0;
while( j < lst.size() ) {
    System.out.println( lst.get(j) );
    j ++;
}
```

```
ListIterator<Integer> itr = lst.listIterator(0);
while( itr.hasNext() ) {
    System.out.println(itr.next());
}
ListIterator<Integer> itr1 = lst.listIterator(0);
while( itr1.hasNext() ) {
    System.out.println(itr1.next());
}
```

```
while( itr.hasPrevious() ) {  
    System.out.println(itr.previous());  
}
```

```
itr.add(9);  
itr.add(16);  
System.out.println(itr.next());  
itr.set(21);  
System.out.println(itr.previous());
```

```
System.out.println(itr1.previous());
```

Exception in thread "main" java.util.

ConcurrentModificationException

at java.util.AbstractList\$Itr.checkForComodification

(AbstractList.java:372)

at java.util.AbstractList\$Itr.previous(AbstractList.java:
386)

at ListExp1.main(ListExp1.java:41)

A New Program

```
ArrayList<Integer> lst = new ArrayList<Integer>();  
for (int i = 0; i < 10; i ++ ) {  
    lst.add(i*i);  
}  
int total = 0;  
  
while( itr.hasNext() ) {  
total = total + itr.next();  
}  
System.out.println(total);
```



```
while( itr.hasPrevious() ) {
```

```
    if (itr.previous() % 2 == 0)
```

```
        itr.set(100);
```

```
}
```

```
while( itr.hasNext() ) {
```

```
    System.out.println(itr.next());
```

```
}
```

285

100

1

100

9

100

25

100

49

100

81

Specs for LinkedList and ArrayList

```
public interface Collection<AnyType> extends
Iterable<AnyType>
{
    int size();
    boolean isEmpty();
    void clear();
    boolean contains(AnyType x);
    boolean add(AnyType x);
    boolean remove(AnyType x);
    java.util.Iterator<AnyType> iterator();
}
```

```
public interface Iterator<AnyType>
{
    boolean hasNext();
    AnyType next();
    void remove();
}
```

```
public interface List<AnyType> extends Collection<AnyType>
{
    AnyType get (int idx);
    AnyType set (int idx, AnyType newVal);
    void add(int idx, AnyType x);
    void remove(int idx);

    ListIterator<AnyType> listIterator(int pos);
}
```

```
public interface ListIterator<AnyType> extends
Iterator<AnyType>
{
    boolean hasPrevious();
    AnyType previous();
    void add(AnyType x);
    void set(AnyType newVal);
}
```

A puzzle for your homework

```
LinkedList<Integer> lst1 = new LinkedList<Integer>();  
LinkedList<IntHolder> lst2 = new LinkedList<IntHolder>();
```

```
lst1.add(1);
```

```
lst1.add(4);
```

```
lst1.add(9);
```

```
System.out.println(lst1.contains(new Integer(4)));
```

```
class IntHolder {  
    int i;  
    public IntHolder(int i1) {  
        i = i1;  
    }  
}
```



```
[kvaradar@p-Inx204 lists]$ java HW3Q  
true  
false  
[kvaradar@p-Inx204 lists]$
```

Other HW question

```
import java.util.*;
```

```
class HW3Q1 {
```

```
    public int position(List<String> lst, String s1, String s2) {
```

```
        /* Should return 0 if either s1 or s2 or both are not  
        present in lst.
```

```
        Should return 1 if the first occurrence of s1 is before the  
        first occurrence of s2; and -1 otherwise. Use an iterator.
```

```
        */
```

```
    }
```

```
}
```