CS202: IT Workshop

Java

Arrays and ArrayList

Ref:

1. Harvey Deitel, Paul Deitel, **Java: How to Program**, 9/e, Prentice Hall India.

2. https://docs.oracle.com/

Limitations of array

- Array has many usages; any limitation of array?

Limitations of array: ArrayList

- ❑One of the limitation of array is its FIXED size
- ❑ArrayList is a data structure where elements can be added, removed dynamically
- ❑In addition, it supports various in-built methods
- ❑We can create an *ArrayList* of only **reference** types (as Objects)

*Java also supports other data structures like LinkedList, Stack, Vector, Set, etc.

ArrayList in Java

- ❑ One of the limitation of array is its FIXED size
- ❑ ArrayList is a data structure where elements can be added, removed dynamically
- ❑ In addition, it supports various in-built methods
- ❑ We can create an *ArrayList* of only **reference** types
- ❑ **What about primitive types (int, double etc.)?**

ArrayList in Java

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

items.add("red");
items.add(0, "yellow");
items.add("green");

for ( int i = 0; i < items.size(); i++ )
System.out.print( items.get( i ) );

items.remove(1);

for ( int i = 0; i < items.size(); i++ )
System.out.print( items.get( i ) );

System.out.print ( items.indexOf("green") );

items.clear();
```

```
$ yellow red green

$ yellow green

$ 1
```

add(item) inserts item at rear end

add (pos, item) inserts at the pos

remove(index) removes item given by index

get(index) returns item of index

indexOf(item) returns the index of the first occurrence of item

clear() removes all items

*we need to import "**java.util.ArrayList**"

ArrayList in Java

Code Demonstration (ArrayListExample.java)

ArrayList in Java

❑ **We can also create ArrayList of the Class we create in our code (e.g. Student)**

Code Demonstration (StudentArrayList.java**)**

Wrapper class

- Many supported data structures in Java (such as Arraylist) works only with Reference type
- Wrapper class converts (wraps) primitive type to its equivalent Reference type (int → Integer, double → Double, etc.)
- Class "Integer" contains the equivalent int as a field within it (along with various other fields).

```
public final class Integer extends Number implements Comparable<Integer> {
        . . . . .
        private final int value;
        . . . . .
        public Integer (int value) {
                this.value = value;
                }
        . . . . .
```

Definition of Class Integer

(present in Java)

Wrapper class example

- To create an **Integer object** from a primitive **int**

```
Integer myIntObj = new Integer(5); //old version

Integer myIntObjNew = 7; //new version
```

This is called Autoboxing

- To get the equivalent **int** value from an **Integer** obj

```
int myPrimInt = myIntObj.intValue();

int myPrimInt = myIntObj; //new version
```

This is called Unboxing

Wrapper class

- We can also use wrapper class for other primitive data types

```
Character myCharObj = 'x';
char myChar = myCharObj;

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

Code Demonstration (WrapperDemo.java)

ArrayList and Wrapper class

- Now, if we want to create an ArrayList of "integers" (i.e. int)?

ArrayList and Wrapper class

- Now, if we want to create an ArrayList of "integers" (i.e. int)?
- We can create that using the wrapper class **Integer**

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

integerList.add(24);
```

Code Demonstration (ArrayListExample.java)