

### 3. ArrayList Class: Using Wrapper Classes

To create an *ArrayList* that stores **ints** we can use the *java.lang.Integer* wrapper class with autoboxing.

```
ArrayList<Integer> listOfInt = new ArrayList<>();  
listOfInt.add(1);  
listOfInt.add(x);  
listOfInt.add(2 * x - 1);  
int x = listOfInt.get(2);
```

### 3. ArrayList Class: PrintReverse Application

We can now write the program that reads *ints-in.txt* and prints the numbers in reverse order to *ints-out.txt*.

```
//*****  
// CLASS: PrintReverse  
//*****  
import java.io.File;  
import java.io.FileNotFoundException;  
import java.io.PrintWriter;  
import java.lang.Integer;  
import java.util.ArrayList;  
import java.util.Scanner;  
  
/**  
 * Reads a file named "ints-in.txt" containing integers and writes the integers to  
 * "ints-out.txt" in reverse order.  
 */  
class PrintReverse {  
    public static void main(String[] args) throws FileNotFoundException {  
        // Create an ArrayList of Integers.  
        ArrayList<Integer> list = new ArrayList<>();
```

### 3. ArrayList Class: PrintReverse Application (continued)

```
// Open "ints-in.txt" for reading.
Scanner scanner = new Scanner(new File("ints-in.txt"));

// Read the integers from "ints-in.txt" placing them in the ArrayList.
while (scanner.hasNext()) {
    list.add(scanner.nextInt());
}

// Close the input file.
scanner.close();

// Open "ints-out.txt" for writing.
PrintWriter out = new PrintWriter(new File("ints-out.txt"));

// Walk through the ArrayList in reverse order, printing each element to the
// output file.
for (int i = list.size() - 1; i >= 0; --i) {
    out.println(list.get(i));
}

// Close the output file.
out.close();
}
}
```