```java
// an example of Java wrapper classes of the primitive types

class typeWrapDemo {
  public static void main ( String args[] ) {
    int x = 5;
    Integer x_obj = new Integer(x);
    Double y_obj = new Double("3.141");

    showOutput(x_obj);
    showOutput(y_obj);
  }

  public static void showOutput(Number n) {
    short s;
    System.out.println( "Number as a string: " + n.toString() );

    s = n.shortValue();
    System.out.println( "Number as a short: " + s );
  }
}
```

```
// an example of some Java arrays.

class arrayDemo {
  public static void main( String args[] ) {
    int i;

    System.out.println("Arguments:");
    for (i = 0; i < args.length; i++) {
      System.out.println( "   " + args[i] );
    }

    int c[];

    c = new int[12];
    for (i = 0; i < c.length; i++) {
      c[i] = i * i;
    }

    System.out.println("Array c[]:");
    printIntArray(c);

    int d[] = { 3, 4, 5, 6, 7, 8 };

    System.out.println("Array d[]:");
    printIntArray(d);

    modifyIntArray(d);

    System.out.println("Array d[] after modification:");
    printIntArray(d);
  }

  public static void printIntArray( int array[] ) {
    int i;

    System.out.println("Element\tValue");
    for (i = 0; i < array.length; i++) {
      System.out.println( i + "\t" + array[i] );
    }
  }

  public static void modifyIntArray( int array[] ) {
    int i;

    for (i = 0; i < array.length; i++) {
      array[i] = array[i] * (array[i] - 1);
    }
  }
```

```
}
```

```
// Driver for Employee hierarchy -- illustrating polymorphism
// and arrays
import java.text.DecimalFormat;

public class Test {
  public static void main( String args[] ) {
    DecimalFormat precision2 = new DecimalFormat( "#.00" );
    Employee workers[];

    workers = new Employee[4];

    workers[0] = new Boss( "John", "Smith", 800.00 );
    workers[1] = new CommissionWorker( "Sue", "Jones",
                                 400.0, 3.0, 150);
    workers[2] = new PieceWorker( "Bob", "Lewis", 2.5, 200 );
    workers[3] = new HourlyWorker( "Karen", "Price", 13.75, 40 );

    for (int i = 0; i < workers.length; i++) {
      System.out.println(workers[i].toString() + " earned $" +
                 precision2.format( workers[i].earnings() ) );
    }
  }
}
```

```java
// Fig. 8.2: StringMisc.java
// This program demonstrates the length, charAt and getChars
// methods of the String class.
//
// Note: Method getChars requires a starting point
// and ending point in the String. The starting point is the
// actual subscript from which copying starts. The ending point
// is one past the subscript at which the copying ends.

public class StringMisc {
  public static void main( String args[] ) {
    String s1;
    char charArray[];

    s1 = new String( "hello there" );
    charArray = new char[ 5 ];

    // output the string
    System.out.println( "s1: " + s1 );

    // test the length method
    System.out.println( "Length of s1: " + s1.length() );

    // loop through the characters in s1 and display reversed
    System.out.print( "The string reversed is: " );

    for ( int i = s1.length() - 1; i >= 0; i-- ) {
      System.out.print( String.valueOf( s1.charAt( i ) ) );
    }
    System.out.println();

    // copy characters from string into char array
    s1.getChars( 0, 5, charArray, 0 );
    System.out.print( "The character array is: " );

    for (int i = 0; i < charArray.length; i++) {
      System.out.print( charArray[i] );
    }
    System.out.println();
  }
}
```

```java
// Fig. 8.14: StringBufferChars.java
// The charAt, setCharAt, getChars, and reverse methods
// of class StringBuffer.

public class StringBufferChars {
  public static void main( String args[] ) {
    StringBuffer buf;

    buf = new StringBuffer( "hello there" );

    System.out.println( "buf = " + buf.toString() );
    System.out.println( "Character at 0: " + buf.charAt( 0 ) );
    System.out.println( "Character at 4: " + buf.charAt( 4 ) );

    char charArray[] = new char[ buf.length() ];
    buf.getChars( 0, buf.length(), charArray, 0 );

    System.out.print( "The characters are: " );
    for (int i = 0; i < charArray.length; i++) {
      System.out.print( charArray[i] );
    }
    System.out.println();

    buf.setCharAt( 0, 'H' );
    buf.setCharAt( 6, 'T' );
    System.out.println( "buf = " + buf.toString() );

    buf.reverse();
    System.out.println( "buf = " + buf.toString() );
  }
}
```

```
// First demonstration of exception handling

import java.io.*;

public class Exceptions1 {
  public static void main ( String args[] ) {
    BufferedReader in = new BufferedReader(
         new InputStreamReader( System.in ) );

    try {
      String s;
      while (true) {
        s = in.readLine();
        System.out.println(s);
      }
    }
    catch ( EOFException e ) {
      System.out.println("EOF Detected");
    }
    catch ( IOException e ) {
      System.out.println("Some other I/O Exception detected");
    }
  }
}
```

```java
// Second demonstration of exception handling

public class Exceptions2 {
  public static void main ( String args[] ) {
    int x[] = {3, 4, 5, 6};

    try {
      for (int i = 0; i < x.length + 1; i++) {
        System.out.println( x[i] );
      }
    }
    catch ( ArrayIndexOutOfBoundsException e ) {
      System.out.println("Array index out of bounds");
    }
  }
}
```