Objectives:
1. To understand the difference between atomic and composite data types.
2. To understand the difference between unstructured and structured composite data types.
3. To understand how java implements arrays.
4. To understand the process of passing an array as an argument.
5. To understand the difference between an array and the information stored within the array.
6. To understand the role of arrays of Objects.

Reading Assignment:
1. Nell/Chip/Mark, Chapter 10

Concepts:
1. Atomic and Composite data types..
2. Two forms of Composite data types.
3. One-Dimensional Arrays.
4. Accessing individual components in a one-dimensional.
5. Arrays of Objects.
1. Atomic and Composite Data Types:
   - Atomic data types: Elements that have no component parts.
   - Composite data types: A data type that allows a collection of values to be associated with an identifier of that type.

2. Two forms of composite data types:
   - Unstructured Data Type
   - Structured Data Type.

3. One Dimensional Arrays:
   - Declaring an Array: Data-Type [] Array-Name;
   - Creating an Array: Array-Name= new Data-Type [IntExpression];
   - Declaring and Creating an Array with an Initializer List

4. Accessing individual components in a one-dimensional.
   - Array-Component-Access:
   - Array-Name [index-Expression]

5. Arrays of Objects:
   - Arrays of Strings
   - Arrays of User-Defined Objects.
Objectives:
   1. To be familiar with concepts of inheritance.

Student Activities:
   1. Perform the Nell Dale chapter 10 exercises 1 and 2.

Lab Solution:

```java
//lesson 10-2
//Program reverse reads numbers into an array
//and prints them out in reverse order.

import java.io.*;
public class Reverse
{
   public static void main(String[] args) throws IOException
   {

      final int MAX=10;
      BufferedReader inFile = new BufferedReader( new
      FileReader("reverse.dat"));
      int[] numbers;
      numbers = new int[MAX];

      int value;
      int index;
      int sum;
      sum =0;
```
```java
for (index=0; index<numbers.length; index++)
{
    value = Integer.parseInt(inFile.readLine());
    numbers[index] = value;
    numbers[index] = Integer.parseInt(inFile.readLine());
    sum = sum + numbers[index];
}

for (index = MAX-1; index>=0; index--)
{
    System.out.println(" "+ numbers[index]);
}
System.out.println("Sum is " + sum);
```