```
1  //for-each loop (Enhanced for loop)
2
3  /*
4  For-each is another array traversing technique like for loop, while loop,
5  do-while loop introduced in Java5 (1.5). It's commonly used to iterate
6  over an array or a Collections class (eg, ArrayList). It is known as the
7  for-each loop because it traverses each element one by one.
8
9  The drawback of the enhanced for loop is that it cannot traverse the
10 elements in reverse order. Here, you do not have the option to skip any
11 element because it does not work on an index basis. Moreover, you
12 cannot traverse the odd or even elements only.
13
14 In this for-loop, we don't have access to the index, so we cannot replace
15 elements at any specific index.
16
17 Enhanced for loop will execute in a sequential manner i.e counter will
18 always increase by one.
19  */
20
21 /*
22 Syntax:
23 for (datatype variableName : array/collection)
24 {
25    //statement/s;
26 }
27  */
28 public class Main {
29    public static void main(String[] args) {
30       int marks[]=new int[]{80,88,92,78,96};
31       for (int i:marks)
32          System.out.println(i);
33    }
34 }
35
36 /*
37 Sequence– the order we want the computer to execute the instructions
38 we provide as programmers. For example, do this first, then do this,
39 then do that, and so forth.
40
41 Selection– selecting which path of an algorithm to execute depending
42 on some criteria. For example, if you passed a class in school, then
43 we execute the operations that clap and cheer and play a song. But if
44 you didn't pass the class, then maybe we would say, "Better luck next
```

time!"

Iteration– looping or repeating. Many times, we want to be able to repeat a set of operations a specific number of times or until some condition occurs.
*/