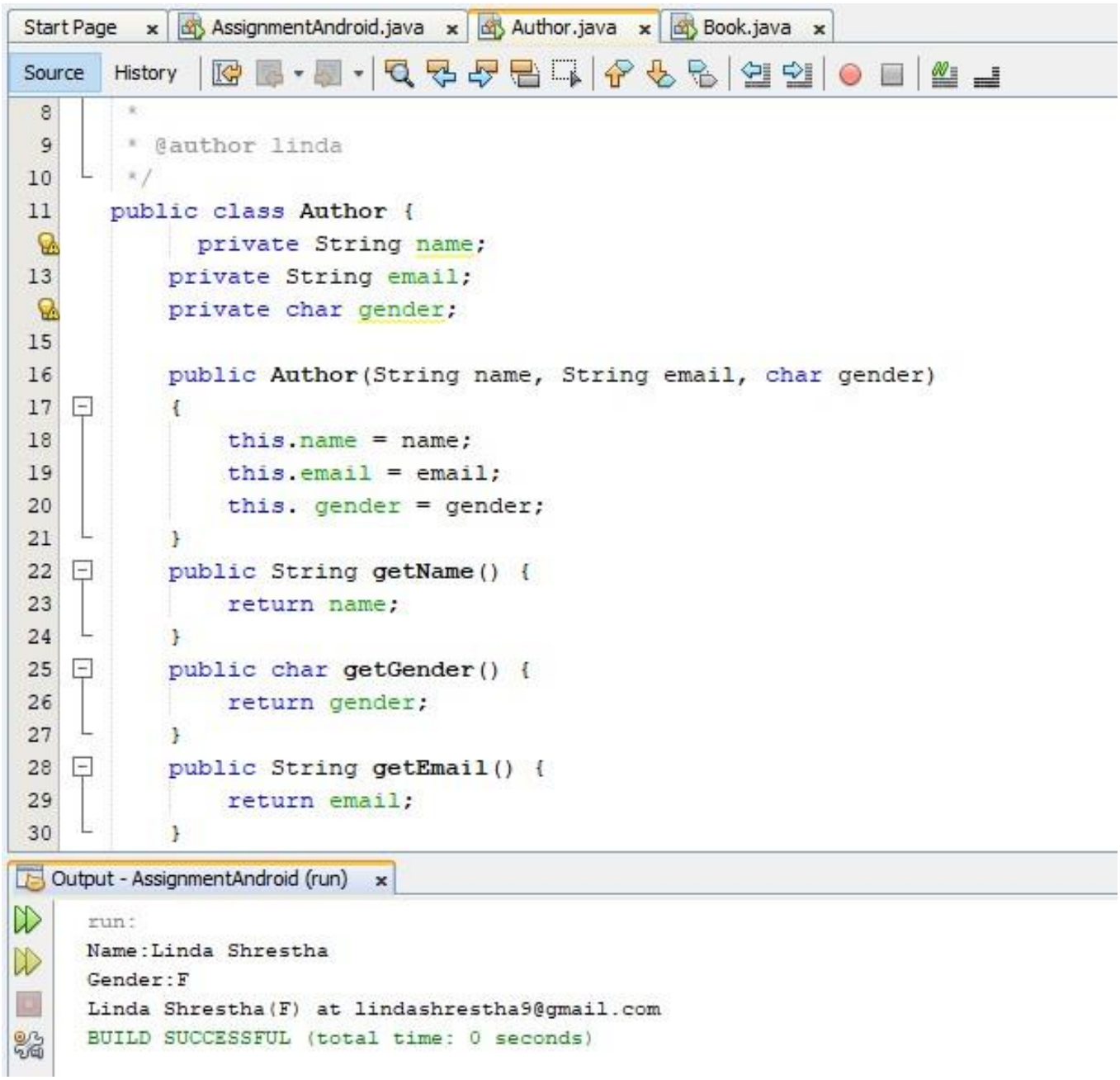
1. ANS


Start Page
AssignmentAndroid.java
Author.java
Book.java
Source
History
*
* @author linda
*/
public class Author {
private String name;
private String email;
private char gender;
public Author(String name, String email, char gender)
{
this.name = name;
this.email = email;
this. gender = gender;
}
public String getName() {
return name;
}
public char getGender() {
return gender;
}
public String getEmail() {
return email;
}
Output - AssignmentAndroid (run)
run:
Name:Linda Shrestha
Gender:F
Linda Shrestha(F) at lindashrestha9@gmail.com
BUILD SUCCESSFUL (total time: 0 seconds)

```
31      public void setEmail(String email) {
32          this.email = email;
33      }
35      public String toString() {
36          return name + " (" + gender + ") at " + email;
37      }
38      public static void main(String[] args) {
39          // Test constructor and toString()
40          Author auth = new Author("Linda Shrestha","lindashrestha9@gmail.com",'F');
41          System.out.println("Name:" + auth.getName());
42           System.out.println("Gender:" + auth.getGender());
43           System.out.println(auth.getName()+"("+auth.getGender()+") at "+auth.getEmail());
44      }
45
46
47  }
```

Output - AssignmentAndroid (run)

```
run:
Name:Linda Shrestha
Gender:F
Linda Shrestha(F) at lindashrestha9@gmail.com
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. ANS

```
 *
 * @author linda
 */
public class Book {
    private String name;
    private Author author;
    private double price;
    private int qty;

    public Book(String name, Author author, double price, int qty) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.qty = qty;
    }
    public String getName() {
        return name;
    }
    public Author getAuthor() {
        return author;
    }
    public double getPrice() {
        return price;
```

```
Output - AssignmentAndroid (run)
run:
War and Peace by Linda Shrestha (F) at lindashrestha9@gmail.com
BUILD SUCCESSFUL (total time: 1 second)
```

```
31      }
32      public void setPrice(double price) {
33          this.price = price;
34      }
35      public int getQty() {
36          return qty;
37      }
38      public void setQty(int qty) {
39          this.qty = qty;
40      }
42      public String toString() {
43          return "'" + name + "' by " + author;
44      }
45      public static void main(String[] args) {
46          Author auth = new Author("Linda Shrestha", "lindashrestha9@gmail.com", 'F');
47          Book bookibook = new Book("War and Peace",auth,1500,10);
48          System.out.println(bookibook.getName()+" by "+auth);  // Book's toString()
49
50      }
    }
```

Output - AssignmentAndroid (run)

```
run:
War and Peace by Linda Shrestha (F) at lindashrestha9@gmail.com
BUILD SUCCESSFUL (total time: 1 second)
```

‘

3. ANS

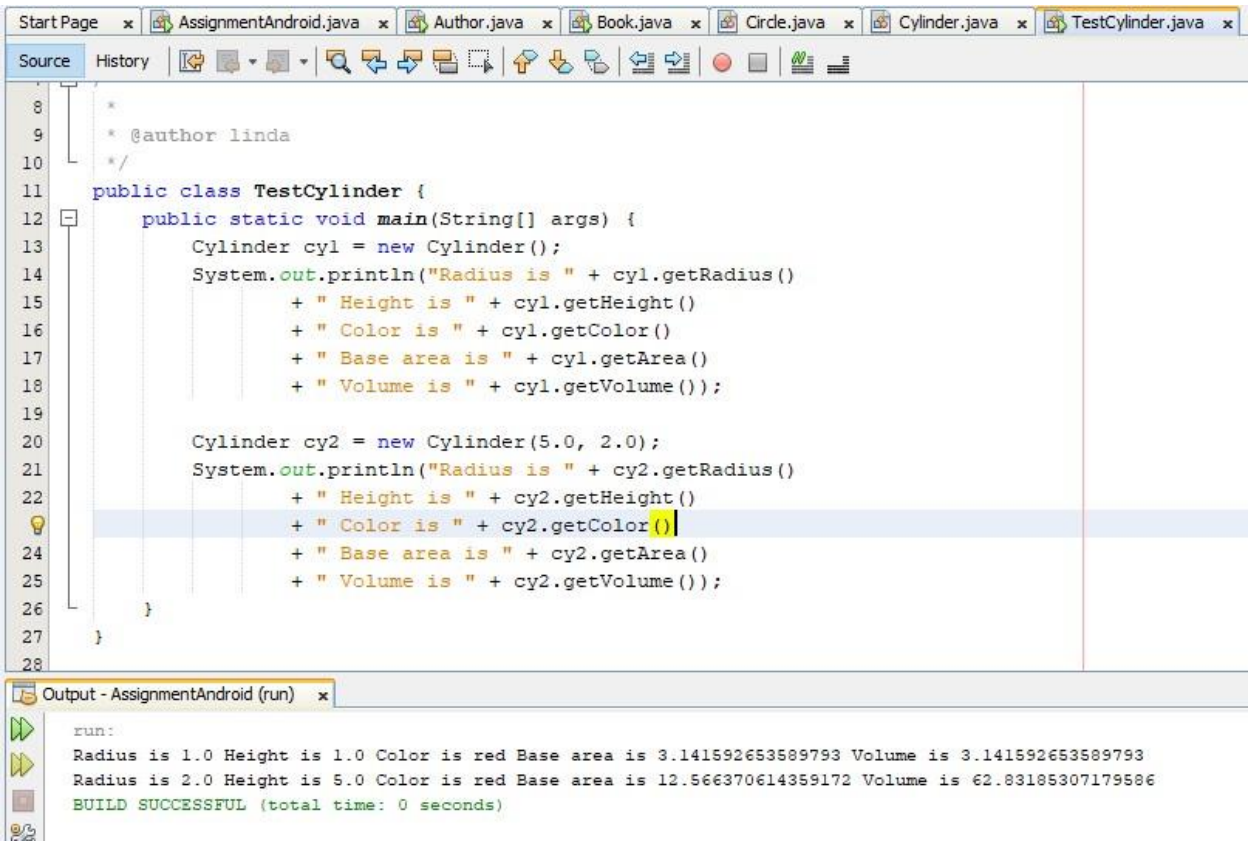
```
 8   *
 9   * @author linda
10   */
     public class Circle {
12       private double radius;
13       private String color;
14       public Circle() {
15           this.radius = 1.0;
16           this.color = "red";
17       }
18       public Circle(double radius) {
19           this.radius = radius;
20           this.color = "red";
21       }
22       public Circle(double radius, String color) {
23           this.radius = radius;
24           this.color = color;
25       }
26       public double getRadius() {
27           return this.radius;
28       }
29       public String getColor() {
30           return this.color;
31       }
32       public void setRadius(double radius) {
33           this.radius = radius;
34       }
35       public void setColor(String color) {
36           this.color = color;
37       }
39       public String toString() {
40           return "Circle[radius=" + radius + ",color=" + color + "]";
41       }
42       public double getArea() {
43           return radius * radius * Math.PI;
44       }
     }
```

```java
 *
 * @author linda
 */
public class Cylinder extends Circle{
    private double height;
    public Cylinder() {
        super();
        this.height = 1.0;
    }
    public Cylinder(double height) {
        super();
        this.height = height;
    }
    public Cylinder(double height, double radius) {
        super(radius);
        this.height = height;
    }
    public Cylinder(double height, double radius, String color) {
        super(radius, color);
        this.height = height;
    }

    public double getHeight() {
        return this.height;
    }
    public void setHeight(double height) {
        this.height = height;
    }

    public double getVolume() {
        return getArea()*height;
    }


    public String toString() {
        return "This is a Cylinder";
    }
}
```

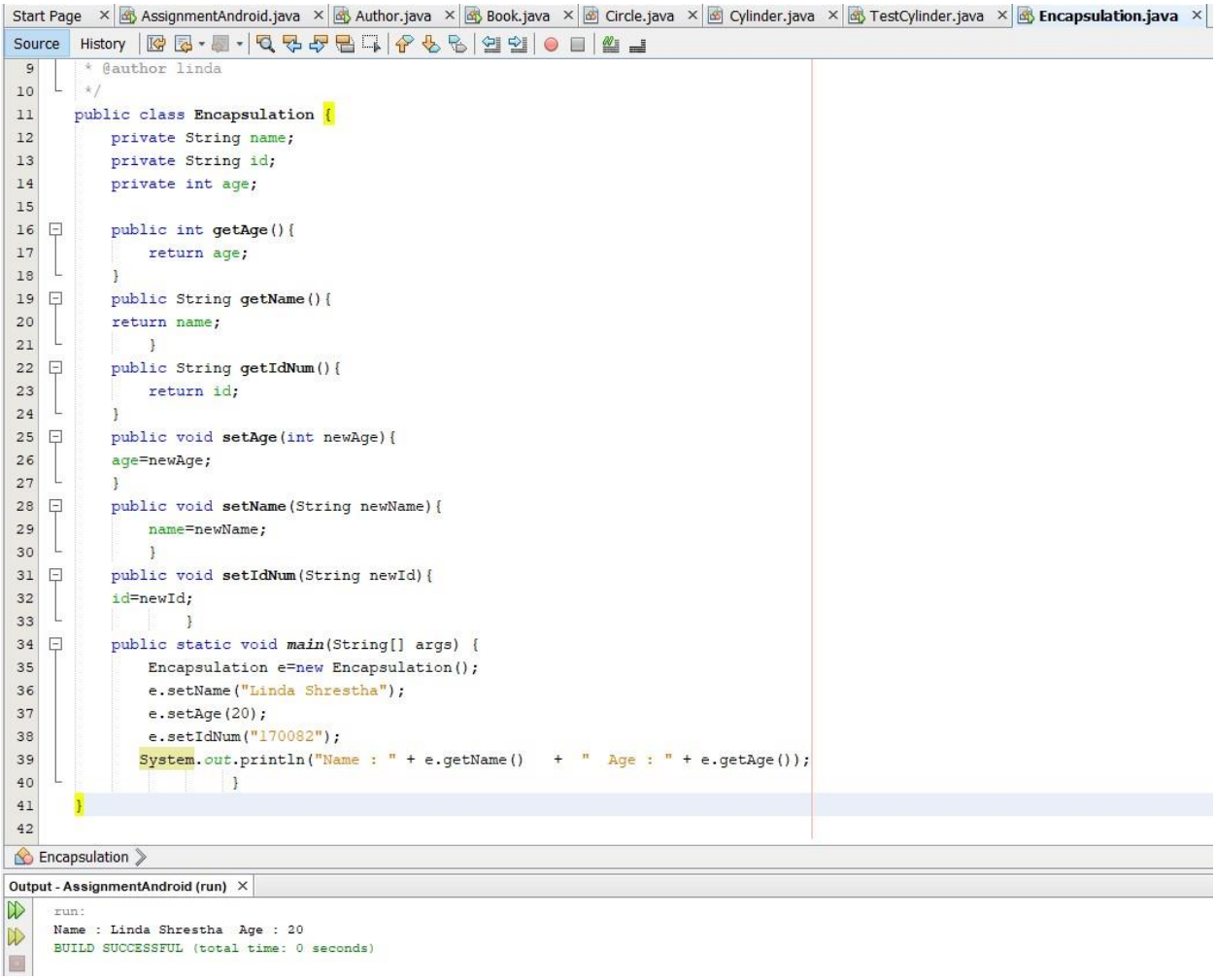

Start Page
AssignmentAndroid.java
Author.java
Book.java
Circle.java
Cylinder.java
TestCylinder.java
Source
History
*
* @author linda
*/
public class TestCylinder {
public static void main(String[] args) {
Cylinder cyl = new Cylinder();
System.out.println("Radius is " + cyl.getRadius()
+ " Height is " + cyl.getHeight()
+ " Color is " + cyl.getColor()
+ " Base area is " + cyl.getArea()
+ " Volume is " + cyl.getVolume());
Cylinder cy2 = new Cylinder(5.0, 2.0);
System.out.println("Radius is " + cy2.getRadius()
+ " Height is " + cy2.getHeight()
+ " Color is " + cy2.getColor()
+ " Base area is " + cy2.getArea()
+ " Volume is " + cy2.getVolume());
}
}
Output - AssignmentAndroid (run)
run:
Radius is 1.0 Height is 1.0 Color is red Base area is 3.141592653589793 Volume is 3.141592653589793
Radius is 2.0 Height is 5.0 Color is red Base area is 12.566370614359172 Volume is 62.83185307179586
BUILD SUCCESSFUL (total time: 0 seconds)

4. ANS

Encapsulation is the hiding of data implementation by restricting access to assessors and mutators (getters and setters)

The advantages of encapsulation are as follows:

- Make a flexible code which is easy to change and maintain.
- Increase usability
- Easy to test the code



5. ANS

Abstraction hide useless function of class to show the necessary function.it is that class which we cannot make the object but can make reference variable.

For example,

```
 9     * @author linda
10     */
11    public abstract class DeclareAbstract {
12        // declare fields
13        // declare nonabstract methods
14        abstract void draw();
15
16
17    }
```

6. ANS

Interface looks like a class but it is not a class. An interface can have methods and variables just like the class but the methods declared in interface are by default abstract. Also, the variables declared in an interface are public, static & final by default.

```
 9     * @author linda
10     */
      public interface Vehicle {
          public void model();
          public void enginetype();
14
15    }
```

```
7  /**
8   *
9   * @author linda
10  */
11 public class Car implements Vehicle{
12 public void model(){
13     System.out.println("Model of car");
14 }
15
16 public void enginetype(){
17     System.out.println("enginetype of car");
18 }
19 public int ccrating(){
20     System.out.println("CC of car");
21     return 0;
22 }
23 public static void main (String args[]){
24 Car c= new Car();
25 c.model();
26 c.enginetype();
27 c.ccrating();
28 }
29 }
30
```

Car

Output - AssignmentAndroid (run)

```
run:
Model of car
enginetype of car
CC of car
BUILD SUCCESSFUL (total time: 0 seconds)
```

7. ANS

Maximum and minimum value of an array

```
 * @author linda
 */
public class MaxMin {
     public static void main(String[] args)
    {
        int a[] = new int[] { 25, 30, 15, 67, 78, 80, 13, 15, 6, 33 };
        int min = a[0];
        int max = a[0];

        for (int i = 1; i < a.length; i++)
        {
            if (a[i] > max)
            {
                max = a[i];
            }
            if (a[i] < min)
            {
                min = a[i];
            }
        }
        System.out.println("Largest Number in a given array is : " + max);
        System.out.println("Smallest Number in a given array is : " + min);
    }
}
```

MaxMin > main > a >

Output - AssignmentAndroid (run)

```
run:
Largest Number in a given array is : 80
Smallest Number in a given array is : 6
BUILD SUCCESSFUL (total time: 0 seconds)
```

8. ANS

Java program to reverse an array of integer value

```
2  import java.util.Scanner;
3  /**
4  * @author linda
5   */
6  public class ReserveArray {
7      public static void main(String[] args) {
8         int size, i, j, temp;
9         int arr[] = new int[50];
10        Scanner scan = new Scanner(System.in);
11        System.out.print("Enter Array Size : ");
12        size = scan.nextInt();
13        System.out.print("Enter Array Elements : ");
14        for(i=0; i<size; i++)
15        {
16            arr[i] = scan.nextInt();
17        }
18        j = i - 1;
19        while(i<j)
20        {
21            temp = arr[i];
22            arr[i] = arr[j];
23            arr[j] = temp;
24            i++;
25           j--;
26        }
27        System.out.print("Now the Reverse of Array is : \n");
28        for(i=0; i<size; i++)
29        {
30        System.out.print(arr[i]+ "  ");
31        }
32    }
33      }
34
```

ReserveArray > main > for (i = 0; i < size; i++) >

Output - AssignmentAndroid (run)

```
run:
Enter Array Size : 5
Enter Array Elements : 1
2
3
4
5
Now the Reverse of Array is :
1  2  3  4  5  BUILD SUCCESSFUL (total time: 2 minutes 28 seconds)
```

9. ANS

Java program to find second largest element in an arrary

```
 * @author linda
 */
public class SecondLargestNumber {
    public static void main(String[] args) {

        int arr[] = { 14, 46, 47, 86, 92, 52, 48, 36, 66, 85 };
        int largest = arr[0];
        int secondLargest = arr[0];

        System.out.println("The given array is:" );
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i]+"\t");
        }
        for (int i = 0; i < arr.length; i++) {

            if (arr[i] > largest) {
                secondLargest = largest;
                largest = arr[i];

            } else if (arr[i] > secondLargest) {
                secondLargest = arr[i];
            }
        }
        System.out.println("\n Second largest number is:" + secondLargest);
    }

}
```

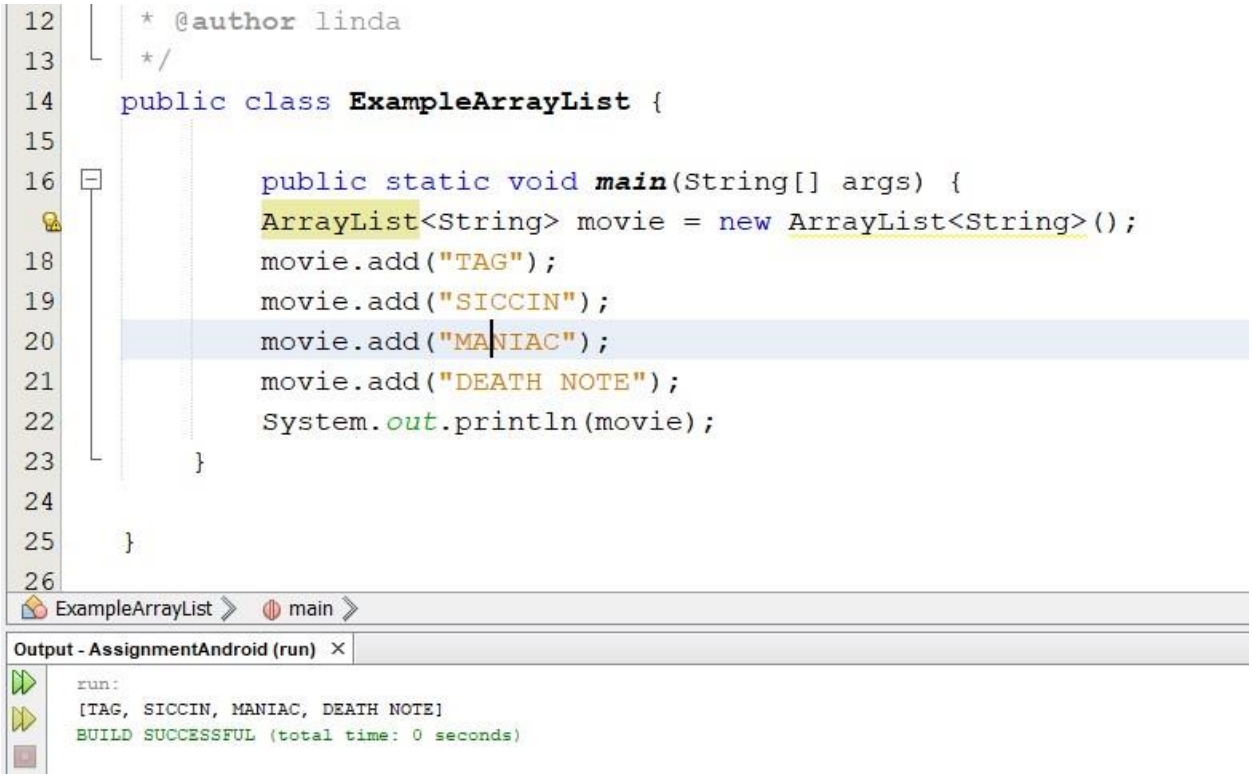
SecondLargestNumber > main > for (int i = 0; i < arr.length; i++) > if (arr[i] > largest) else if (arr[i] > secondLargest)

Output - AssignmentAndroid (run)

```
run:
The given array is:
14      46      47      86      92      52      48      36      66      85
 Second largest number is:86
BUILD SUCCESSFUL (total time: 0 seconds)
```

10. ANS

Array List is a resizable array in which the element can be added and removed whenever we want.



```
 * @author linda
 */
public class ExampleArrayList {

        public static void main(String[] args) {
        ArrayList<String> movie = new ArrayList<String>();
        movie.add("TAG");
        movie.add("SICCIN");
        movie.add("MANIAC");
        movie.add("DEATH NOTE");
        System.out.println(movie);
    }

}
```

ExampleArrayList > main

Output - AssignmentAndroid (run)

```
run:
[TAG, SICCIN, MANIAC, DEATH NOTE]
BUILD SUCCESSFUL (total time: 0 seconds)
```