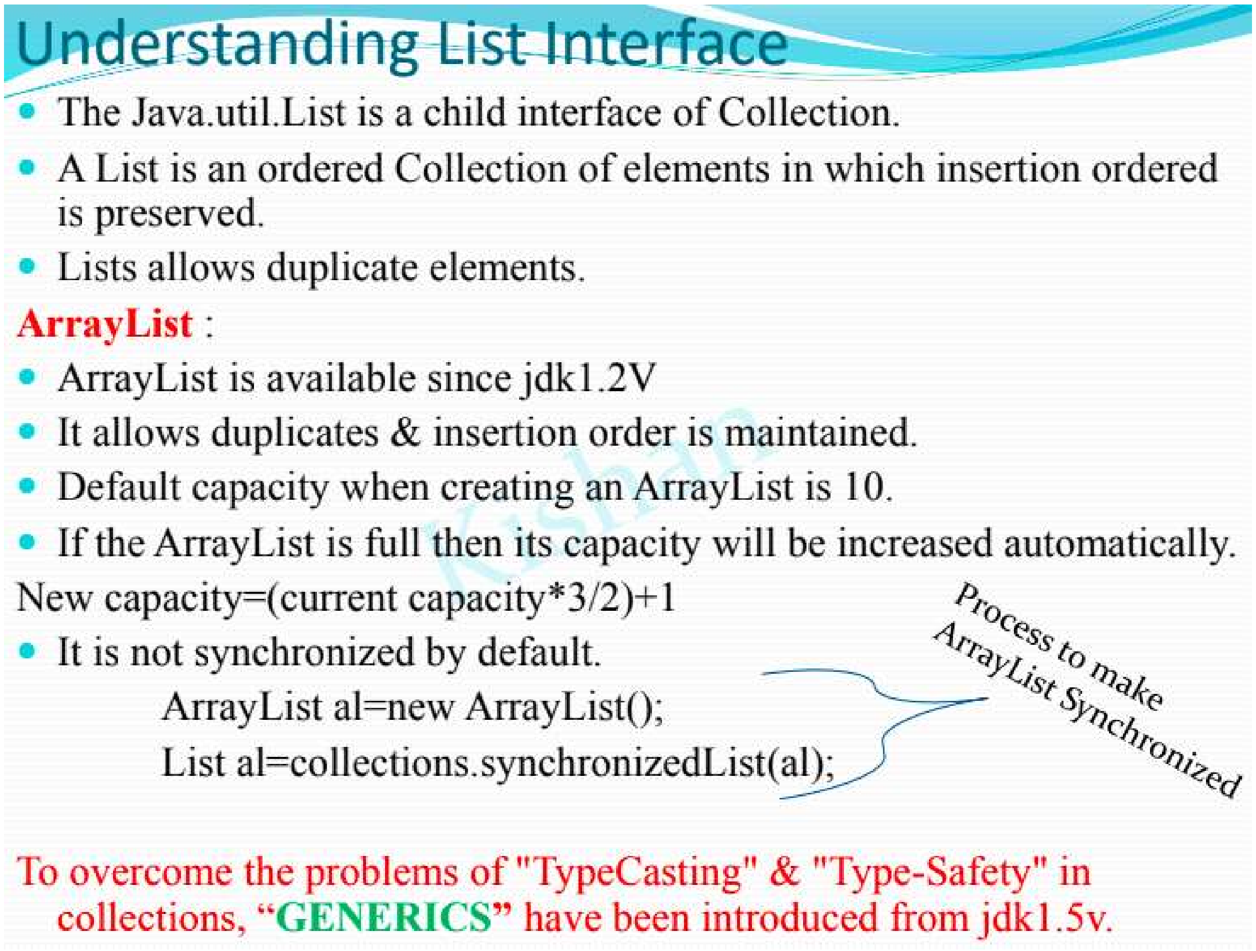


Understanding List Interface

- The Java.util.List is a child interface of Collection.
- A List is an ordered Collection of elements in which insertion ordered is preserved.
- Lists allows duplicate elements.

ArrayList :

- ArrayList is available since jdk1.2V
- It allows duplicates & insertion order is maintained.
- Default capacity when creating an ArrayList is 10.
- If the ArrayList is full then its capacity will be increased automatically.

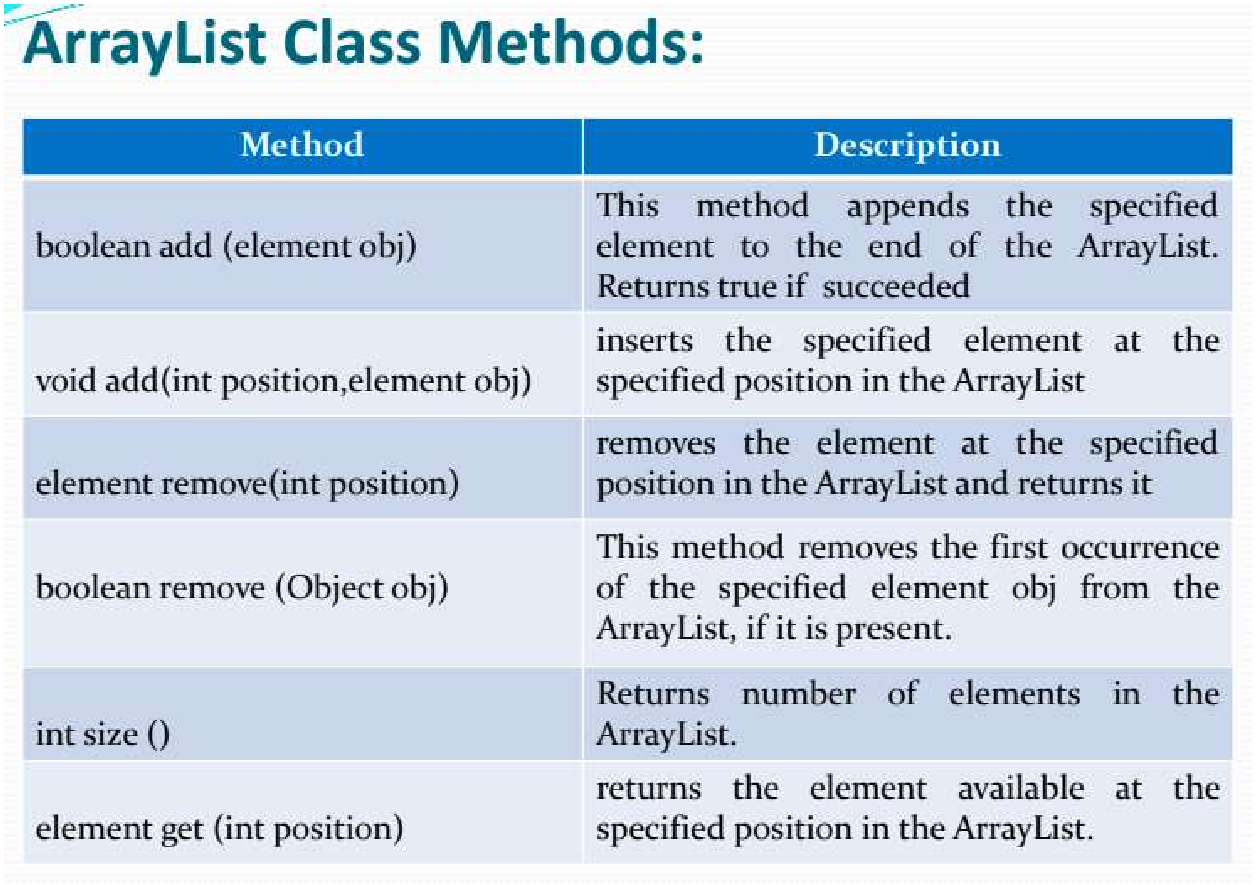
New capacity=(current capacity*3/2)+1

- It is not synchronized by default.

```
ArrayList al=new ArrayList();
List al=collections.synchronizedList(al);
```

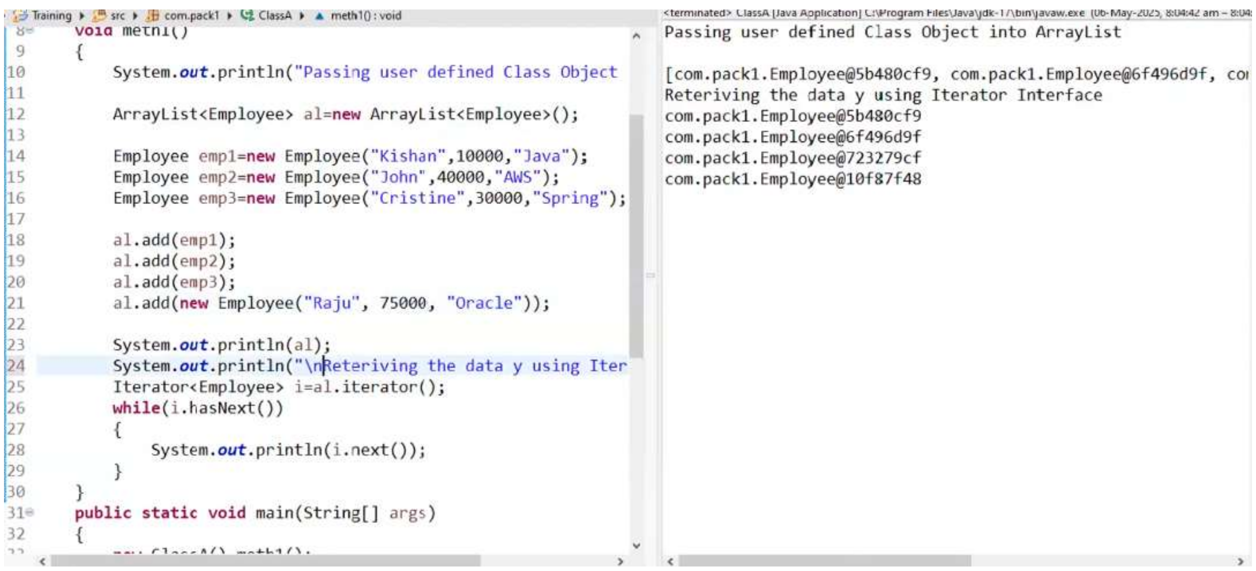
Process to make ArrayList Synchronized

To overcome the problems of "TypeCasting" & "Type-Safety" in collections, "**GENERICS**" have been introduced from jdk1.5v.



ArrayList Class Methods:

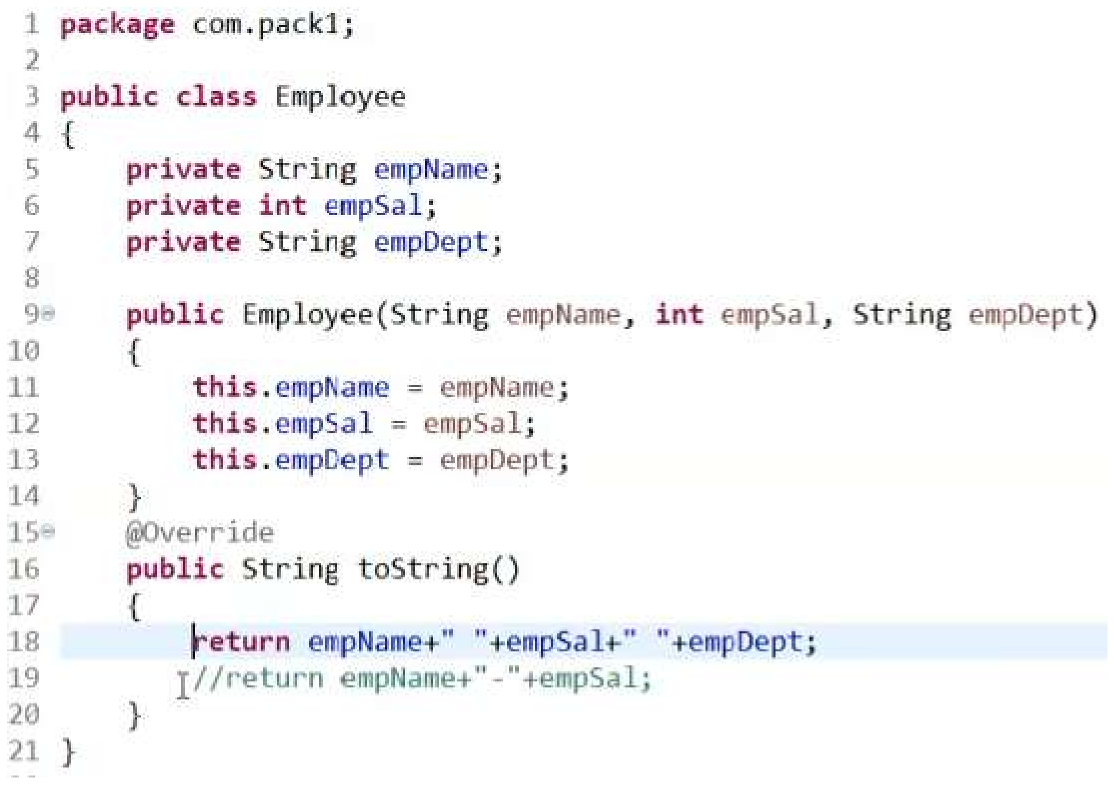
Method	Description
boolean add (element obj)	This method appends the specified element to the end of the ArrayList. Returns true if succeeded
void add(int position,element obj)	inserts the specified element at the specified position in the ArrayList
element remove(int position)	removes the element at the specified position in the ArrayList and returns it
boolean remove (Object obj)	This method removes the first occurrence of the specified element obj from the ArrayList, if it is present.
int size ()	Returns number of elements in the ArrayList.
element get (int position)	returns the element available at the specified position in the ArrayList.



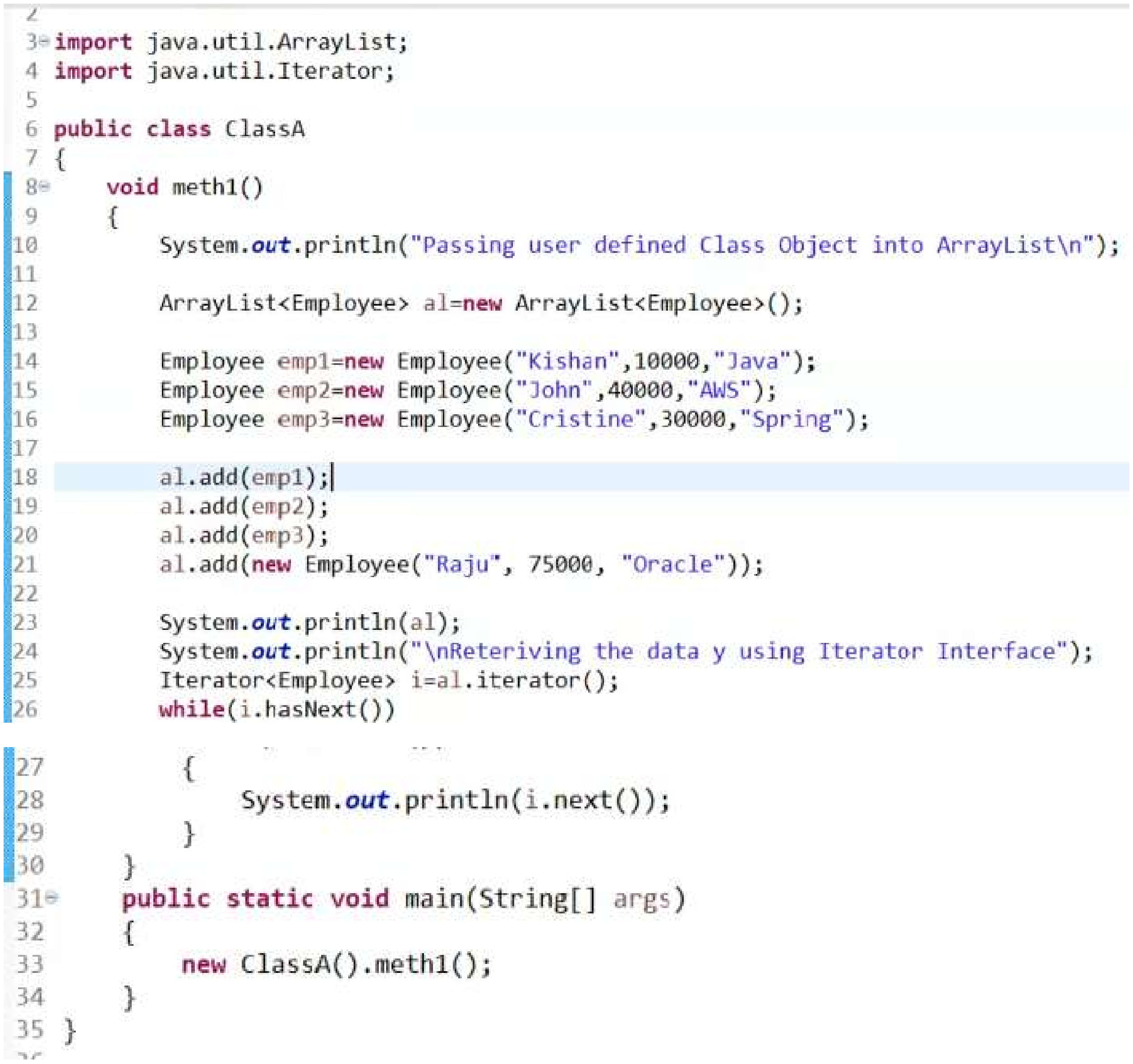
```
    void meth1()
    {
        System.out.println("Passing user defined Class Object

        ArrayList<Employee> al=new ArrayList<Employee>();

        Employee emp1=new Employee("Kishan",10000,"Java");
        Employee emp2=new Employee("John",40000,"AWS");
        Employee emp3=new Employee("Cristine",30000,"Spring");

        al.add(emp1);
        al.add(emp2);
        al.add(emp3);
        al.add(new Employee("Raju", 75000, "Oracle"));

        System.out.println(al);
        System.out.println("\nReteriving the data y using Iter
        Iterator<Employee> i=al.iterator();
        while(i.hasNext())
        {
            System.out.println(i.next());
        }
    }
    public static void main(String[] args)
    {
```

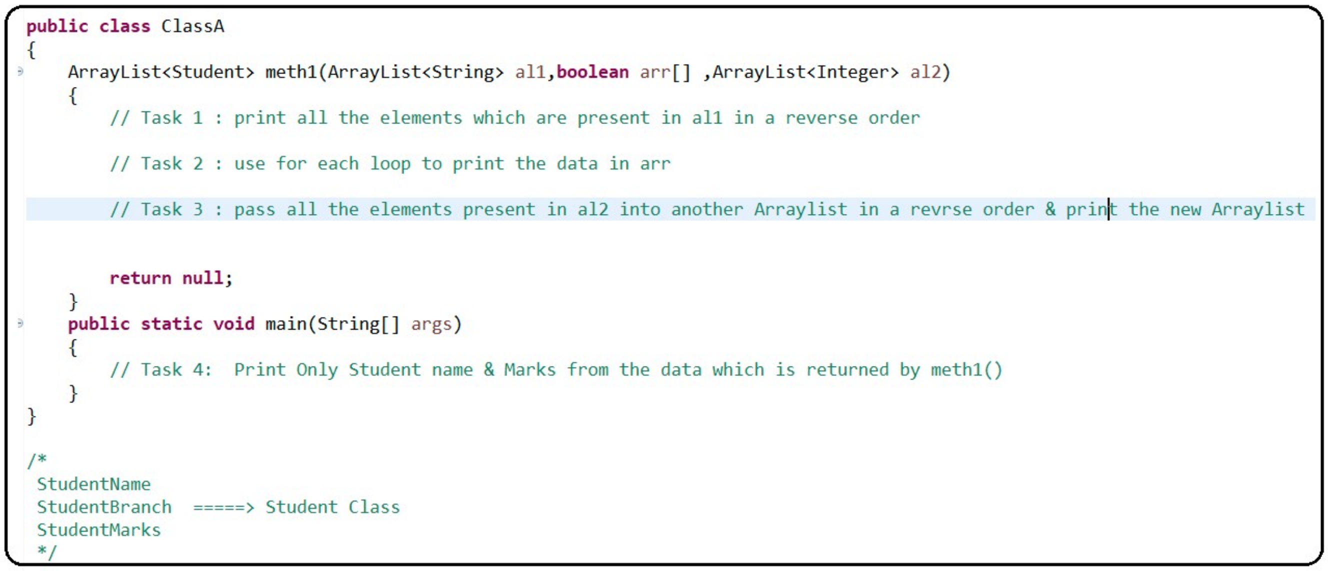
```
<terminated> ClassA [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (06-May-2025, 8:04:42 am – 8:04
Passing user defined Class Object into ArrayList

[com.pack1.Employee@5b480cf9, com.pack1.Employee@6f496d9f, co
Reteriving the data y using Iterator Interface
com.pack1.Employee@5b480cf9
com.pack1.Employee@6f496d9f
com.pack1.Employee@723279cf
com.pack1.Employee@10f87f48
```

Passing user defined object in array list and retrieving it back

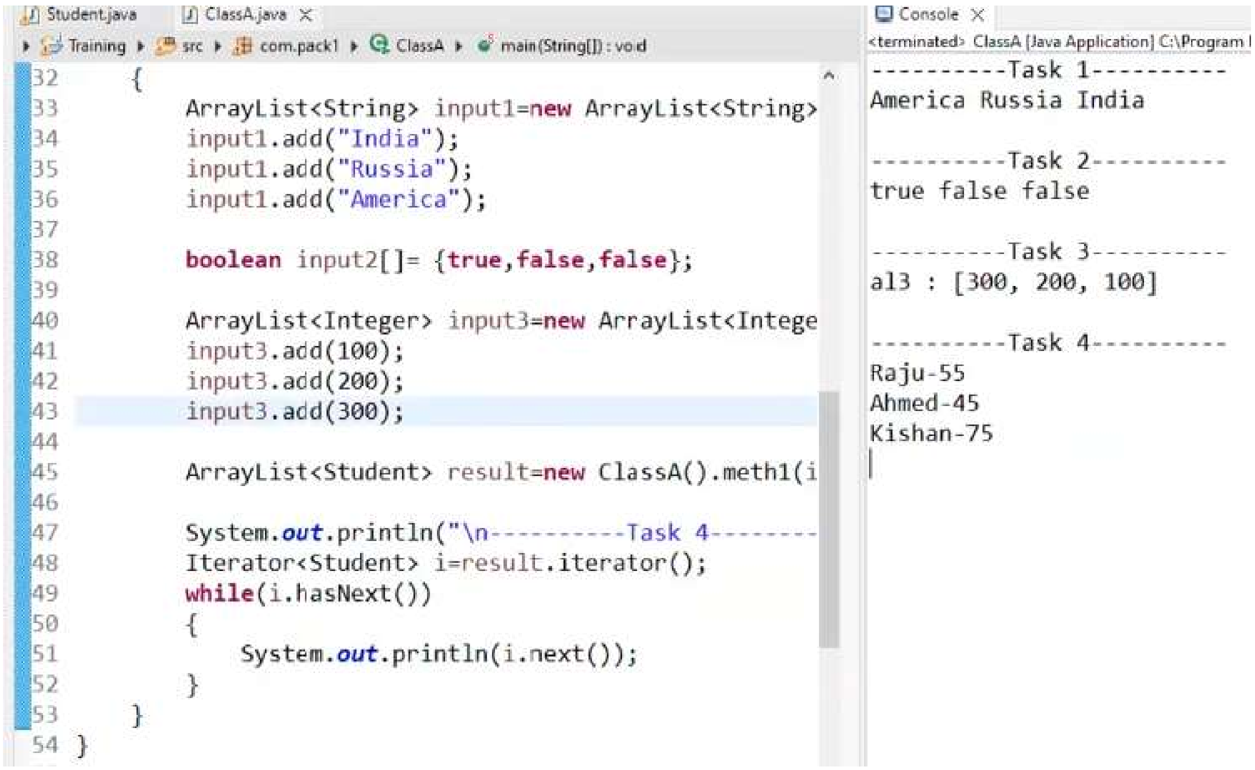


```java
package com.pack1;

public class Employee
{
    private String empName;
    private int empSal;
    private String empDept;

    public Employee(String empName, int empSal, String empDept)
    {
        this.empName = empName;
        this.empSal = empSal;
        this.empDept = empDept;
    }
    @Override
    public String toString()
    {
        return empName+" "+empSal+" "+empDept;
        //return empName+"-"+empSal;
    }
}
```


```java
import java.util.ArrayList;
import java.util.Iterator;

public class ClassA
{
    void meth1()
    {
        System.out.println("Passing user defined Class Object into ArrayList\n");

        ArrayList<Employee> al=new ArrayList<Employee>();

        Employee emp1=new Employee("Kishan",10000,"Java");
        Employee emp2=new Employee("John",40000,"AWS");
        Employee emp3=new Employee("Cristine",30000,"Spring");

        al.add(emp1);
        al.add(emp2);
        al.add(emp3);
        al.add(new Employee("Raju", 75000, "Oracle"));

        System.out.println(al);
        System.out.println("\nReteriving the data y using Iterator Interface");
        Iterator<Employee> i=al.iterator();
        while(i.hasNext())
        {
            System.out.println(i.next());
        }
    }
    public static void main(String[] args)
    {
        new ClassA().meth1();
    }
}
```



```java
public class ClassA
{
    ArrayList<Student> meth1(ArrayList<String> al1,boolean arr[] ,ArrayList<Integer> al2)
    {
        // Task 1 : print all the elements which are present in al1 in a reverse order

        // Task 2 : use for each loop to print the data in arr

        // Task 3 : pass all the elements present in al2 into another Arraylist in a revrse order & print the new Arraylist


        return null;
    }
    public static void main(String[] args)
    {
        // Task 4:  Print Only Student name & Marks from the data which is returned by meth1()
    }
}

/*
 StudentName
 StudentBranch  =====> Student Class
 StudentMarks
 */
```

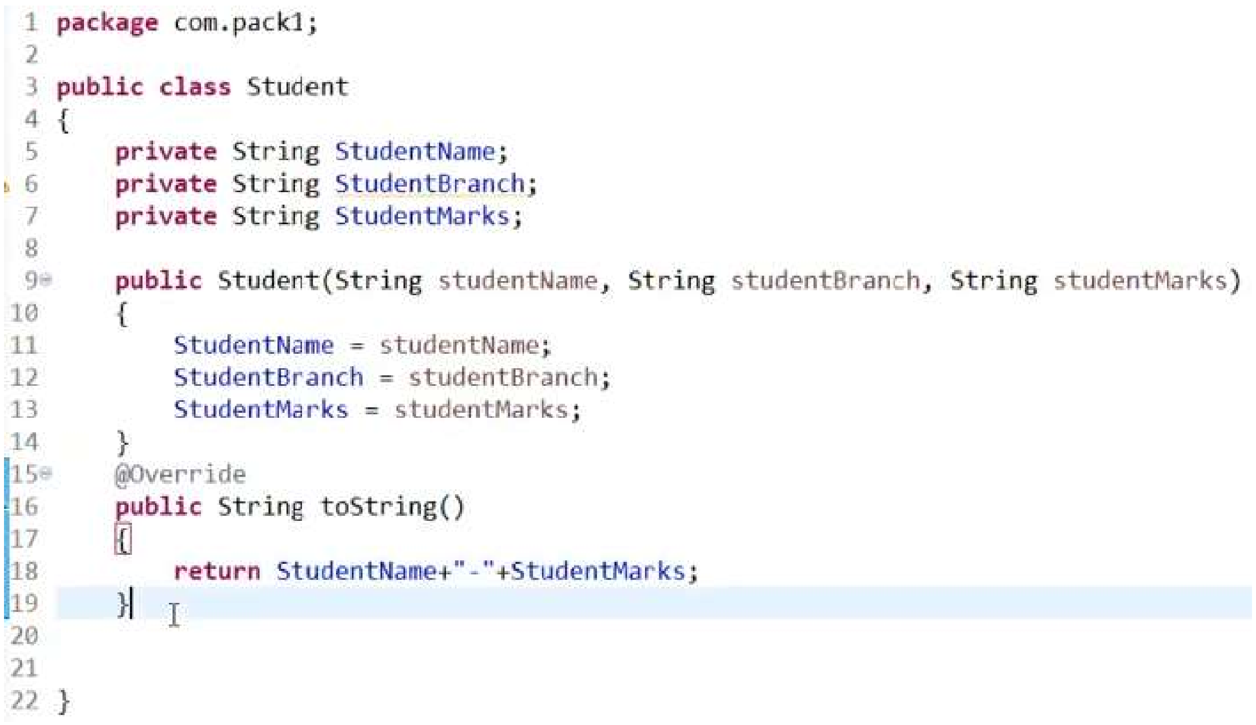


Student.java   ClassA.java
Training ▸ src ▸ com.pack1 ▸ ClassA ▸ main(String[]) : void

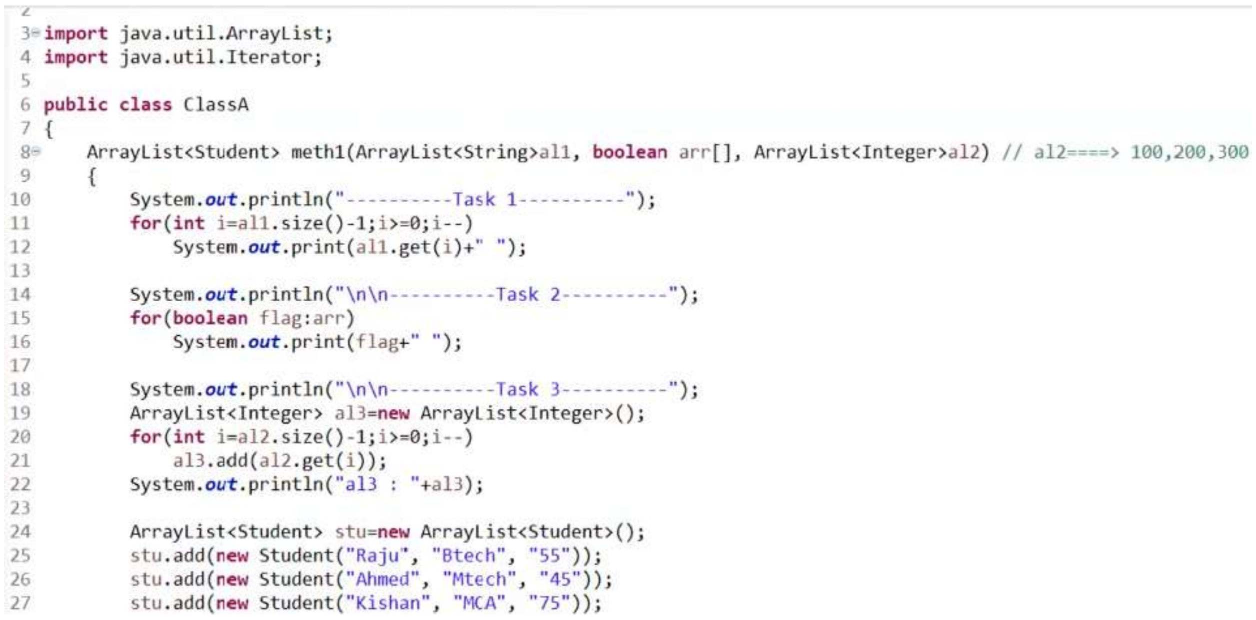
```java
    {
        ArrayList<String> input1=new ArrayList<String>
        input1.add("India");
        input1.add("Russia");
        input1.add("America");

        boolean input2[]= {true,false,false};

        ArrayList<Integer> input3=new ArrayList<Intege
        input3.add(100);
        input3.add(200);
        input3.add(300);

        ArrayList<Student> result=new ClassA().meth1(i

        System.out.println("\n----------Task 4--------
        Iterator<Student> i=result.iterator();
        while(i.hasNext())
        {
            System.out.println(i.next());
        }
    }
}
```

Console
<terminated> ClassA [Java Application] C:\Program

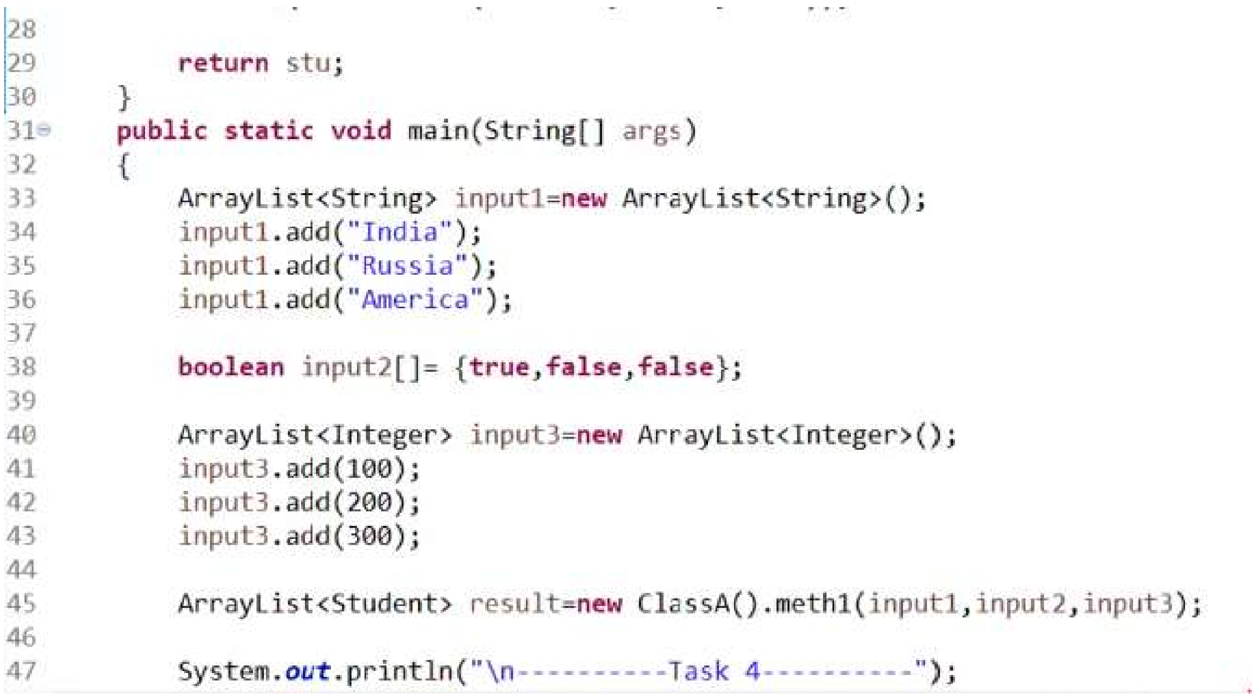
```
----------Task 1----------
America Russia India

----------Task 2----------
true false false

----------Task 3----------
al3 : [300, 200, 100]

----------Task 4----------
Raju-55
Ahmed-45
Kishan-75
```



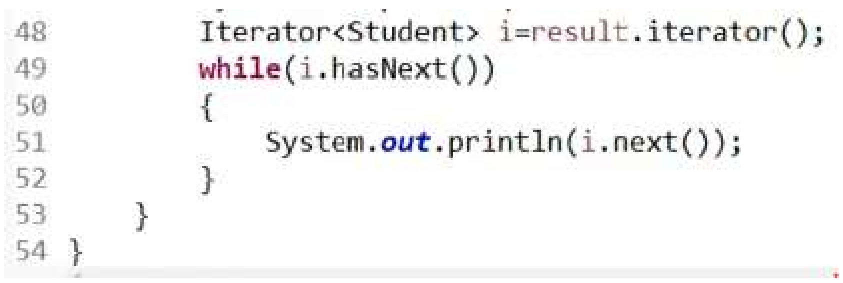
```java
package com.pack1;

public class Student
{
    private String StudentName;
    private String StudentBranch;
    private String StudentMarks;

    public Student(String studentName, String studentBranch, String studentMarks)
    {
        StudentName = studentName;
        StudentBranch = studentBranch;
        StudentMarks = studentMarks;
    }
    @Override
    public String toString()
    {
        return StudentName+"-"+StudentMarks;
    }


}
```



```java
import java.util.ArrayList;
import java.util.Iterator;

public class ClassA
{
    ArrayList<Student> meth1(ArrayList<String>al1, boolean arr[], ArrayList<Integer>al2) // al2====> 100,200,300
    {
        System.out.println("----------Task 1----------");
        for(int i=al1.size()-1;i>=0;i--)
            System.out.print(al1.get(i)+" ");

        System.out.println("\n\n----------Task 2----------");
        for(boolean flag:arr)
            System.out.print(flag+" ");

        System.out.println("\n\n----------Task 3----------");
        ArrayList<Integer> al3=new ArrayList<Integer>();
        for(int i=al2.size()-1;i>=0;i--)
            al3.add(al2.get(i));
        System.out.println("al3 : "+al3);

        ArrayList<Student> stu=new ArrayList<Student>();
        stu.add(new Student("Raju", "Btech", "55"));
        stu.add(new Student("Ahmed", "Mtech", "45"));
        stu.add(new Student("Kishan", "MCA", "75"));
```

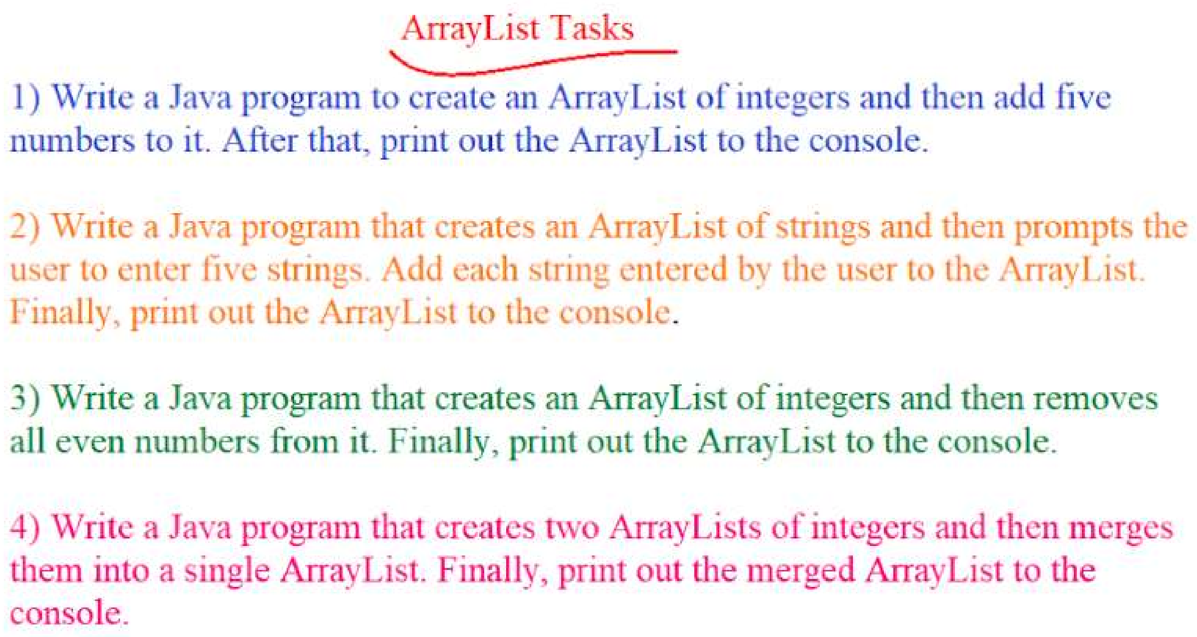


```java

        return stu;
    }
    public static void main(String[] args)
    {
        ArrayList<String> input1=new ArrayList<String>();
        input1.add("India");
        input1.add("Russia");
        input1.add("America");

        boolean input2[]= {true,false,false};

        ArrayList<Integer> input3=new ArrayList<Integer>();
        input3.add(100);
        input3.add(200);
        input3.add(300);

        ArrayList<Student> result=new ClassA().meth1(input1,input2,input3);

        System.out.println("\n----------Task 4----------");
```



```java
        Iterator<Student> i=result.iterator();
        while(i.hasNext())
        {
            System.out.println(i.next());
        }
    }
}
```

Assignment



ArrayList Tasks

1) Write a Java program to create an ArrayList of integers and then add five numbers to it. After that, print out the ArrayList to the console.

2) Write a Java program that creates an ArrayList of strings and then prompts the user to enter five strings. Add each string entered by the user to the ArrayList. Finally, print out the ArrayList to the console.

3) Write a Java program that creates an ArrayList of integers and then removes all even numbers from it. Finally, print out the ArrayList to the console.

4) Write a Java program that creates two ArrayLists of integers and then merges them into a single ArrayList. Finally, print out the merged ArrayList to the console.