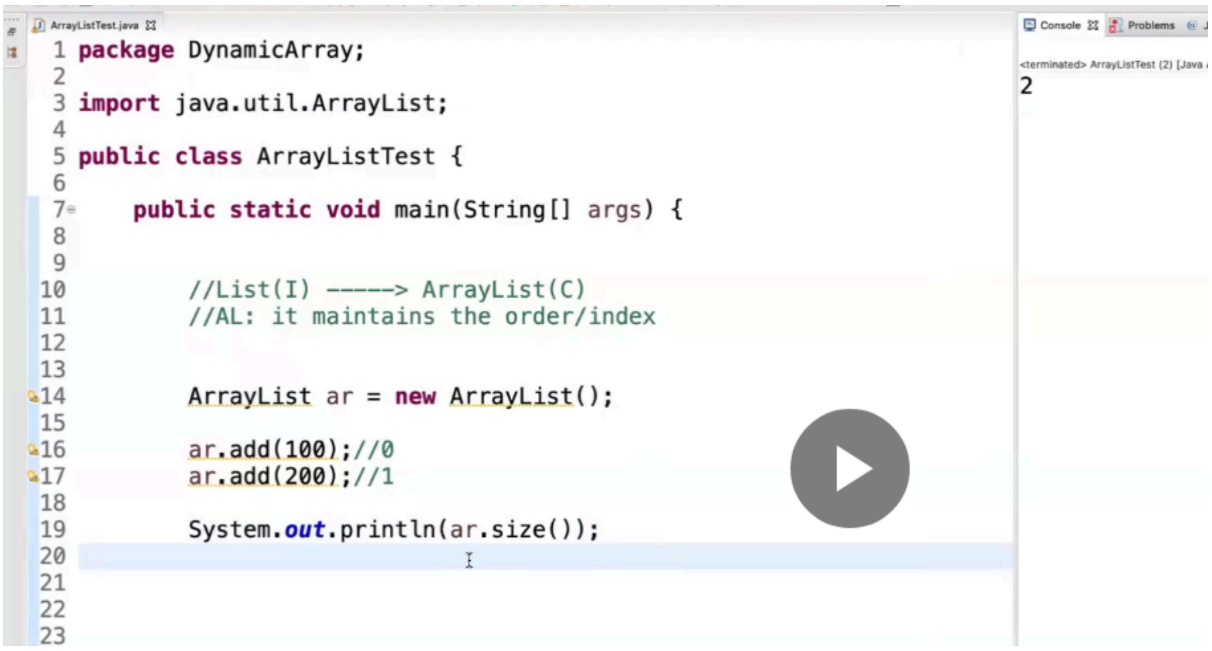
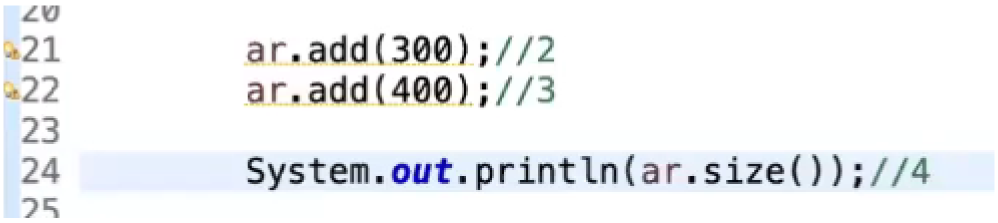


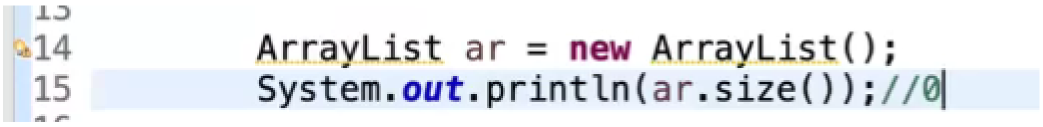
```java
package DynamicArray;

import java.util.ArrayList;

public class ArrayListTest {

    public static void main(String[] args) {


        //List(I) -----> ArrayList(C)
        //AL: it maintains the order/index


        ArrayList ar = new ArrayList();

        ar.add(100);//0
        ar.add(200);//1

        System.out.println(ar.size());
```

```
<terminated> ArrayListTest (2) [Java
2
```



```java
        ar.add(300);//2
        ar.add(400);//3

        System.out.println(ar.size());//4
```

4



```java
        ArrayList ar = new ArrayList();
        System.out.println(ar.size());//0
```

0

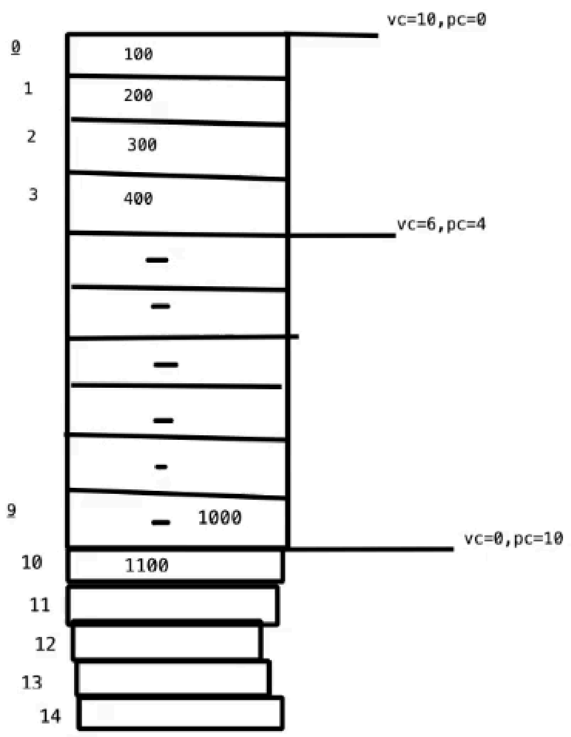
~~8.00~~

10 virtual segments created for array list by default in heap.

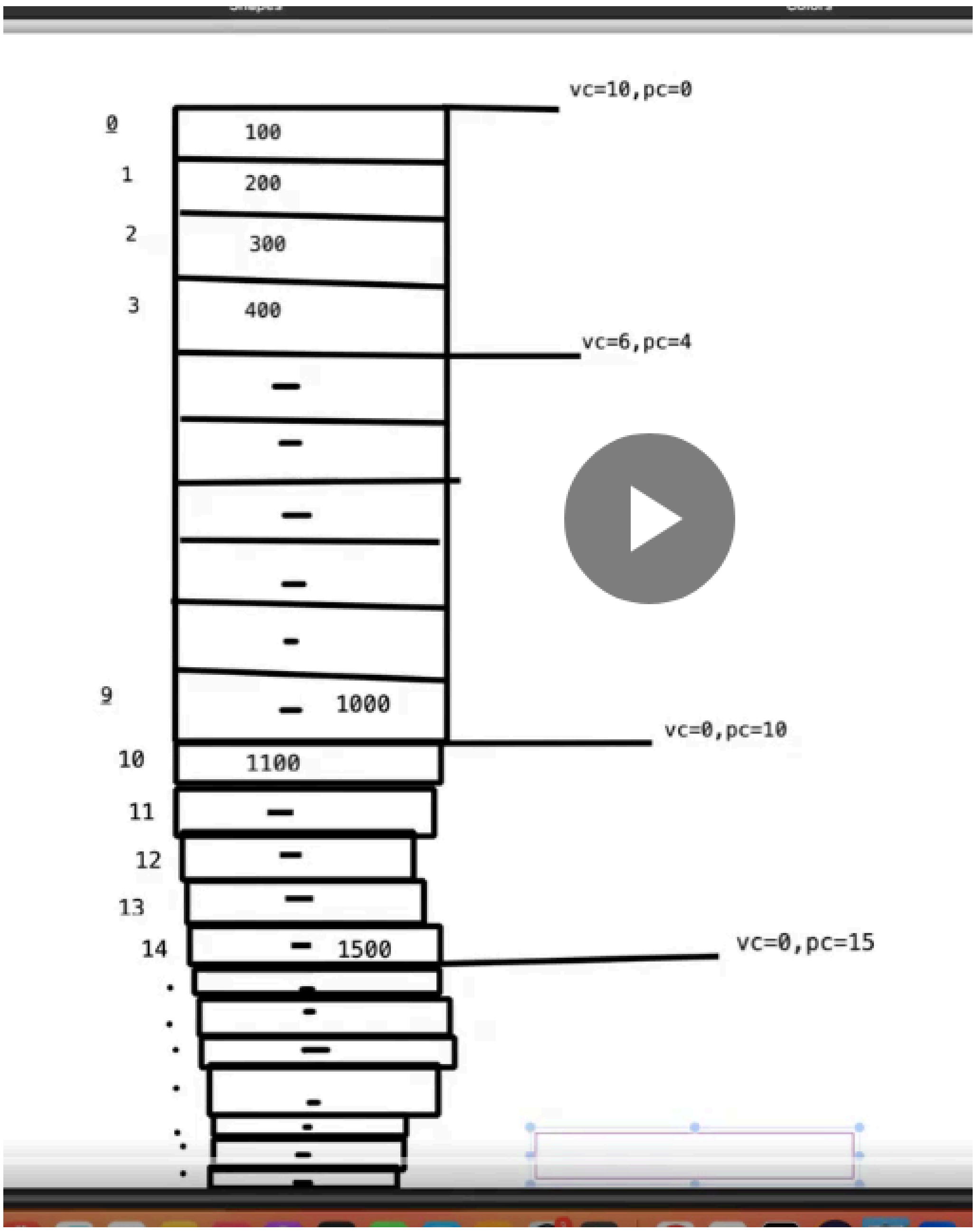
Physical capacity is depending on how many elements present. All methods on array list operate on physical capacity.

No method to check virtual capacity.

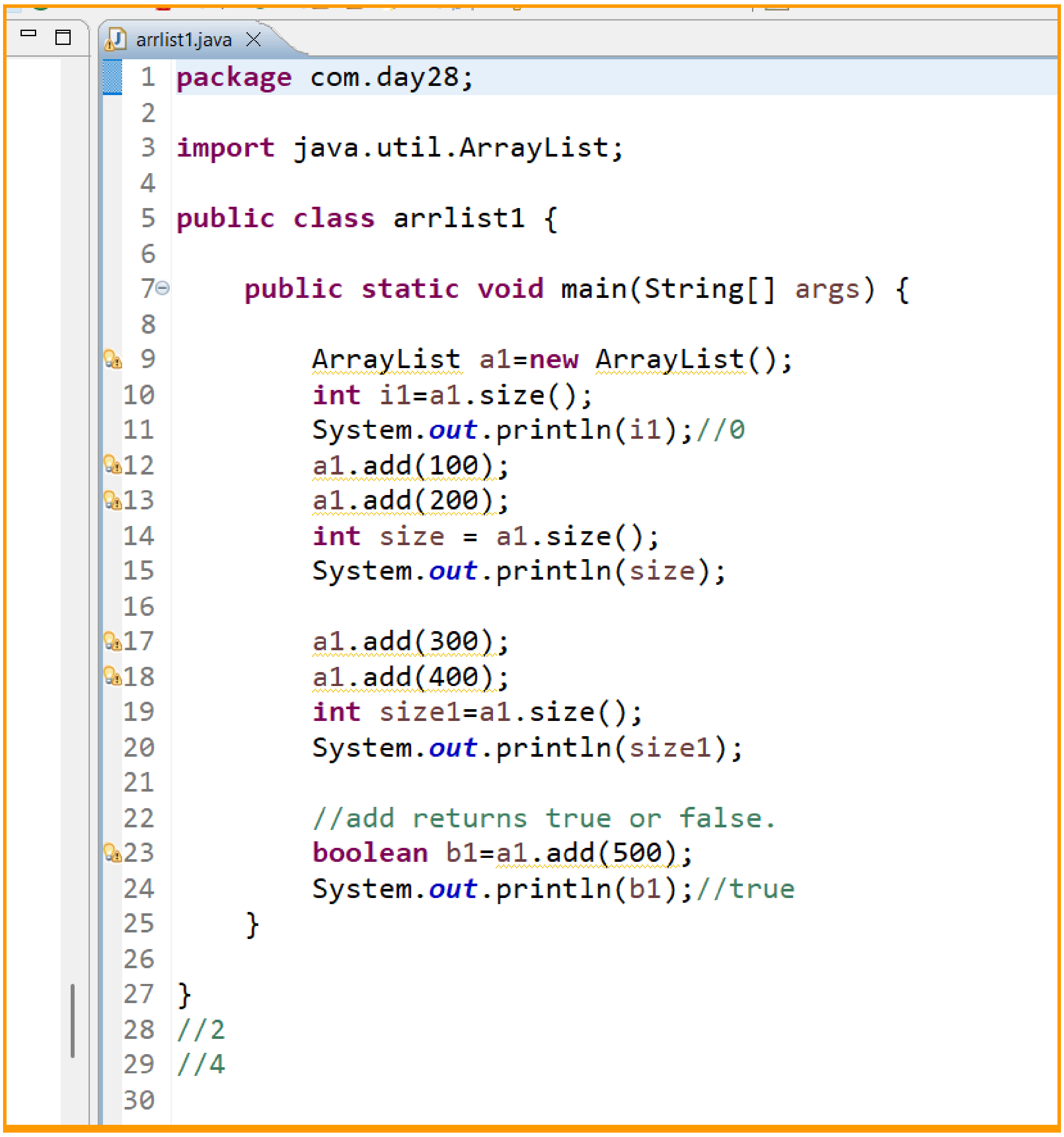
```
ArrayList ar = new ArrayList();
 default vc = 10

   0 to 9: 10 values are filled
   [0] to [9]: full
   [10]:11th value: ar.add(1100)

    LF = pc/2 = 10/2 = 5 (vc)
```



```
[10] to [14]: filled
[15]: 16th value:
LF = pc/2=15/2=7 (vc)
```



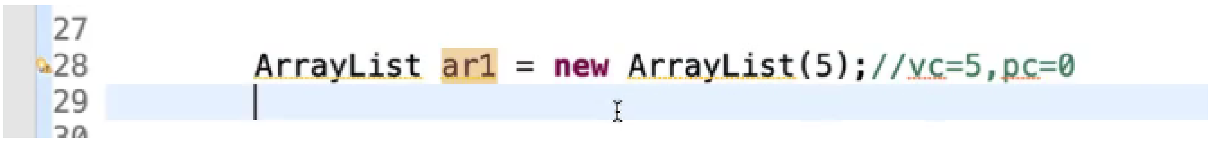
paste arrlist1



arrlist1.java

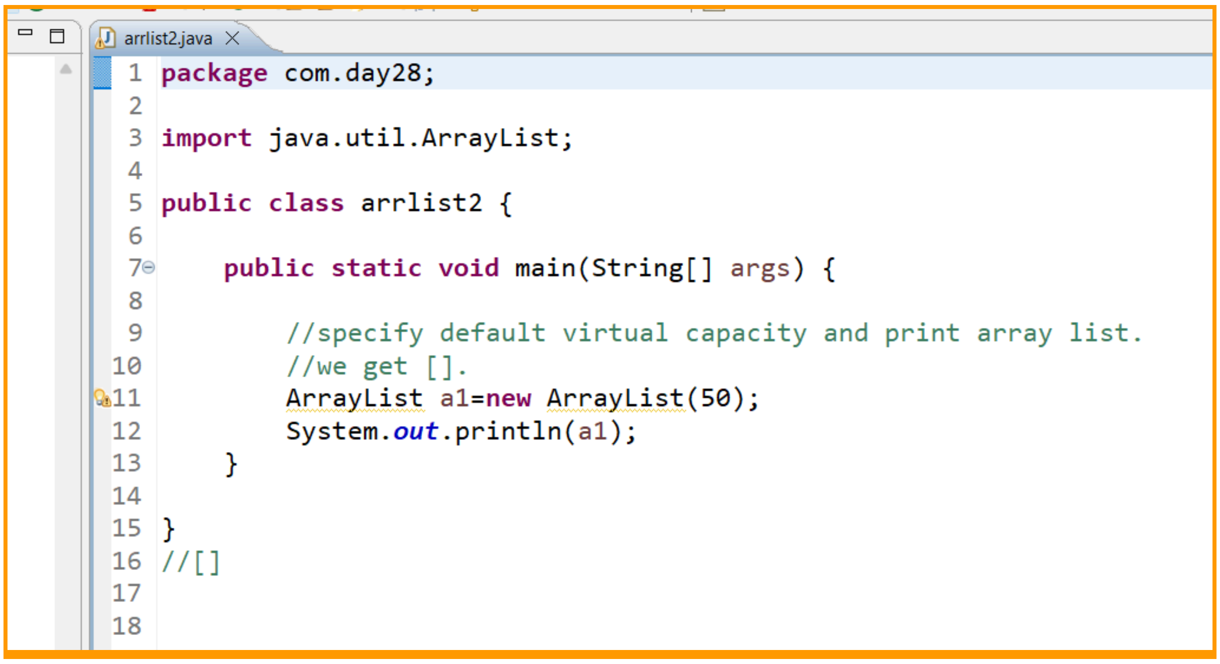
```java
package com.day28;

import java.util.ArrayList;

public class arrlist1 {

	public static void main(String[] args) {

		ArrayList a1=new ArrayList();
		int i1=a1.size();
		System.out.println(i1);//0
		a1.add(100);
		a1.add(200);
		int size = a1.size();
		System.out.println(size);

		a1.add(300);
		a1.add(400);
		int size1=a1.size();
		System.out.println(size1);

		//add returns true or false.
		boolean b1=a1.add(500);
		System.out.println(b1);//true
	}

}
//2
//4
```

We can specify default virtual capacity-



```java
		ArrayList ar1 = new ArrayList(5);//vc=5,pc=0
```

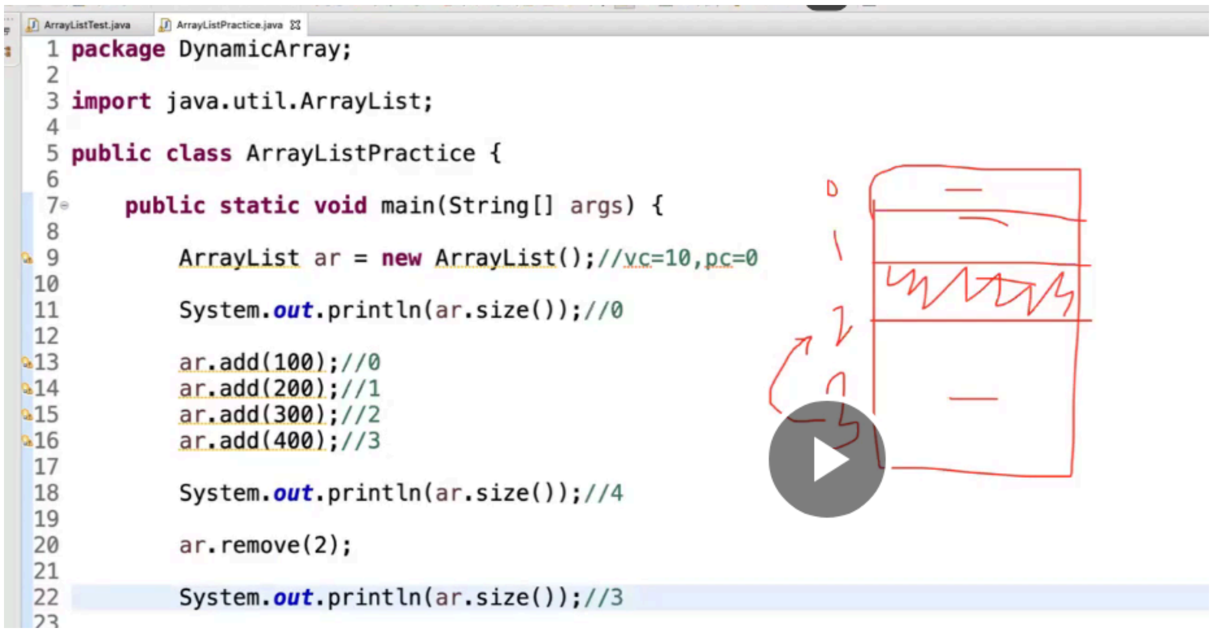
paste arrlist2



```java
package com.day28;

import java.util.ArrayList;

public class arrlist2 {

    public static void main(String[] args) {

        //specify default virtual capacity and print array list.
        //we get [].
        ArrayList a1=new ArrayList(50);
        System.out.println(a1);
    }

}
//[]
```

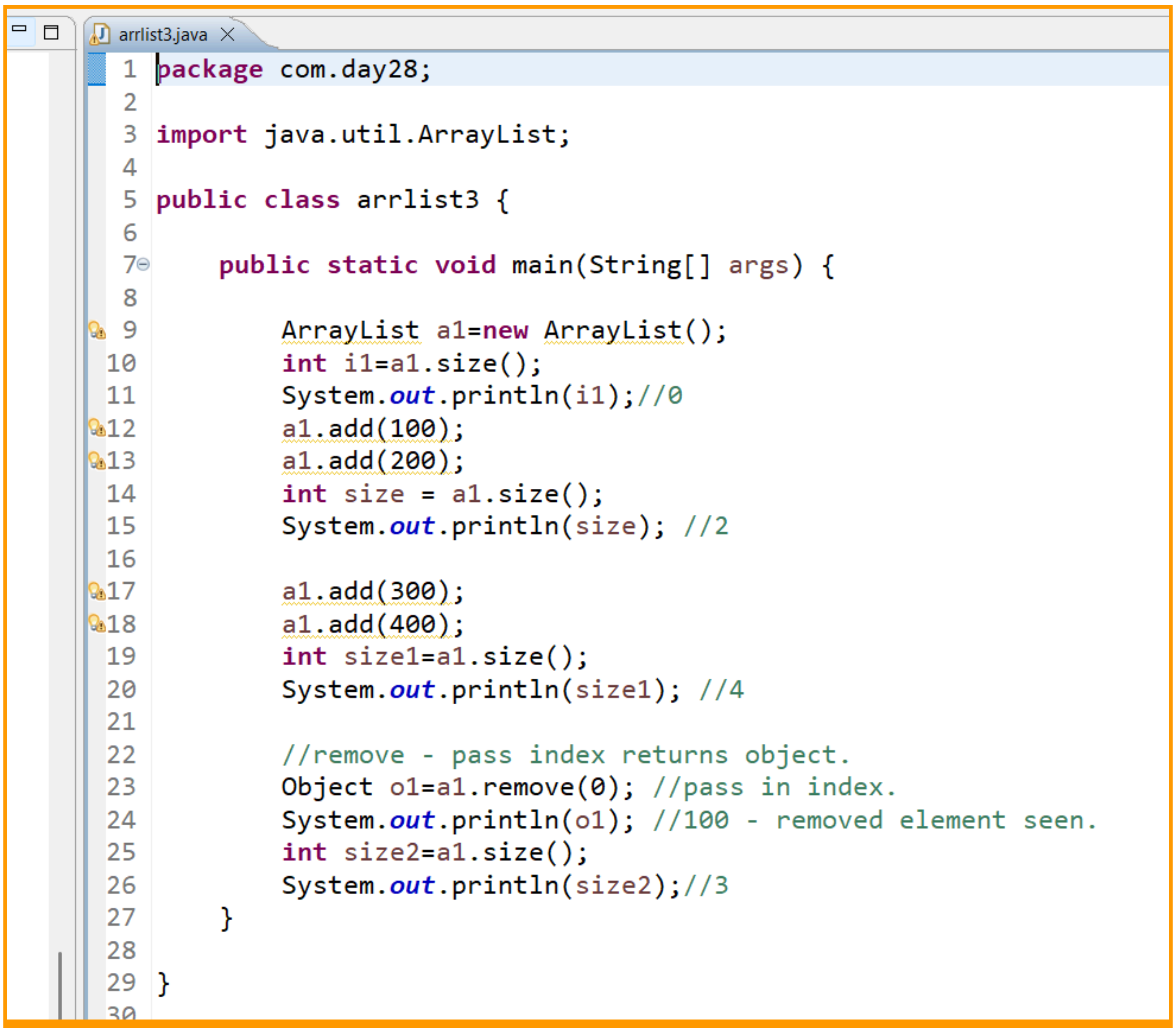
Remove-

Entire index removed and lower indexes shifted up.



```java
package DynamicArray;

import java.util.ArrayList;

public class ArrayListPractice {

    public static void main(String[] args) {

        ArrayList ar = new ArrayList();//vc=10,pc=0

        System.out.println(ar.size());//0

        ar.add(100);//0
        ar.add(200);//1
        ar.add(300);//2
        ar.add(400);//3

        System.out.println(ar.size());//4

        ar.remove(2);

        System.out.println(ar.size());//3
```

paste arrlist3

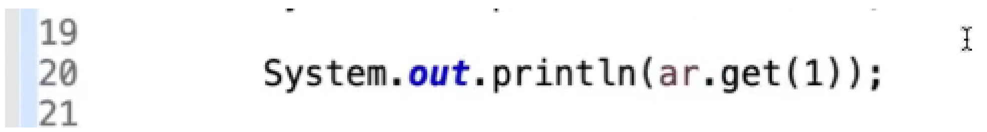


arrlist3.java

```java
package com.day28;

import java.util.ArrayList;

public class arrlist3 {

    public static void main(String[] args) {

        ArrayList a1=new ArrayList();
        int i1=a1.size();
        System.out.println(i1);//0
        a1.add(100);
        a1.add(200);
        int size = a1.size();
        System.out.println(size); //2

        a1.add(300);
        a1.add(400);
        int size1=a1.size();
        System.out.println(size1); //4

        //remove - pass index returns object.
        Object o1=a1.remove(0); //pass in index.
        System.out.println(o1); //100 - removed element seen.
        int size2=a1.size();
        System.out.println(size2);//3
    }

}
```

Get-

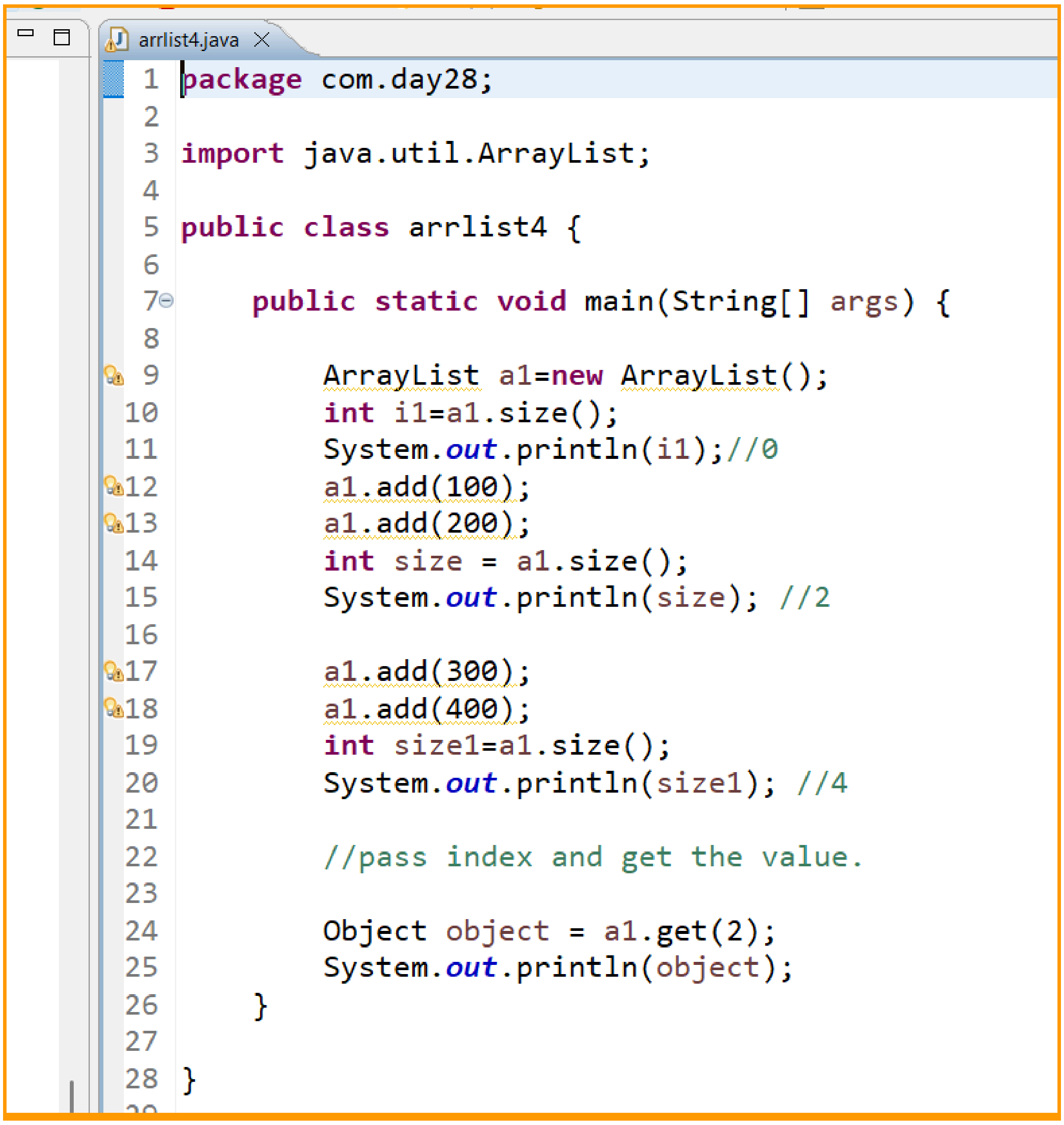
Pass index.



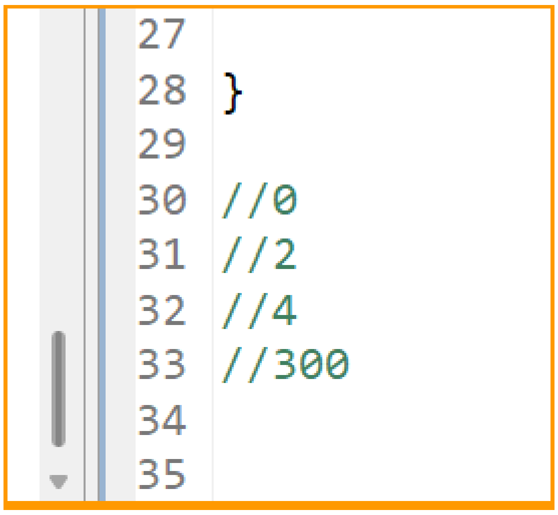
```java
        System.out.println(ar.get(1));
```

200

paste arrlist4

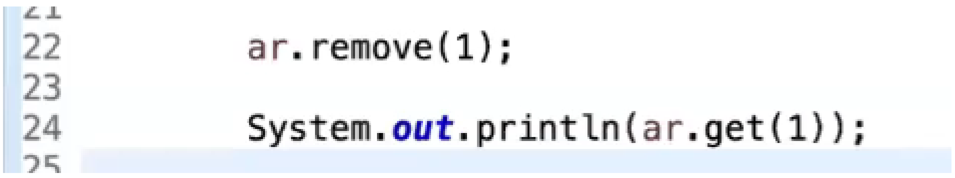

arrlist4.java

```
package com.day28;

import java.util.ArrayList;

public class arrlist4 {

    public static void main(String[] args) {

        ArrayList a1=new ArrayList();
        int i1=a1.size();
        System.out.println(i1);//0
        a1.add(100);
        a1.add(200);
        int size = a1.size();
        System.out.println(size); //2

        a1.add(300);
        a1.add(400);
        int size1=a1.size();
        System.out.println(size1); //4

        //pass index and get the value.

        Object object = a1.get(2);
        System.out.println(object);
    }

}
```

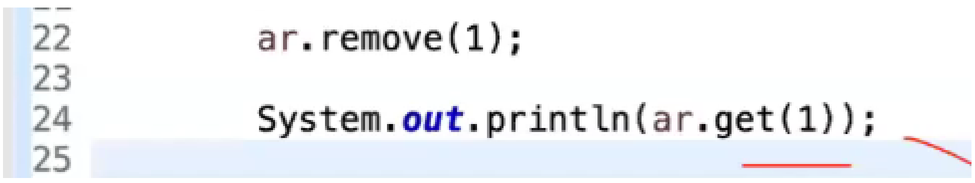


```

}

//0
//2
//4
//300


```

Remove based on index number-



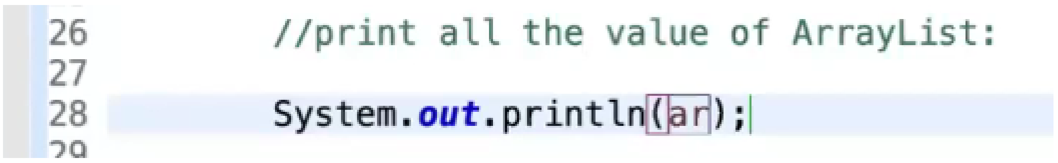
```
        ar.remove(1);

        System.out.println(ar.get(1));
```



```
        ar.remove(1);

        System.out.println(ar.get(1));
```

300

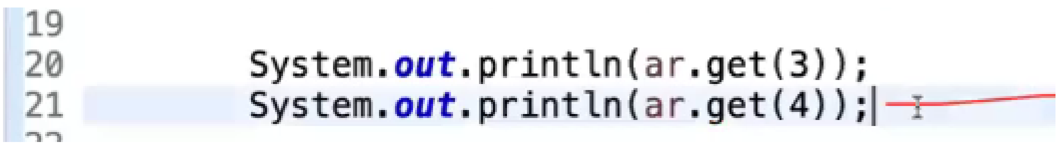
Print all values-



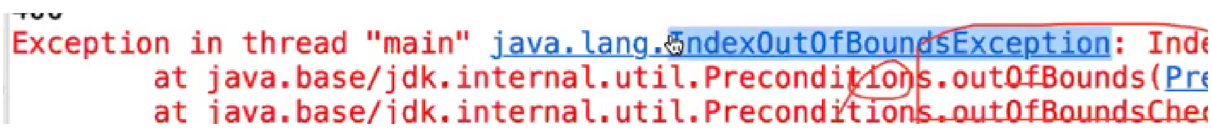
```
        //print all the value of ArrayList:

        System.out.println(ar);
```

[100, 200, 300, 400]

Access out of index-

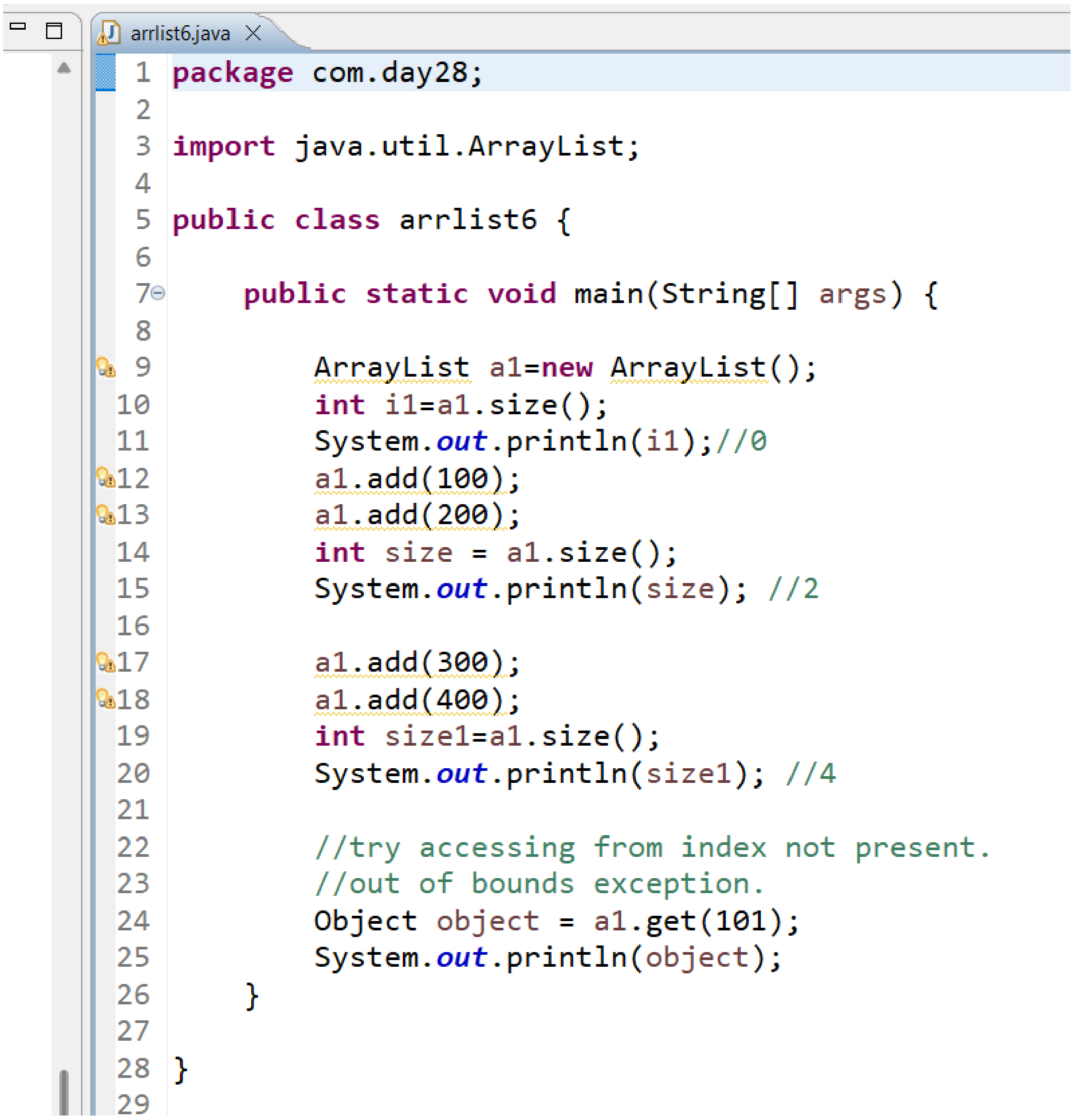


```
        System.out.println(ar.get(3));
        System.out.println(ar.get(4));
```



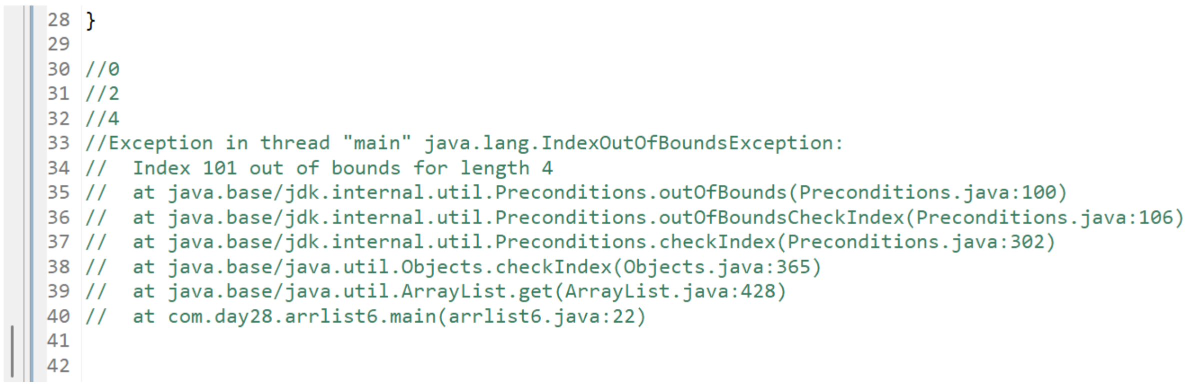
```
Exception in thread "main" java.lang.IndexOutOfBoundsException: Inde
        at java.base/jdk.internal.util.Preconditions.outOfBounds(Pre
        at java.base/jdk.internal.util.Preconditions.outOfBoundsChec
```

paste arrlist6



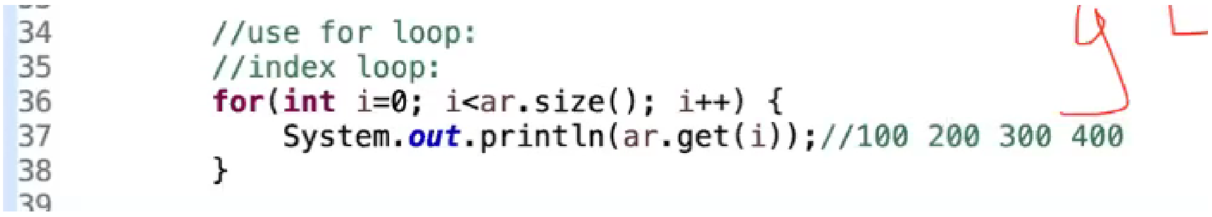
arrlist6.java

```java
package com.day28;

import java.util.ArrayList;

public class arrlist6 {

	public static void main(String[] args) {

		ArrayList a1=new ArrayList();
		int i1=a1.size();
		System.out.println(i1);//0
		a1.add(100);
		a1.add(200);
		int size = a1.size();
		System.out.println(size); //2

		a1.add(300);
		a1.add(400);
		int size1=a1.size();
		System.out.println(size1); //4

		//try accessing from index not present.
		//out of bounds exception.
		Object object = a1.get(101);
		System.out.println(object);
	}

}

```

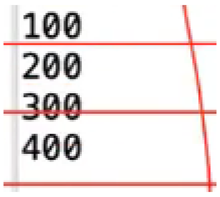


```
}

//0
//2
//4
//Exception in thread "main" java.lang.IndexOutOfBoundsException:
//  Index 101 out of bounds for length 4
//  at java.base/jdk.internal.util.Preconditions.outOfBounds(Preconditions.java:100)
//  at java.base/jdk.internal.util.Preconditions.outOfBoundsCheckIndex(Preconditions.java:106)
//  at java.base/jdk.internal.util.Preconditions.checkIndex(Preconditions.java:302)
//  at java.base/java.util.Objects.checkIndex(Objects.java:365)
//  at java.base/java.util.ArrayList.get(ArrayList.java:428)
//  at com.day28.arrlist6.main(arrlist6.java:22)


```

Print values one by one-

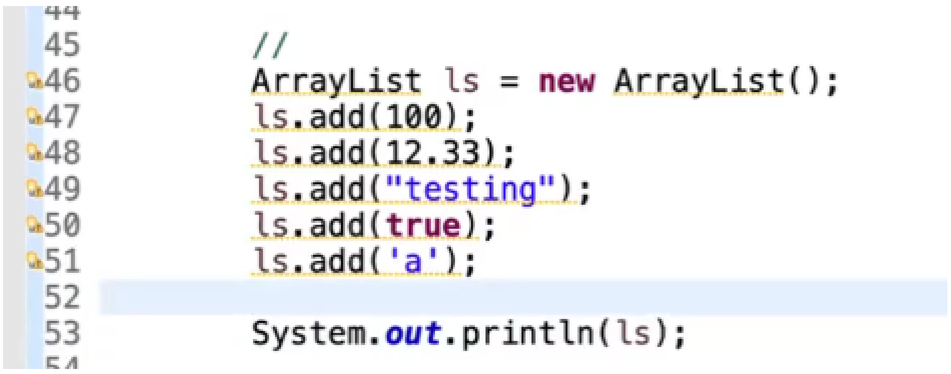


```
        //use for loop:
        //index loop:
        for(int i=0; i<ar.size(); i++) {
            System.out.println(ar.get(i));//100 200 300 400
        }
```

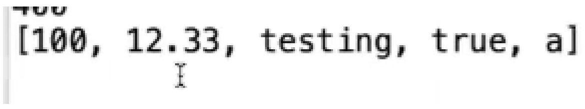


```
100
200
300
400
```

Storing any value-



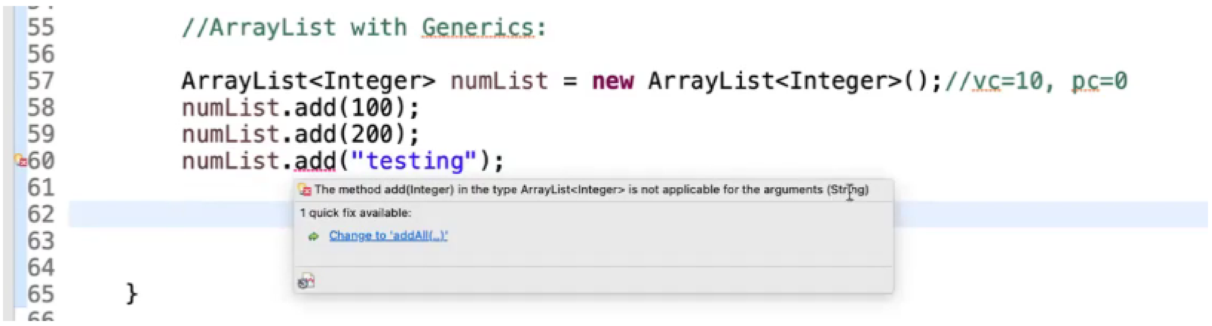
```
        //
        ArrayList ls = new ArrayList();
        ls.add(100);
        ls.add(12.33);
        ls.add("testing");
        ls.add(true);
        ls.add('a');

        System.out.println(ls);
```



```
[100, 12.33, testing, true, a]
```

Generics-

Cant add any other value.



```
        //ArrayList with Generics:

        ArrayList<Integer> numList = new ArrayList<Integer>();//vc=10, pc=0
        numList.add(100);
        numList.add(200);
        numList.add("testing");
```

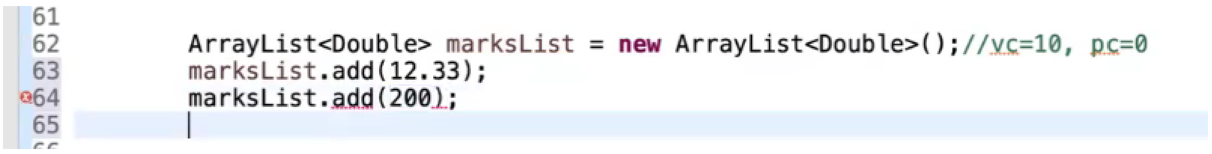
The method add(Integer) in the type ArrayList<Integer> is not applicable for the arguments (String)

1 quick fix available:

Change to 'addAll(..)'

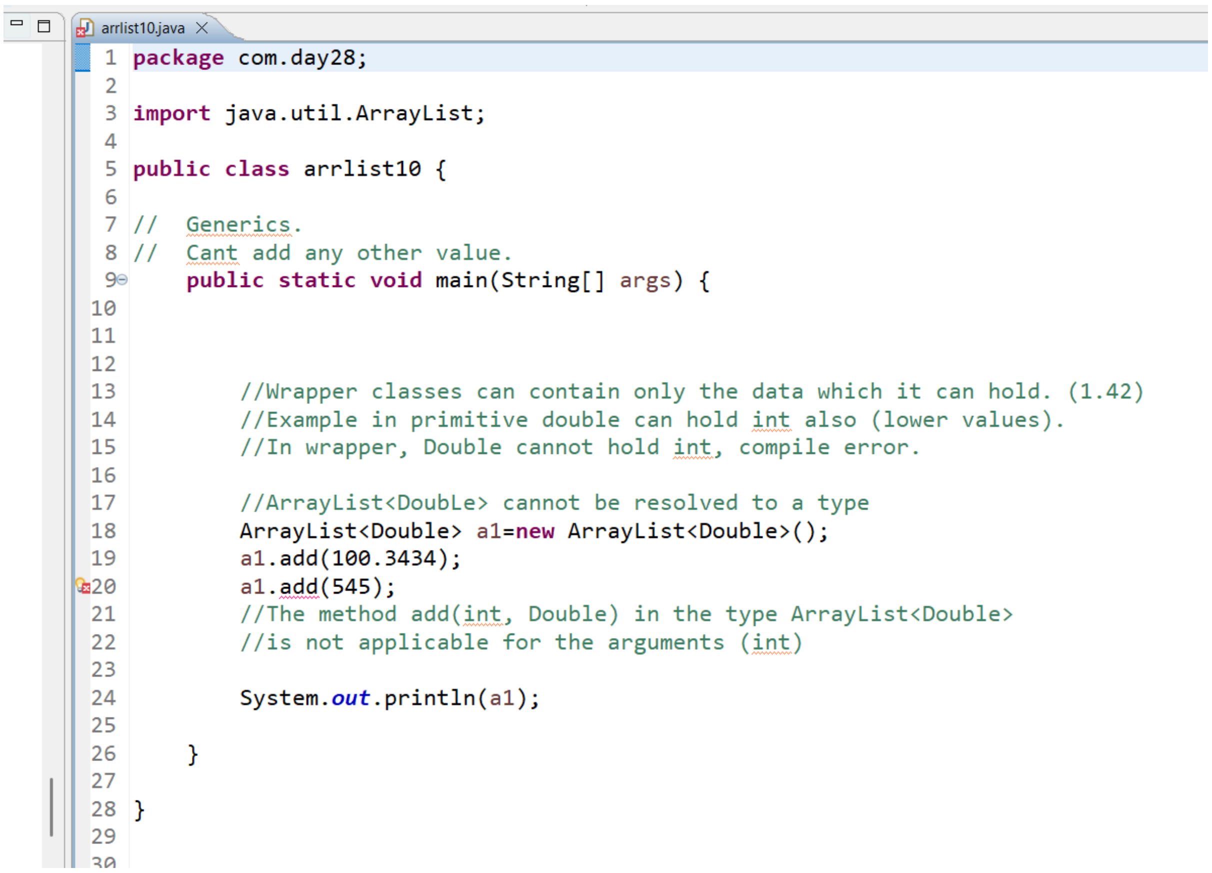
```
    }
```

Double cannot hold integer-



```
        ArrayList<Double> marksList = new ArrayList<Double>();//vc=10, pc=0
        marksList.add(12.33);
        marksList.add(200);

```

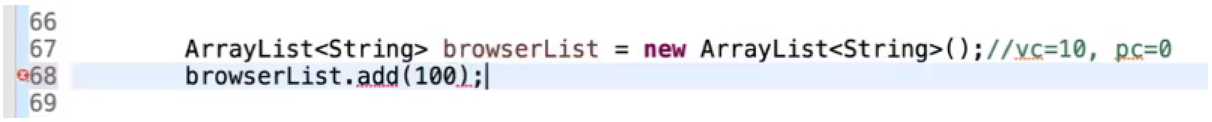
paste arrlist10



arrlist10.java

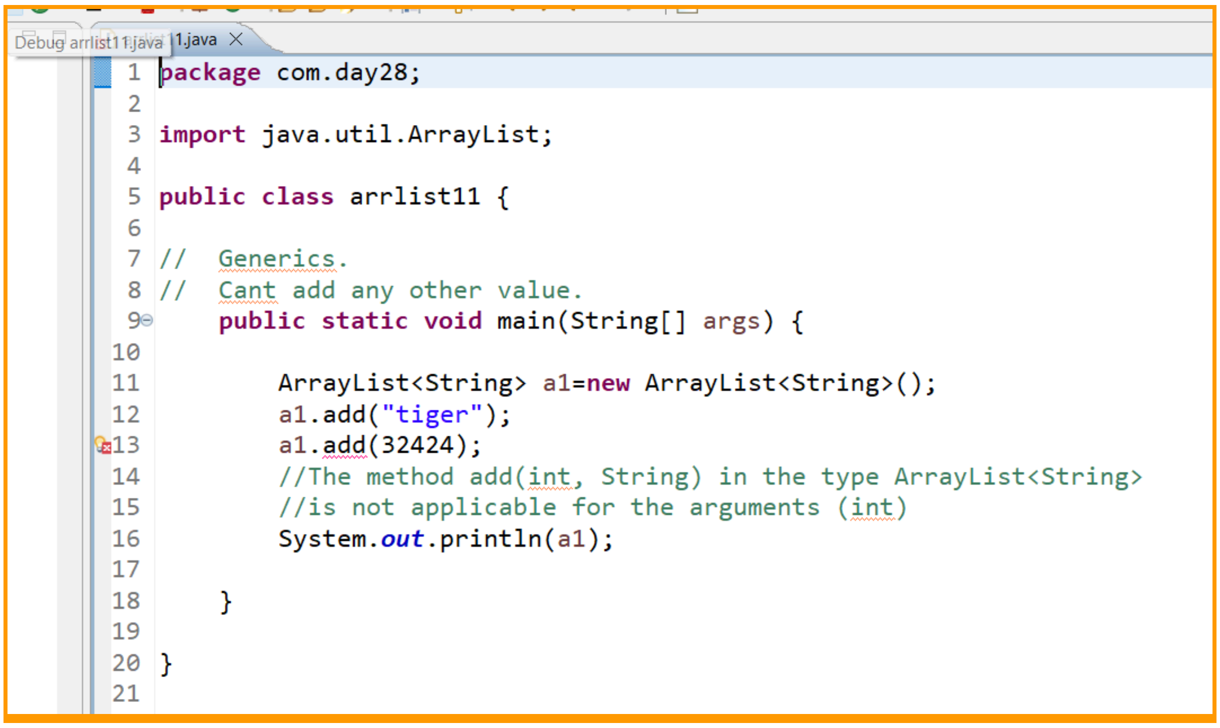
```
package com.day28;

import java.util.ArrayList;

public class arrlist10 {

//   Generics.
//   Cant add any other value.
    public static void main(String[] args) {



        //Wrapper classes can contain only the data which it can hold. (1.42)
        //Example in primitive double can hold int also (lower values).
        //In wrapper, Double cannot hold int, compile error.

        //ArrayList<DoubLe> cannot be resolved to a type
        ArrayList<Double> a1=new ArrayList<Double>();
        a1.add(100.3434);
        a1.add(545);
        //The method add(int, Double) in the type ArrayList<Double>
        //is not applicable for the arguments (int)

        System.out.println(a1);

    }

}
```

String-

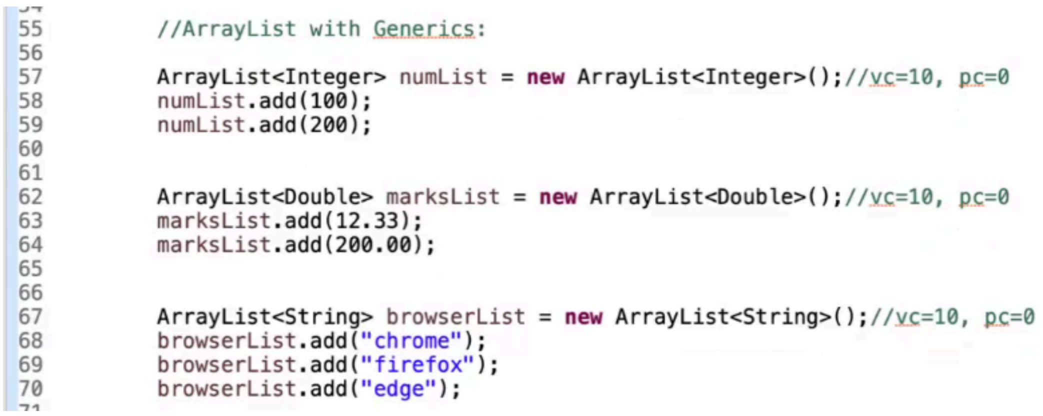


```
        ArrayList<String> browserList = new ArrayList<String>();//vc=10, pc=0
        browserList.add(100);
```

paste arrlist11

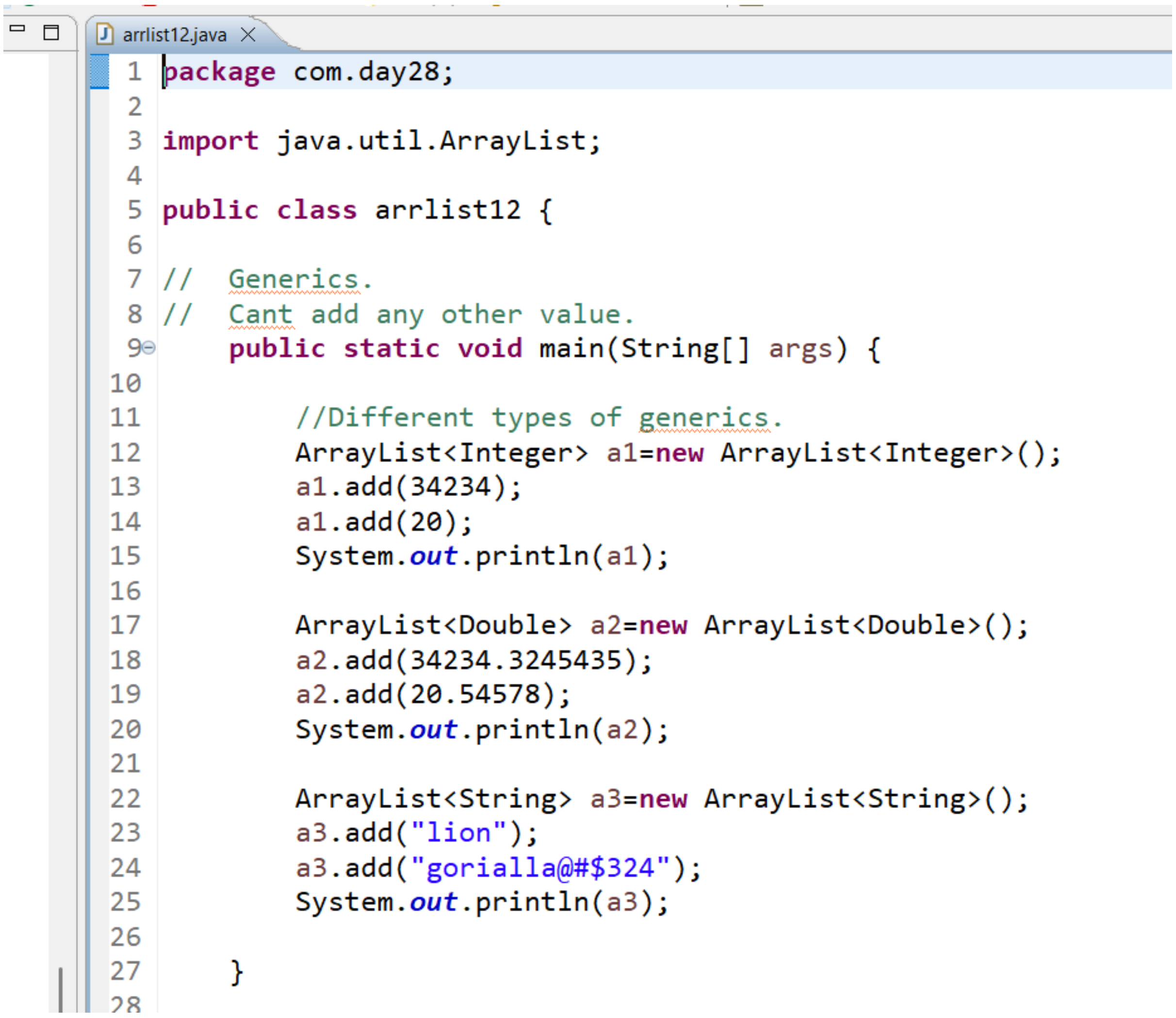


```java
package com.day28;

import java.util.ArrayList;

public class arrlist11 {

//  Generics.
//  Cant add any other value.
    public static void main(String[] args) {

        ArrayList<String> a1=new ArrayList<String>();
        a1.add("tiger");
        a1.add(32424);
        //The method add(int, String) in the type ArrayList<String>
        //is not applicable for the arguments (int)
        System.out.println(a1);

    }

}
```



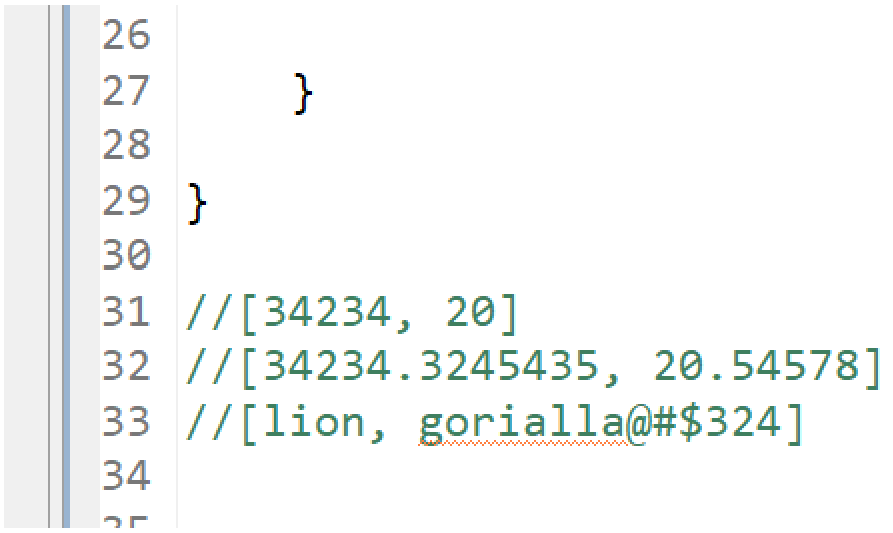
```java
        //ArrayList with Generics:

        ArrayList<Integer> numList = new ArrayList<Integer>();//vc=10, pc=0
        numList.add(100);
        numList.add(200);


        ArrayList<Double> marksList = new ArrayList<Double>();//vc=10, pc=0
        marksList.add(12.33);
        marksList.add(200.00);


        ArrayList<String> browserList = new ArrayList<String>();//vc=10, pc=0
        browserList.add("chrome");
        browserList.add("firefox");
        browserList.add("edge");
```

paste arrlist12



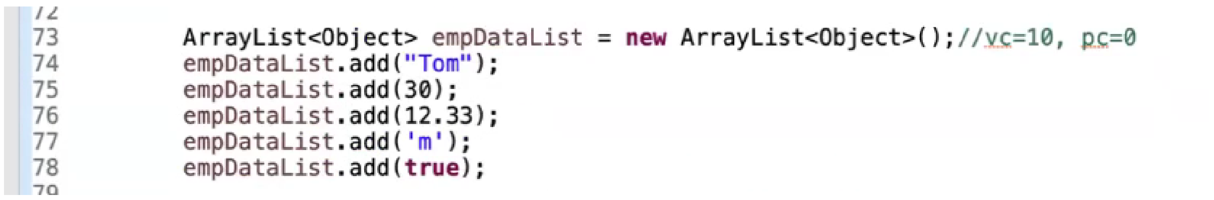
arrlist12.java

```
package com.day28;

import java.util.ArrayList;

public class arrlist12 {

//   Generics.
//   Cant add any other value.
    public static void main(String[] args) {

        //Different types of generics.
        ArrayList<Integer> a1=new ArrayList<Integer>();
        a1.add(34234);
        a1.add(20);
        System.out.println(a1);

        ArrayList<Double> a2=new ArrayList<Double>();
        a2.add(34234.3245435);
        a2.add(20.54578);
        System.out.println(a2);

        ArrayList<String> a3=new ArrayList<String>();
        a3.add("lion");
        a3.add("gorialla@#$324");
        System.out.println(a3);

    }
```



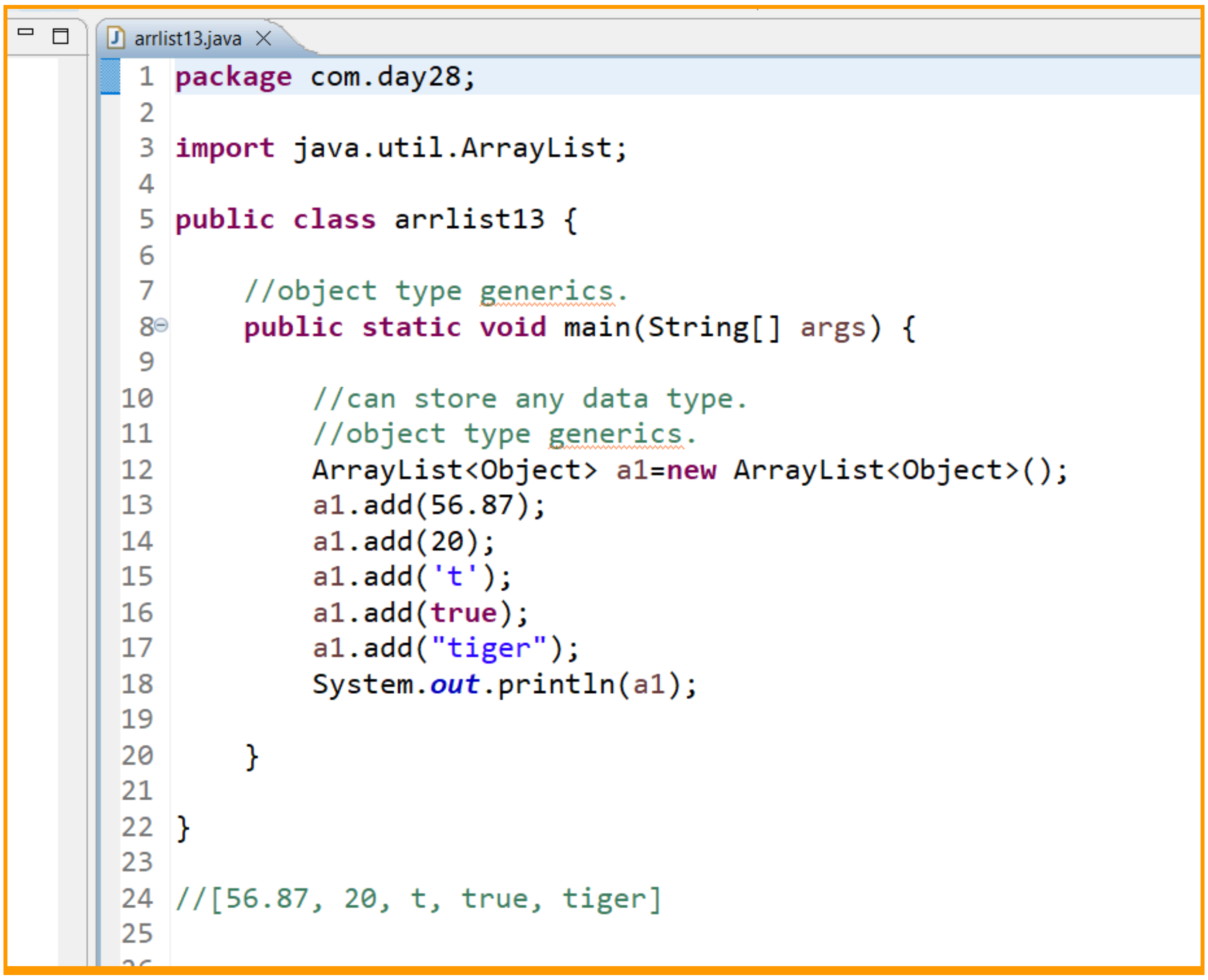
```


    }

}

//[34234, 20]
//[34234.3245435, 20.54578]
//[lion, gorialla@#$324]
```

Object type-



```
        ArrayList<Object> empDataList = new ArrayList<Object>();//vc=10, pc=0
        empDataList.add("Tom");
        empDataList.add(30);
        empDataList.add(12.33);
        empDataList.add('m');
        empDataList.add(true);
```

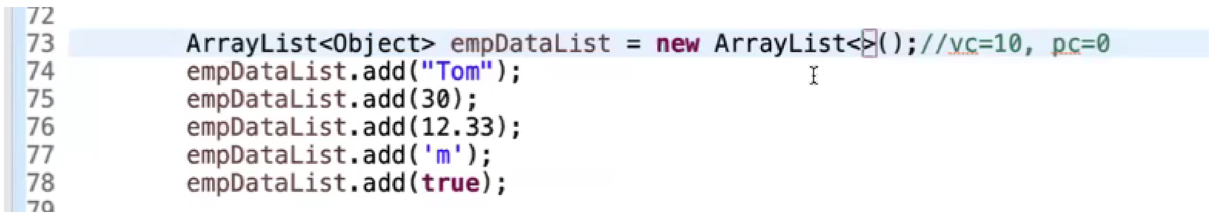
paste arrlist13



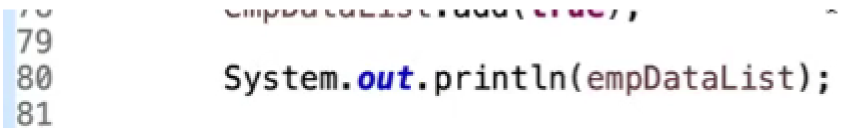
arrlist13.java

```
package com.day28;

import java.util.ArrayList;

public class arrlist13 {

    //object type generics.
    public static void main(String[] args) {

        //can store any data type.
        //object type generics.
        ArrayList<Object> a1=new ArrayList<Object>();
        a1.add(56.87);
        a1.add(20);
        a1.add('t');
        a1.add(true);
        a1.add("tiger");
        System.out.println(a1);

    }

}

//[56.87, 20, t, true, tiger]

```

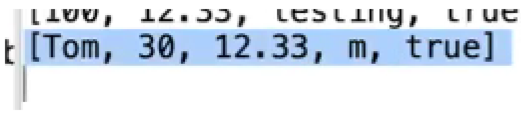
Right side is not mandatory but if we write no harm –



```
        ArrayList<Object> empDataList = new ArrayList<>();//vc=10, pc=0
        empDataList.add("Tom");
        empDataList.add(30);
        empDataList.add(12.33);
        empDataList.add('m');
        empDataList.add(true);
```

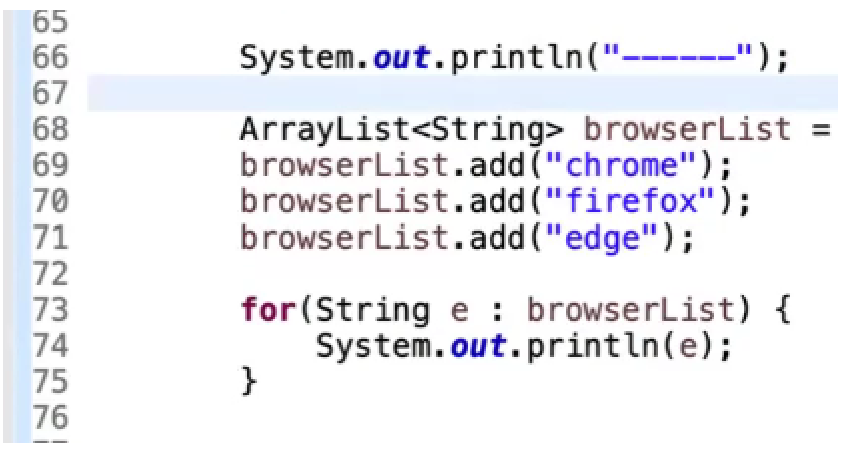


```
        empDataList.add(true);

        System.out.println(empDataList);

```



```
[100, 12.33, testing, true, a]
[Tom, 30, 12.33, m, true]
```

For each loop-



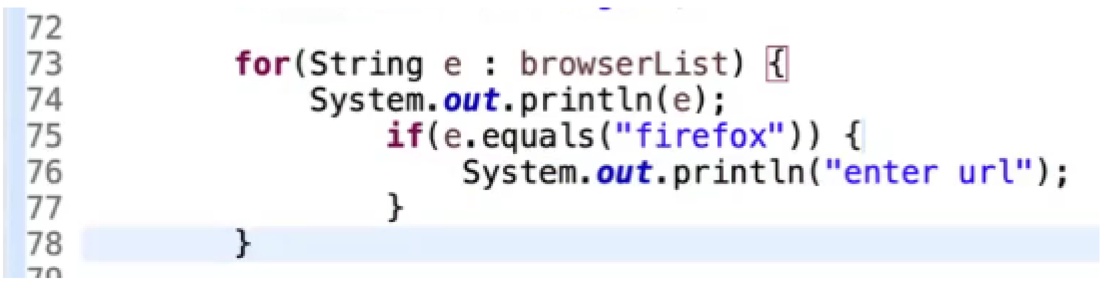
```
        System.out.println("------");

        ArrayList<String> browserList =
        browserList.add("chrome");
        browserList.add("firefox");
        browserList.add("edge");

        for(String e : browserList) {
            System.out.println(e);
        }

```

Chrome

Ff

Edge

With loops under loops in for each-



```

        for(String e : browserList) {
            System.out.println(e);
                if(e.equals("firefox")) {
                    System.out.println("enter url");
                }
        }
```

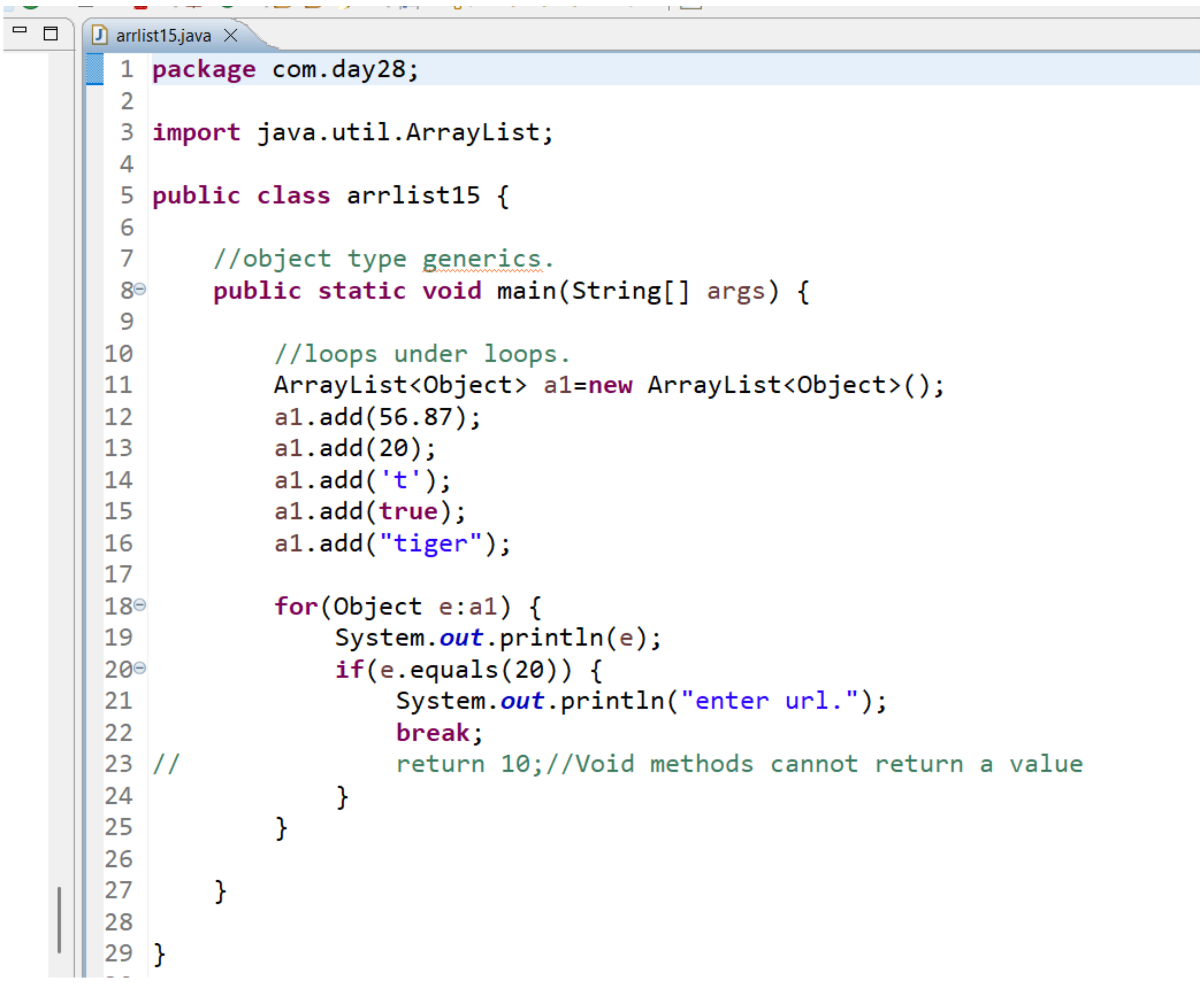
Chrome

Ff

Enter url

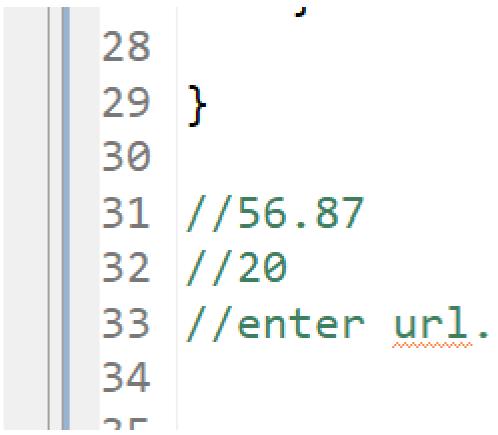
Edge

paste arrlist15



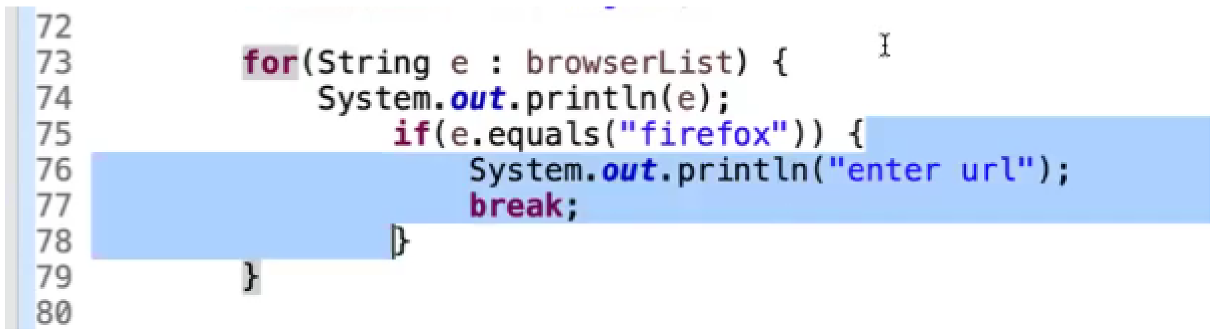
arrlist15.java

```
package com.day28;

import java.util.ArrayList;

public class arrlist15 {

    //object type generics.
    public static void main(String[] args) {

        //loops under loops.
        ArrayList<Object> a1=new ArrayList<Object>();
        a1.add(56.87);
        a1.add(20);
        a1.add('t');
        a1.add(true);
        a1.add("tiger");

        for(Object e:a1) {
            System.out.println(e);
            if(e.equals(20)) {
                System.out.println("enter url.");
                break;
//              return 10;//Void methods cannot return a value
            }
        }

    }

}
```



```
}

//56.87
//20
//enter url.
```

With break-



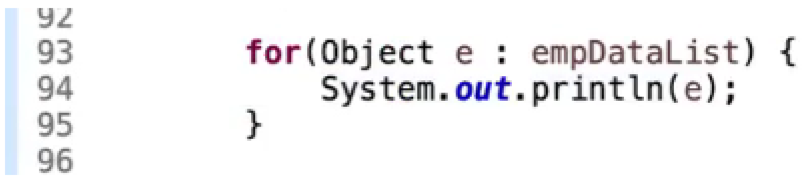
```
for(String e : browserList) {
    System.out.println(e);
        if(e.equals("firefox")) {
            System.out.println("enter url");
            break;
        }
}
```

Chrome

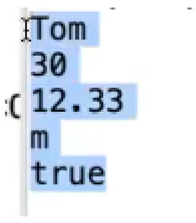
Ff

Enter url

Print object-



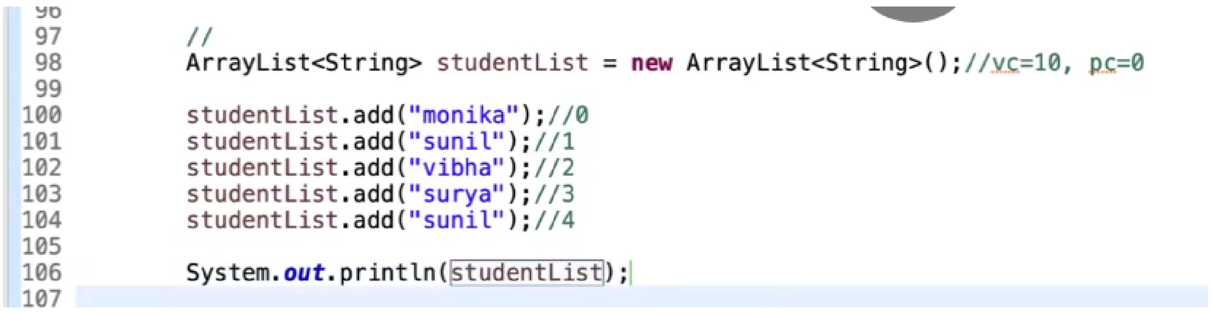
```
for(Object e : empDataList) {
    System.out.println(e);
}
```



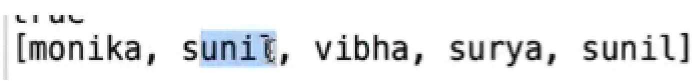
```
Tom
30
12.33
m
true
```

Duplicate value-

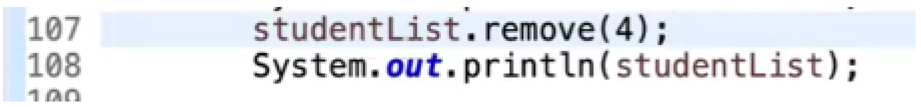
Allowed.



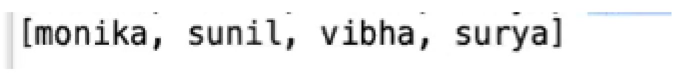
```java
        //
        ArrayList<String> studentList = new ArrayList<String>();//vc=10, pc=0

        studentList.add("monika");//0
        studentList.add("sunil");//1
        studentList.add("vibha");//2
        studentList.add("surya");//3
        studentList.add("sunil");//4

        System.out.println(studentList);
```



```
[monika, sunil, vibha, surya, sunil]
```

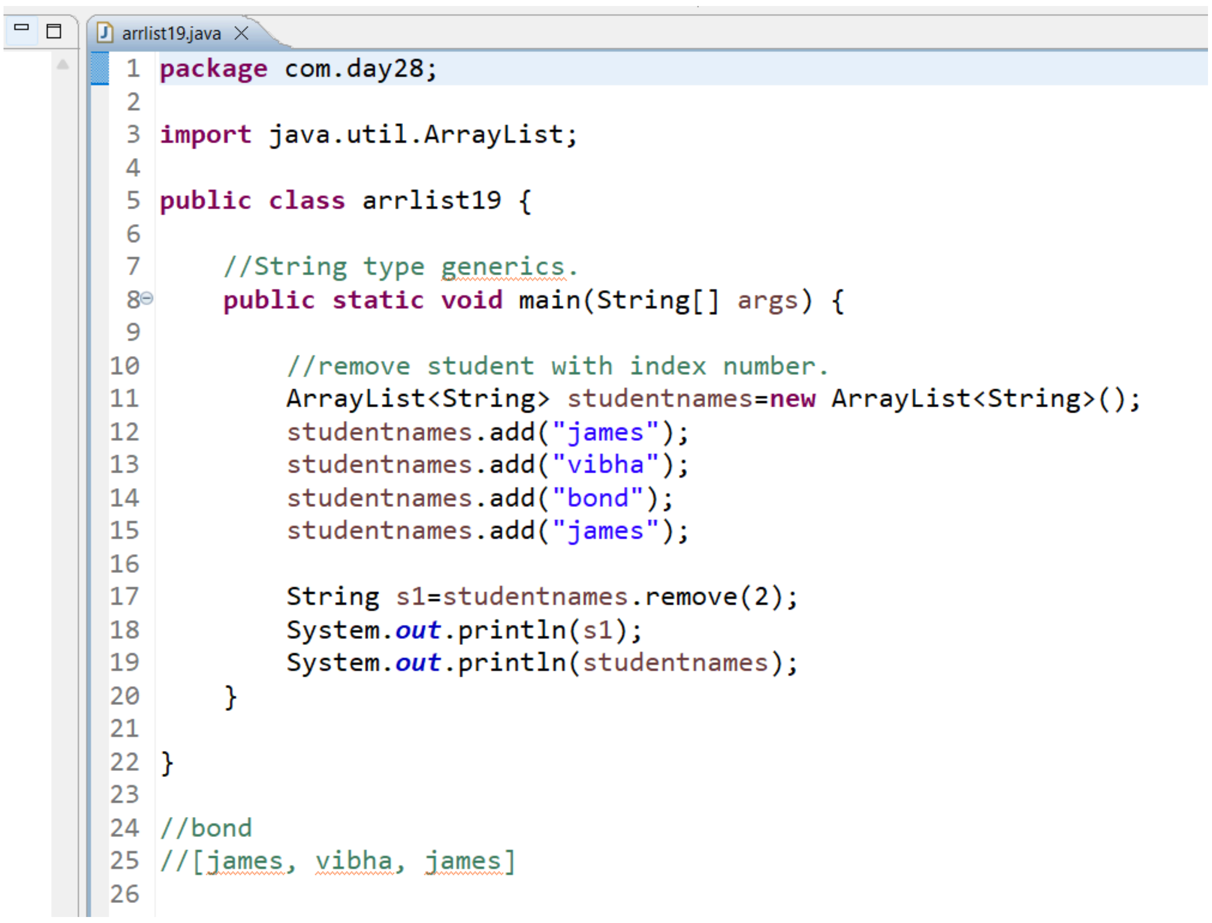


```java
        studentList.remove(4);
        System.out.println(studentList);
```



```
[monika, sunil, vibha, surya]
```

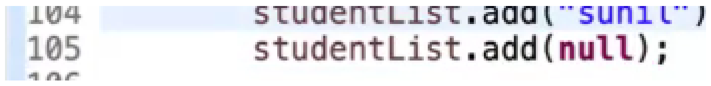
paste arrlist19



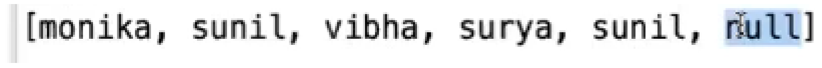
```java
package com.day28;

import java.util.ArrayList;

public class arrlist19 {

    //String type generics.
    public static void main(String[] args) {

        //remove student with index number.
        ArrayList<String> studentnames=new ArrayList<String>();
        studentnames.add("james");
        studentnames.add("vibha");
        studentnames.add("bond");
        studentnames.add("james");

        String s1=studentnames.remove(2);
        System.out.println(s1);
        System.out.println(studentnames);
    }

}

//bond
//[james, vibha, james]

```

Null-

Allowed.

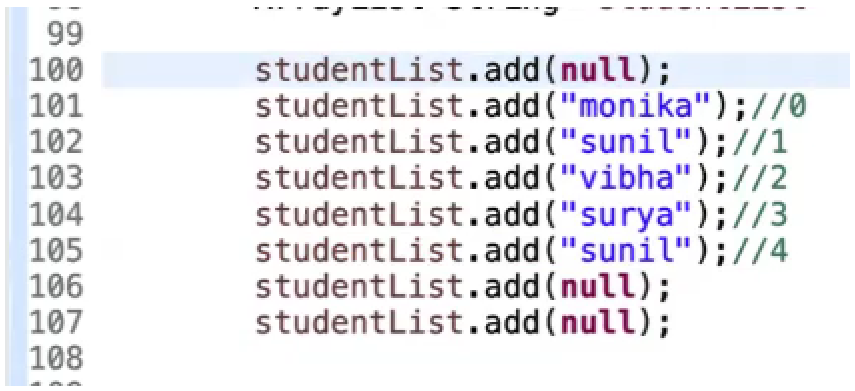


```
        studentList.add("sunil")
        studentList.add(null);
```



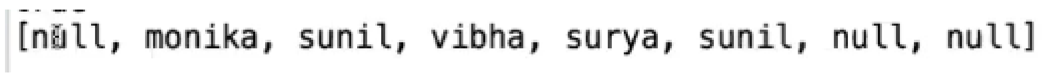
```
[monika, sunil, vibha, surya, sunil, null]
```

Multiple null-



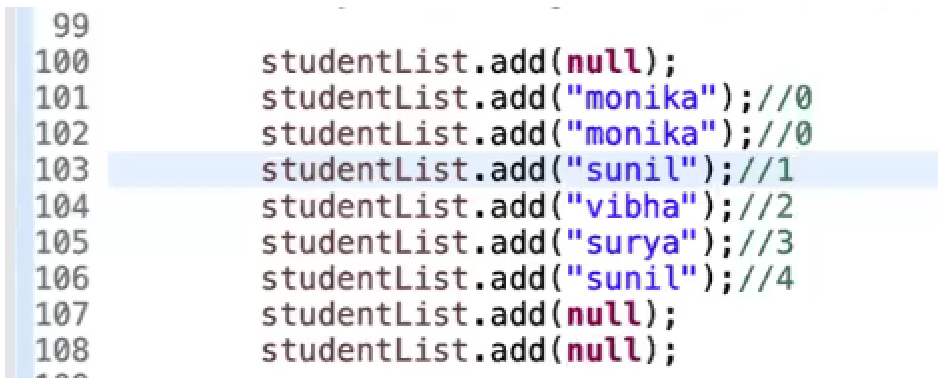
```
        studentList.add(null);
        studentList.add("monika");//0
        studentList.add("sunil");//1
        studentList.add("vibha");//2
        studentList.add("surya");//3
        studentList.add("sunil");//4
        studentList.add(null);
        studentList.add(null);
```

Allowed.

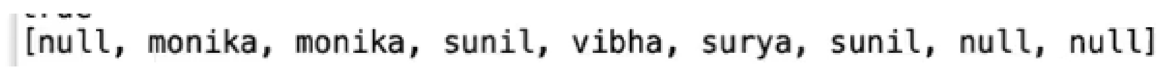


```
[null, monika, sunil, vibha, surya, sunil, null, null]
```

Back to back duplicates-



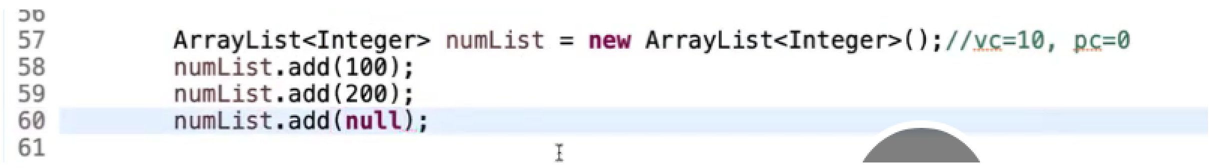
```
        studentList.add(null);
        studentList.add("monika");//0
        studentList.add("monika");//0
        studentList.add("sunil");//1
        studentList.add("vibha");//2
        studentList.add("surya");//3
        studentList.add("sunil");//4
        studentList.add(null);
        studentList.add(null);
```



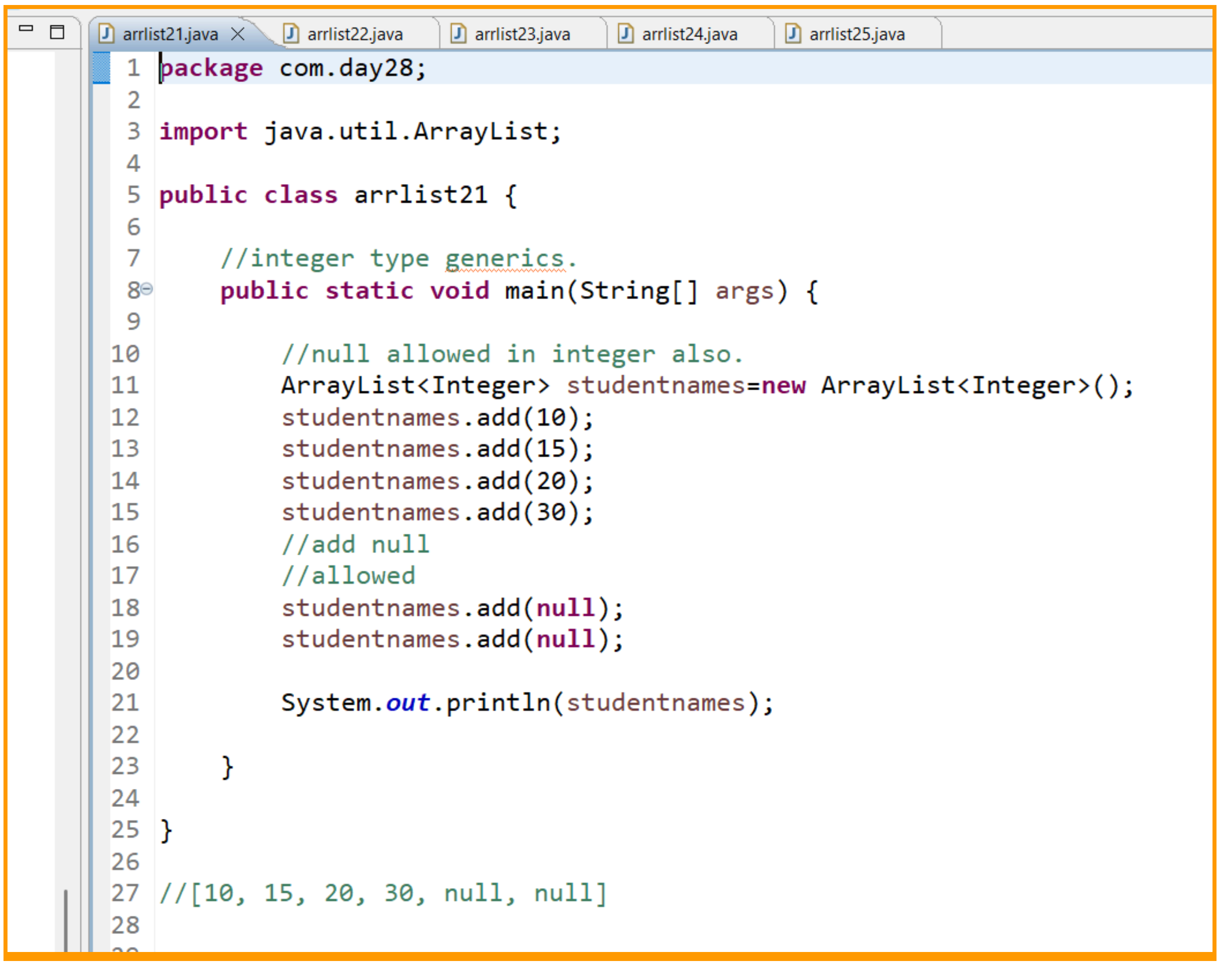
```
[null, monika, monika, sunil, vibha, surya, sunil, null, null]
```

Integer non primitive-

Null allowed.

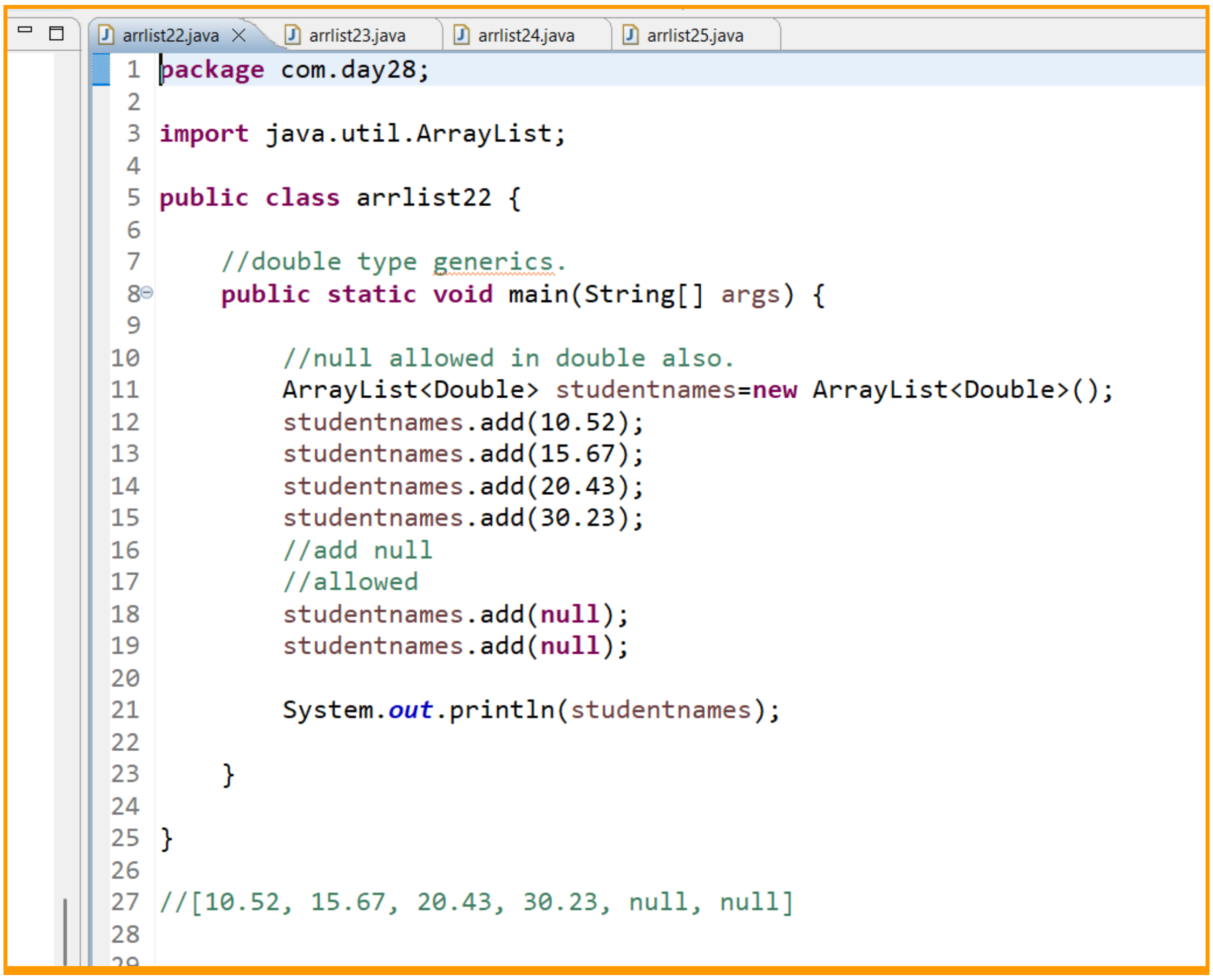

```
        ArrayList<Integer> numList = new ArrayList<Integer>();//vc=10, pc=0
        numList.add(100);
        numList.add(200);
        numList.add(null);
```

paste arrlist21



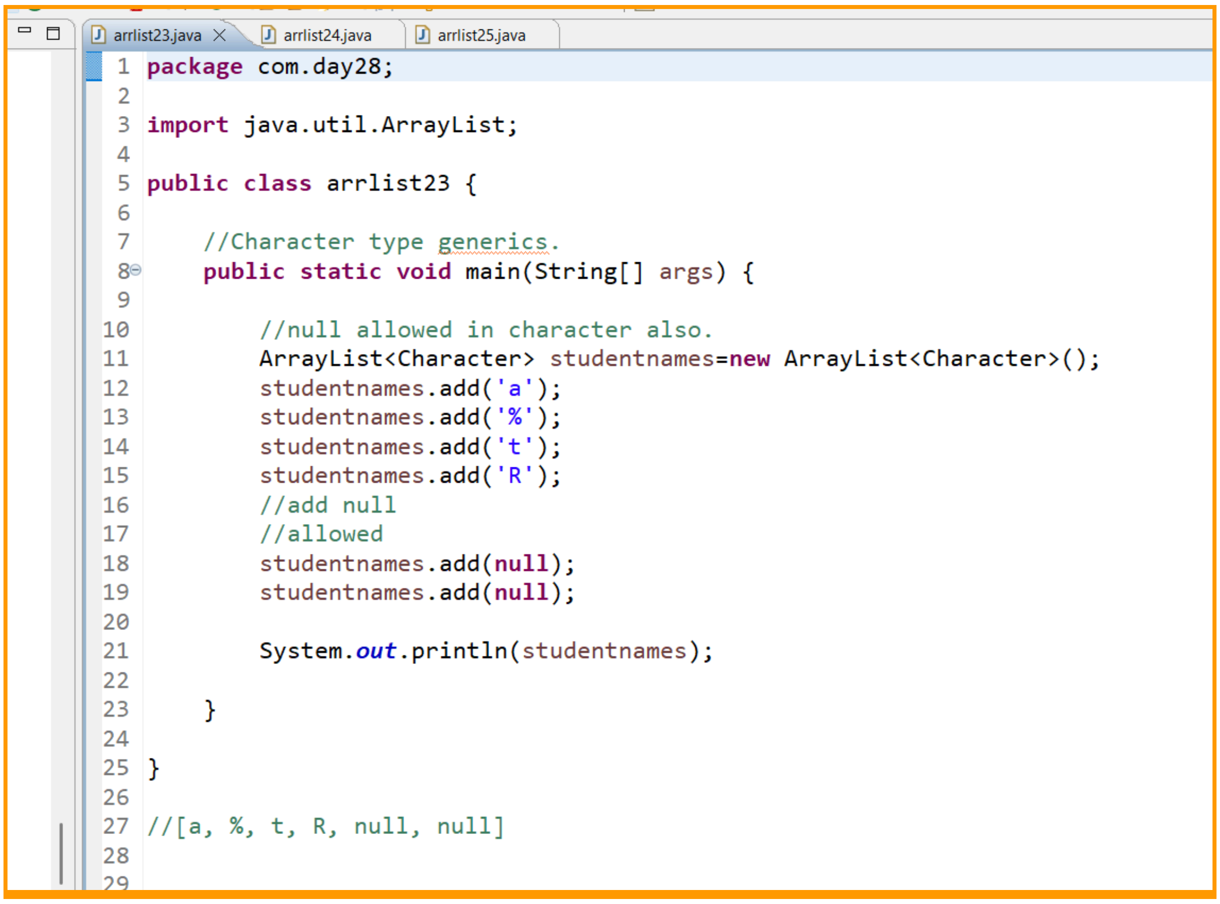
```
package com.day28;

import java.util.ArrayList;

public class arrlist21 {

    //integer type generics.
    public static void main(String[] args) {

        //null allowed in integer also.
        ArrayList<Integer> studentnames=new ArrayList<Integer>();
        studentnames.add(10);
        studentnames.add(15);
        studentnames.add(20);
        studentnames.add(30);
        //add null
        //allowed
        studentnames.add(null);
        studentnames.add(null);

        System.out.println(studentnames);

    }

}

//[10, 15, 20, 30, null, null]

```

paste arrlist22



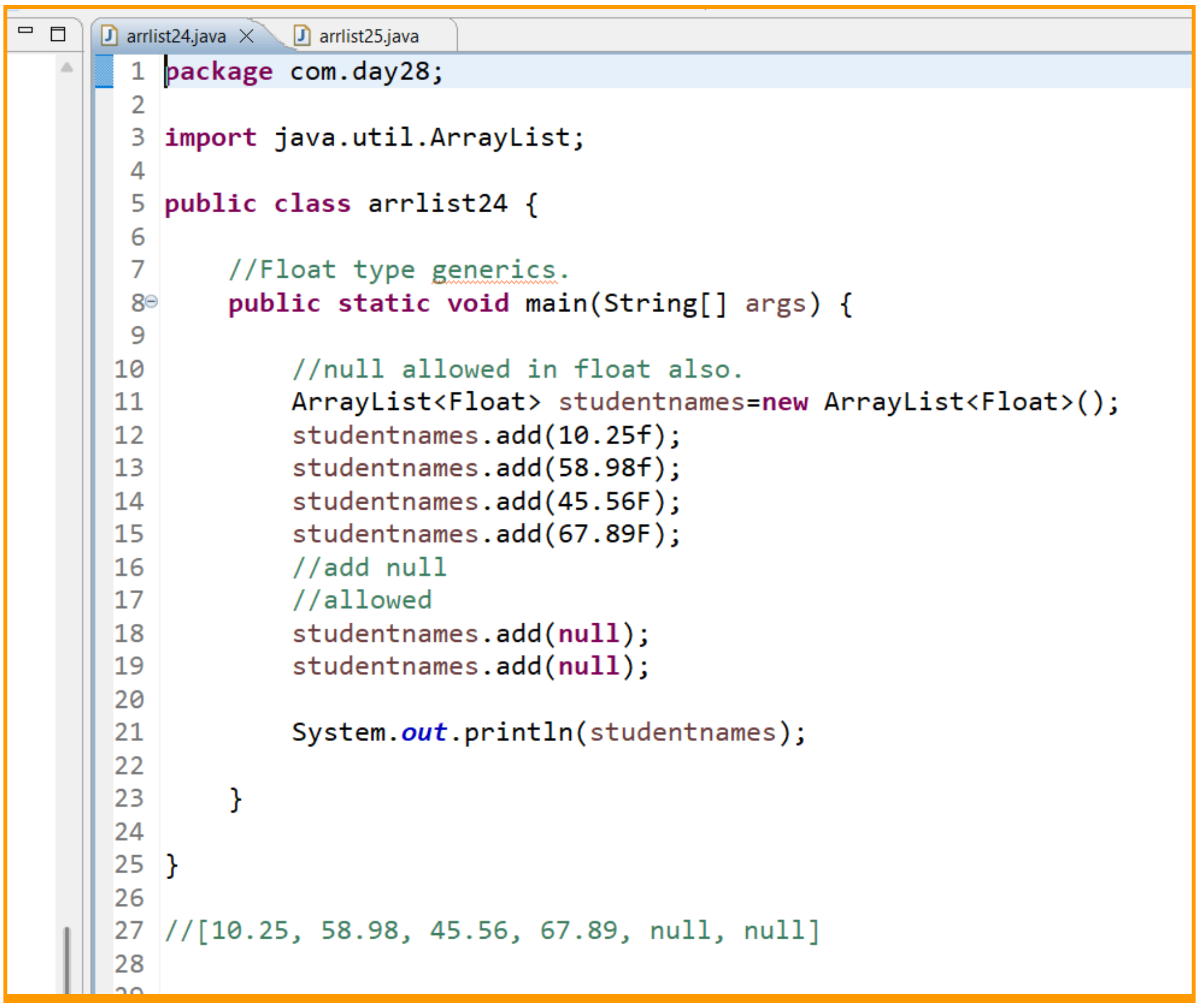
```
package com.day28;

import java.util.ArrayList;

public class arrlist22 {

    //double type generics.
    public static void main(String[] args) {

        //null allowed in double also.
        ArrayList<Double> studentnames=new ArrayList<Double>();
        studentnames.add(10.52);
        studentnames.add(15.67);
        studentnames.add(20.43);
        studentnames.add(30.23);
        //add null
        //allowed
        studentnames.add(null);
        studentnames.add(null);

        System.out.println(studentnames);

    }

}

//[10.52, 15.67, 20.43, 30.23, null, null]
```

paste arrlist23



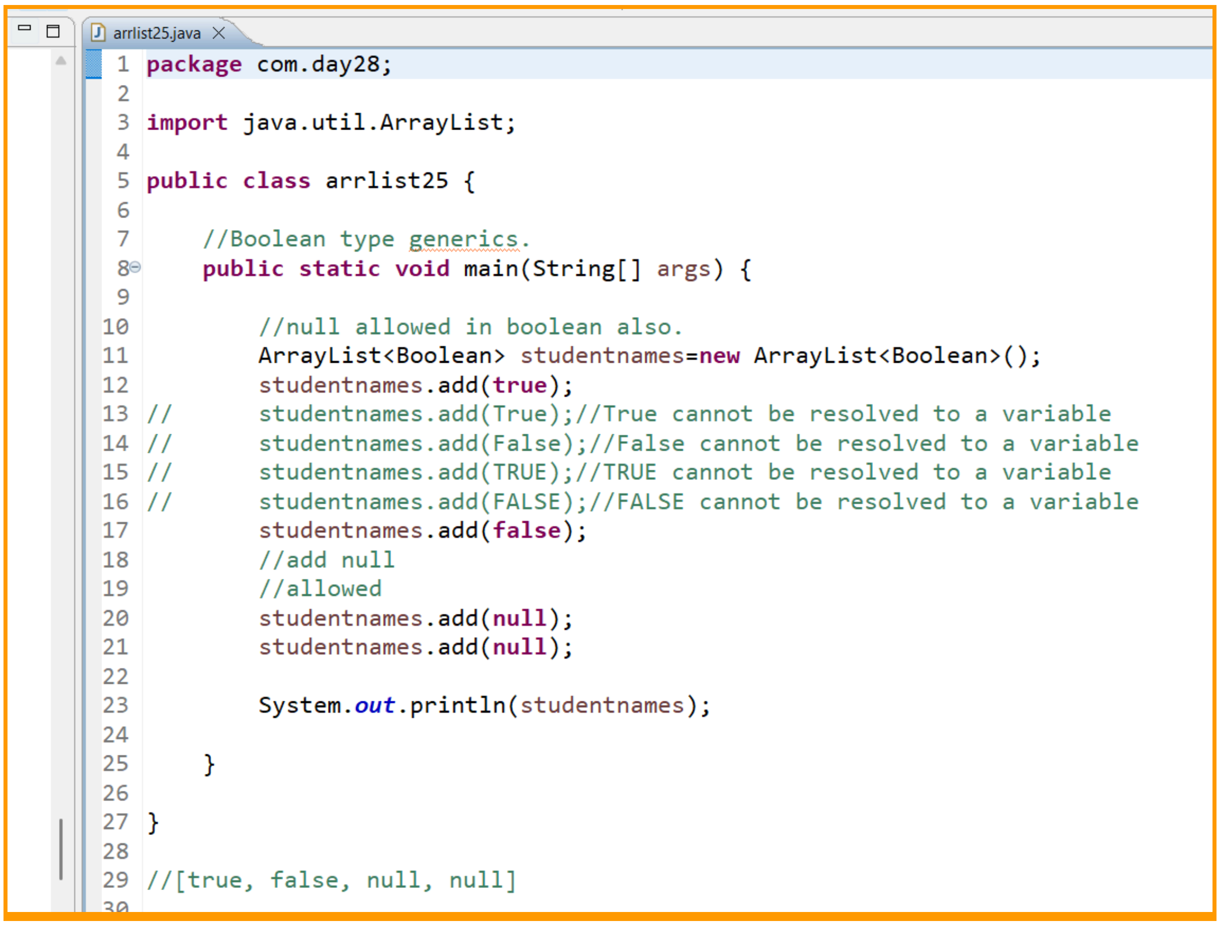
```java
package com.day28;

import java.util.ArrayList;

public class arrlist23 {

    //Character type generics.
    public static void main(String[] args) {

        //null allowed in character also.
        ArrayList<Character> studentnames=new ArrayList<Character>();
        studentnames.add('a');
        studentnames.add('%');
        studentnames.add('t');
        studentnames.add('R');
        //add null
        //allowed
        studentnames.add(null);
        studentnames.add(null);

        System.out.println(studentnames);

    }

}

//[a, %, t, R, null, null]
```

paste arrlist24

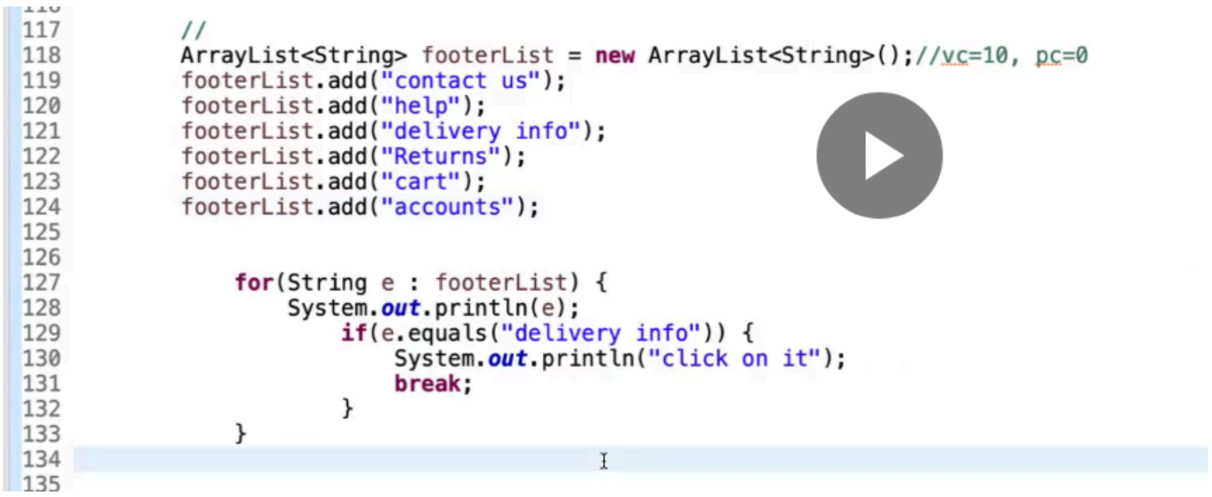


```java
package com.day28;

import java.util.ArrayList;

public class arrlist24 {

    //Float type generics.
    public static void main(String[] args) {

        //null allowed in float also.
        ArrayList<Float> studentnames=new ArrayList<Float>();
        studentnames.add(10.25f);
        studentnames.add(58.98f);
        studentnames.add(45.56F);
        studentnames.add(67.89F);
        //add null
        //allowed
        studentnames.add(null);
        studentnames.add(null);

        System.out.println(studentnames);

    }

}

//[10.25, 58.98, 45.56, 67.89, null, null]

```

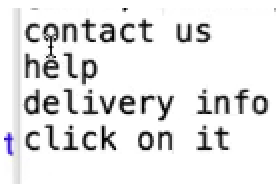
paste arrlist25



```java
package com.day28;

import java.util.ArrayList;

public class arrlist25 {

    //Boolean type generics.
    public static void main(String[] args) {

        //null allowed in boolean also.
        ArrayList<Boolean> studentnames=new ArrayList<Boolean>();
        studentnames.add(true);
//      studentnames.add(True);//True cannot be resolved to a variable
//      studentnames.add(False);//False cannot be resolved to a variable
//      studentnames.add(TRUE);//TRUE cannot be resolved to a variable
//      studentnames.add(FALSE);//FALSE cannot be resolved to a variable
        studentnames.add(false);
        //add null
        //allowed
        studentnames.add(null);
        studentnames.add(null);

        System.out.println(studentnames);

    }

}

//[true, false, null, null]
```



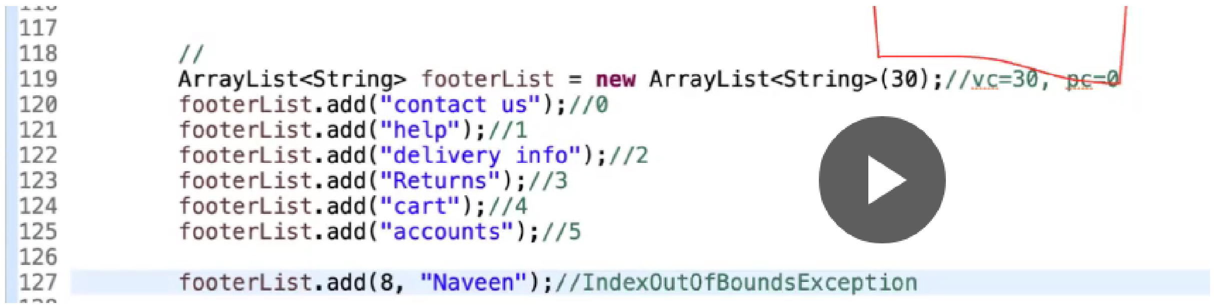
```java
        //
        ArrayList<String> footerList = new ArrayList<String>();//vc=10, pc=0
        footerList.add("contact us");
        footerList.add("help");
        footerList.add("delivery info");
        footerList.add("Returns");
        footerList.add("cart");
        footerList.add("accounts");


            for(String e : footerList) {
                System.out.println(e);
                    if(e.equals("delivery info")) {
                        System.out.println("click on it");
                        break;
                    }
            }

```



```
contact us
help
delivery info
click on it
```

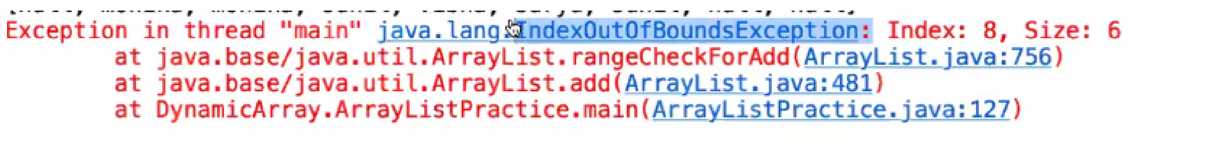
Generics cannot be of primitive data types.(1.07) and (1.15)

Order based collection-



```
        //
        ArrayList<String> footerList = new ArrayList<String>(30);//vc=30, pc=0
        footerList.add("contact us");//0
        footerList.add("help");//1
        footerList.add("delivery info");//2
        footerList.add("Returns");//3
        footerList.add("cart");//4
        footerList.add("accounts");//5

        footerList.add(8, "Naveen");//IndexOutOfBoundsException
```

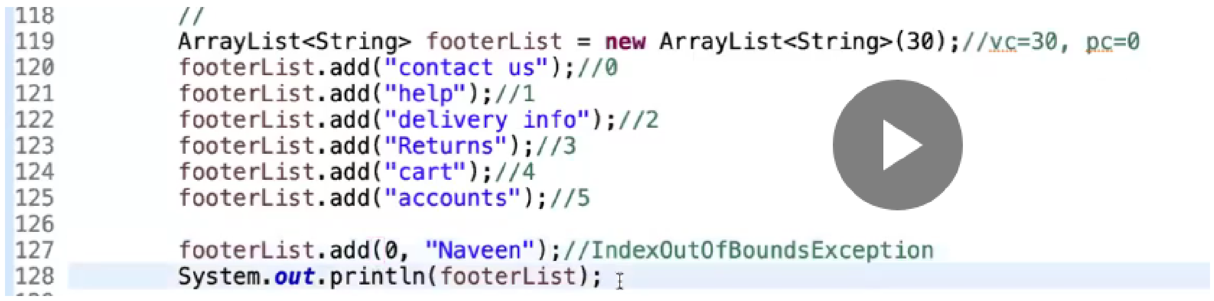
We cant randomly add at any index.



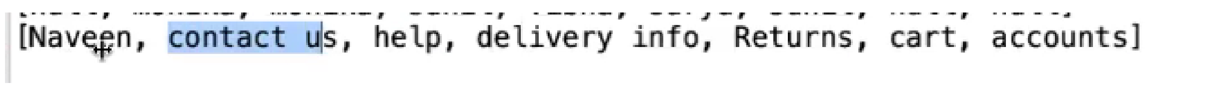
```
Exception in thread "main" java.lang.IndexOutOfBoundsException: Index: 8, Size: 6
	at java.base/java.util.ArrayList.rangeCheckForAdd(ArrayList.java:756)
	at java.base/java.util.ArrayList.add(ArrayList.java:481)
	at DynamicArray.ArrayListPractice.main(ArrayListPractice.java:127)
```

~~1,20~~

Add new value using index-

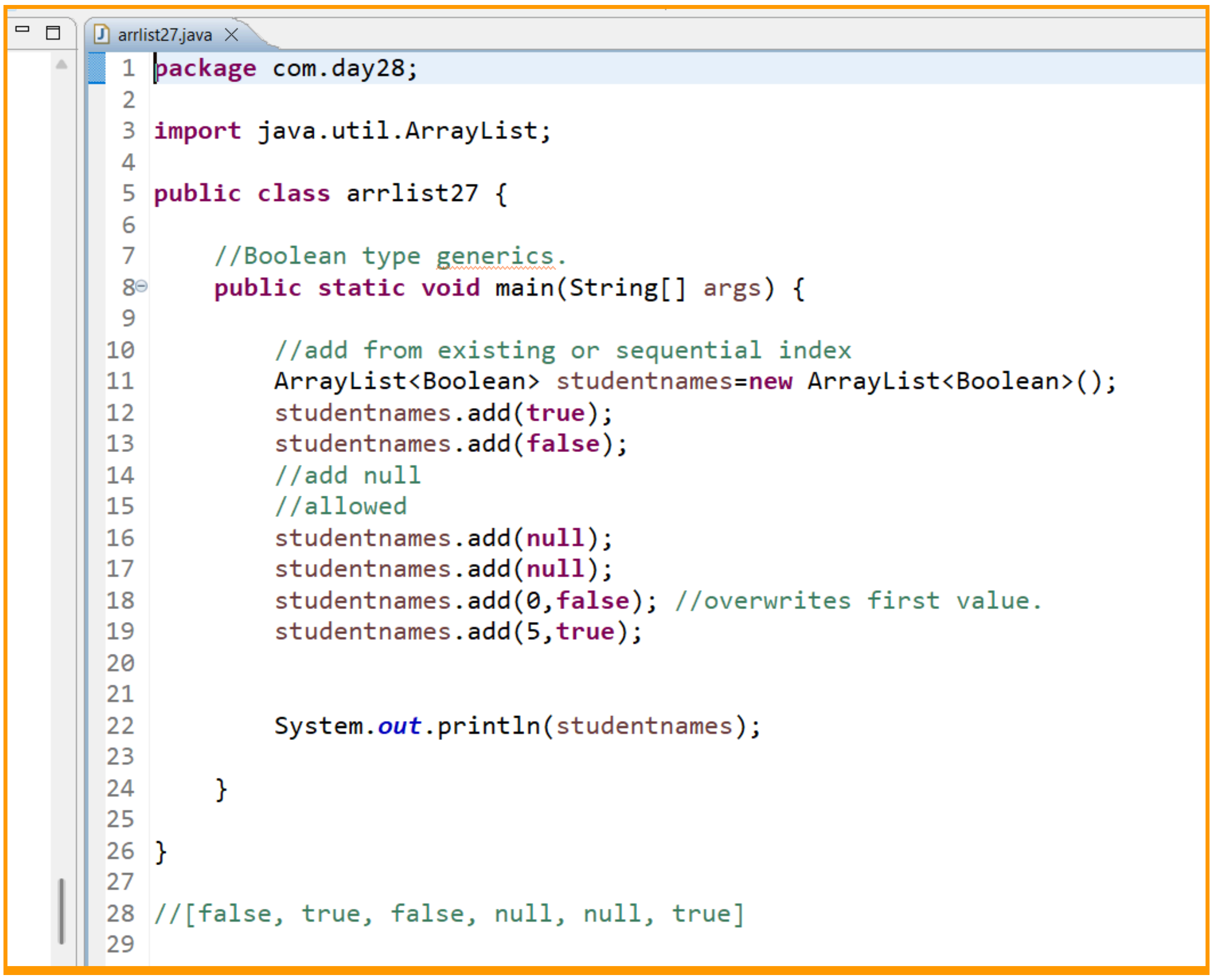


```
        //
        ArrayList<String> footerList = new ArrayList<String>(30);//vc=30, pc=0
        footerList.add("contact us");//0
        footerList.add("help");//1
        footerList.add("delivery info");//2
        footerList.add("Returns");//3
        footerList.add("cart");//4
        footerList.add("accounts");//5

        footerList.add(0, "Naveen");//IndexOutOfBoundsException
        System.out.println(footerList);
```



```
[Naveen, contact us, help, delivery info, Returns, cart, accounts]
```

paste arrlist27

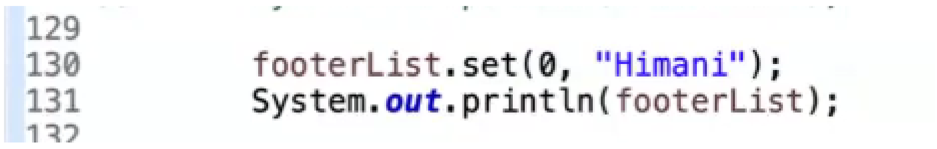


arrlist27.java

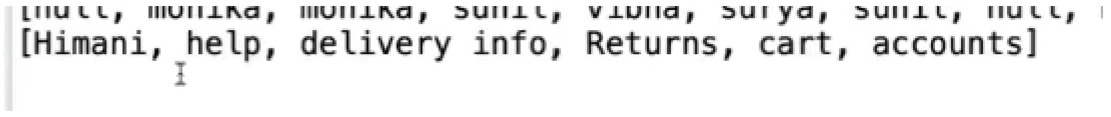
```java
package com.day28;

import java.util.ArrayList;

public class arrlist27 {

    //Boolean type generics.
    public static void main(String[] args) {

        //add from existing or sequential index
        ArrayList<Boolean> studentnames=new ArrayList<Boolean>();
        studentnames.add(true);
        studentnames.add(false);
        //add null
        //allowed
        studentnames.add(null);
        studentnames.add(null);
        studentnames.add(0,false); //overwrites first value.
        studentnames.add(5,true);


        System.out.println(studentnames);

    }

}

//[false, true, false, null, null, true]

```

Set-

To update.

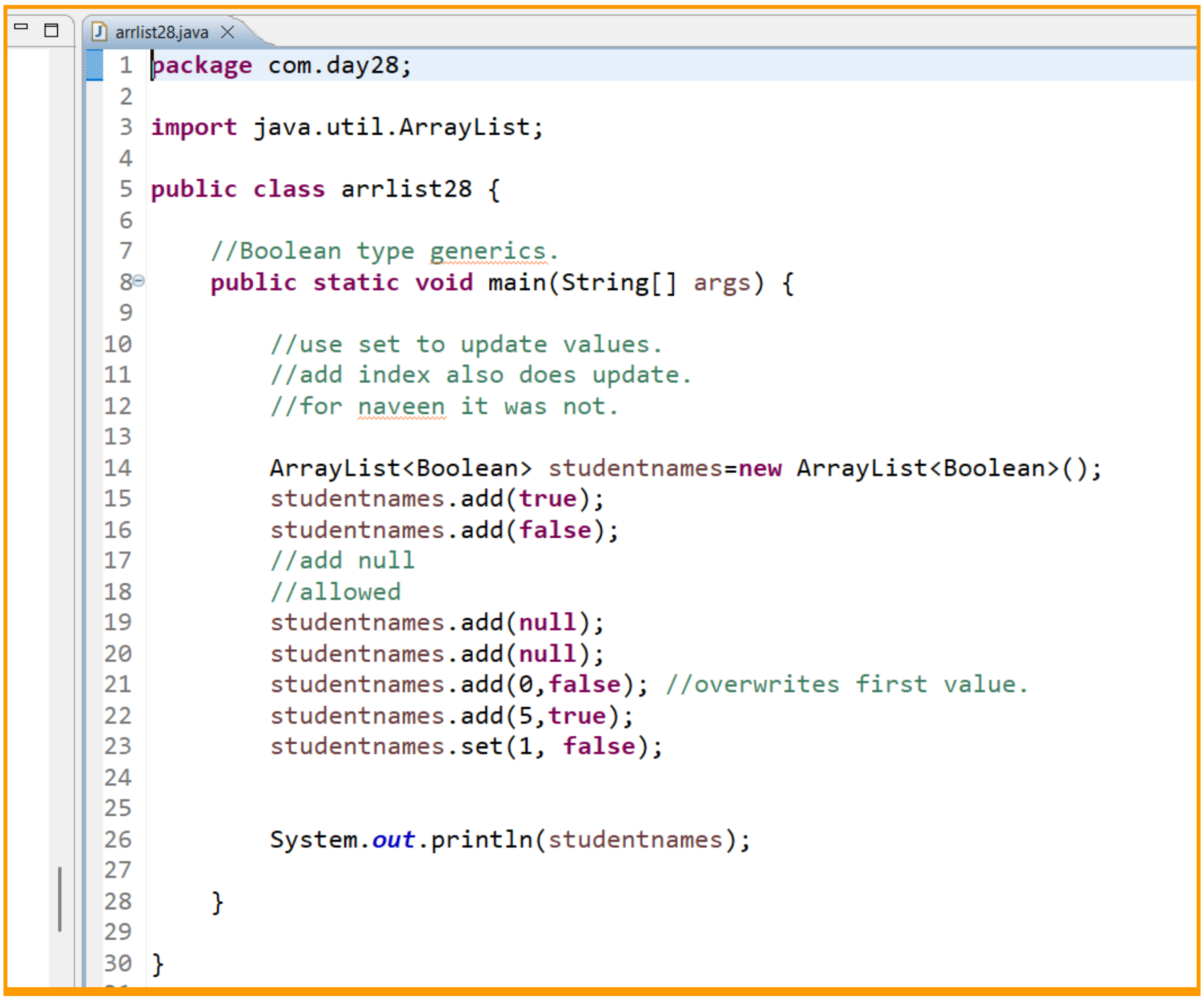


```java

        footerList.set(0, "Himani");
        System.out.println(footerList);

```



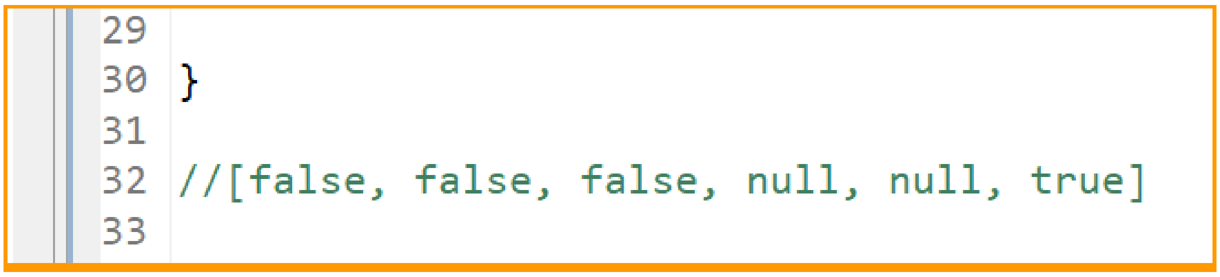
```
[null, monika, monika, sunil, vibha, surya, sunil, null,
[Himani, help, delivery info, Returns, cart, accounts]
```

paste arrlist28



arrlist28.java

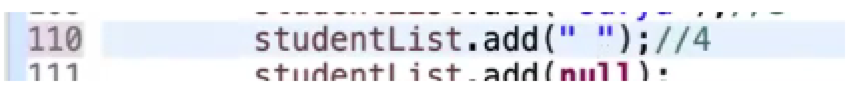
```
package com.day28;

import java.util.ArrayList;

public class arrlist28 {

    //Boolean type generics.
    public static void main(String[] args) {

        //use set to update values.
        //add index also does update.
        //for naveen it was not.

        ArrayList<Boolean> studentnames=new ArrayList<Boolean>();
        studentnames.add(true);
        studentnames.add(false);
        //add null
        //allowed
        studentnames.add(null);
        studentnames.add(null);
        studentnames.add(0,false); //overwrites first value.
        studentnames.add(5,true);
        studentnames.set(1, false);


        System.out.println(studentnames);

    }

}
```



```

}

//[false, false, false, null, null, true]

```

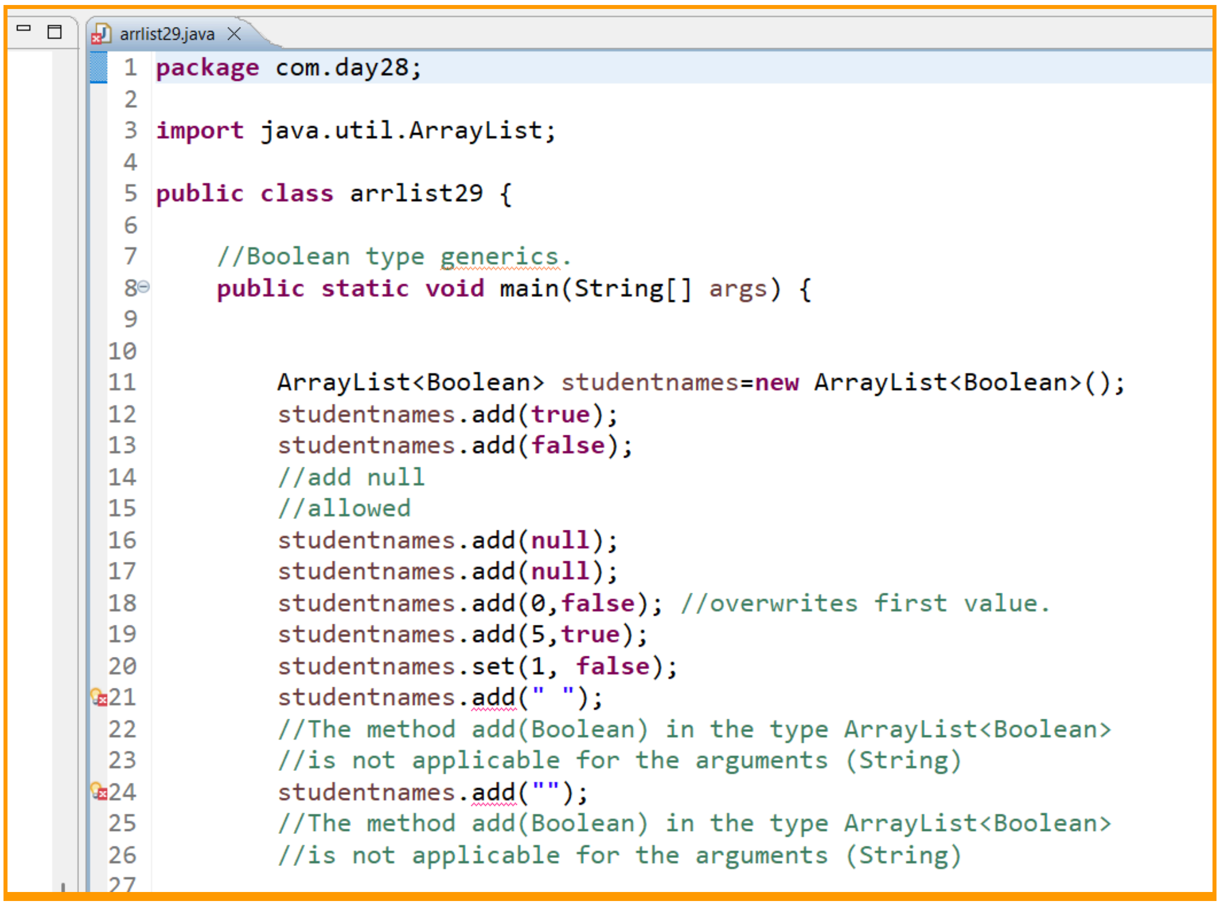
No access modifiers for collections.

We can add "" or " " in array list.



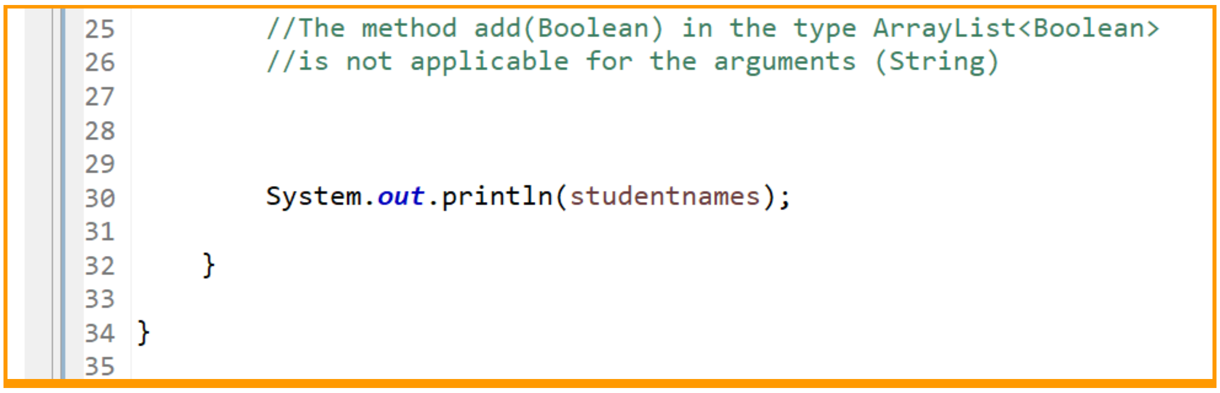
```
        studentList.add(" ");//4
        studentList.add(null);
```

paste arrlist29

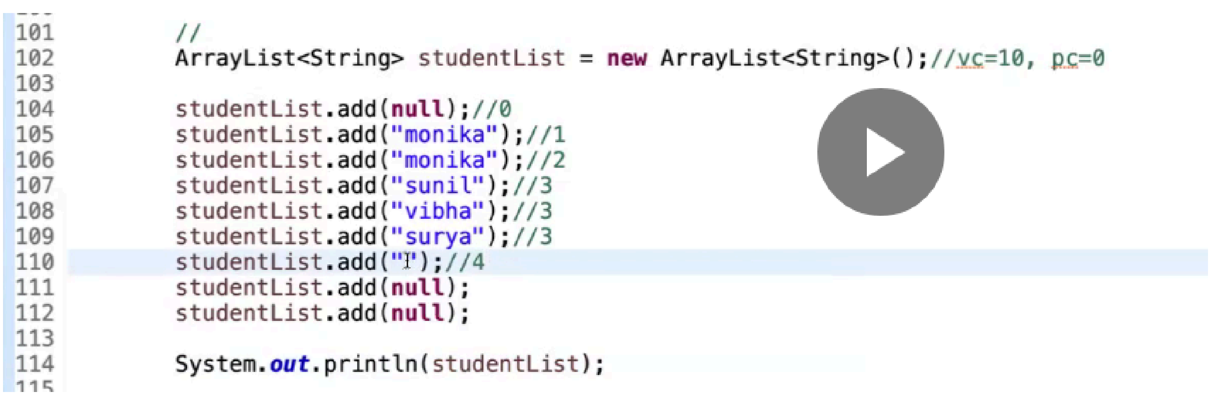


arrlist29.java

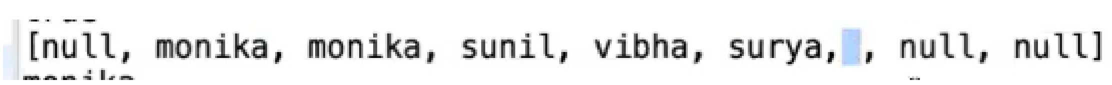
```java
package com.day28;

import java.util.ArrayList;

public class arrlist29 {

    //Boolean type generics.
    public static void main(String[] args) {


        ArrayList<Boolean> studentnames=new ArrayList<Boolean>();
        studentnames.add(true);
        studentnames.add(false);
        //add null
        //allowed
        studentnames.add(null);
        studentnames.add(null);
        studentnames.add(0,false); //overwrites first value.
        studentnames.add(5,true);
        studentnames.set(1, false);
        studentnames.add(" ");
        //The method add(Boolean) in the type ArrayList<Boolean>
        //is not applicable for the arguments (String)
        studentnames.add("");
        //The method add(Boolean) in the type ArrayList<Boolean>
        //is not applicable for the arguments (String)
```



```java
        //The method add(Boolean) in the type ArrayList<Boolean>
        //is not applicable for the arguments (String)



        System.out.println(studentnames);

    }

}
```

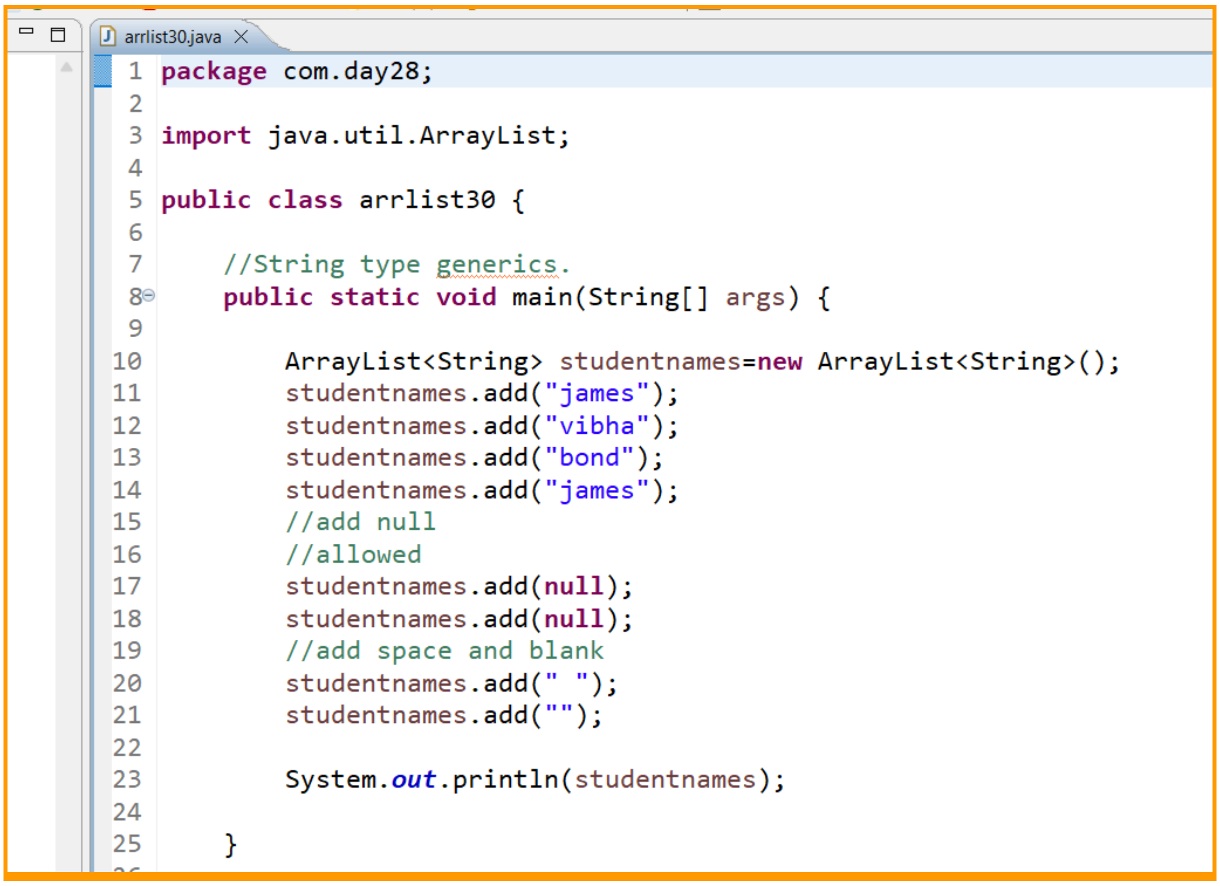


```java
        //
        ArrayList<String> studentList = new ArrayList<String>();//vc=10, pc=0

        studentList.add(null);//0
        studentList.add("monika");//1
        studentList.add("monika");//2
        studentList.add("sunil");//3
        studentList.add("vibha");//3
        studentList.add("surya");//3
        studentList.add("");//4
        studentList.add(null);
        studentList.add(null);

        System.out.println(studentList);
```

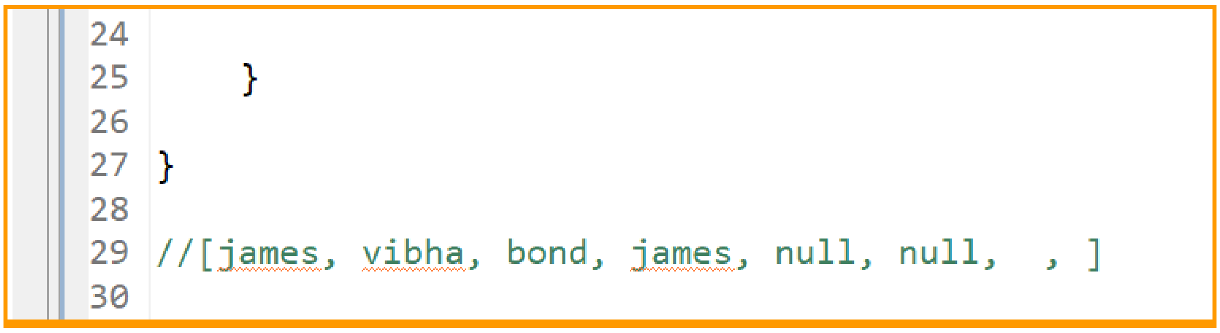


```
[null, monika, monika, sunil, vibha, surya, , null, null]
```

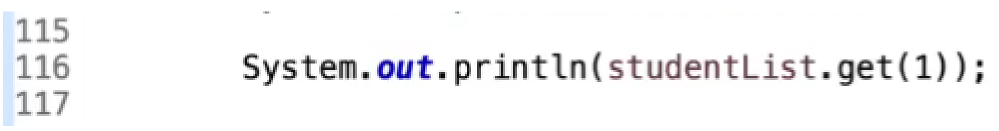
paste arrlist30



```java
package com.day28;

import java.util.ArrayList;

public class arrlist30 {

    //String type generics.
    public static void main(String[] args) {

        ArrayList<String> studentnames=new ArrayList<String>();
        studentnames.add("james");
        studentnames.add("vibha");
        studentnames.add("bond");
        studentnames.add("james");
        //add null
        //allowed
        studentnames.add(null);
        studentnames.add(null);
        //add space and blank
        studentnames.add(" ");
        studentnames.add("");

        System.out.println(studentnames);

    }
```

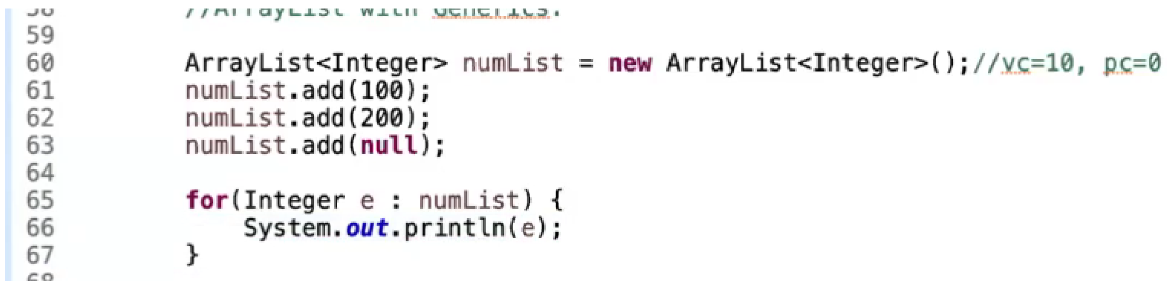


```java

    }

}

//[james, vibha, bond, james, null, null,  , ]

```



```java

        System.out.println(studentList.get(1));

```

monika



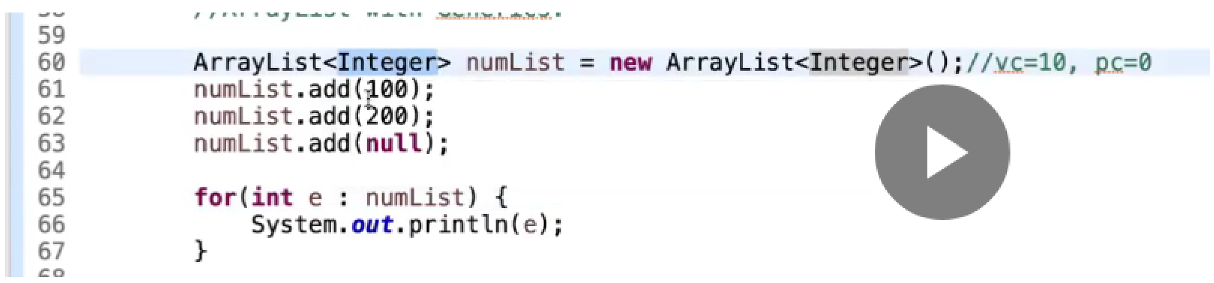
```
//ArrayList with Generics:

ArrayList<Integer> numList = new ArrayList<Integer>();//vc=10, pc=0
numList.add(100);
numList.add(200);
numList.add(null);

for(Integer e : numList) {
    System.out.println(e);
}
```

100

200

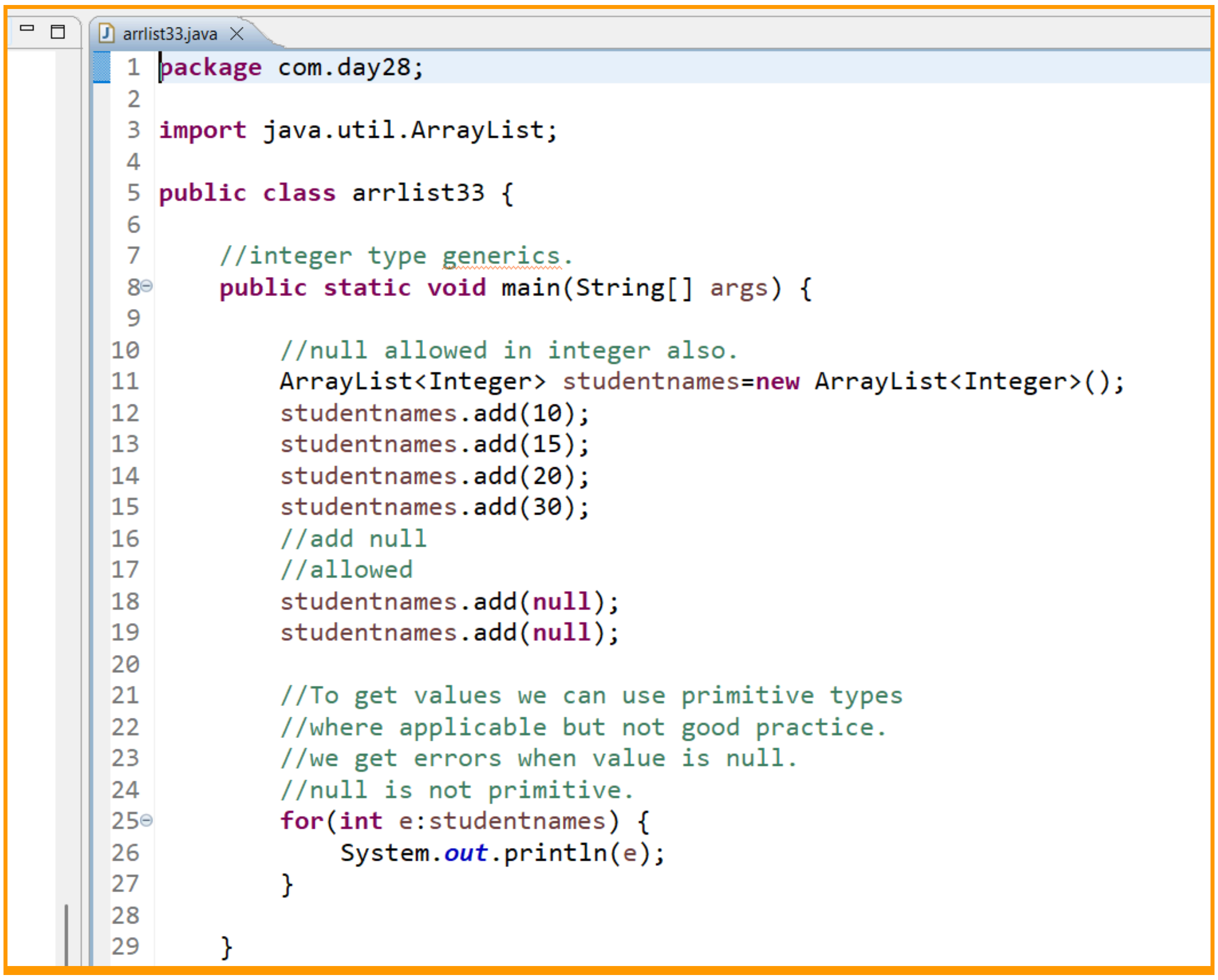
Null

To get values we can use primitive types where applicable but not good practice-



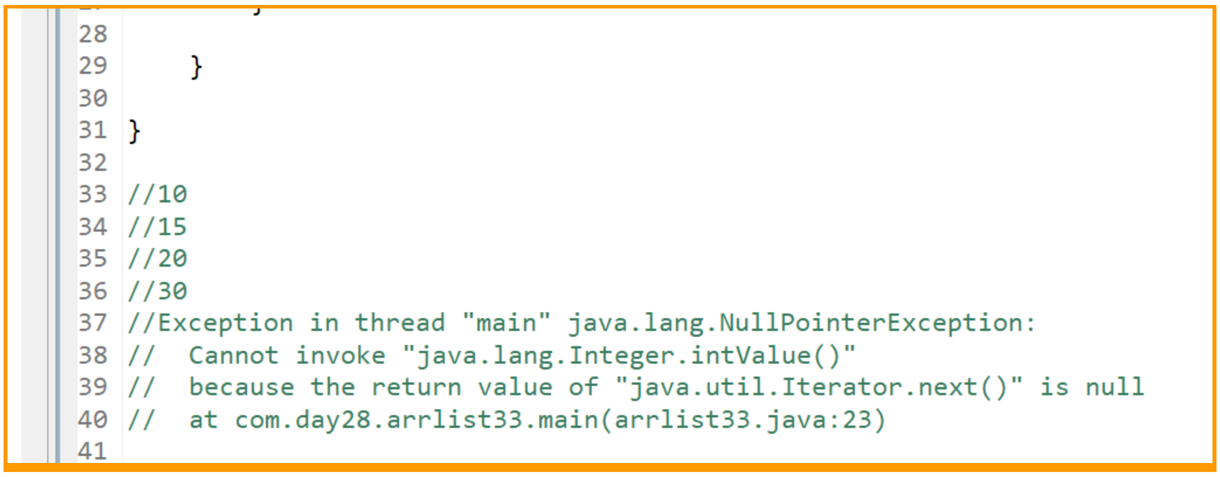
```
//ArrayList with Generics:

ArrayList<Integer> numList = new ArrayList<Integer>();//vc=10, pc=0
numList.add(100);
numList.add(200);
numList.add(null);

for(int e : numList) {
    System.out.println(e);
}
```

paste arrlist33



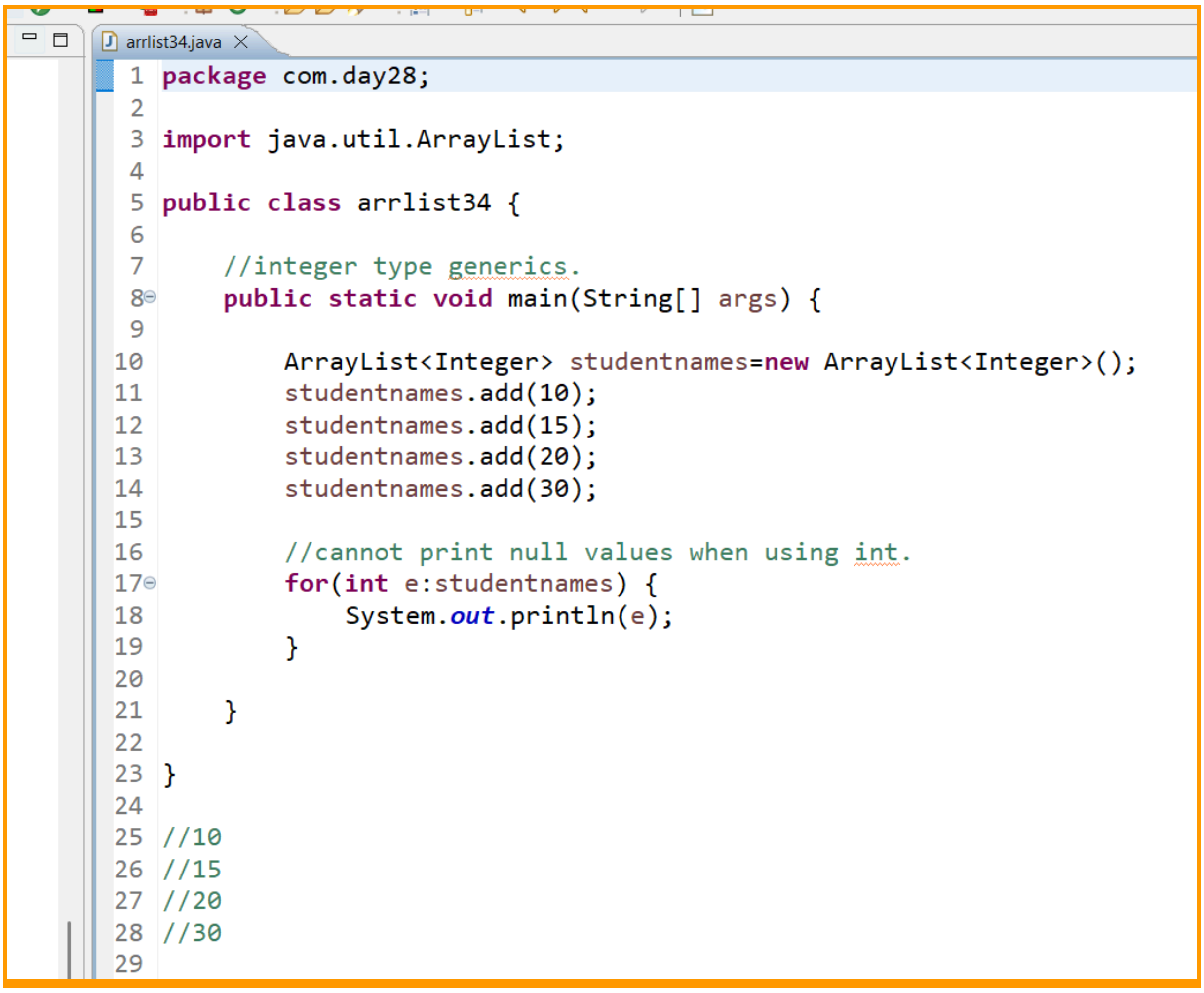
arrlist33.java

```java
package com.day28;

import java.util.ArrayList;

public class arrlist33 {

    //integer type generics.
    public static void main(String[] args) {

        //null allowed in integer also.
        ArrayList<Integer> studentnames=new ArrayList<Integer>();
        studentnames.add(10);
        studentnames.add(15);
        studentnames.add(20);
        studentnames.add(30);
        //add null
        //allowed
        studentnames.add(null);
        studentnames.add(null);

        //To get values we can use primitive types
        //where applicable but not good practice.
        //we get errors when value is null.
        //null is not primitive.
        for(int e:studentnames) {
            System.out.println(e);
        }

    }
```



```java

    }

}

//10
//15
//20
//30
//Exception in thread "main" java.lang.NullPointerException:
//   Cannot invoke "java.lang.Integer.intValue()"
//   because the return value of "java.util.Iterator.next()" is null
//   at com.day28.arrlist33.main(arrlist33.java:23)

```

paste arrlist34



```java
package com.day28;

import java.util.ArrayList;

public class arrlist34 {

    //integer type generics.
    public static void main(String[] args) {

        ArrayList<Integer> studentnames=new ArrayList<Integer>();
        studentnames.add(10);
        studentnames.add(15);
        studentnames.add(20);
        studentnames.add(30);

        //cannot print null values when using int.
        for(int e:studentnames) {
            System.out.println(e);
        }

    }

}

//10
//15
//20
//30
```

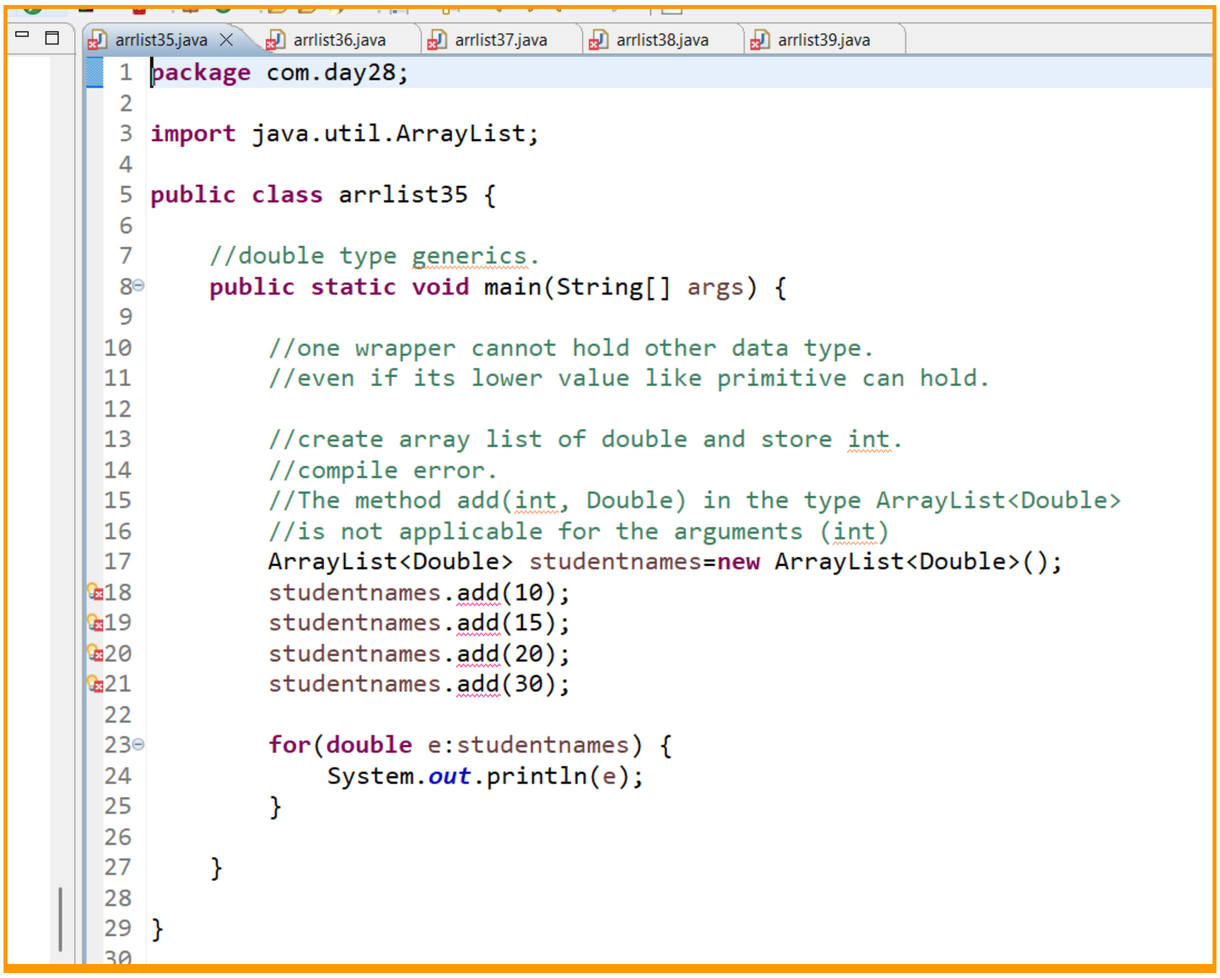
~~1,37~~

Wrapper classes can contain only the data which it can hold. (1.42)

Example in primitive double can hold int also (lower values).

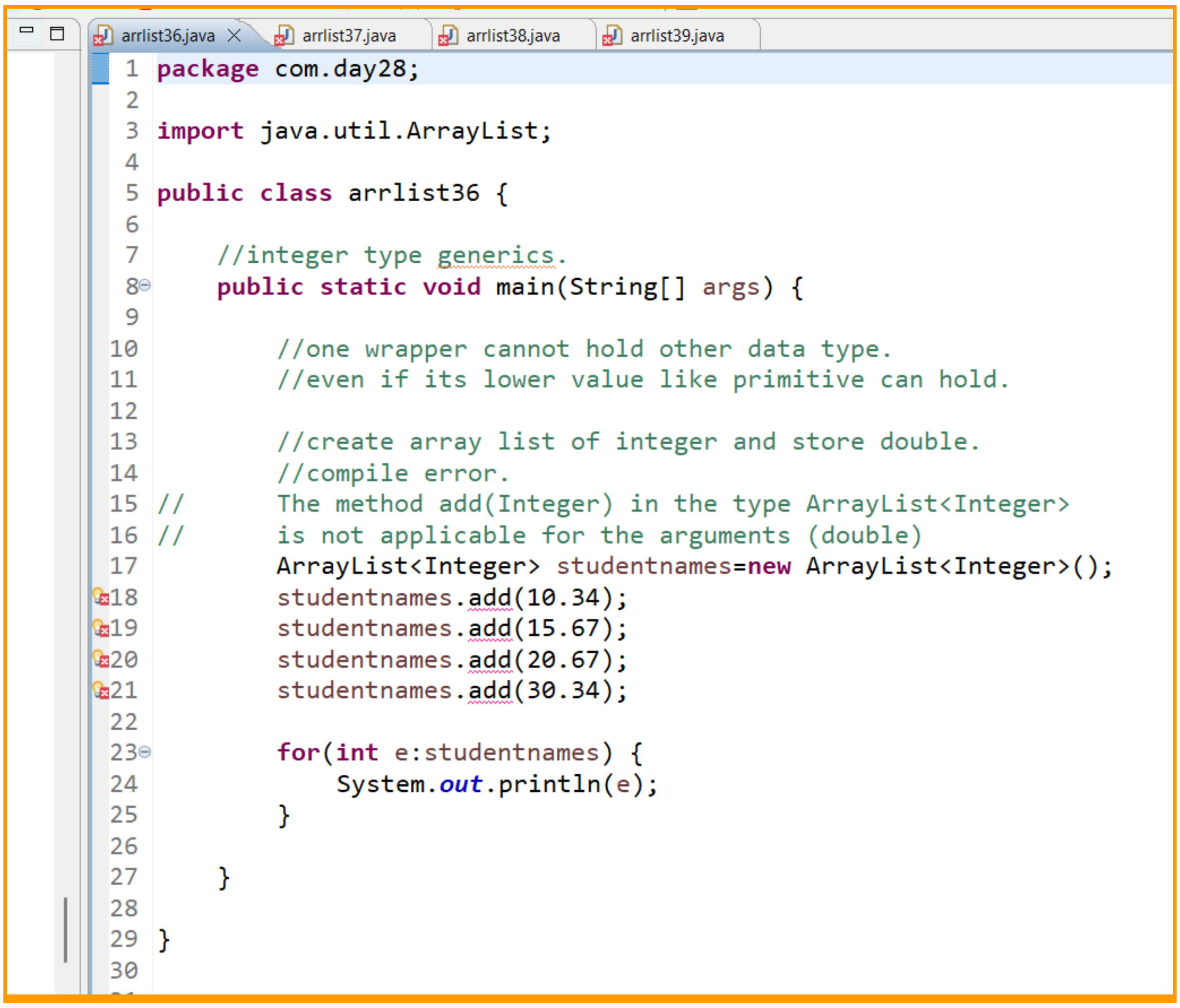
In wrapper, Double cannot hold int, compile error.

paste arrlist35



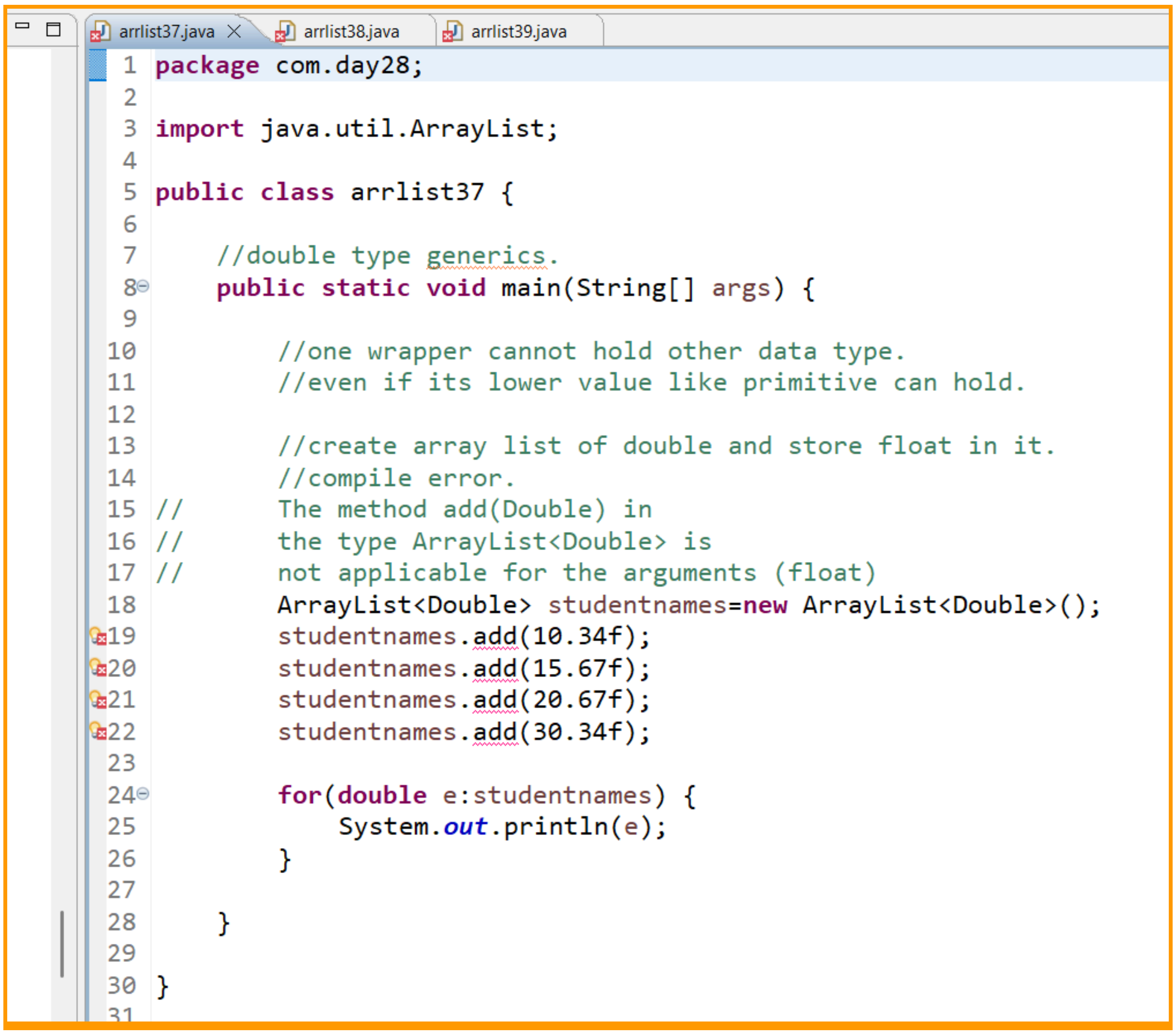
```
package com.day28;

import java.util.ArrayList;

public class arrlist35 {

    //double type generics.
    public static void main(String[] args) {

        //one wrapper cannot hold other data type.
        //even if its lower value like primitive can hold.

        //create array list of double and store int.
        //compile error.
        //The method add(int, Double) in the type ArrayList<Double>
        //is not applicable for the arguments (int)
        ArrayList<Double> studentnames=new ArrayList<Double>();
        studentnames.add(10);
        studentnames.add(15);
        studentnames.add(20);
        studentnames.add(30);

        for(double e:studentnames) {
            System.out.println(e);
        }

    }

}
```

paste arrlist36



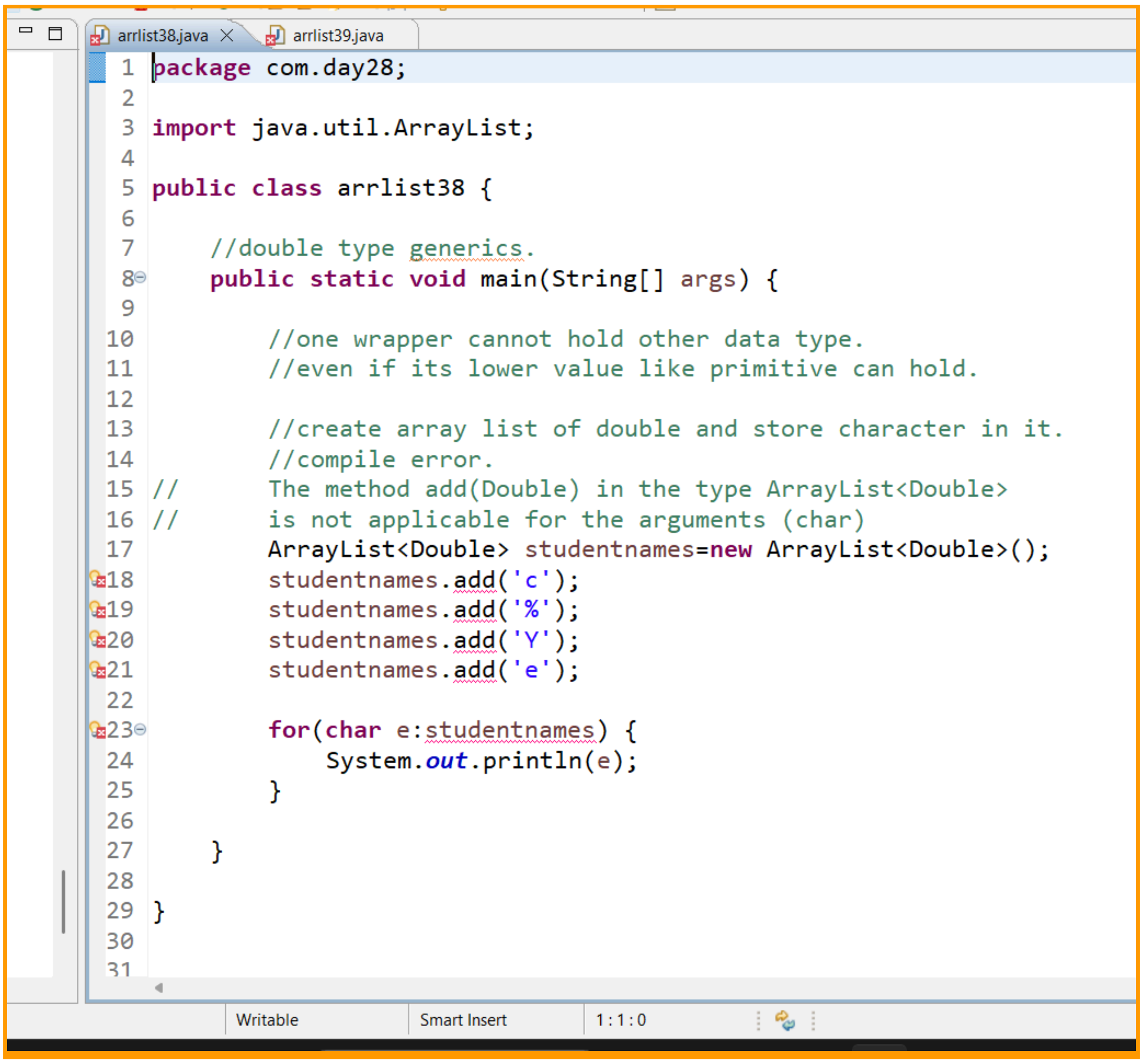
```java
package com.day28;

import java.util.ArrayList;

public class arrlist36 {

    //integer type generics.
    public static void main(String[] args) {

        //one wrapper cannot hold other data type.
        //even if its lower value like primitive can hold.

        //create array list of integer and store double.
        //compile error.
//      The method add(Integer) in the type ArrayList<Integer>
//      is not applicable for the arguments (double)
        ArrayList<Integer> studentnames=new ArrayList<Integer>();
        studentnames.add(10.34);
        studentnames.add(15.67);
        studentnames.add(20.67);
        studentnames.add(30.34);

        for(int e:studentnames) {
            System.out.println(e);
        }

    }

}
```

paste arrlist37



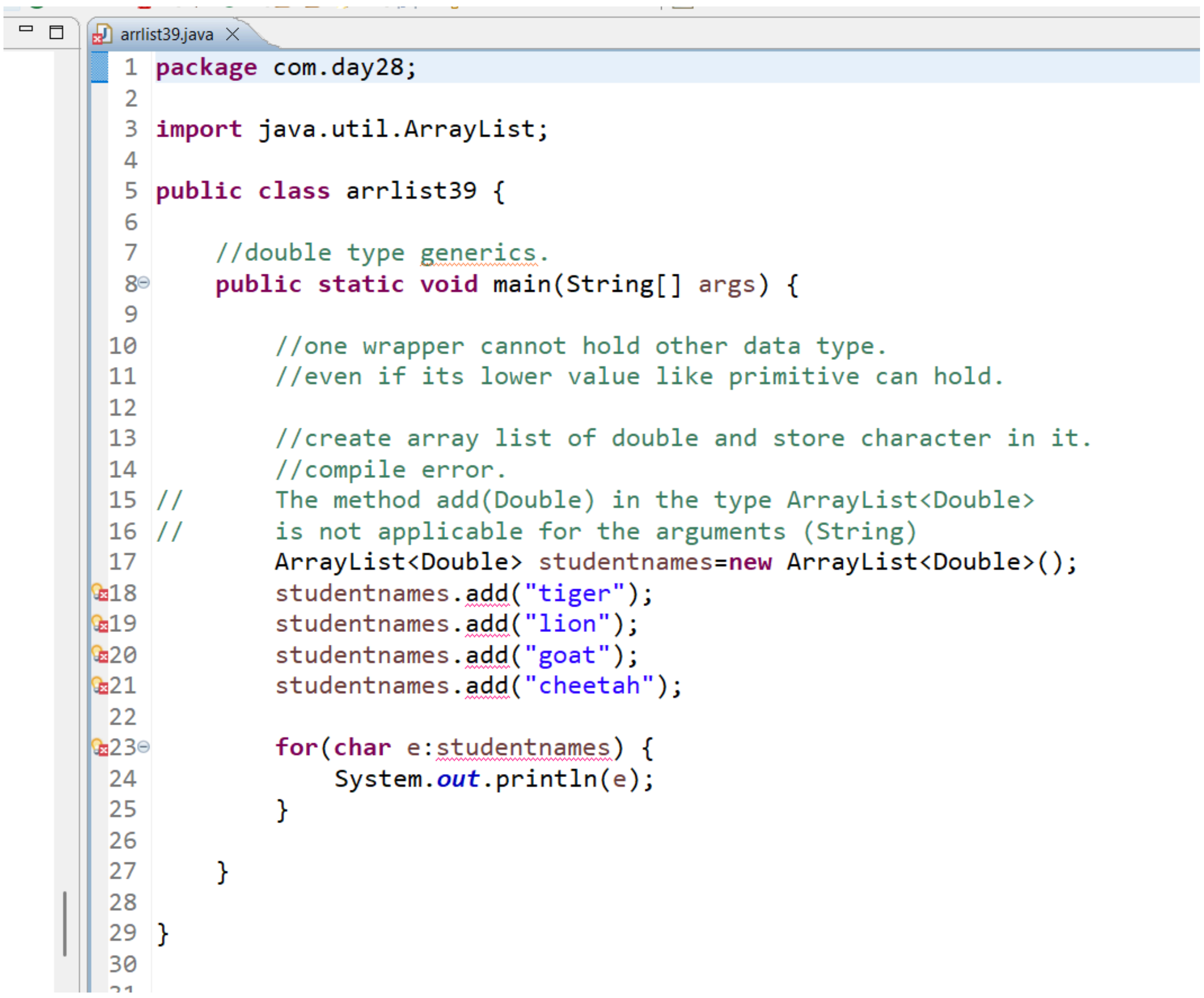
```java
package com.day28;

import java.util.ArrayList;

public class arrlist37 {

    //double type generics.
    public static void main(String[] args) {

        //one wrapper cannot hold other data type.
        //even if its lower value like primitive can hold.

        //create array list of double and store float in it.
        //compile error.
//      The method add(Double) in
//      the type ArrayList<Double> is
//      not applicable for the arguments (float)
        ArrayList<Double> studentnames=new ArrayList<Double>();
        studentnames.add(10.34f);
        studentnames.add(15.67f);
        studentnames.add(20.67f);
        studentnames.add(30.34f);

        for(double e:studentnames) {
            System.out.println(e);
        }

    }

}
```

paste arrlist38



```
package com.day28;

import java.util.ArrayList;

public class arrlist38 {

    //double type generics.
    public static void main(String[] args) {

        //one wrapper cannot hold other data type.
        //even if its lower value like primitive can hold.

        //create array list of double and store character in it.
        //compile error.
//      The method add(Double) in the type ArrayList<Double>
//      is not applicable for the arguments (char)
        ArrayList<Double> studentnames=new ArrayList<Double>();
        studentnames.add('c');
        studentnames.add('%');
        studentnames.add('Y');
        studentnames.add('e');

        for(char e:studentnames) {
            System.out.println(e);
        }

    }

}
```

paste arrlist39


arrlist39.java

```
package com.day28;

import java.util.ArrayList;

public class arrlist39 {

    //double type generics.
    public static void main(String[] args) {

        //one wrapper cannot hold other data type.
        //even if its lower value like primitive can hold.

        //create array list of double and store character in it.
        //compile error.
//      The method add(Double) in the type ArrayList<Double>
//      is not applicable for the arguments (String)
        ArrayList<Double> studentnames=new ArrayList<Double>();
        studentnames.add("tiger");
        studentnames.add("lion");
        studentnames.add("goat");
        studentnames.add("cheetah");

        for(char e:studentnames) {
            System.out.println(e);
        }

    }

}

```