

JAVA ARRAY REFRESHER

v109 by Dave Slemon

Q1. How would you explain in plain English to a friend what an array is.

Provide your friend with a good analogy.

A1. How would you explain in plain English to a friend what an array is.

**Imagine a street with houses on it.
Each house has its own unique
house number starting at "0".
That's what an array is.**

Q2a. What's another word for this?

[]

A2a. What's another word for this?

[]

Array

Array, List or Vector

Q2b. What's another word for this?

[] []

A2b. What's another word for this?

```
[ ] [ ]
```

2-dimensional Array

Q2c. How do you read this?

```
int [ ] num;
```

A2c. How do you read this?

```
int [ ] num;
```

integer array num

Q2c. Instantiate a 1000 element integer array called, **num**.

A2. Instantiate a 1000 element integer array called, **num**.

```
int [ ] num = new int[1000];
```

Q3. Name 4 primitive data types in Java. Note primitive data types all start with a lowercase letter.

A3. Name 4 primitive data types in Java. Note primitive data types all start with a lowercase letter.

- a. **int**
- b. **long**
- c. **double**
- d. **boolean**

Q4. Using a property offered with built-in arrays, output the length of array, **b**

A4. Using a property offered with built-in arrays, output the length of array, **b**

```
System.out.println( b.length );
```

Q5. Write a loop to initialize all 50 elements of the array, **b** below to: false

```
boolean[ ] b = new boolean[50];
```

A5. Write a loop to initialize all 50 elements of the array below to: false

```
boolean[ ] b = new boolean[50];
```

<pre>//standard for for (int i=0; i< 50; i++) b[i] = false;</pre>	<pre>//enhanced for for (boolean x : b) x = false;</pre>
---	---

Q6. When you instantiate a **String array** and before you populate the array with values, what value exists in each element?

A6. When you declare a **String array** and before you populate the array with values, what value exists in each element? **This is true for all objects, not just String**

null

Q7. Add the corresponding values in arrays, **a** & **b** and place their sum in array **c** for each corresponding cell.

Sample Session

	a	b	c
3	0	3	
1	4	5	
-1	6	5	

A7.

	a	b	c
3	0	3	
1	4	5	
-1	6	5	

```
for(int i=0; i < a.length; i++)
    c[i] = a[i] + b[i];
```

Q8. Consider the

String array, **s**.

Write a Java output statement which displays the following using array elements where possible

The fox chased the cat

s
ape
fox
hog
cat
dog
bat

A8a. Consider the String array, **s**. Write a Java output statement which displays the following

The fox chased the cat

```
System.out.println
```

```
("The " + s[1] + " chased the " + s[3] );
```

s
ape
fox
hog
cat
dog
bat

Q8b. Using an array property, output the number of elements in array, **s**

s
ape
fox
hog
cat
dog
bat

A8b. Using an array property, output the number of elements in array, **S**

System.out.println(s.length);

	s
ape	
fox	
hog	
cat	
dog	
bat	

Q9a. Instantiate only array, **S**

	s
21	2.7
22	3.1
23	4.8

A9a. Instantiate only array, **S**
double [] s = new double[24];

	s
21	2.7
22	3.1
23	4.8

Q9b. Write 3 assignment statements which populate s[21], s[22] & s[23]

	s
21	2.7
22	3.1
23	4.8

A9b. Write 3 assignment statements which populate s[21], s[22] & s[23]

s[21] = 2.7;
s[22] = 3.1;
s[23] = 4.8;

	s
21	2.7
22	3.1
23	4.8

Q10. Instantiate and populate array, **"a"** in as few java code lines as possible.

a
3
1
-1

A10. Instantiate and populate array, "a" in as few java code lines as possible.

```
int [] a = { 3, 1, -1 };
```

3	a
1	
-1	

Q11. Using a single variable, **temp**, switch the 3 and 4 values as shown below in the array **r**

3	r	4	r
8		8	
4		3	
BEFORE SHOT		AFTER SHOT	

A11. Using a single variable, **temp**, switch the 3 and 4 values as shown below in the

array **r**

```
int temp = r[0];
r[0] = r[2];
r[2] = temp;
```

3	r	4	r
8		8	
4		3	
BEFORE SHOT		AFTER SHOT	

Q12a Write a loop to calculate the "sum" all of the entries in the **marks** array.

	marks	sum	
0	30.2		
1	71.5		
2	80.0	count	
75	85.0		

A12a Write a loop to calculate the "sum" all of the entries in the **marks** array.

```
double sum = 0.0;
for(int i=0; i < marks.length; i++)
    sum = sum + marks[i];
Now write using an ENHANCED for
```

	marks	sum	
0	30.2		
1	71.5		
2	80.0	count	
75	85.0		

A12a Write a loop to calculate the "sum" all of the entries in the **marks** array.

```
double sum = 0.0;
for(double m : marks)
    sum = sum + m;
```

	marks	sum	
0	30.2		
1	71.5		
2	80.0	count	
75	85.0		

Q12b Write a loop to calculate “**count**” which is the number of entries where the array’s value is greater or equal to 50.0

marks		sum	count
0	30.2	<input type="text"/>	<input type="text"/>
1	71.5		
2	80.0		
75	85.0		

A12b Write a loop to calculate “**count**” which is the number of entries where the array’s value is greater or equal to 50.0

```
int count = 0;
for(int i=0; i < marks.length; i++)
    if (marks[i] >= 50.0)
        count = count + 1;
OR
for(double m : marks)
    if (m >= 50.0)
        count = count + 1;
```

marks		sum	count
0	30.2	<input type="text"/>	<input type="text"/>
1	71.5		
2	80.0		
75	85.0		

Q13 What is displayed if the following are output to the console?

a) c[2] _____

b) c[a] _____

c) c.length _____

c		a
0	2	<input type="text" value="3"/>
1	4	
2	1	
3	8	

A13 What is displayed if the following are output to the console?

a) c[2] 1

b) c[a] 8

c) c.length 4

c		a
0	2	<input type="text" value="3"/>
1	4	
2	1	
3	8	

Q13 What is displayed if the following are output to the console?

d) c[a-1] _____

e) a + c[3] _____

f) c[0] + c[3] _____

c		a
0	2	<input type="text" value="3"/>
1	4	
2	1	
3	8	

A13 What is displayed if the following are output to the console?

d) c[a-1] 1

e) a + c[3] 11

f) c[0] + c[3] 10

c		a
0	2	<input type="text" value="3"/>
1	4	
2	1	
3	8	

Q14 Instantiate and then calculate the average mark in the array of student grades. Assume the array is populated with grades.

grades	
0	70
1	75
2	90
3	55
99	66

A14

```
double[] grades = new double[100];

double sum=0.0, count = 0.0, avg = 0.0;
for(int i=0; i < grades.length; i++)
{
    sum = sum + grades[i];
    count = count + 1;
}
avg = 0.0;
if (count > 0.0)
    avg = sum / count;
```

grades	
0	70
1	75
2	90
3	55
99	66

Q15 Instantiate a 25 element **String** array called, **friends**. Also create a 25 element integer array called, **ages**. Assume that the two arrays are populated with data.

For example, Sue's age is 17

a) Write code which outputs your youngest friend(s) by name.

friends		ages	
0	Sue	0	17
1	Bill	1	18
2	Bob	2	17
3	Mary	3	16
24	Joe	24	19

A15 Write code which outputs your youngest friend(s) by name.

```
String[] friends = new String[25];
int[] ages = new int[25];

//first find the youngest age
int youngest = 999;
for(int i=0; i < ages.length; i++)
    if (ages[i] < youngest) youngest = ages[i];

//output all friends with the youngest age
for(int i=0; i < friends.length; i++)
    if (ages[i] == youngest)
        System.out.println(friends[i]);
```

friends		ages	
0	Sue	0	17
1	Bill	1	18
2	Bob	2	17
3	Mary	3	16
24	Joe	24	19

Q15 Assume that the two arrays are populated with data.
For example, Sue's age is 17

b) Output the names of your friends who are exactly 19 year old.

friends		ages	
0	Sue	0	17
1	Bill	1	18
2	Bob	2	17
3	Mary	3	16
24	Joe	24	19

A15b Output the names of your friends who are exactly 19 year old.

```
for(int i=0; i < friends.length; i++)
    if (ages[i] == 19)
        System.out.println( friends[i] );
```

friends		ages	
0	Sue	0	17
1	Bill	1	18
2	Bob	2	17
3	Mary	3	16
24	Joe	24	19

Q15c A year goes by, write code which increases the age of each friend by 1 year.

friends		ages	
0	Sue	0	17
1	Bill	1	18
2	Bob	2	17
3	Mary	3	16
24	Joe	24	19

A15c A year goes by, write code which increases the age of each friend by 1 year.

```
for(int i=0; i < ages.length; i++)
    ages[i] = ages[i] + 1;
```

friends		ages	
0	Sue	0	17
1	Bill	1	18
2	Bob	2	17
3	Mary	3	16
24	Joe	24	19

Q16 You have to store 3 colours into an array called, **colours**. The colours are #FFFFFF, #000000 and #DEDEDE.

Instantiate an array with an appropriate datatype and populate it with these 3 colours.

A16 You have to store 3 colours into an array called, colours. The colours are #FFFFFF, #000000 and #DEDEDE.

```
String [ ] colours = new String[3];
colours[0] = "#FFFFFF";
colours[1] = "#000000";
colours[2] = "#DEDEDE";
```

(there's also another better answer ... can you guess what it may be?)

A16 You have to store 3 colours into an array called, colours. There are #FFFFFF, #000000 and #DEDEDE.

```
import javafx.scene.paint.Color;
```

```
Color[ ] colours = new Color[3];
```

```
colours[0] = Color.web("#FFFFFF",1.0);
colours[1] = Color.web("#000000",1.0);
colours[2] = Color.web("#DEDEDE",1.0);
```

Note: The alpha value defines the transparency of a color and can be represented by a float value in the range [0.0,1.0] or [0,255]

```
colours[0] = Color.web("#FFFFFF",1.0);
colours[1] = Color.web("#000000",1.0);
colours[2] = Color.web("#DEDEDE",1.0);
```

Q17 a) instantiate the “a” array

(do not populate it)

	a	b	c	d	e
0	y	bob	7	1.2	false
1	x	rat	8	7.4	true
2	z	cat	1	-1.1	false
3	-	dog	2	0.0	true
4	?	ape	4	3.3	true
			7	8.8	true

A17 a) instantiate the “a” array

(do not populate it)

char [] a = new char[5];

	a	b	c	d	e
0	y	bob	7	1.2	false
1	x	rat	8	7.4	true
2	z	cat	1	-1.1	false
3	-	dog	2	0.0	true
4	?	ape	4	3.3	true
			7	8.8	true

Q17 b) instantiate and populate the “c” array in one line of java

	a	b	c	d	e
0	y	bob	7	1.2	false
1	x	rat	8	7.4	true
2	z	cat	1	-1.1	false
3	-	dog	2	0.0	true
4	?	ape	4	3.3	true
			7	8.8	true

A17 b) instantiate and populate the “c” array in one line of java

int [] c = { 7, 8, 1, 2, 4, 7 };

	a	b	c	d	e
0	y	bob	7	1.2	false
1	x	rat	8	7.4	true
2	z	cat	1	-1.1	false
3	-	dog	2	0.0	true
4	?	ape	4	3.3	true
			7	8.8	true

Q17 c) instantiate but don’t populate the “d” array

	a	b	c	d	e
0	y	bob	7	1.2	false
1	x	rat	8	7.4	true
2	z	cat	1	-1.1	false
3	-	dog	2	0.0	true
4	?	ape	4	3.3	true
			7	8.8	true

17 c) instantiate but don’t populate the “d” array

double [] d = new double[6];

	a	b	c	d	e
0	y	bob	7	1.2	false
1	x	rat	8	7.4	true
2	z	cat	1	-1.1	false
3	-	dog	2	0.0	true
4	?	ape	4	3.3	true
			7	8.8	true

Q17 d) Instantiate but don't populate the "e" array

		a	b	c	d	e
0	y	bob	7	1.2	false	
1	x	rat	8	7.4	true	
2	z	cat	1	-1.1	false	
3	-	dog	2	0.0	true	
4	?	ape	4	3.3	true	
			7	8.8	true	

A17 d) Instantiate but don't populate the "e" array

boolean [] e = new boolean[6];

		a	b	c	d	e
0	y	bob	7	1.2	false	
1	x	rat	8	7.4	true	
2	z	cat	1	-1.1	false	
3	-	dog	2	0.0	true	
4	?	ape	4	3.3	true	
			7	8.8	true	

Q17 e) what would be displayed

b[4] _____ d[2] _____

		a	b	c	d	e
0	y	bob	7	1.2	false	
1	x	rat	8	7.4	true	
2	z	cat	1	-1.1	false	
3	-	dog	2	0.0	true	
4	?	ape	4	3.3	true	
			7	8.8	true	

A17 e) what would be displayed

b[4] **ape** d[2] **-1.1**

		a	b	c	d	e
0	y	bob	7	1.2	false	
1	x	rat	8	7.4	true	
2	z	cat	1	-1.1	false	
3	-	dog	2	0.0	true	
4	?	ape	4	3.3	true	
			7	8.8	true	

Q17 f) what would be displayed

e[2] _____ d[c[2]] _____

		a	b	c	d	e
0	y	bob	7	1.2	false	
1	x	rat	8	7.4	true	
2	z	cat	1	-1.1	false	
3	-	dog	2	0.0	true	
4	?	ape	4	3.3	true	
			7	8.8	true	

A17 f) what would be displayed

e[2] **false** d[c[2]] **7.4**

		a	b	c	d	e
0	y	bob	7	1.2	false	
1	x	rat	8	7.4	true	
2	z	cat	1	-1.1	false	
3	-	dog	2	0.0	true	
4	?	ape	4	3.3	true	
			7	8.8	true	

Q18 Write a loop to subtract 1 from each element of the "c" array

		a	b	c	d	e
0	y	bob	7	1.2	false	
1	x	rat	8	7.4	true	
2	z	cat	1	-1.1	false	
3	-	dog	2	0.0	true	
4	?	ape	4	3.3	true	
			7	8.8	true	

A18 Write a loop to subtract 1 from each element of the "c" array

```
for(int i=0; i< c.length; i++)
    c[i] = c[i] - 1;
```

		a	b	c	d	e
0	y	bob	7	1.2	false	
1	x	rat	8	7.4	true	
2	z	cat	1	-1.1	false	
3	-	dog	2	0.0	true	
4	?	ape	4	3.3	true	
			7	8.8	true	

Q19 Output how many "true" values exist in the "e" array

		a	b	c	d	e
0	y	bob	7	1.2	false	
1	x	rat	8	7.4	true	
2	z	cat	1	-1.1	false	
3	-	dog	2	0.0	true	
4	?	ape	4	3.3	true	
			7	8.8	true	

A19 Output how many "true" values exist in the "e" array

```
int count = 0;
for(int i=0; i< e.length; i++)
    if ( e[i] )
        count++;

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

		a	b	c	d	e
0	y	bob	7	1.2	false	
1	x	rat	8	7.4	true	
2	z	cat	1	-1.1	false	
3	-	dog	2	0.0	true	
4	?	ape	4	3.3	true	
			7	8.8	true	

Q20 The large array called, **friends** contains a list of every student in the school. For a given student, his/her actual friend list is defined in separate individual boolean arrays named by their name.

	friends	jimHill	paulaHo
0	Jim Hill	false	true
1	Paul Snow	true	true
2	Paula Ho	false	false
3	May Jones	true	false
1000	Ron Black	false	false

Q20 For example, Paula Ho's friends are: **Jim Hill** and **Paul Snow**

	friends	jimHill	paulaHo
0	Jim Hill	false	true
1	Paul Snow	true	true
2	Paula Ho	false	false
3	May Jones	true	false
1000	Ron Black	false	false

Q20 Write a Java main program which outputs only Jim Hill's friends.

	friends	jimHill
0	Jim Hill	false
1	Paul Snow	true
2	Paula Ho	false
3	May Jones	true
1000	Ron Black	false

A20 Write a Java main program which outputs only Jim Hill's friends.

```
System.out.println("Jim Hill's friends are: ");
for(int i=0; i< friends.length; i++)
{
    if ( jimHill [ i ] ) {
        System.out.print ( friends[i] + " ";
    }
}
```

	friends	jimHill
0	Jim Hill	false
1	Paul Snow	true
2	Paula Ho	false
3	May Jones	true
1000	Ron Black	false

Q21 Begin by instantiating the **animals** and **order** arrays. Next, populate the two arrays with the exact values shown in the sketch below. The integer **order** array is populated in such a way that its values are the subscripts belonging to the **animals** array such that it puts the **animals** array in alphabetical order.

Write a for loop which prints the animal names out in alphabetical order.

	animals	order
0	cat	1
1	ape	3
2	dog	0
3	bat	2
4	rat	4

A21 Write a for loop which prints the animal names out in alphabetical order.

```
String [ ] animals = {"cat","ape","dog","bat","rat"};
```

```
int [ ] order = {1,3,0,2,4};
```

```
int index = 0;
```

```
for (int i=0; i< order.length; i++) {
    index = order[i];
    System.out.println( animals[index] );
}
```

	animals	order
0	cat	1
1	ape	3
2	dog	0
3	bat	2
4	rat	4

Q22 The array **words** contains every word in the dictionary, i.e. all 172,710 of them. Write a loop which outputs ONLY the words which end in "ed", for example, selected

	words
0	aahed
1	aahing
2	aahs
3	aal
172,709	zyzzyvas

```
String w = "";
int lastPos = 0;
```

```
for (int i=0; i < words.length; i++)
{
    w = words[i];
    lastPos = w.length() - 1;
    if (w.charAt(lastPos-1) == 'e' && w.charAt(lastPos) == 'd' )
        System.out.println(w);
}
```

	words
0	aahed
1	aahing
2	aahs
3	aal
172,709	zyzzyvas

Q23 Instantiate a 50 element integer array called, "a". Display the contents of the array backwards. (assume the array is populated). For example, if a[49] = 19 and a[48] = 7 and a[47] = 2 and a[0] = -1, then the output would look like this:

```
19
7
2
and so on
-1
```

A23 Instantiate a 50 element integer array called, "a". Display the contents of the array backwards. (assume the array is populated). For example, if a[49] = 19 and a[48] = 7 and a[47] = 2 and a[0] = -1, then the output would look like this:

```
int [ ] a = new int[50];
for(int i = a.length-1; i >= 0; i--)
    System.out.println( a[i] );
```

19
7
2
and so on
-1

Q24 Copy each letter from the secretWord into a char array. Output the char array with two spaces between each letter both forwards and backwards in lower case, then output the number of vowels, i.e. aeiou

String secretWord = "HALLOWEEN"

```
h a l l o w e e n
n e e w o l l a h
vowels = 4
```

```
char[ ] ch = new char[ secretWord.length( ) ];
int vowels = 0;
for(int c=0; c < secretWord.length( ); c++) {
    ch[c] = secretWord.toLowerCase().charAt(c);
    if (ch[c] == 'a' || ch[c] == 'e' || ch[c] == 'i' || ch[c] == 'o' || ch[c] == 'u')
        vowels++;
}
//forwards
for(int i=0; i < ch.length; i++)
    System.out.print ( ch[i] + " " );
System.out.println();
//backwards
for(int i=ch.length-1; i >= 0; i--)
    System.out.print ( ch[i] + " " );
System.out.println("Vowels = "+vowels);
```

A24

```
h a l l o w e e n
n e e w o l l a h
Vowels = 4
```

Q25 What is display if the following program is executed?

```
public static void main()
{
    int[] ch = {1,2,1,2,1,2,1,0,0};
    int amount = 0;
    for(int j = ch.length - 1; j > 0; j--)
        amount = amount + ch[j];
    System.out.println("amount: " + amount);
}
```

A25 What is display if the following program is executed?

```
public static void main()
{
    int[] ch = {1,2,1,2,1,2,1,0,0};
    int amount = 0;
    for(int j = ch.length - 1; j > 0; j--)
        amount = amount + ch[j];
    System.out.println("amount: " + amount);
}
```

amount: 9

Q26 In the array below output any value within the array that repeats? A value in the array can only repeat at most ONCE.

for example,

Original: 1 2 3 5 5 7 8 8 9 9 2

Repeats: 2 5 8 9

A26 In the array below output any value within the array that repeats? A value in the array can only repeat ONCE.

```
public static void main()
{
    int[] A = {1,2,3,5,5,7,8,8,9,9,2};

    //output the original
    System.out.print("Original: ");
    for(int i=0; i< A.length; i++)
        System.out.print("%d ", A[i]);
    System.out.println();

    System.out.print("Repeats: ");
    for(int i=0; i< A.length; i++)
        for(int j=i+1; j< A.length; j++)
            if (A[i] == A[j])
                System.out.print("%d ", A[i]);
    System.out.println();
}
```

Original: 1 2 3 5 5 7 8 8 9 9 2

Repeats: 2 5 8 9

Q27 Separate the 1s and 0s from the array below.

```
public static void main()
{
    int[] nums = {0,1,0,0,1,1,1,0,1,0};

    System.out.print("Original: ");
    for(int i=0; i< nums.length; i++)
        System.out.print("%d ", nums[i]);
    System.out.println();
}
```

sample output,

Original: 0 1 0 0 1 1 1 0 1 0

Separated: 0 0 0 0 1 1 1 1 1 1

A27 Separate the 1s and 0s from the array below.

```
public static void main() {
    int[] nums = {0,1,0,0,1,1,1,0,1,0};

    System.out.print("Original: ");
    for(int i=0; i< nums.length; i++)
        System.out.print("%d ", nums[i]);
    System.out.println();

    separate(nums);

    System.out.print("Separated: ");
    for(int i=0; i< nums.length; i++)
        System.out.print("%d ", nums[i]);
    System.out.println();
}
```

```
public static void separate(int[] nums)
{
    //count number of 0s
    int zeros = 0;
    for(int i=0; i< nums.length; i++)
        if (nums[i] == 0) zeros++;

    for(int i=0; i< zeros; i++)
        nums[i] = 0;

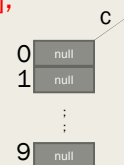
    for(int i=zeros; i< nums.length; i++)
        nums[i] = 1;
}
```

Q28 Instantiate an array called, "c" of **Circle** to 10 elements.

Draw a picture of this array.

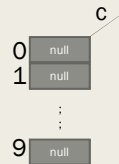
A28 Instantiate an array called, "c" of **Circle** to 10 elements. **Draw a picture of this array.**

Circle [] c = new Circle [10];



Q29 Assume the **Circle** class has a constructor which accepts the radius of the circle. Populate the array `c`, with 10 circles with a radius of 2.

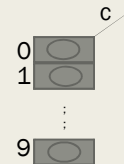
```
Circle [ ] c = new Circle [10];
```



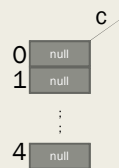
A29 Assume the **Circle** class has a constructor which accepts the radius of the circle. Populate the array `c`, with 10 circles with a radius of 2.

```
Circle [ ] c = new Circle [10];
```

```
for(int i = 0; i < c.length; i++)
{
    c[i] = new Circle(2);
}
```

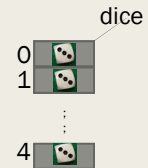


Q30 Assume the **Dice** class has a constructor which specifies the number of faces a die has. Populate the array `dice`, with 5 six-sided dice.



A30 Assume the **Dice** class has a constructor which specifies the number of faces a die has. Populate the array `dice`, with 5 six-sided dice.

```
Dice [ ] dice = new Dice [5];
for(int i = 0; i < dice.length; i++)
{
    dice[i] = new Dice(6);
}
```



Q31 Write a single Java statement to declare and create an array of 5 integers without specifying values to those integers.

A31 Write a single Java statement to declare and create an array of 5 integers without specifying values to those integers.

```
int[ ] a = new int[5];
```


Q32 Write a single Java statement to declare and create an array containing the following 3 strings, "Bat", "Bee", "Bug"

A32 Write a single Java statement to declare and create an array containing the following 3 strings, "Bat", "Bee", "Bug"

```
String[] a = { "Bat", "Bee", "Bug" };
```

Q33 Write a single Java statement to print the length of an array named, a.

A33 Write a single Java statement to print the length of an array named, a.

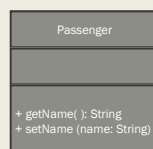
```
System.out.println (a.length);
```

Write a statement to print the 3rd element of an array named, a

Write a statement to print the 3rd element of an array named, a

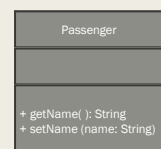
```
System.out.println ( a[2] );
```

Q34 Suppose you have a variable named `passengers` which stores an array of objects of type `Passenger`. Write a statement which calls the `getName()` method for the first `Passenger` object in the `passengers` array.

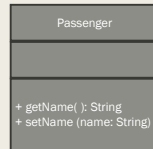


A34 Suppose you have a variable named `passengers` which stores an array of objects of type `Passenger`. Write a statement which calls the `getName()` method for the first `Passenger` object in the `passengers` array.

```
System.out.println( passengers[0].getName() );
```



Q35 Using the `passengers` variable, write an enhanced for loop to set the name of every `Passenger` object in the array to "Sally"



A35 Using the `passengers` variable, write an enhanced for loop to set the name of every `Passenger` object in the array to "Sally"

```
for ( Passenger p: passengers ) {
    p.setName("Sally");
}
```

