Creating a New Array

For creating a new array **new** keyword is used.

Syntax:

```
data_type[] array_name = new data_type[size];
```

- Replace 'size' with an integer. This denotes how many elements our array can hold.
- Replace data_type according to the type of values the array will hold. E.g. data types are int, char, etc.
- Replace array_name with the name of your array.

If we have to create an array of int with size 3 then:

```
int[] arr= new int[3]           //an integer array named arr is created with size 3
```

Example 1: Store first 5 multiples of 3.

```
1  int[] arr = new int[5];             //since we need only 5 multiples
2  int len = arr.length;
3  int num = 1;
4
5  for(int i = 0; i < len; i++)
6  {
7     arr[i] = 3 * num;
8     num++;
9  }
10
11 System.out.println("Array:");
12
13 for(int i = 0; i < len; i++)
14 {
15    System.out.println(arr[i]);
16 }
```

Output:

```
Array:
3
6
9
12
15
```

Dry Run:

Before loop	arr = {0,0,0,0,0} len = 5 num = 1		
loop variable	**num**	**Inside loop**	**Print**
i = 0	1	⇨ arr[i] = 3 * num ⇨ arr[0] = 3 * 1 ⇨ arr[0] = 3	-
i = 1	2	⇨ arr[1] = 3 * 2 ⇨ arr[1] = 6	-
i = 2	3	⇨ arr[2] = 3 * 3 ⇨ arr[2] = 9	-
i = 3	4	⇨ arr[3] = 3 * 4 ⇨ arr[3] = 12	-
i = 4	5	⇨ arr[4] = 3 * 5 ⇨ arr[4] = 15	-
i = 5	-	-	-
Outside Loop			Array: 3 6 9 12 15

Problem 1	**Give output of following code.**

```
1  int p = 4;
2  int[] ajr = new int[p];
3
4  int x = 0;
5
6  for(int i = 0; i < p; i++)
7  {
8     ajr[i] = x;
9     x = x + i;
10 }
11
12 for(int i = 0; i < p; i++)
13 {
14    System.out.println(ajr[i]);
15 }
```

Removing Elements from Array

Once the array is defined with a specific size, we cannot change its size again. To add/remove certain elements or to change the size of array you need to create a new array with updated size and copy the elements you want to retain from previous array.

Example 2: Removing first 2 elements

```
1  int[] arr = {3,1,2,9,5};
2  int len = arr.length;
3
4  int size = len - 2;                 //size after removing 2 elements
5
6  int[] arrNew = new int[size];       //new array with new size
7
8  int index = 0;                      //separate 'index' variable for new array
9
10 for(int i = 2; i < len; i++)        //starting from 2nd index value (3rd element)
11 {
12    arrNew[index] = arr[i];
13    index++;
14 }
15
16 System.out.println("New Array:");
17
18 for(int i = 0; i < arrNew.length; i++)
19 {
20    System.out.println(arrNew[i]);
21 }
```

Output:

```
New Array:
2
9
5
```

Explanation:

1. Given:

```
arr: {3,1,2,9,5}
len: 5
```

arr:

arr	3	1	2	9	5	Element
	0	1	2	3	4	Index

2. We create a new array with size len – 2 (removing 2 elements)

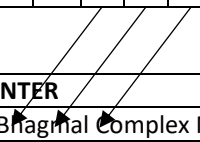
```
arrNew size = len – 2 = 5 - 2 = 3
```

arrNew:

arrNew	0	0	0	Element
	0	1	2	Index

3. We copy each element from 'arr' (from 3rd index) to 'newArr'. We leave the first 2 elements from 'arr'.

arr	3	1	2	9	5	Element
	0	1	2	3	4	Index



arrNew	2	9	5	Element
	0	1	2	Index

Dry Run:

Before loop	len = 5 size = 5 – 2 = 3 index = 0 arrNew = {0,0,0}		
loop variable	**Index**	**Inside loop**	**Print**
i = 2	0	⇨ newArr[index] = arr[i] ⇨ newArr[0] = arr[2] ⇨ newArr[0] = 2	-
i = 3	1	⇨ newArr[index] = arr[i] ⇨ newArr[1] = arr[3] ⇨ newArr[1] = 9	-
i = 4	2	⇨ newArr[index] = arr[i] ⇨ newArr[2] = arr[4] ⇨ newArr[2] = 5	-
i = 5	-	-	-
Outside Loop			New Array: 2 9 5

Problem 2	**Give output of following code.**

```
1  int[] ar1 = {10,90,50,70,30};
2  int m = ar1.length;
3
4  int n = m – 3;
5
6  int[] ar2 = new int[n];
7
8  int k = 0;
9
10 for(int i = 3; i < m; i++)
11 {
12     ar2[k] = ar1[i] + 5;
13     k++;
14 }
15
16 for(int i = 0; i < n; i++)
17 {
18     System.out.println(ar2[i]);
19 }
```

Shifting Elements to End of Array

To shift specific elements to the end array, we first create a new array of same size. We then create two index variables one pointing to front of new array and the other to the end of it. We then traverse our original array and check if the value is to be shifted then we copy this value at the index pointing to end and decrement this index. Otherwise we put the value from the front index and increment this index.

Example 3: Shifting all 3's to the end

```
1  int[] arr = {3,1,3,5,6};
2  int len = arr.length;
3  int[] arr1 = new int[len];
4  int index1 = 0;
5  int index2 = len - 1;
6
7  for(int i = 0; i < len; i++)
8  {
9     int num = arr[i];
10
11    if(num == 3)
12    {
13          arr1[index2] = 3;
14          index2--;
15    }
16    else
17    {
18          arr1[index1] = num;
19          index1++;
20    }
21 }
22
23 System.out.println("New Array:");
24 for(int i = 0; i < len; i++)
25 {
26    System.out.println(arr1[i]);
27 }
```

Output:

```
1
5
6
3
3
```

Explanation:

1. Given:

arr: {3,1,3,5,6}
len: 5

arr:

arr	3	1	3	5	6	Element
	0	1	2	3	4	Index

2. We create a new array with size len (same length)

```
arr1 size = 5
```

arr1:

arr1	0	0	0	0	0	Element
	0	1	2	3	4	Index

3. We start the algorithm:

i = 0

arr	3	1	3	5	6	Element
	0	1	2	3	4	Index
num = arr[0] = 3 Copy num at index2 index2--						
arr1	0	0	0	0	3	Element
	0	1	2	3	4	Index
	index1				index2	

i = 1

arr	3	1	3	5	6	Element
	0	1	2	3	4	Index
num = arr[1] = 1 Copy num at index1 index1++						
arr1	1	0	0	0	3	Element
	0	1	2	3	4	Index
	index1			index2		

i = 2

arr	3	1	3	5	6	Element
	0	1	2	3	4	Index
num = arr[2] = 3 Copy num at index2 index2--						
arr1	1	0	0	3	3	Element
	0	1	2	3	4	Index
		index1		index2		

i = 3

arr	3	1	3	5	6	Element
	0	1	2	3	4	Index
num = arr[3] = 5 Copy num at index1 index1++						
arr1	1	5	0	3	3	Element
	0	1	2	3	4	Index
		index1	index2			

i = 4

arr	3	1	3	5	6	Element
	0	1	2	3	4	Index
num = arr[4] = 6 Copy num at index1 index1++						
arr1	1	5	6	3	3	Element
	0	1	2	3	4	Index
			index1			

Problem 3	**Give output of following code.**

```
1  int[] ar1 = {1,5,3,2,4,9};
2  int x = ar1.length;
3
4  int[] ar2 = new int[x];
5  int a = 0;
6  int b = x - 1;
7
8  for(int i = 0; i < x; i++)
9  {
10    int m = ar1[i];
11
12    if(m <= 3)
13    {
14          ar2[b] = m;
15          b--;
16    }
17    else
18    {
19          ar2[a] = m;
20          a++;
21 }
22
23 for(int i = 0; i < x; i++)
24 {
25    System.out.println(ar2[i]);
26 }
```