

## Chapter 7 Array Worksheet

**1. How are the elements of an array accessed?**

The elements of an array are accessed using square brackets `[]` and the index value. The index is the position of an element in the array, starting from 0 for the first element. To access an element, you specify the array name and the index of the value inside the square brackets. For example, `arrayName[index]` will access the element at the specified index in the array.

**2. What is the syntax for an array initializer?**

```
arrayName = {element1, element2, ..., elementN};
```

**3. What is the syntax to declare an array reference variable?**

```
dataType[] arrayName;
```

**4. How do you create an array?**

```
dataType[] arrayName = new dataType[size];
```

**5. When is the memory allocated for an array?**

memory for an array is allocated at runtime, when the keyword `new` is used to create the array.

**6. What is the data type for an array index?**

the data type for an array index is `int`.

**7. What is the first index value in an array?**

the first index value in an array is always 0.

**8. If you declare an array `double[] list = {3.4, 2.0, 3.5, 5.5}`, the value of `list[1]` is \_\_\_\_\_.**

2.0

**9. Write Java statements that will assign all the values in an array, named `beta` with a size of 10, to 3.**

```
int[] beta = new int[10];  
Arrays.fill(beta, 3);
```

**10. What is the output of the following code?**

```
int size = 10;
int [] grades = new int [size];
size = 20;
System.out.println("Size variable = " + size);
System.out.println("The size of the grades array is = " +
grades.length);
```

*Output:*  
*Size variable = 20*  
*The size of the grades array is = 10*

**11. What is the output of the following code?**

```
public class Homework{
    public static void main(String[] args){
        int [] list = {5, 10, 15, 20};
        for(int index = 0; index < list.length; index++){
            System.out.println(list[index]);
        } // end main
    } // end class
```

*Output:*  
*5*  
*10*  
*15*  
*20*

**12. What is stored in memory for the array `list` after the following code executes?**

```
for (int i = 0; i < list.length; i++){
    list[i] = 2*i;
}
```

the `list` array will store the following elements: `[0, 2, 4, 6, 8, ...]`. The values are assigned using the expression `2 * i`, where `i` represents the index of each element in the loop.

**13. When an array is passed to a method, what is passed?**

when an array is passed to a method, a reference to the array is passed. In other words, the memory address of the array is passed to the method. This allows the method to access and modify the elements of the array directly. Changes made to the array inside the method will be reflected in the original array as well.

**14. Perform the linear search on the following numbers: 10, 15, 17, 25, 35, 41, 55. The target value is 35. Show all steps and index values for credit.**

In a linear search on the given numbers to find the target value of 35, we would iterate through the list of numbers one by one until a match is found or until the list ends.

Here are the steps and index values for each iteration:

Step 1: Start the linear search.

Index: 0

Current Value: 10

Target Value: 35

Match: No ( $10 \neq 35$ )

Step 2:

Index: 1

Current Value: 15

Target Value: 35

Match: No ( $15 \neq 35$ )

Step 3:

Index: 2

Current Value: 17

Target Value: 35

Match: No ( $17 \neq 35$ )

Step 4:

Index: 3

Current Value: 25

Target Value: 35

Match: No ( $25 \neq 35$ )

Step 5:

Index: 4

Current Value: 35

Target Value: 35

Match: Yes ( $35 = 35$ )

Target value found at index 4.

linear search stops at this point because we found the target value of 35.  
The index value of the target value is 4.

**Perform the binary search on the following numbers: 10, 15, 17, 25, 35, 41, 55. The target value is 35. Show all steps and index values for credit.**

In a binary search on the given sorted list of numbers and find the target value of 35, we divide the list in half repeatedly until we either find a match or determine that the target value is not present.

Here are the steps and index values for each iteration:

Step 1: Start the binary search.

List: [10, 15, 17, 25, 35, 41, 55]

Left Index: 0

Right Index: 6

Mid Index:  $(0 + 6) / 2 = 3$

Mid Value: 25

Target Value: 35

Match: No ( $25 < 35$ )

Step 2:

List: [35, 41, 55]

Left Index: 4

Right Index: 6

Mid Index:  $(4 + 6) / 2 = 5$

Mid Value: 41

Target Value: 35

Match: No ( $41 > 35$ )

Step 3:

List: [35]

Left Index: 4

Right Index: 4

Mid Index:  $(4 + 4) / 2 = 4$

Mid Value: 35

Target Value: 35

Match: Yes ( $35 = 35$ )

Target value found at index 4.

binary search stops at this point because we have found the target value of 35.

The index value of the target value is 4.

.