Chapter 7 Check Point Questions

Section 7.2

▼7.2.1

How do you declare an array reference variable and how do you create an array?

See the section "Array Basics."

Hide Answer

▼7.2.2

When is the memory allocated for an array?

The memory is allocated when an array is created.

Hide Answer

▼7.2.3

What is the output of the following code?

```java
int x = 30;
int[] numbers = new int[x];
x = 60;
System.out.println("x is " + x);
System.out.println("The size of numbers is " + numbers.length);
```

```
x is 60
The size of numbers is 30
```

Hide Answer

▼7.2.4

Indicate true or false for the following statements:
(a) Every element in an array has the same type.
(b) The array size is fixed after an array reference variable is declared.
(c) The array size is fixed after it is created.
(d) The elements in an array must be of a primitive data type.

(a) Answer: True
(b) Answer: False
(c) Answer: True
(d) Answer: False

Hide Answer

▼7.2.5

Which of the following statements are valid?

```
int i = new int(30);
double d[] = new double[30];
char[] r = new char(1..30);
int i[] = (3, 4, 3, 2);
float f[] = {2.3, 4.5, 6.6};
char[] c = new char();

int i = new int(30);
Answer: Invalid

double d[] = new double[30];
Answer: Valid

char[] r = new char(1..30);
Answer: Invalid

int i[] = (3, 4, 3, 2);
Answer: Invalid

float f[] = {2.3, 4.5, 5.6};
Answer: Valid

char[] c = new char();
Answer: Invalid
```

Hide Answer

▼7.2.6

How do you access elements in an array?

You access an array using its index. arrayRefVar[index] is known as an array indexed variable.

Hide Answer

▼7.2.7

What is the array index type? What is the lowest index? What is the representation of the third element in an array named a?

The array index type is int and its lowest index is 0.

```
a[2]
```

Hide Answer

▼7.2.8

Write statements to do the following:
a. Create an array to hold 10 double values.
b. Assign the value 5.5 to the last element in the array.
c. Display the sum of the first two elements.
d. Write a loop that computes the sum of all elements in the array.
e. Write a loop that finds the minimum element in the array.
f. Randomly generate an index and display the element of this index in the array.
g. Use an array initializer to create another array with the initial values 3.5, 5.5, 4.52, and 5.6.

```
(a)      double[] list = new double[10];
(b)      list[list.length - 1] = 5.5;
(c)      System.out.println(list[0] + list[1]);
(d)      double sum = 0;
         for (int i = 0; i < list.length; i++)
           sum += list[i];
(e)      double min = list[0];
         for (int i = 1; i < list.length; i++)
           if (min > list[i])  min = list[i];
(f)      System.out.println(list[(int)(Math.random() * list.length)]);
(g)      double[] list = {3.5, 5.5, 4.52, 5.6};
```

Hide Answer

▼7.2.9

What happens when your program attempts to access an array element with an invalid index?

A runtime exception ArrayIndexOutOfBounds occurs.

Hide Answer

▼7.2.10

Identify and fix the errors in the following code:

```
1  public class Test {
2    public static void main(String[] args) {
3      double[100] r;
4
5        for (int i = 0; i < r.length(); i++);
6          r(i) = Math.random * 100;
7    }
8  }
```

Line 3: the array declaration is wrong. It should be double[]. The array needs to be created before its been used. e.g. new double[10]
Line 5: The semicolon (;) at the end of the for loop heading should be removed.
Line 5: r.length() should be r.length.
Line 6: random should be random()
Line 6: r(i) should be r[i].

Hide Answer

▼7.2.11

What is the output of the following code?

```
1  public class Test {
2    public static void main(String[] args) {
3      int list[] = {1, 2, 3, 4, 5, 6};
4      for (int i = 1; i < list.length; i++)
5        list[i] = list[i - 1];
6
7      for (int i = 0; i < list.length; i++)
8        System.out.print(list[i] + " ");
9    }
10 }
```

1 1 1 1 1 1

Hide Answer

Section 7.4

▼**7.4.1**

Will the program pick four random cards if you replace lines 22-27 in Listing 7.2 DeckOfCards.java with the following code?

```java
for (int i = 0; i < 4; i++) {
  int cardNumber = (int)(Math.random() * deck.length);
  String suit = suits[cardNumber / 13];
  String rank = ranks[cardNumber % 13];
  System.out.println("Card number " + cardNumber + ": "
    + rank + " of " + suit);
}
```

No. You may have the same card appearing more than once.

Hide Answer

Section 7.5

▼**7.5.1**

Use the arraycopy method to copy the following array to a target array t:

```java
int[] source = {3, 4, 5};
```

```java
System.arraycopy(source, 0, t, 0, source.length);
```
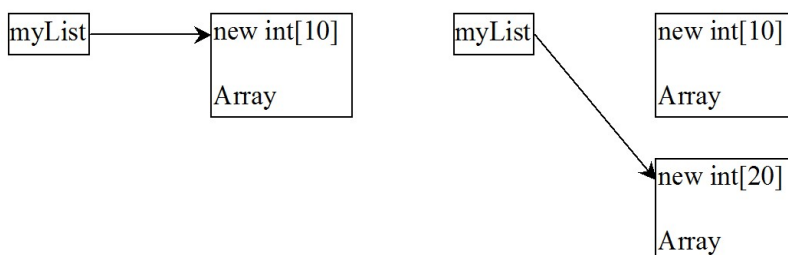
Hide Answer

▼**7.5.2**

Once an array is created, its size cannot be changed. Does the following code resize the array?

```java
int[] myList;
myList = new int[10];
// Sometime later you want to assign a new array to myList
myList = new int[20];
```

The second assignment statement myList = new int[20] creates a new array and assigns its reference to myList.

Hide Answer

Section 7.7

▼ 7.7.1

Suppose the following code is written to reverse the contents in an array, explain why it is wrong. How do you fix it?

```java
int[] list = {1, 2, 3, 5, 4};

for (int i = 0, j = list.length - 1; i < list.length; i++, j--) {
  // Swap list[i] with list[j]
  int temp = list[i];
  list[i] = list[j];
  list[j] = temp;
}
```

It swaps the elements twice. For example, the first element is swapped with the last element once and then the last element is swapped with the first element another time. In effect, the elements are not swapped. Here is the fix:

```java
int[] list = {1, 2, 3, 5, 4};

for (int i = 0, j = list.length - 1; i < list.length / 2; i++, j--) {
  // Swap list[i] with list[j]
  int temp = list[i];
  list[i] = list[j];
  list[j] = temp;
}
```

Hide Answer

Section 7.8

▼ 7.8.1

True or false? When an array is passed to a method, a new array is created and passed to the method.

False. When an array is passed to a method, the reference value of the array is passed. No new array is created. Both argument and parameter point to the same array.

Hide Answer

▼ 7.8.2

Show the output of the following two programs:

```java
(a)
public class Test {
  public static void main(String[] args) {
    int number = 0;
    int[] numbers = new int[1];

    m(number, numbers);

    System.out.println("number is " + number
      + " and numbers[0] is " + numbers[0]);
  }

  public static void m(int x, int[] y) {
    x = 3;
    y[0] = 3;
```

```
      }
    }

    (b)
    public class Test {
      public static void main(String[] args) {
        int[] list = {1, 2, 3, 4, 5};
        reverse(list);
        for (int i = 0; i < list.length; i++)
          System.out.print(list[i] + " ");
      }

      public static void reverse(int[] list) {
        int[] newList = new int[list.length];

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

        list = newList;
      }
    }
```
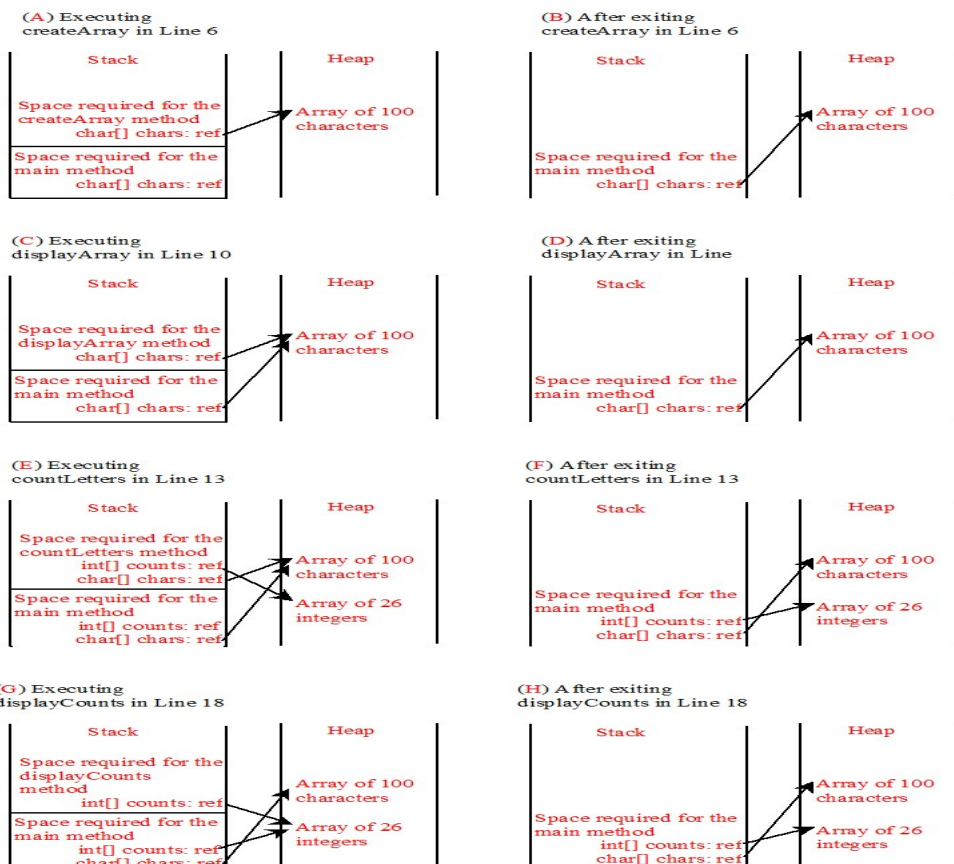
(a) number is 0 and numbers[0] is 3
(b) 1 2 3 4 5

Hide Answer

▼ 7.8.3

Where are the arrays stored during execution? Show the contents of the stack and heap during and after executing displayArray, countLetters, displayCounts in Listing 7.4.

Section 7.9

▼7.9.1

What is wrong with each of the following method headers?

```java
public static void print(String... strings, double... numbers)
public static void print(double... numbers, String name)
public static double... print(double d1, double d2)
```

Only one variable-length parameter may be specified in a method and this parameter must be the last parameter. The method return type cannot be a variable-length parameter.

▼7.9.2

Can you invoke the printMax method in Listing 7.5 using the following statements?

```java
printMax(1, 2, 2, 1, 4);
printMax(new double[]{1, 2, 3});
printMax(new int[]{1, 2, 3});
```

The last one

```java
printMax(new int[]{1, 2, 3});
```

is incorrect, because the array must of the double[] type.

Section 7.10

▼7.10.1

If high is a very large integer such as the maximum int value 2147483647, (low + high) / 2 may cause overflow. How do you fix it to avoid overflow?

Replace (low + high) / 2 with (-low + high) / 2 + low

▼7.10.2

Use Figure 7.9 as an example to show how to apply the binary search approach to a search for key 10 and key 12 in list {2, 4, 7, 10, 11, 45, 50, 59, 60, 66, 69, 70, 79}.

Omitted

▼7.10.3

If the binary search method returns -4, is the key in the list? Where should the key be inserted if you wish to insert the key into the list?

The key is not in the list. The key should be inserted at -(-4 + 1) = 3.

Section 7.11

▼7.11.1

Use Figure 7.11 as an example to show how to apply the selection-sort approach to sort {3.4, 5, 3, 3.5, 2.2, 1.9, 2}.

Omitted

▼7.11.2

How do you modify the selectionSort method in Listing 7.8 to sort numbers in decreasing order?

Omitted

Section 7.12

▼7.12.1

What types of array can be sorted using the java.util.Arrays.sort method? Does this sort method create a new array?

You can sort an array of any primitive types except boolean. The sort method is void, so it does not return a new array.

▼7.12.2

To apply java.util.Arrays.binarySearch(array, key), should the array be sorted in increasing order, in decreasing order, or neither?

To apply java.util.Arrays.binarySearch(array, key), the array must be sorted in increasing order.

▼7.12.3

Show the output of the following code:

```java
int[] list1 = {2, 4, 7, 10};
java.util.Arrays.fill(list1, 7);
System.out.println(java.util.Arrays.toString(list1));

int[] list2 = {2, 4, 7, 10};
System.out.println(java.util.Arrays.toString(list2));
System.out.print(java.util.Arrays.equals(list1, list2));
```

```
[7, 7, 7, 7]
[2, 4, 7, 10]
False
```

Section 7.13

This book text declares the main method as

```
public static void main(String[] args)
```

Can it be replaced by one of the following lines?

```
public static void main(String args[])
public static void main(String[] x)
public static void main(String x[])
static void main(String x[])
```

public static void main(String[] args)

can be replaced by

public static void main(String args[])

public static void main(String[] x)

public static void main(String x[])

but not

static void main(String x[])

because it is not public.

Hide Answer

Show the output of the following program when invoked using
1. java Test I have a dream
2. java Test "1 2 3"
3. java Test

```java
public class Test {
  public static void main(String[] args) {
    System.out.println("Number of strings is " + args.length);
    for (int i = 0; i < args.length; i++)
      System.out.println(args[i]);
  }
}
```

(1)
Number of strings is 4
I
have
a
dream

(2)
Number of strings is 1
1 2 3

(3)
Number of strings is 0

Hide Answer