

Solutions to Quick Check Questions

9

Arrays

9.1 Array Basics



Quick Check

1. Which of the following statements are invalid?

- 1 —> a. `float number[23];`
- 2 —> b. `float number = { 1.0f, 2.0f, 3.0f };`
- 3 —> c. `int number;`
`number = new Array[23];`
- 4 —> d. `int[] number = [1, 2, 3, 4];`

1. Invalid. You cannot set the size of an array at declaration. You do that when you create an array.

2. Invalid. The data type declaration must be `float[]`.

3. Invalid. `Array` is an invalid term. It should be `int[23]` for creating an array of `int`.

4. *Invalid. The parentheses are used for designating array elements as { 1, 2, 3, 4 }.*

2. Write a code fragment to compute the sum of all positive real numbers stored in the following array.

```
double[] number = new double[25];

double sum = 0;
for (int count = 0; count < number.length; count++) {
    if (number[count] > 0) {
        sum += number[count];
    }
}
```

3. Describe the difference between the following two code fragments.

```
//code fragment 1
for (int i = 0; i < number.length; i++) {
    if ( i % 2 == 0 ) {
        outputBox.println( number[i] );
    }
}

//code fragment 2
for (int i = 0; i < number.length; i++) {
    if ( number[i] % 2 == 0 ) {
        outputBox.println( number[i] );
    }
}
```

Fragment 1 prints out elements stored in the even index, i.e., number[0], number[2], number[4], and so forth.

Fragment 2 prints out even numbers in the array, regardless of the position they are stored in the array.

9.2 Arrays of Objects



Quick Check

1. Which of the following statements are invalid?

invalid → a. `Person[25] person;`
b. `Person[] person;`
c. `Person person[] = new Person[25];`
invalid → d. `Person person[25] = new Person[25];`

2. Write a code fragment to print out the name of those who are older than 20. Assume the following declaration and that the array is already set up correctly.

```
Person[ ] friend = new Person[100];

for (int count = 0; count < friend.length; count++) {
    if (friend[count].getAge() > 20) {
        System.out.println(friend[count].getName());
    }
}
```

9.3 Passing Arrays to Methods



Quick Check

1. What will be an output from the following code?

```
int[] list = {10, 20, 30, 40 };
myMethod( list );
outputBox.println( list[1] );
outputBox.println( list[3] );
...
public void myMethod(int[] intArray)
{
    for (int i = 0; i < intArray.length; i+=2) {
        intArray[i] = i;
    }
}
```

Output:

20
40

2. If we replace myMethod of question 1 with the following, what will be an output?

```
public void myMethod(int[] intArray)
{
    int[] local = intArray;
    for (int i = 0; i < local.length; i+=2) {
        local[i] = i;
    }
}
```

Output:

```
20
40
```

9.4 MultiInputBox

No Quick Check Questions.

9.5 Self-Referencing Pointer



Quick Check

1. What will be an output from the following code?

```
Tester testOne = new Tester();
testOne.methodOne(15);
...
class Tester
{
    private int x;

    public void methodTwo( )
    {
        System.out.println(x);
    }

    public void methodOne(int x)
    {
        this.x = x + 1;
        methodTwo();
    }
}
```

Output:

16

2. What's wrong with the following code?

```
public void setAge( int anAge )
{
    age = this.anAge;
}
```

The term anAge is a parameter to the method and not a data member, so the expression this.anAge is invalid.

9.6 Sample Development: The Address Book

No Quick Check questions.

9.7 Two-Dimensional Arrays



Quick Check

1. Write a code fragment to compute the average pay of the pays stored in the payScaleTable array.

```
int count = 0;
double sum = 0;
for (int row = 0; row < payScaleTable.length; row++) {
    for (int col = 0; col < payScaleTable[row].length;
        col++) {
        count++;
        sum += payScale[row][col];
    }
}
double average = sum / count;
```

2. Write a code fragment that finds the largest integer in the following two-dimensional array.

```
int[][] table = new int[10][10];
```

```

int max = table[0][0];
for (int row = 0; row < table.length; row++) {
    for (int col = 0; col < table[row].length; col++) {
        if (max < table[row][col]) {
            max = table[row][col];
        }
    }
}

```

3. What is an output from the following code.

```

int[][] table = new int[10][5];

outputBox.println(table.length);
outputBox.println(table[4].length);

```

Output:

```

10
5

```

9.8 Vectors



Quick Check

1. What is the output from the following code?

```

Vector vector = new Vector( 3, 2 );

for (int i = 0; i < 6; i++ ) {
    vector.add( "element " + i );
    System.out.println( vector.capacity( ) + "    "
                        + vector.size( ) );
}

```

Output:

```

7 6

```

2. What is the output from the following code?

```

Vector vector = new Vector( );

for (int i = 0; i < 6; i++ ) {

```

```
        vector.add( "element " + i );  
    }  
  
    vector.remove( 1 );  
    vector.remove( 3 );  
  
    System.out.println( vector.elementAt( 2 ) );  
  
}
```

Output:

```
element 3
```

