S

Single-Dimensional arrays

Chapter 7

November 4, 2019

Thomas J. Short

7.2.2 Memory gets allocated when the array is created not when it’s initialized.

7.2.3 What is the output of the following code?

int x = 30;

int[] numbers = new int[x];

x = 60;

System.out.println("x is " + x);  **x is 60**

System.out.println("The size of numbers is " + numbers.length); **The size of numbers is 30**

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

7.2.5 Which of the following statements are valid?

int i = new int(30); **Invalid**

double d[] = new double[30]; **Valid**

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

int i[] = (3, 4, 3, 2); **invalid**

float f[] = {2.3, 4.5, 6.6}; **Valid**

char[] c = new char(); **Invalid**

7.2.8 Write statements to do the following:  
a. Create an array to hold 10 double values.

**double[] playableCharacters = new double playableCharacters[10];**

b. Assign the value 5.5 to the last element in the array.

**playerHitPoints[playerHitPoints - 1] = 5.5;**

c. Display the sum of the first two elements.

**System.out.println(playerHitPoints[0] + playerHitPoints[1]);**

d. Write a loop that computes the sum of all elements in the array.

**Double[] mylist = new double {1.0,2.0,3.0};**

**double total = 0;**

**for(i = 0; i < mylist.length; i++;){**

**Total += mylist[i];**

**}**  
e. Write a loop that finds the minimum element in the array.

**double min = playerHitPoints[0];**

**for(int countMyList = 1; countMyList < playerHitPoints.length; i++ )**

**{**

**if(min > playerHitPoints[countMyList])**

**{**

**min = playerHitPoints[countMyList]**

**}**

**}**

f. Randomly generate an index and display the element of this index in the array.

**System.out.println(myList[(int)(math.random() \* myList.length)]);**g. Use an array initializer to create another array with the initial values 3.5, 5.5, 4.52, and 5.6.

**Double[] playerHealth = {3.5, 5.5, 4.52, 5.6};**

7.8.2 Show the output of the following two programs:

**(a) number is 0 and numbers[0] is 3**

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) 1 2 3 4 5**

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];

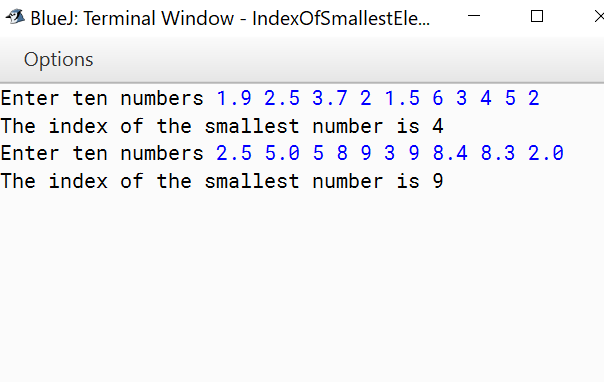
list = newList;

}

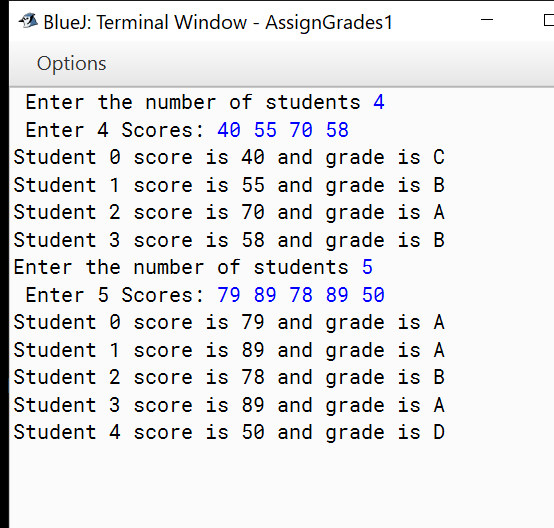
}

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.

**index of the smallest element**

****

**Assign Grade Result:**



**Print Distinct Numbers result:**

