

Quizlet

chapter 6 part1

48 terms | muk

1) Consider the following line of code:

```
int[] somearray = new int[30];
```

Which one of the following options is a valid line of code for displaying the twenty-eighth element of somearray?

- a)
System.out.println(somearray[28]);
- b)
System.out.println(somearray(28));
- c)
System.out.println(somearray(27));
- d)
System.out.println(somearray[27]);

d) System.out.println(somearray[27]);



2) Identify the correct statement for defining an integer array named numarray of ten elements.

- a) int[] numarray = new int[9];
- b) int[] numarray = new int[10];
- c) int[10] numarray;
- d) int numarray[10];

b) int[] numarray = new int[10];



3) Which one of the following

a) int[] somearray = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };



statements is a valid initialization of an array named somearray of ten elements?

- a) `int[] somearray = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };`
- b) `int somearray[] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };`
- c) `int[10] somearray = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };`
- d) `int somearray[10] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };`

4) What is the output of the following code snippet?

```
int[] myarray = { 10, 20, 30, 40, 50
};
System.out.print(myarray[2]);
System.out.print(myarray[3]);
```

- a) 1050
- b) 2030
- c) 3040
- d) 4050

c) 3040



5) What is the result of executing this code snippet?

```
int[] marks = { 90, 45, 67 };
for (int i = 0; i <= 3; i++)
{
System.out.println(marks[i]);
}
```

- a) The code snippet does not give any output.
- b) The code snippet displays all

c) The code snippet causes a bounds error.



the marks stored in the array without any redundancy.

- c) The code snippet causes a bounds error.
- d) The code snippet executes an infinite loop.

6) What is the valid range of index values for an array of size 10?

- a) 0 to 10
- b) 1 to 9
- c) 1 to 10
- d) 0 to 9

d) 0 to 9



7) Which one of the following statements is correct about the given code snippet?

```
int[] somearray = new int[6];
for (int i = 1; i < 6; i++)
{
    somearray[i] = i + 1;
}
```

- a) The for loop initializes all the elements of the array.
- b) The for loop initializes all the elements except the first element.
- c) The for loop initializes all the elements except the last element.
- d) The for loop initializes all the elements except the first and last elements

b) The for loop initializes all the elements except the first element.



8) What is the output of the

a) 2345



given code snippet?

```
int[] mynum = new int[5];  
for (int i = 1; i < 5; i++)  
{  
    mynum[i] = i + 1;  
    System.out.print(mynum[i]);  
}
```

- a) 2345
- b) 1234
- c) 1345
- d) 1111

9) Which one of the following statements is correct for the given code snippet?

```
int[] number = new int[3]; // Line  
1  
number[3] = 5; // Line 2
```

- a) Line 2 declares and initializes an array of three elements with value 5.
- b) Line 2 causes a bounds error because 3 is an invalid index number.
- c) Line 2 initializes the third element of the array with value 5.
- d) Line 2 initializes all the elements of the array with value 5.

b) Line 2 causes a bounds error because 3 is an invalid index number



10) What is displayed after executing the given code snippet?

c) The code snippet causes a bounds error.



```
int[] mymarks = new int[10];
int total = 0;
Scanner in = new
Scanner(System.in);
for (int cnt = 1; cnt <= 10; cnt++)
{
System.out.print("Enter the
marks: ");
mymarks[cnt] = in.nextInt();
total = total + mymarks[cnt];
}
System.out.println(total);
```

- a) The code snippet displays the total marks of all ten subjects.
- b) The for loop causes a run-time error on the first iteration.
- c) The code snippet causes a bounds error.
- d) The code snippet displays zero.

11) When an array myArray is only partially filled, how can the programmer keep track of the current number of elements?

- a) access myArray.length()
- b) maintain a companion variable that stores the current number of elements
- c) access myArray.currentElements()
- d) access myArray.length() - 1

b) maintain a companion variable that stores the current number of elements



12) Suppose you wish to write a

c) public static int sum(int[] values, int currSize)



method that returns the sum of the elements in the partially filled array myArray. Which is a reasonable method header?

- a) public static int sum(int[] values)
- b) public static int sum()
- c) public static int sum(int[] values, int currSize)
- d) public static int sum(int[] values, int size, int currSize)

13) Which code snippet prints out the elements in a partially filled array of integers?

- a)
for (int i = 0; i < myArray.length(); i++)
{
System.out.print(myArray[i]);
}
- b)
for (int i = 0; i < myArray.currLength(); i++)
{
System.out.print(myArray[i]);
}
- c)
for (int i = 0; i < currLength; i++)
{
System.out.print(myArray[i]);
}
- d)
for (int i = 0; i < myArray.length(); i++)
{

- c)
for (int i = 0; i < currLength; i++)
{



```
System.out.print(myArray[currLe
ngth]);
}
```

14) In a partially filled array, the number of slots in the array that are not currently used is

- a) the length of the array minus the number of elements currently in the array
- b) the number of elements currently in the array minus the length of the array
- c) the length of the array plus the number of elements currently in the array
- d) the number of elements currently in the array

a) the length of the array minus the number of elements currently in the array



15) Complete the following code snippet with the correct enhanced for loop so it iterates over the array without using an index variable.

```
String[] arr = { "abc", "def", "ghi",
"jkl" };
_____
```

```
{
System.out.print(str);
}
```

- a) for (String str : arr)
- b) for (str : String arr)
- c) for (str[] : arr)
- d) for (arr[] : str)

a) for (String str : arr)



16) Which statements are true

a) I, II



about the buffer overrun attack launched over the Internet in 1988?

- I. The buffer overrun exploited a program that was written in C running on the Unix operating system
- II. The Java programming language generates a run-time exception when buffer overrun occurs
- III. In recent years computer attacks have lessened

- a) I, II
- b) I, III
- c) II, III
- d) I, II, III

17) What is the result of the following code?

```
for (double element : values)
{
    element = 0;
}
```

- a) The elements of the array values are initialized to zero.
- b) The elements of the array element are initialized to zero.
- c) The first element of the array values is initialized to zero.
- d) The array values is not modified.

d) The array values is not modified.



18) Which statements about the

c) II, III



enhanced for loop are true?

I. It is suitable for all array algorithms

II. It does not allow the contents of the array to be modified

III. It does not require the use of an index variable

- a) I, II
- b) I, III
- c) II, III
- d) I, II, III

19) The enhanced for loop

a) is convenient for traversing all elements in an array

b) is convenient for traversing elements in a partially filled array

c) is only used for arrays of integers

d) is used to initialize all array elements to a common value

a) is convenient for traversing all elements in an array



20) When the order of the elements is unimportant, what is the most efficient way to remove an element from an array?

a) Delete the element and move each element after that one to a lower index.

b) Replace the element to be deleted with the last element in the array.

b) Replace the element to be deleted with the last element in the array.



- c) Replace the element to be deleted with the first element in the array.
- d) Replace the element with the next element

22) It may be necessary to "grow" an array when reading inputs because

- a) the number of inputs may not be known in advance.
- b) arrays in Java must be resized after every 100 elements.
- c) arrays are based on powers of two.
- d) the only alternative is a bounds exception.

a) the number of inputs may not be known in advance.



21) When an array reading and storing input runs out of space

- a) the program must be recompiled with a bigger size for the array.
- b) the array must be "grown" using the growArray method.
- c) it automatically resizes to accommodate new elements.
- d) the array must be "grown" using the new command and the copyOf method.

d) the array must be "grown" using the new command and the copyOf method.



23) The binary search is faster than the linear search, providing

- a) the size of the array is a power of two.
- b) the elements of the array are

b) the elements of the array are ordered.



ordered.

c) the elements of the array are unordered.

d) the element being searched for is actually in the array.

24) Which statements about array algorithms are true?

I. The array algorithms are building blocks for many programs that process arrays

II. Java contains ready-made array algorithms for every problem situation

III. It is inefficient to make multiple passes through an array if you can do everything in one pass

a) I, II

b) I, III

c) II, III

d) I, II, III

b) I, III



25) Suppose you wish to use an array to solve a new problem. What is the first step to take in finding a solution?

a) structure a program using methods

b) adapt a built-in array algorithm

c) decompose the problem into steps

d) assemble a test program and run it

c) decompose the problem into steps





26.) Suppose you wish to process an array of values and eliminate any potential duplicate values stored in the array. Which array algorithms might be adapted for this?

- a) find the minimum
- b) remove an element
- c) add an element
- d) calculate the sum of the elements

b) remove an element

27) Which code snippet finds the largest value in an array that is only partially full?

a)

```
double largest = values[0];
for (int i = 1; i < values.length; i++)
{
    if (values[i] > largest)
    {
        largest = values[i];
    }
}
```

b)

```
double largest = values[0];
for (int i = 1; i < values.length; i++)
{
    if (values[i] < largest)
    {
        largest = values[i];
    }
}
```

c)

```
double largest = values[0];
for (int i = 1; i < currSize; i++)
```

c)

```
double largest = values[0];
for (int i = 1; i < currSize; i++)
{
    if (values[i] > largest)
    {
        largest = values[i];
    }
}
```



```
{  
if (values[i] > largest)  
{  
largest = values[i];  
}  
}  
d)  
double largest = values[0];  
for (int i = 1; i < currSize; i++)  
{  
if (values[i] < largest)  
{  
largest = values[i];  
}  
}
```

28) When a Java program terminates and reports an exception, the error message contains which pieces of useful information?

- I. The compile and revision control history of the source code changes that caused the error
- II. The name of the exception that occurred
- III. The stack trace

- a) I, II
- b) I, III
- c) II, III
- d) I, II, III

c) II, III



29) Which code snippet calculates the sum of all the even elements in an array

a.



values?

a)

```
int sum = 0;
for (int i = 0; i < values.length; i++)
{
    if ((values[i] % 2) == 0)
    {
        sum += values[i];
    }
}
```

b)

```
int sum = 0;
for (int i = 0; i < values.length; i++)
{
    if ((values[i] % 2) == 0)
    {
        sum++;
    }
}
```

c)

```
int sum = 0;
for (int i = 0; i < values.length; i++)
{
    if ((values[i] / 2) == 0)
    {
        sum += values[i];
    }
}
```

d)

```
int sum = 0;
for (int i = 0; i < values.length; i++)
{
    if ((values[i] / 2) == 0)
    {
        sum++;
    }
}
```

```
}  
}
```

30) Which code snippet calculates the sum of all the elements in even positions in an array?

a)

```
int sum = 0;  
for (int i = 1; i < values.length;  
i+=2)  
{  
sum++;  
}
```

b)

```
int sum = 0;  
for (int i = 0; i < values.length; i++)  
{  
sum++;  
}
```

c)

```
int sum = 0;  
for (int i = 0; i < values.length; i++)  
{  
sum += values[i];  
}
```

d)

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

d)

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



31) Consider using a deck of cards as a way to visualize a shuffle algorithm. When two

c) it stays the same



cards shuffle their position, what has to happen to the size of the array holding them?

- a) it increases by one
- b) it decreases by one
- c) it stays the same
- d) it increases by two

32) Consider the telephone book as a physical object that can help understand algorithms. What kind of algorithm might be visualized using it?

- a) Sorting
- Searching
- b) Searching
- c) Finding the maximum
- d) Monte Carlo methods

b.) Searching



33) Why is the use of physical objects helpful in algorithm design?

- a) It simulates the way the computer actually implements the algorithm
- b) It is more abstract than using a pencil and paper
- c) Because the constraints on physical things are the same as the constraints on bits and bytes
- d) Many people feel it is less intimidating than drawing diagrams

d) Many people feel it is less intimidating than drawing diagrams



34) Which statement(s) about the size of a Java array, array list,

c) III



and string are true?

I. The syntax for determining the size of an array, an array list, and a string in Java is consistent among the three

II. The string uses `s.size()`, while the array list uses `a.length()`

III. The array uses `a.length`, which is not a method call

- a) I
- b) II
- c) III
- d) II, III

35) If a programmer confuses the method required for checking the length of a string and uses `size()` instead of `length()`, what will happen?

- a) The program will crash.
- b) The program will not compile.
- c) The program will run but will produce an uncertain result.
- d) The compiler will automatically correct the error.

b) The program will not compile



36) Babbage's machine for automatically producing printed tables was called

- a) the slide rule.
- b) the difference engine.
- c) the mechanical calculator.
- d) the cog wheel.

b) the difference engine.



37) Java 7 introduced enhanced

c) diamond syntax.



syntax for declaring array lists,
which is termed

- a) angle brackets.
- b) method lists.
- c) diamond syntax.
- d) symmetric slants.

38) The following statement
gets an element from position 4
in an array:

```
x = a[4];
```

What is the equivalent operation
using an array list?

- a) `x = a.get(4);`
- b) `x = a.get();`
- c) `x = a.get[4];`
- d) `x = a[4];`

a) `x = a.get(4);`



39) Which statements are true
regarding the differences
between arrays and array lists?

- I. Arrays are better if the size of
a collection will not change
- II. Array lists are more efficient
than arrays
- III. Array lists are easier to use
than arrays

- a) I, II
- b) I, III
- c) II, III
- d) I, II, III

b) I, III



40) What is the value of the

a) 1



count variable after the execution of the given code snippet?


```
ArrayList<Integer> num = new
ArrayList<Integer>();
num.add(1);
num.add(2);
num.add(1);
int count = 0;
for (int i = 0; i < num.size(); i++)
{
if (num.get(i) % 2 == 0)
{
count++;
}
}
```

- a) 1
- b) 2
- c) 0
- d) 3

41) Which one of the following code snippets accepts the integer input in an array named num1 and stores the odd integers of num1 in another array named oddnum?

a)

```
ArrayList<Integer> num1 = new
ArrayList<Integer>();
ArrayList<Integer> oddnum =
new ArrayList<Integer>();
int data;
Scanner in = new
Scanner(System.in);
```

b) 

```
ArrayList<Integer> num1 = new ArrayList<Integer>();
ArrayList<Integer> oddnum = new
ArrayList<Integer>();
int data;
Scanner in = new Scanner(System.in);
for (int i = 0; i < 10; i++)
{
data = in.nextInt();
num1.add(data);
if (num1.get(i) % 2 != 0)
{
oddnum.add(num1.get(i));
}
}
```

```
for (int i = 0; i < 10; i++)
{
    data = in.nextInt();
    numl.add(data);
    if (numl.get(i) % 2 == 0)
    {
        oddnum.add(numl.get(i));
    }
}
```

b)

```
ArrayList<Integer> numl = new
ArrayList<Integer>();
ArrayList<Integer> oddnum =
new ArrayList<Integer>();
int data;
Scanner in = new
Scanner(System.in);
for (int i = 0; i < 10; i++)
{
    data = in.nextInt();
    numl.add(data);
    if (numl.get(i) % 2 != 0)
    {
        oddnum.add(numl.get(i));
    }
}
```

c)

```
ArrayList<Integer> numl = new
ArrayList<Integer>();
ArrayList<Integer> oddnum =
new ArrayList<Integer>();
int data;
Scanner in = new
Scanner(System.in);
for (int i = 0; i < 10; i++)
{
    data = in.nextInt();
```

```
num1.add(data);
if (num1.get(i) % 2 == 0)
{
    oddnum.add(num1[i]);
}
}
d)
ArrayList<Integer> num1;
ArrayList<Integer> oddnum =
new ArrayList<Integer>();
int data;
Scanner in = new
Scanner(System.in);
for (int i = 0; i < 10; i++)
{
    data = in.nextInt();
    num1.add(data);
    if (num1[i] % 2 != 0)
    {
        oddnum.add(num1[i]);
    }
}
```

Answer: b

42) Is there any thing wrong with the following code snippet?

```
String[] data = { "abc", "def",
"ghi", "jkl" };
String searchedValue = "ghi";
int pos = 0;
boolean found = false;
while (pos < data.length)
{
    if
(data[pos].equals(searchedValu
```

d) There is a logic error.



```
e))
{
found = true;
}
else
{
found = false;
}
pos++;
}
if (found)
{
System.out.println("Found at
position: " + pos);
}
else
{
System.out.println("Not found");
}
```

- a) There is nothing wrong.
- b) There is compile-time error.
- c) There is a bounds error.
- d) There is a logic error.

43) Consider the following code snippet:

```
String[] data = { "abc", "def",
"ghi", "jkl" };
String [] data2;
```

In Java 6 and later, which statement copies the data array to the data2 array?

- a) `data2 = Arrays.copyOf(data, data2.length);`

b) `data2 = Arrays.copyOf(data, data.length);`



- b) `data2 = Arrays.copyOf(data, data.length);`
- c) `data2 = Arrays.copyOf(data, data.size());`
- d) `data2 = Arrays.copyOf(data);`

44) Consider the following code snippet in Java 6 or later:

```
String[] data = { "abc", "def",  
"ghi", "jkl" };  
String[] data2 =  
Arrays.copyOf(data, data.length  
- 1);
```

What does the last element of `data2` contain?

- a) "abc"
- b) "def"
- c) "ghi"
- d) "jkl"

c) "ghi"



45) Consider the following code snippet:

```
String[] data = { "abc", "def",  
"ghi", "jkl" };
```

Using Java 6 or later, which statement grows the `data` array to twice its size?

- a) `data = Arrays.copyOf(data, 2 * data.length);`
- b) `data = Arrays.copyOf(data, 2);`
- c) `data = Arrays.copyOf(data, 2 * data);`
- d) `data = Arrays.copyOf(data, 2 * data.size());`

a) `data = Arrays.copyOf(data, 2 * data.length);`



46) Consider the following

b) `Arrays.sort(data);`



code snippet:

```
String[] data = { "123", "ghi", "jkl",  
"def", "%&*" };
```

Which statement sorts the data array in ascending order?

- a) data = Arrays.sort();
- b) Arrays.sort(data);
- c) data = Arrays.sort(data.length);
- d) data = Arrays.sort(data);

47) Consider the following method:

```
public static int mystery(int  
length, int n)  
{  
int[] result = new int[length];  
for (int i = 0; i < result.length; i++)  
{  
result[i] = (int) (n *  
Math.random());  
}  
return result;  
}
```

Which statement is correct about the code?

- a) The method works perfectly
- b) The method returns a random number
- c) The method return type should be changed to int[]
- d) The method has a bounds

c) The method return type should be changed to int[]



error when the array size is too large

Answer: c

49) Which one of the following is the correct header for a method named arrMeth that is called like this:

arrMeth(intArray); // intArray is an integer array of size 3

- a) public static void arrMeth(int[] ar, int n)
- b) public static void arrMeth(r[], n)
- c) public static void arrMeth(int n, int[] ar)
- d) public static void arrMeth(int[] ar)

d) public static void arrMeth(int[] ar)

