

ARRAY LIST

1. Which of the following correctly initializes an array arr to contain four elements each with value 0?

I int[] arr = {0, 0, 0, 0};
II int[] arr = new int[4];
III int[] arr = new int[4];
 for (int i = 0; i < arr.length; i++)
 arr[i] = 0;

- (A) I only
(B) III only
(C) I and III only
(D) II and III only
(E) I, II, and III

2. The following program segment is intended to find the index of the first negative integer in arr[0] ... arr[N-1], where arr is an array of N integers.

```
int i = 0;
while (arr[i] >= 0)
{
    i++;
}
location = i;
```

This segment will work as intended

- (A) always.
(B) never.
(C) whenever arr contains at least one negative integer.
(D) whenever arr contains at least one nonnegative integer.
(E) whenever arr contains no negative integers.

3. Refer to the following code segment. You may assume that arr is an array of values.

```
int sum = arr[0], i = 0;
while (i < arr.length)
{
    i++;
    sum += arr[i];
}
```

Which of the following will be the result of executing the segment?

- (A) Sum of arr[0], arr[1], ..., arr[arr.length-1] will be stored in sum.
(B) Sum of arr[1], arr[2], ..., arr[arr.length-1] will be stored in sum.
(C) Sum of arr[0], arr[1], ..., arr[arr.length] will be stored in sum.
(D) An infinite loop will occur.
(E) A run-time error will occur.

4. The following code fragment is intended to find the smallest value in arr[0] ... arr[n-1].

```
//Precondition: arr[0]...arr[n-1] initialized with integers.
//                                         arr is an array, arr.length = n.
//Postcondition: min = smallest value in arr[0]...arr[n-1].
int min = arr[0];
int i = 1;
while (i < n)
{
    i++;
    if (arr[i] < min)
        min = arr[i];
}
```

This code is incorrect. For the segment to work as intended, which of the following modifications could be made?

- I Change the line

int i = 1; 

to

int i = 0;

Make no other changes.

- II Change the body of the while loop to

```
{
    if (arr[i] < min)
        min = arr[i];
    i++;
}
```



Make no other changes.

- III Change the test for the while loop as follows:

while (i <= n) 

Make no other changes.

I only

II only

III only

IV only

V only

VI only

VII only

VIII only

IX only

X only

XI only

XII only

XIII only

XIV only

XV only

XVI only

XVII only

XVIII only

XVIX only

XVII only

5. Refer to the following code segment. You may assume that array arr1 contains elements arr1[0], arr1[1], ..., arr1[N-1], where N = arr1.length.

```

int count = 0;
for (int i = 0; i < N; i++)
    if (arr1[i] != 0)
    {
        arr1[count] = arr1[i];
        count++;
    }
int[] arr2 = new int[count];
for (int i = 0; i < count; i++)
    arr2[i] = arr1[i];

```

If array arr1 initially contains the elements 0, 6, 0, 4, 0, 0, 2 in this order, what will arr2 contain after execution of the code segment?

- (A) 6, 4, 2
 (B) 0, 0, 0, 0, 6, 4, 2
 (C) 6, 4, 2, 4, 0, 0, 2
 (D) 0, 6, 0, 4, 0, 0, 2
 (E) 6, 4, 2, 0, 0, 0, 0

6. Consider this program segment:

```

for (int i = 2; i <= k; i++)
    if (arr[i] < someValue)
        System.out.print("SMALL");

```

What is the maximum number of times that SMALL can be printed?

- (A) 0
 (B) 1
 (C) k - 1
 (D) k - 2
 (E) k

7. What will be output from the following code segment, assuming it is in the same class as the `doSomething` method?

```
int[] arr = {1, 2, 3, 4};  
doSomething(arr);  
System.out.print(arr[1] + " ");  
System.out.print(arr[3]);  
...  
public void doSomething(int[] list)  
{  
    int[] b = list;  
    for (int i = 0; i < b.length; i++)  
        b[i] = i;  
}
```

- (A) 0 0
(B) 2 4
 (C) 1 3
(D) 0 2
(E) 0 3

8. Consider writing a program that reads the lines of any text file into a sequential list of lines. Which of the following is a good reason to implement the list with an `ArrayList` of `String` objects rather than an array of `String` objects?

- (A) The `get` and `set` methods of `ArrayList` are more convenient than the `[]` notation for arrays.
(B) The `size` method of `ArrayList` provides instant access to the length of the list.
(C) An `ArrayList` can contain objects of any type, which leads to greater generality.
 (D) If any particular text file is unexpectedly long, the `ArrayList` will automatically be resized. The array, by contrast, may go out of bounds.
(E) The `String` methods are easier to use with an `ArrayList` than with an array.

9. Consider writing a program that produces statistics for long lists of numerical data. Which of the following is the best reason to implement each list with an array of `int` (or `double`), rather than an `ArrayList` of `Integer` (or `Double`) objects?

- (A) An array of primitive number types is more efficient to manipulate than an `ArrayList` of wrapper objects that contain numbers.
(B) Insertion of new elements into a list is easier to code for an array than for an `ArrayList`.
(C) Removal of elements from a list is easier to code for an array than for an `ArrayList`.
(D) Accessing individual elements in the middle of a list is easier for an array than for an `ArrayList`.
(E) Accessing all the elements is more efficient in an array than in an `ArrayList`.

Multiple-Choice Questions on Arrays and Array Lists

10. A client method has this declaration, followed by code to initialize the list:

```
Address[] list = new Address[100];
```

Here is a code segment to generate a list of *names only*.

```
for (Address a : list)  
/* line of code */
```

Which is a correct /* line of code */?

- (A) System.out.println(Address[i].getName());
- (B) System.out.println(list[i].getName());
- (C) System.out.println(a[i].getName());
- (D) System.out.println(a.getName());
- (E) System.out.println(list.getName());

11. The following code segment is to print out a list of addresses:

```
for (Address addr : list)  
{  
    /* more code */  
}
```

Which is a correct replacement for /* more code */?

I System.out.println(list[i].getName());
System.out.println(list[i].getStreet());
System.out.print(list[i].getCity() + ", ");
System.out.print(list[i].getState() + " ");
System.out.println(list[i].getZip());

II System.out.println(addr.getName());
System.out.println(addr.getStreet());
System.out.print(addr.getCity() + ", ");
System.out.print(addr.getState() + " ");
System.out.println(addr.getZip());

III System.out.println(addr);

- (A) I only
- (B) II only
- (C) III only
- (D) I and II only
- (E) I, II, and III

12. A client method has this declaration:

```
Student[] allStudents = new Student[NUM_STUDS]; //NUM_STUDS is  
//an int constant
```

Here is a code segment to generate a list of Student names only. (You may assume
that allStudents has been initialized.)

```
for (Student student : allStudents)  
/* code to print list of names */
```

Which is a correct replacement for /* code to print list of names */?

- (A) System.out.println(allStudents.getName());
- (B) System.out.println(student.getName());
- (C) System.out.println(student.getAddress().getName());
- (D) System.out.println(allStudents.getAddress().getName());
- (E) System.out.println(student[i].getAddress().getName());

13. Here is a method that locates the Student with the highest idNum:

```
//Precondition: Array stuArr of Student is initialized.  
//Postcondition: Student with highest idNum has been returned.  
public static Student locate(Student[] stuArr)  
{  
    /* method body */  
}
```

Which of the following could replace */* method body */* so that the method works as intended?

```
I int max = stuArr[0].getIdNum();  
for (Student student : stuArr)  
    if (student.getIdNum() > max)  
    {  
        max = student.getIdNum();  
        return student;  
    }  
return stuArr[0];
```



```
II Student highestSoFar = stuArr[0];  
int max = stuArr[0].getIdNum();  
for (Student student : stuArr)  
    if(student.getIdNum() > max)  
    {  
        max = student.getIdNum();  
        highestSoFar = student;  
    }  
return highestSoFar;
```



```
III int maxPos = 0;  
for(int i = 1; i < stuArr.length; i++)  
    if(stuArr[i].getIdNum() > stuArr[maxPos].getIdNum())  
        maxPos = i;  
return stuArr[maxPos];
```

- (A) I only
- (B) II only
- (C) III only
- (D) I and III only
- (E) II and III only

14. Which of the following correctly replaces */* more code */* in the Transaction constructor to initialize the tickList array?

- (A) `tickList[i] = new Ticket(getRow(), getSeat(), getPrice());`
- (B) `tickList[i] = new Ticket(row, seat, price);`
- (C) `tickList[i] = new tickList(getRow(), getSeat(), getPrice());`
- (D) `tickList[i] = new tickList(row, seat, price);`
- (E) `tickList[i] = new tickList(numTicks);`

15. Which represents correct */* code to calculate amount */* in the totalPaid method?

- (A) `for (Ticket t : tickList)
 total += t.myPrice;`
- (B) `for (Ticket t : tickList)
 total += tickList.getPrice();`
- (C) `for (Ticket t : tickList)
 total += t.getPrice();`
- (D) `Transaction T;
for (Ticket t : T)
 total += t.getPrice();`
- (E) `Transaction T;
for (Ticket t : T)
 total += t.myPrice;`

16. Suppose it is necessary to keep a list of all ticket transactions. A suitable declaration would be

- (A) `Transaction[] listofSales = new Transaction[NUMSALES];`
- (B) `Transaction[] listofSales = new Ticket[NUMSALES];`
- (C) `Ticket[] listofSales = new Transaction[NUMSALES];`
- (D) `Ticket[] listofSales = new Ticket[NUMSALES];`
- (E) `Transaction[] Ticket = new listofSales[NUMSALES];`

17. Refer to method `match` below:

```

//Precondition: v[0]..v[N-1] and w[0]..w[M-1] initialized with
//               integers. v[0] < v[1] < .. < v[N-1] and
//               w[0] < w[1] < .. < w[M-1].
//Postcondition: Returns true if there is an integer k that occurs
//                in both arrays, otherwise returns false.
//public static boolean match(int[] v, int[] w, int N, int M)
{
    int vIndex = 0, wIndex = 0;
    while (vIndex < N && wIndex < M)
    {
        if (v[vIndex] == w[wIndex])
            return true;
        else if (v[vIndex] < w[wIndex])
            vIndex++;
        else
            wIndex++;
    }
    return false;
}

```

Assuming that the method has not been exited, which assertion is true at the end of every execution of the `while` loop?

- A) v[0]..v[vIndex-1] and w[0]..w[wIndex-1] contain no common value,
vIndex \leq N and wIndex \leq M.
- B) v[0]..v[vIndex] and w[0]..w[wIndex] contain no common value,
vIndex \leq N and wIndex \leq M.
- C) v[0]..v[vIndex-1] and w[0]..w[wIndex-1] contain no common value,
vIndex \leq N-1 and wIndex \leq M-1.
- D) v[0]..v[vIndex] and w[0]..w[wIndex] contain no common value,
vIndex \leq N-1 and wIndex \leq M-1.
- E) v[0]..v[N-1] and w[0]..w[M-1] contain no common value,
vIndex \leq N and wIndex \leq M.

18. Consider this class:

```

public class Book
{
    private String myTitle;
    private String myAuthor;
    private boolean myCheckoutStatus;

    //constructor
    public Book(String title, String author)
    {
        myTitle = title;
        myAuthor = author;
        myCheckoutStatus = false;
    }

    //Change checkout status.
    public void changeStatus()
    { myCheckoutStatus = !myCheckoutStatus; }

    //other methods not shown ...
}

```

A client program has this declaration:

```
Book[] bookList = new Book[SOME_NUMBER];
```

Suppose bookList is initialized so that each Book in the list has a title, author, and checkout status. The following piece of code is written, whose intent is to change the checkout status of each book in bookList.

```
for (Book b : bookList)
    b.changeStatus();
```

Which is *true* about this code?

- (A) The bookList array will remain unchanged after execution.
- (B) Each book in the bookList array will have its checkout status changed, as intended.
- (C) A NullPointerException may occur.
- (D) A run-time error will occur because it is not possible to modify objects using the for-each loop.
- (E) A logic error will occur because it is not possible to modify objects in an array without accessing the indexes of the objects.

Consider this class for Questions 19 and 20:

```
public class BingoCard
{
    private int[] myCard;
    /* Default constructor: Creates BingoCard with
     * 20 random digits in the range 1 - 90. */
    public BingoCard()
    { /* implementation not shown */ }

    /* Display BingoCard. */
    public void display()
    { /* implementation not shown */ }
    ...
}
```

A program that simulates a bingo game declares an array of BingoCard. The array has NUMPLAYERS elements, where each element represents the card of a different player. Here is a code segment that creates all the bingo cards in the game:

```
/* declare array of BingoCard */
/* construct each BingoCard */
```

19. Which of the following is a correct replacement for

```
/* declare array of BingoCard */?
```

- (A) int[] BingoCard = new BingoCard[NUMPLAYERS];
- (B) BingoCard[] players = new int[NUMPLAYERS];
- (C) BingoCard[] players = new BingoCard[20];
- (D) BingoCard[] players = new BingoCard[NUMPLAYERS];
- (E) int[] players = new BingoCard[NUMPLAYERS];

20. Assuming that players has been declared as an array of BingoCard, which of the following is a correct replacement for

```
/* construct each BingoCard */
```

I for (BingoCard card : players)
 card = new BingoCard();

II for (BingoCard card : players)
 players(card) = new BingoCard();

III for (int i = 0; i < players.length; i++)
 players[i] = new BingoCard();

- (A) I only
- (B) II only
- (C) III only
- (D) I and III only
- (E) I, II, and III

21. Wh

I

II

(A)

(C)

(D)

(E)

22. Con

L:

S1

II

Whi

(A)

(B)

(C)

(D)

(E)

23. Let 1:

Whic

that e

(A) t

(B) 1

(C) 1

(D) 0

(E) 0

Multiple-Choice

11. Which declaration will cause an error?

- I List<String> stringList = new ArrayList<String>();
- II List<int> intList = new ArrayList<int>();
- III ArrayList<Comparable> compList = new ArrayList<Comparable>();

- (A) I only
- II only
- (C) III only
- (D) I and III only
- (E) II and III only

12. Consider these declarations:

```
List<String> strList = new ArrayList<String>();
String ch = " ";
Integer intOb = new Integer(5);
```

array
layer.

Which statement will cause an error?

- (A) strList.add(ch);
- (B) strList.add(new String("handy andy"));
- (C) strList.add(intOb.toString());
- (D) strList.add(ch + 8);
- (E) strList.add(intOb + 8);

13. Let list be an ArrayList<Integer> containing these elements:

2 5 7 6 0 1

Which of the following statements would *not* cause an error to occur? Assume that each statement applies to the given list, independent of the other statements.

- (A) Object ob = list.get(6);
- (B) Integer intOb = list.add(3.4);
- (C) list.add(6, 9);
- (D) Object x = list.remove(6);
- (E) Object y = list.set(6, 8);

24. Refer to method `insert` below:

```

    /* Precondition: List list is an ArrayList that contains
     * Comparable values sorted in decreasing order.
     * Postcondition: Element inserted in its correct position
     * in list. */
    public void insert(List<Comparable> list, Comparable element)
    {
        int index = 0;
        while (element.compareTo(list.get(index)) < 0)
            index++;
        list.add(index, element);
    }

```

- Assuming that the type of `element` is compatible with the objects in the list, which is a *true* statement about the `insert` method?
- (A) It works as intended for all values of `element`.
 - (B) It fails for all values of `element`.
 - (C) It fails if `element` is greater than the first item in `list` and works in all other cases.
 - (D) It fails if `element` is smaller than the last item in `list` and works in all other cases.
 - (E) It fails if `element` is either greater than the first item or smaller than the last item in `list` and works in all other cases.

25. Consider the following code segment, applied to `list`, an `ArrayList` of `Integer` values.

```

int len = list.size();
for (int i = 0; i < len; i++)
{
    list.add(i + 1, new Integer(i));
    Object x = list.set(i, new Integer(i + 2));
}

```

If `list` is initially 6 1 8, what will it be following execution of the code segment?

- (A) 2 3 4 2 1 8
- (B) 2 3 4 6 2 2 0 1 8
- (C) 2 3 4 0 1 2
- (D) 2 3 4 6 1 8
- (E) 2 3 3 2

7 3 4 2 8