

Computer Science Practice Exam #2Multiple Choice

1. Assume that an array of integer values has been declared as follows and has been initialized.

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

Which of the following code segments correctly interchanges the value of `arr[0]` and `arr[5]`?

- A.

```
arr[0] = 5;  
arr[5] = 0;
```
 - B.

```
arr[0] = arr[5];  
arr[5] = arr[0];
```
 - C.

```
int k = arr[5];  
arr[0] = arr[5];  
arr[5] = k;
```
 - D.

```
int k = arr[0];  
arr[0] = arr[5];  
arr[5] = k;
```
 - E.

```
int k = arr[5];  
arr[5] = arr[0];  
arr[0] = arr[5];
```
-

2. Consider the following code segment.

```
List<String> items = new ArrayList<String>();  
items.add("A");  
items.add("B");  
items.add("C");  
items.add(0, "D");  
items.remove(3);  
items.add(0, "E");  
System.out.println(items);
```

What is printed as a result of executing the code segment?

- A. [A, B, C, E]
- B. [A, B, D, E]
- C. [E, D, A, B]
- D. [E, D, A, C]
- E. [E, D, C, B]

Questions 3 - 4 refer to the following code segment.

```
int k = a random number such that  $1 \leq k \leq n$ ;  
  
for(int p = 2; p <=k; p++)  
    for(int r = 1; r < k; r++)  
        System.out.println("Hello");
```

3. What is the minimum number of times that Hello will be printed?

- A. 0
- B. 1
- C. 2
- D. $n-1$
- E. $n-2$

4. What is the maximum number of times that Hello will be printed?

- A. 2
 - B. $n-1$
 - C. $n-2$
 - D. $(n-1)^2$
 - E. n^2
-

5. Consider the following instance variable and incomplete method. The method `calcTotal` is intended to return the sum of all values in `vals`.

```
private int[] vals;

public int calcTotal()
{
    int total = 0;

    /* missing code */

    return total;
}
```

Which of the following code segments shown below can be used to replace `/* missing code */` so that `calcTotal` will work as intended?

- I.

```
for(int pos = 0; pos < vals.length; pos++)
{
    total += vals[pos];
}
```
- II.

```
for(int pos = vals.length; pos > 0; pos--)
{
    total += vals[pos];
}
```
- III.

```
int pos = 0;
while(pos < vals.length)
{
    total += vals[pos];
    pos++;
}
```

- A. I only
 - B. II only
 - C. III only
 - D. I and III
 - E. II and III
-

6. Consider the following method that is intended to modify its parameter `nameList` by replacing all occurrences of `name` with `newValue`.

```
public void replace (ArrayList<String> nameList,  
                   String name, String newValue)  
{  
    for(int j = 0; j < nameList.size(); j++)  
    {  
        if( /* expression */ )  
        {  
            nameList.set(j, newValue);  
        }  
    }  
}
```

Which of the following can be used to replace `/* expression */` so that `replace` will work as intended?

- A. `nameList.get(j).equals(name)`
 - B. `nameList.get(j) == name`
 - C. `nameList.remove(name)`
 - D. `nameList[j] == name`
 - E. `nameList[j].equals(name)`
-

7. Consider the following class declaration.

```
public class SomeClass
{
    private int num;

    public SomeClass(int n)
    {
        num = n;
    }

    public void increment(int more)
    {
        num = num + more;
    }

    public int getNum()
    {
        return num;
    }
}
```

The following code segment appears in another class.

```
SomeClass one = new SomeClass(100);
SomeClass two = new SomeClass(100);
SomeClass three = one;

one.increment(200);

System.out.println(one.getNum() + " " +
                    two.getNum() + " " +
                    three.getNum());
```

What is printed as a result of executing the code segment?

- A. 100 100 100
 - B. 300 100 100
 - C. 300 100 300
 - D. 300 300 100
 - E. 300 300 300
-

8. The following incomplete method is intended to sort its array parameter `arr` in increasing order.

```
// postcondition: arr is sorted in increasing order
public static void sortArray(int[] arr)
{
    int j, k;

    for(j = arr.length - 1; j > 0; j--)
    {
        int pos = j;

        for(/* missing code */)
        {
            if (arr[k] > arr[pos])
            {
                pos = k;
            }
        }
        swap(arr, j, pos);
    }
}
```

Assume that `swap(arr, j, pos)` exchanges the values of `arr[j]` and `arr[pos]`. Which of the following could be used to replace `/* missing code */` so that executing the code segment sorts the values in array `arr`?

- A. `k = j - 1; k > 0; k--`
 - B. `k = j - 1; k >= 0; k--`
 - C. `k = 1; k < arr.length; k++`
 - D. `k = 1; k > arr.length; k++`
 - E. `k = 0; k <= arr.length; k++`
-

9. Assume that the following variable declarations have been made.

```
double d = Math.random();
double r;
```

Which of the following assigns a value to `r` from the uniform distribution over the range $0.5 \leq r < 5.5$?

- A. `r = d + 0.5;`
- B. `r = d + 0.5 * 5.0;`
- C. `r = d * 5.0;`
- D. `r = d * 5.0 + 0.5;`
- E. `r = d * 5.5;`

10. Consider the following method. Method `allEven` is intended to return `true` if all elements in array `arr` are even numbers; otherwise it should return `false`.

```
public boolean allEven(int[] arr)
{
    boolean isEven = /* expression */;

    for(int k = 0; k < arr.length; k++)
    {
        /* loop body */
    }
    return isEven;
}
```

Which of the following replacements for `/* expression */` and `/* loop body */` should be used so that method `allEven` will work as intended?

- | | <u>/* expression */</u> | <u>/* loop body */</u> |
|----|-------------------------|--|
| A. | <code>false</code> | <code>if((arr[k] % 2) == 0)
 isEven = true;</code> |
| B. | <code>false</code> | <code>if((arr[k] % 2) != 0)
 isEven = false;
else
 isEven = true;</code> |
| C. | <code>true</code> | <code>if((arr[k] % 2) != 0)
 isEven = false;</code> |
| D. | <code>true</code> | <code>if((arr[k] % 2) != 0)
 isEven = false;
else
 isEven = true;</code> |
| E. | <code>true</code> | <code>if((arr[k] % 2) == 0)
 isEven = false;
else
 isEven = true;</code> |
-

Questions 11 – 12 refer to the code from the Picture Lab

11. Which of the following will compile without error?

- I. `DigitalPicture p = new Picture();`
- II. `DigitalPicture p = new SimplePicture();`
- III. `Picture p = new SimplePicture();`

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

12. If the following code is in a method in the `Picture` class, what will the value of `count` be after the following code executes?

```
int count = 0;
for (int row = 5; row < 12; row++)
{
    for (int col = 8; col < 13; col++)
    {
        count++;
    }
}
```

- A. 13
- B. 25
- C. 35
- D. 42
- E. 48

Questions 13 – 14 refer to the code from the Elevens Lab

13. Consider the first line of the **ElevensBoard** class declaration below.

```
public class ElevensBoard extends Board
```

Which of the following methods must be implemented as a result of the way the **ElevensBoard** class has been declared?

- I. toString
- II. anotherPlayIsPossible
- III. isLegal
- IV. dealMyCards

- A. I only
 - B. I, II, and IV only
 - C. II and III only
 - D. I and IV only
 - E. I, II, III, and IV
-

14. Consider the following declarations where ... is a valid Board constructor parameter list.

- I. Board board = new Board(...);
- II. Board board = new ElevensBoard();
- III. ElevensBoard board = new ElevensBoard();

Which of these declarations is(are) legal?

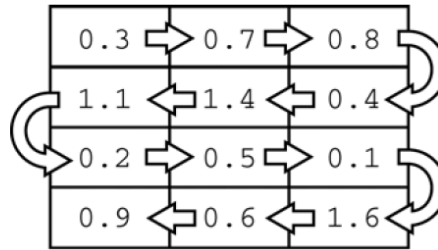
- A. I only
 - B. I and II only
 - C. I and III only
 - D. II and III only
 - E. I, II, and III
-

15. A car dealership needs a program to store information about the cars for sale. For each car, they want to keep track of the following information: number of doors (2 or 4), whether the car has air conditioning, and its average number of miles per gallon. Which of the following is the best object-oriented program design?

- A. Use one class, Car, with three instance variables: int numDoors, boolean hasAir, and double milesPerGallon.
- B. Use four unrelated classes: Car, Doors, AirConditioning, and MilesPerGallon.
- C. Use a class Car with three subclasses: Doors, AirConditioning, and MilesPerGallon.
- D. Use a class Car, with a subclass Doors, with a subclass AirConditioning, with a subclass MilesPerGallon.
- E. Use three classes: Doors, AirConditioning, and MilesPerGallon, each with a subclass Car.

16.

A telescope scans a rectangular area of the night sky and collects the data into a 1-dimensional array. Each data value scanned is a number representing the amount of light detected by the telescope. The telescope scans back and forth across the sky (alternating between left to right and right to left) in the pattern indicated below by the arrows. The back-and-forth ordering of the values received from the scan is called *telescope order*.



The telescope records the data in telescope order into a 1-dimensional array of `double` values. This 1-dimensional array of information received from a single scan will be transferred into a 2-dimensional array, which reconstructs the original view of the rectangular area of the sky. This 2-dimensional array is part of the `SkyView` class, shown below. In this question you will write a constructor and a method for this class.

```
public class SkyView
{
    /** A rectangular array that holds the data representing a rectangular area of the sky. */
    private double[][] view;

    /** Constructs a SkyView object from a 1-dimensional array of scan data.
     * @param numRows the number of rows represented in the view
     * Precondition: numRows > 0
     * @param numCols the number of columns represented in the view
     * Precondition: numCols > 0
     * @param scanned the scan data received from the telescope, stored in telescope order
     * Precondition: scanned.length == numRows * numCols
     * Postcondition: view has been created as a rectangular 2-dimensional array
     * with numRows rows and numCols columns and the values in
     * scanned have been copied to view and are ordered as
     * in the original rectangular area of sky.
     */
    public SkyView(int numRows, int numCols, double[] scanned)
    { /* to be implemented in part (a) */ }

    /** Returns the average of the values in a rectangular section of view.
     * @param startRow the first row index of the section
     * @param endRow the last row index of the section
     * @param startCol the first column index of the section
     * @param endCol the last column index of the section
     * Precondition: 0 <= startRow <= endRow < view.length
     * Precondition: 0 <= startCol <= endCol < view[0].length
     * @return the average of the values in the specified section of view
     */
    public double getAverage(int startRow, int endRow,
                             int startCol, int endCol)
    { /* to be implemented in part (b) */ }

    // There may be instance variables, constructors, and methods that are not shown.
}
```

- (a) Write the constructor for the `SkyView` class. The constructor initializes the `view` instance variable to a 2-dimensional array with `numRows` rows and `numCols` columns. The information from `scanned`, which is stored in the telescope order, is copied into `view` to reconstruct the sky view as originally seen by the telescope. The information in `scanned` must be rearranged as it is stored into `view` so that the sky view is oriented properly.

For example, suppose `scanned` contains values, as shown in the following array.

	0	1	2	3	4	5	6	7	8	9	10	11
scanned	0.3	0.7	0.8	0.4	1.4	1.1	0.2	0.5	0.1	1.6	0.6	0.9

Using the `scanned` array above, a `SkyView` object created with `new SkyView(4, 3, scanned)`, would have `view` initialized with the following values.

view	0	1	2
0	0.3	0.7	0.8
1	1.1	1.4	0.4
2	0.2	0.5	0.1
3	0.9	0.6	1.6

For another example, suppose `scanned` contains the following values.

	0	1	2	3	4	5
scanned	0.3	0.7	0.8	0.4	1.4	1.1

A `SkyView` object created with `new SkyView(3, 2, scanned)`, would have `view` initialized with the following values.

view	0	1
0	0.3	0.7
1	0.4	0.8
2	1.4	1.1

Complete the `SkyView` constructor below.

```
/** Constructs a SkyView object from a 1-dimensional array of scan data.
 * @param numRows the number of rows represented in the view
 * Precondition: numRows > 0
 * @param numCols the number of columns represented in the view
 * Precondition: numCols > 0
 * @param scanned the scan data received from the telescope, stored in telescope order
 * Precondition: scanned.length == numRows * numCols
 * Postcondition: view has been created as a rectangular 2-dimensional array
 * with numRows rows and numCols columns and the values in
 * scanned have been copied to view and are ordered as
 * in the original rectangular area of sky.
 */
public SkyView(int numRows, int numCols, double[] scanned)
```

- (b) Write the `SkyView` method `getAverage`, which returns the average of the elements of the section of `view` with row indexes from `startRow` through `endRow`, inclusive, and column indexes from `startCol` through `endCol`, inclusive.

For example, if `nightSky` is a `SkyView` object where `view` contains the values shown below, the call `nightSky.getAverage(1, 2, 0, 1)` should return `0.8`. (The average is $(1.1 + 1.4 + 0.2 + 0.5) / 4$, which equals `0.8`). The section being averaged is indicated by the dark outline in the table below.

view	0	1	2
0	0.3	0.7	0.8
1	1.1	1.4	0.4
2	0.2	0.5	0.1
3	0.9	0.6	1.6

Complete method `getAverage` below.

```
/** Returns the average of the values in a rectangular section of view.
 * @param startRow the first row index of the section
 * @param endRow the last row index of the section
 * @param startCol the first column index of the section
 * @param endCol the last column index of the section
 * Precondition: 0 <= startRow <= endRow < view.length
 * Precondition: 0 <= startCol <= endCol < view[0].length
 * @return the average of the values in the specified section of view
 */
public double getAverage(int startRow, int endRow,
                        int startCol, int endCol)
```

17. JavaBat Array 2 Type of Question

(More practice of these types of questions can be found on www.javabat.com)

18. JavaBat String 2 Type of Question

(More practice of these types of questions can be found on www.javabat.com)

19. JavaBat AP 1 Type of Question

(More practice of these types of questions can be found on www.javabat.com)

Answers for Multiple Choice

1. D
2. C
3. A
4. D
5. D
6. A
7. C
8. B
9. D
10. C
11. D
12. C
13. C
14. D
15. A

Solution for Free Response

16. Note: these are one solution to the practice free response. If you do not understand these solutions then you should write your own solutions that you do understand and bring them in so it can be checked.

Part (a):

```
public SkyView(int numRows, int numCols, double[] scanned)
{
    view = new double[numRows][numCols];
    int i = 0;
    for (int row = 0; row < numRows; row++) {
        if (row % 2 == 0) {
            for (int col = 0; col < numCols; col++) {
                view[row][col] = scanned[i];
                i++;
            }
        }
        else {
            for (int col = numCols - 1; col >= 0; col--) {
                view[row][col] = scanned[i];
                i++;
            }
        }
    }
}
```

Part (b):

```
public double getAverage(int startRow, int endRow, int startCol,
                        int endCol)
{
    double sum = 0.0;
    int count = 0;
    for (int row = startRow; row <= endRow; row++){
        for (int col = startCol; col <= endCol; col++){
            sum += view[row][col];
            count++;
        }
    }
    return sum / count;
}
```