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 arr is an array of integers.

int sum = arr[0],

i = 0;

while(i < arr.length) {

i++;

sum += arr[i];

}

Which of the following will be 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 < arr.length) {

i++;

if(arr[i] < min)

min = arr[i];

}

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

I. Change the line

int i = 1;

to

int i = 0;

II. Change the body of the while loop to

{

if(arr[i] < min)

min = arr[i];

i++;

}

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

while(i <=n)

(A) I only (B) II only

(C) III only (D) I and II only

(E) I, II and III

5. 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. Refer to the following class:

public class Tester {

private int[] testArray = {3, 4, 5};

// add 1 to n

public void increment(int n) {

n ++;

}

public void testMethod() {

for(int i = 0; i < testArray.length; i++) {

increment(testArray[i]);

System.out.print(testArray[i] + “ “);

}

}

}

If testMethod is called on a Tester object, what is printed?

(A) 3 4 5 (B) 4 5 6

(C) 5 6 7 (D) 0 0 0

(E) No output will be produced. An ArrayIndexOutOfBoundsException will be thrown.

8. Consider the getData method below:

public void getData(int[] myList) {

System.out.println(“Enter 10 integers “);

myList = new int[10];

for(int i = 0; i < 10; i++)

myList[i] = IO.readInt(); // read user input

}

A method in the same class calls getData as follows:

int[] myArray = new int[10];

getData(myArray);

for(int i = 0; i < 10; i++)

System.out.print(myArray[i] + “ “);

What will be the effect of running this code?

(A) The ten integers enter by the user will be the output.

(B) Ten zeroes will be the output.

(C) Ten distinct integers, different from those entered by the user, will be the output.

(D) A NullPointerException will be thrown.

(E) An ArrayIndexOutOfBoundsException will be thrown.

9. 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 static void doSomething(int[] myList) {

int[] b = myList;

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

Refer to the following code for Questions 10-11.

public class Address {

private String myName;

private String myStreet;

private String myCity;

private String myState;

private String myZip;

// constructors

…

// accessors

public String getName() {

return myName;

}

public String getStreet() {

return myStreet;

}

public String getCity() {

return myCity;

}

public String getState() {

return myState;

}

public String getZip() {

return myZip;

}

}

10. A client method has this declaration:

Address[] myList = new Address [100];

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

for(int i = 0; i < myList.length; i++)

< *line of code* >

Which is a correct < *line of code* >?

(A) System.out.println(Address[i].getName());

(B) System.out.println(Address.myList[i].getName());

(C) System.out.println(myList[i].Address.getName());

(D) System.out.println(myList[i].getName());

(E) System.out.println(myList.getName[i]);

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

for(int i = 0; i < myList.length; i++) {

< *more code*>

}

Which is a correct replacement for < *more code* >?

I. System.out.println(myList[i].getName());

System.out.println(myList[i].getStreet());

System.out.print(myList[i].getCity() + “, “);

System.out.print(myList[i].getState() + “ “);

System.out.println(myList[i].getZip());

II. System.out.println(myList[i]);

III. System.out.println(myList[i].Address);

(A) I only (B) II only

(C) III only (D) I and II only

(E) I, II and III

12. Consider this class:

public class Book {

private String myTitle;

private String myAuthor;

// constructor

public Book(String title, String author) {

myTitle = title;

myAuthor = author;

}

// display title, author

public void display() {

//implementation code not shown

}

…

}

A program has this declaration:

Book[] booklist = new Book[SOME\_NUMBER];

Suppose booklist is initialized so that each Book in the list has a title and author. Which of the following will display the title and author of each book in booklist?

(A) for(int i = 0; i < booklist[i].length; i++)

booklist[i].display();

(B) for(int i = 0; i < booklist.length; i++)

booklist[i].display();

(C) for(int i = 0; i < booklist.length; i++)

booklist.display();

(D) for(int i = 0; i < booklist.length; i++)

Book.display();

(E) for(int i = 0; i < booklist.length; i++)

Book[i].display();

13. Refer to method insert below:

/\* Precondition: List myList is any ArrayList which contains Comparable

\* values sorted in decreasing order

\* Postcondition: element inserted in its correct position in list

\*/

public void insert(List myList, Comparable element) {

int index = 0;

while(element.compareTo(myList.get(index)) < 0)

index++;

myList.add(index, element);

}

Assuming that the type of element is compatible with the objects in myList 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 myList and works in all other cases.

(D) It fails if element is smaller than the last item in myList 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 myList and works in all other cases.

14. Consider the following code segment, applied to myList, an ArrayList of Integer values.

int len = myList.size();

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

myList.add(i+1, i);

int x = myList.set(i, (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

15. Refer to the following declarations:

String[] colors = {“red”, “green”, “black”};

ArrayList<String> colorList = new ArrayList<String>();

Which of the following correctly assigns the elements of the colors array to colorList? The final ordering of the colors in colorList should be the same as in the colors array.

I. for(int i = 0; i < colors.length; i++)

colorList.add(i, colors.get(i));

II. for(int i = 0; i < colors.length; i++)

colorList.add(colors[i]);

III. for(int i = colors.length – 1; i >= 0; i--)

colorList.add(i, colors.[i]);

(A) I only (B) II only

(C) III only (D) II and III only

(E) I, II and III

16. Consider writing a program that reads the lines of any text file into 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 that the [] notations 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, will go out of bounds.

(E) The String methods are easier to use with an ArrayList than with an array.

17. 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.

Use the following code for question 18.

/\* A simple coin class \*/

public class Coin {

private double myValue;

private String myName;

// constructor

public Coin(double value, String name) {

myValue = value;

myName = name;

}

// Return the value and name of this coin

public double getValue() {

return myValue;

}

public String getName() {

return myName;   
 }

// Define equals method for Coin objects

public Boolean equals(Object obj) {

/\* implementation not shown \*/

}

// Other methods not shown

…

}

/\* A purse holds a collection of coins \*/

public class Purse {

private List<Coin> coins;

// constructor

// creates an empty purse

public Purse() {

coins = new ArrayList<Coin>();

}

// Adds aCoin to the purse

public void add(Coin aCoin) {

coins.add(aCoin);

}

// Returns total value of coins in purse

public double getTotal() {

*implementation code*

}

}

Here is the getTotal method from the Purse class:

// returns total value of coins in purse

public double getTotal() {

double total = 0;

< *more code* >

return total;

}

18. Which of the following is a correct replacement for < *more code* >?

(A) for(int i = 0; i < coins.length; i++) {

Coin c = coins.get(i);

total += coins.getValue();

}

(B) for(int i = 0; i < coins.length(); i++) {

Coin c = coins.[i];

total += c.getValue();

}

(C) for(int i = 0; i < coins.length(); i++) {

Coin c = coins.get(i);

total += c.getValue();

}

(D) for(int i = 0; i < coins.size(); i++) {

Coin c = coins.get(i);

total += coins.getValue();

}

(E) for(int i = 0; i < coins.size(); i++) {

Coin c = coins.get(i);

total += c.getValue();

}

19. Which of the following initializes an 8 x 10 matrix with integer values that are perfect squares? (0 is a perfect square.)

I. int[][] mat = new int[8][10];

II. int[][] mat = new int[8][10];

for(int r = 0; r < mat.length; r++)

for(int c = 0; c < mat[r].length; c++)

mat[r][c] = r \* r;

III. int[][] mat = new int[8][10];

for(int c = 0; c < mat[r].length; c++)

for(int r = 0; r < mat.length; r++)

mat[r][c] = c \* c;

(A) I only (B) II only

(C) III only (D) I and II only

(E) I, II, and III

20. A square matrix is declared as

int[][] mat = new int[SIZE][SIZE];

where SIZE is an appropriate integer constant. Consider the following method:

public void mystery(in[][] mat, int value, int top, int left, int bottom, int right) {

for(int i = left; i <= right; i++) {

mat[top][i] = value;

mat[bottom][i] = value;

}

for(int i = top + 1; i <= bottom – 1; i++) {

mat[i][left] = value;

mat[i][right] = value;

}

}

Assuming that there are no out-of-range errors, which best describes what method mystery does?

(A) Places value in corners of the rectangle with corners (top, left) and (bottom, right).

(B) Places value in diagonals of the square with corners (top, left) and (bottom, right).

(C) Places value in each element of the rectangle with corners (top, left) and (bottom, right).

(D) Places value in each element of the border of the rectangle with corners (top, left) and (bottom, right).

(E) Places value in the topmost and bottommost rows of the rectangle with corners (top, left) and (bottom, right).

Use the following method for Question 21.

public static Boolean isThere(String[][] mat, int row, int col, String symbol) {

boolean yes;

int i, count = 0;

for(i = 0; i < SIZE; i++)

if(mat[i][col].equals(symbol))

count++;

yes = (count == SIZE);

count = 0;

for(i = 0; i < SIZE; i++)

if(mat[row][i].equals(symbol))

count++;

return(yes || count == SIZE);

}

21. Now consider this code segment:

public final int SIZE = 8;

String[][] mat = new String[SIZE][SIZE];

Which of the following conditions on a matrix mat of the type declared in the code segment will by itself guarantee that

isThere(mat, 2, 2, “$”)

will have the value true when evaluated?

I. The element in row 2 and column 2 is “$”.

II. All elements in both diagonals are “$”.

III. All elements in column 2 are “$”.

(A) I only (B) III only

(C) I and II only (D) I and III only

(E) II and III only

22. A two-dimensional array of double, rainfall, will be used to represent the daily rainfall for a given year. In this scheme, rainfall[month][day] represents the amount of rain on the given day and month. For example,

rainfall[1][15] is the amount of rain on January 15th.

rainfall[12][25] is the amount of rain on December 25th.

The array can be declared as follows:

double[][] rainfall = new double[13][32];

This creates 13 rows indexed from 0 to 12 and 32 columns indexed from 0 to 31, all initialized to 0.0. Row 0 and column 0 will be ignored. Column 31 in row 4 will be ignored since April 31 is not a valid day. In years that are not leap years, columns 29, 30 and 31 in row 2 will be ignored since February 29, 30 and 31 are not valid days.

Consider the method averageRainfall below:

/\* Precondition: rainfall is initialized with values representing amounts of

\* rain on all valid days. Invalid days are initialized to

\* 0.0. February 29th is not a valid day.

\* Postcondition: returns average rainfall for the year \*/

public double averageRainfall(double rainfall[][]) {

double total = 0.0;

< *more code* >

}

Which of the following is a correct replacement for < *more code* > so that the postcondition for the method is satisfied?

I. for(int month = 1; month < rainfall.length; month++)

for(int day = 1; day < rainfall[month].length; day++)

total += rainfall[month][day];

return total /(13\*32);

II. for(int month = 1; month < rainfall.length; month++)

for(int day = 1; day < rainfall[month].length; day++)

total += rainfall[month][day];

return total / 365;

III. for(int month = 0; month < rainfall.length; month++)

for(int day = 0; day < rainfall[month].length; day++)

total += rainfall[month][day];

return total / 365;

(A) none (B) I only

(C) II only (D) III only

(E) II and III only