

For Questions 3 - 4, refer to the following code.

```
public static int sum(int[] a)
{
    int result = 0;
    String newName = "name";

    for (int i = 0; i < a.length; i++)
    {
        result = result + a[i];
    }
    newName = name + "1";
    return result;
}
```

3) Consider the following code segment.

```
int[] array= {1, 2, 3, 4, 5, 7};
sum(array);
```

What is the value of result when the sum(array) call is made?

- (A) 22
- (B) 33
- (C) 44
- (D) 55
- (E) 66

Loops &

ARRAYS

pt. #2

6) Consider the incomplete method. The method `indexElement()` is intended to return the element at specified index passed. The possible indices are 0, 1, 2, and 3.

```
public int index(int x)
{
    int[] array = {1, 2, 3, 5};

    // missing code
}
```

Which of the code segments shown below can be used to replace `// missing code` so that `index()` will work as intended?

- (A) `int element = array[x];`
- (B) `int element = array[x];`
`return element;`
- (C) `element = array[x];`
`return element;`
- (D) `int element = array;`
`return element;`
- (E) None of the above

4) Consider the following code segment.

```
int[] array= {1, 2, 3, 4, 5, 7};
sum(array);
```

What is the value of `newName` when the `sum(array)` call is made?

- (A) `name1`
- (B) `name`
- (C) `12345`
- (D) `54321`
- (E) `12111`

7) Consider the following code segment.

```
int values[] = {1, 2, 3, 4, 5, 6, 7, 8};  
for (int i = 0; i < x; i++)  
{  
    System.out.println(values[i]);  
}
```

What value of x will allow all elements in the `values[]` array to be printed out?

- (A) 6
- (B) 7
- (C) 8
- (D) 9
- (E) 3

Practice Test 2

For Questions 8 - 9, refer to the following information.

```
1 public class Test
2 {
3     public static void main(String[] args)
4     {
5         int total = 0;
6         int[] i = new int[3];
7
8         for (int j = 1; j <= i.length; j++)
9         {
10            total += (i[j] = j);
11            System.out.println(total);
12        }
13    }
14 }
```

9) What will be the output if we remove Line 3 of the Test class?

(A) Compile time error

(B) 5

(C) 6

(D) 7

(E) 9

8) What is the output of the above program?

(A) 5

(B) 6

(C) 7

(D) All of the above

(E) None of the above; the system will throw an `ArrayIndexOutOfBoundsException`

12) What will be displayed upon execution of this code segment?

```
int[] myarr = new int[10];
for (int i = 0 ; i < myarr.length; i++)
{
    myarr[i] = i + 1;
}
for (int i = 0; i < myarr.length; i++)
{
    System.out.print(myarr[i]);
}
```

- (A) 0 1 2 3 4 5 6 7 8 9
- (B) 12345678910
- (C) 1 2 3 4 5 6 7 8 9 10
- (D) myarr 1 2 3 4 5 6 7 8 9 10
- (E) None of the above

```
(D) for (Book b : books)
{
    System.out.println(title);
    System.out.println(price);
}
```

- (E) None of the above

For Questions 29 - 30, consider the following sorting method.

```
public static int[] sort(int[] arr)
{
    int temp;
    for (int i = 0; i < arr.length - 1; i++)
    {
        for (int j = 1; j < arr.length - i; j++)
        {
            if (arr[j - 1] > arr[j])
            {
                temp = arr[j - 1];
                arr[j - 1] = arr[j];
                arr[j] = temp;
            }
        }
    }
    return arr;
}
```

29) If we pass an array to the method above, in what form will the method return it?

- (A) In descending order
- (B) In ascending order
- (C) Unsorted
- (D) Run time error
- (E) Random order

30) If [10 9 2 15 11] is passed to the method, what will the array be after the second iteration of the outer loop?

- (A) [9 10 2 15 11]
- (B) [9 2 10 11 15]
- (C) [2 9 10 11 15]
- (D) [10 9 11 15 2]
- (E) [9 2 10 15 11]

For Questions 15 - 16, refer to the following code.

```
public class Employee
{
    private String EmpId;
    private String Name;
    private double Salary;

    public Employee(String EmpId, String Name, double Salary)
    {
        this.EmpId = EmpId;
        this.Name = Name;
        this.Salary = Salary;
    }
    // Setter and Getter methods
}
```

```
Employee[] emp = new Employee[50];
```

15) Which of the following code segments can be used to print the maximum salary?

(A)

```
for (int i = 0; i < emp.length; i++)
{
    int max = 0;
    if (emp.getSalary() > max)
    {
        max = emp.getSalary();
    }
}
System.out.println(max);
```

```
(B) int max = Employee[0];  
    for (Employee e : emp)  
    {  
        if (e.getSalary() > max)  
        {  
            max = e.getSalary();  
        }  
    }  
    System.out.println(max);
```

```
(C) if(emp[i].getSalary > emp[i+1].getSalary())  
    {  
        max = emp[i].getSalary();  
    }  
    System.out.println(max);
```

(D) None of the above

16) Which of the following code segments represents the total salary of all the employees?

(A)

```
double total_sal = 0.0;
for (Employee e : Emp)
{
    total_sal = e.getSalary();
}
```

(B)

```
double total_sal = 0.0;
for (Employee e : Emp)
{
    total_sal = e[i].getSalary();
}
```

(C)

```
double total_sal = 0.0;
for (Employee e : Emp)
{
    total_sal += e.getSalary();
}
```

(D)

```
double total_sal = 0.0;
for (Employee e : Emp)
{
    total_sal += Emp.getSalary();
}
```

17) What will be displayed upon execution of this code segment?

```
int [] a = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
for (int i = 0; i < a.length; i++)
{
    if (a[i] % 2 == 0)
    {
        System.out.print(a[i] + " ");
    }
}
```

(A) 1 3 4 5 6 7

(B) 2 4 6 8 10

(C) 1 2 3 4 5

(D) 2 4 5 6 7

25) Consider the following code segment.

```
1 for (int i = 0; i < 5; i++)
2 {
3     for (int j = 0; j < 5; j++)
4     {
5         System.out.print("*");
6     }
7 }
```

How many times is Line 5 executed?

- (A) 10
- (B) 15
- (C) 20
- (D) 25
- (E) 30

Loops
Arrays
= 2

26) Consider the following output.

```
1 1
2 22
3 333
4 4444
5 55555
```

Which of the following code segments produces the desired output?

```
I. for (int i = 1; i <= 5; i++)
    {
        for (int j = i; j > 0; j--)
            {
                System.out.print(i);
            }
        System.out.println();
    }
```

```
II. for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < i; j++)
            {
                System.out.print(i);
            }
        System.out.println();
    }
```

```
III. for (int i = 1; i <= 5; i++)
    {
        for (int j = i; j > 0; j--)
            {
                System.out.print(i);
            }
    }
```

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

27) Consider the following code segment.

```
for (int k = 0; k < 20; k += 2)
{
    if (k % 3 == 1)
    {
        System.out.print(k + " ");
    }
}
```

What is the output of above code segment?

- (A) 16 4 10
- (B) 4 10 16
- (C) 10 4 16
- (D) 4 10 4
- (E) 4 4 4

28) Consider the following code segment.

```
for (int j = 1; j <= 1; j++)
{
    for (int k = 1; k <= 1; k = k * 2)
    {
        System.out.println(j + " " + k);
    }
}
```

What is the output of above code segment?

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

33) Consider the following code segment.

```
char array_variable [] = new char[10];  
for (int i = 0; i < 10; ++i)  
{  
    array_variable[i] = 'i';  
    System.out.print(array_variable[i] + "");  
}
```

What is the output of the above code segment?

- (A) 1111111111
- (B) iiiiiiiiii
- (C) iii
- (D) iiiii
- (E) iiii

11) Consider the following code segment.

```
public static void main(String[] args)
{
    int[] a = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    for (int i = 0; i < a.length; i++)
    {
        System.out.print(a[i] + " ");
        System.out.print(a[i] + " ");
    }
}
```

(A) 1 2 3 4 5 6 7 8 9 10

(B) 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10

(C) 1 2 3 4 5 6 7 8 9

(D) 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9

(E) None of the above

12) Which of the while loops is equivalent to the for loop below?

```
for (int x = 5; x <= 50; x += 5)
{
    System.out.print(" " + x);
}
```

(A)

```
int x = 5;
while (x <= 50)
{
    System.out.print(" " + x);
    x += 5;
}
```

(B)

```
while (x <= 50)
{
    int x = 5;
    System.out.print(" " + x);
    x += 5;
}
```

(C)

```
while (int x = 5; x <= 50)
{
    System.out.print(" " + x);
    x += 5;
}
```

(D)

```
int x = 5;
while (x <= 50; x += 5)
{
    System.out.print(" " + x);
}
```

18) In an exam, there are 50 questions, each worth 1 point. One point is awarded for every correct answer and one point deducted for every mistake or omission. The exam has been graded and the result is saved as a boolean array called `question[]`. Which of the following code segments correctly calculates and outputs the total score?

```
(A) int sum_c = 0;
    int sum_f = 0;
    for (int i = 0; i < question.length; i++)
    {
        if (question[i] == true)
        {
            sum_c = sum_c + 1;
        }
    }
    sum_f = 50 - sum_c;
    int total = sum_c - sum_f;
    System.out.println(total);
```

```
(B) for (int i = 0; i < question.length; i++)
    {
        if (question[i] == true)
        {
            sum_c = sum_c + 1;
        }
    }
```

```
(C) int sum_c = 0;
    int sum_f = 0;
    for (i = 0; i < question.length; i++)
    {
        if (question[i] = true)
        {
            sum_c = sum_c + 1;
        }
    }
    sum_f = 50 - sum_c;
    int total = sum_c - sum_f;
    System.out.println(total);
```

```
(D) int sum_c = 0;
    int sum_f = 0;
    for (i = 0; i < question.length; i++)
    {
        if (question[i] == false)
        {
            sum_c = sum_c + 1;
        }
    }
    sum_f = 50 - sum_c;
    int total = sum_c - sum_f;
    System.out.println(total);
```

(E) None of the above

32) Consider the following method.

```
public int findele(int[] a, int key)
{
    for (int i = 0; i < a.length; i++)
    {
        if (key == a[i])
        {
            System.out.println("Element found");
        }
    }
}
```

What will be the output if the following array is passed and the key is 30.

```
int[] a = {12, 15, 67, 30, 80, 67};
```

- (A) Compile time Error
- (B) Run time Error
- (C) No output
- (D) Element found
- (E) None of the above

34) Consider the following method.

```
public static void sort(int[] arr)
{
    int n = arr.length;
    for (int j = 1; j < n; j++)
    {
        int k = arr[j];
        int i = j - 1;
        while ((i > -1) && (arr[i] > k))
        {
            arr[i+1] = arr[i];
            i--;
        }
        arr[i+1] = k;
    }
}
```

The following array is passed to the sort method.

```
int[] arr = {10, 3, 4, 2, 1, 14, 12, 11};
sort(arr);
```

What will be the sequence of elements at the completion of the 3rd iteration of the for loop?

- (A) 2, 3, 10, 4, 1, 14, 12, 11
- (B) 2, 3, 4, 10, 1, 14, 12, 11
- (C) 2, 10, 3, 4, 1, 14, 12, 11
- (D) 2, 3, 4, 10, 1, 11, 12, 14
- (E) None of the above

For Questions 35 - 36, refer to the following class.

```
public class Student
{
    private String id;
    private double gpa;
    private int marks;

    public Student(String id, double gpa, int marks)
    {
        this.id = id;
        this.gpa = gpa;
        this.marks = marks;
    }

    // getter and setter methods (not shown)
    // other methods (not shown)
}
```

35) Consider the following code segment.

```
Student[] studs = new Student[10];

public void highestgpa()
{
    /* Implementation goes here */
}
```

Which of the following implementations is a valid method of calculating the highest GPA among students in the studs array?

(A) `double max = 0.0;`

```
for (Student s : studs)
{
    if (s.getGpa() > max)
    {
        max= s.getGpa();
    }
}
System.out.println("Highest gpa: " + max);
```

(B) `if (s.getGpa() > max)`
`{`
`max = s.getGpa();`
`}`

(C) `for (Student s : studs)`
`{`
`double max = 0.0;`

`if (s.getGpa() > max)`
`{`
`max = s.getGpa();`
`}`
`}`
`System.out.println("Highest gpa: " + max);`

(D) for (Student s : studs)

```
{  
  
    double max = 0.0;  
  
    while (s.getGpa() > max)  
    {  
        max = s.getGpa();  
    }  
}  
System.out.println("Highest gpa: " + max);
```

(E) None of the above

36) Consider the following code segment.

```
public void averagemarks()  
{  
    /* Implementation goes here */  
}
```

Which of the following implementations is a valid method of calculating the average marks among students in the studs array?

(A)

```
double avg = 0.0;  
double sum = 0.0;  
  
for (Student s : studs)  
{  
    sum = s.getMarks();  
}  
System.out.println(avg);
```

(B)

```
double avg = 0.0;  
double sum = 0.0;  
  
for (Student s : studs)  
{  
    sum += s.getMarks();  
}  
avg = sum / studs.length;  
System.out.println(avg);
```

(C)

```
for (Student s : studs)  
{  
    sum += s.getMarks();  
}
```

(D)

```
double avg = 0.0;  
double sum = 0.0;  
  
for (Student s : studs)  
{  
    sum += s.getMarks();  
}  
avg = sum / studs.length();  
System.out.println(avg);
```

(E) None of the above

8. What is the value of `v[4]` after the following code is executed?

```
int i;
int d = 1;
int[] v = {1, 1, 1, 1, 1};

for (i = 0; i < v.length; i++)
{
    d *= 2;
    v[i] += d;
}
```

- (A) 16
- (B) 32
- (C) 33
- (D) 64
- (E) 65

13. Consider the following code segment, intended to find the position of an integer `targetValue` in `int[] a`:

```
int i = 0, position;
while (a[i] != targetValue)
{
    i++;
}
position = i;
```

When will this code work as intended?

- (A) Only when $0 \leq \text{targetValue} < \text{a.length}$
- (B) Only when $\text{targetValue} == \text{a}[0]$
- (C) Only when $\text{targetValue} == \text{a}[i]$ for some $i, 0 \leq i < \text{a.length}$
- (D) Only when $\text{targetValue} != \text{a}[i]$ for any $i, 0 \leq i < \text{a.length}$
- (E) Always

19. Consider the following three code segments:

- I.

```
int i = 1;
while (i <= 10)
{
    System.out.println(i);
    i += 2;
}
```
- II.

```
for (int i = 0; i < 5; i++)
{
    System.out.println(2*i + 1);
}
```
- III.

```
for (int i = 0; i < 10; i++)
{
    i++;
    System.out.println(i);
}
```

Which of the three segments produce the same output?

- (A) I and II only
- (B) II and III only
- (C) I and III only
- (D) I, II, and III
- (E) All three outputs are different.

30. Consider the following method:

```
// precondition: a[0] ... a[a.length - 1] are
//              sorted in ascending order
// postcondition: Returns the location of the target
//              value in the array a, or -1 if not found
public static int search(int[] a, int target)
{
    int first = 0;
    int middle;
    int last = a.length - 1;

    while (first <= last)
    {
        middle = (first + last) / 2;
        if (target == a[middle])
            return middle;
        else if (target < a[middle])
            last = middle;
        else
            first = middle;
    }
    return -1;
}
```

This method fails to work as expected under certain conditions. If the array has five elements with values 3 4 35 42 51, which of the following values of target would make this method fail?

- (A) 3
- (B) 4
- (C) 35
- (D) 42
- (E) 51

22. Consider the following method with two missing statements:

```
// precondition: 1 <= n <= arr.length
// postcondition: returns the sum of all positive odd values
//              among the first n elements of arr
public static int addPositiveOddValues(int[] arr, int n)
{
    int i, sum = 0;
    < statement1 >
    {
        < statement2 >
        sum += arr[i];
    }
    return sum;
}
```

Which of the following are appropriate replacements for < statement1 > and < statement2 > so that the method works as specified?

- | < statement1 > | < statement2 > |
|---------------------------------|------------------------------------|
| (A) for (i = 1; i < n; i += 2) | if (arr[i] > 0) |
| (B) for (i = 0; i < n; i++) | if (arr[i] > 0 && arr[i] % 2 != 0) |
| (C) for (i = 1; i <= n; i += 2) | if (arr[i] > 0) |
| (D) for (i = 0; i <= n; i++) | if (arr[i] % 2 != 0) |
| (E) None of the above | |

31. Which of the following best describes the return value for the method `propertyX` below?

```
// precondition: v.length >= 2
boolean propertyX(int[] v)
{
    boolean flag = false;
    int i;

    for (i = 0; i < v.length - 1; i++)
    {
        flag = flag || (v[i] == v[i+1]);
    }

    return flag;
}
```

- (A) Returns `true` if the elements of `v` are sorted in ascending order, `false` otherwise
- (B) Returns `true` if the elements of `v` are sorted in descending order, `false` otherwise
- (C) Returns `true` if `v` has two adjacent elements with the same value, `false` otherwise
- (D) Returns `true` if `v` has two elements with the same value, `false` otherwise
- (E) Returns `true` if all elements in `v` have different values, `false` otherwise

1. What is the output of the following program segment?

```
int num = 5;
while (num >= 0)
{
    num -= 2;
}
System.out.print(num);
```

- (A) -2
- (B) -1
- (C) 0
- (D) 2
- (E) 21

4. What is the output of the following code segment?

```
int sum = 0, count, d = -1;
for (count = 10; count > 0; count--)
{
    sum += d;
    if (d > 0)
        d++;
    else
        d--;
    d = -d;
}
System.out.println(sum);
```

- (A) 0
- (B) 5
- (C) -5
- (D) 10
- (E) -10

5. The following code segment is supposed to calculate and display $1 + 2 + \dots + 20$:

```
int count = 0, sum = 0;
while (count < 20)
{
    sum += count;
}
System.out.println(sum);
```

Which statement best describes the result:

- (A) The total displayed will be correct.
- (B) The total displayed will be 20 too small.
- (C) The output will be the number 0.
- (D) The output will be the number 20.
- (E) There will be no output because the program goes into an infinite loop.

13. Consider the following code segment:

```
int n = IO.readInt(); // read an int value
n = Math.abs(n);

while (n >= 2)
{
    n = n/2 - 1;
}
System.out.print(n);
```

Which of the following is the list of all possible outputs?

- (A) 0
- (B) -1, 0
- (C) 0, 1
- (D) -1, 1
- (E) -1, 0, 1

31. A programmer wants to create a swap method that swaps two integer values. Which of the following three ways of representing the values and corresponding methods successfully swap the values?

I.

```
// a and b are Integer objects that represent the values
// to be swapped
public static void swap(Integer a, Integer b)
{
    Integer temp = a; a = b; b = temp;
}
```

II.

```
// a[0] and a[1] contain the values to be swapped
public static void swap(int[] a)
{
    int temp = a[0]; a[0] = a[1]; a[1] = temp;
}
```

III.

```
// a[0] and b[0] contain the values to be swapped
public static void swap(int[] a, int[] b)
{
    int temp = a[0]; a[0] = b[0]; b[0] = temp;
}
```

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

14) Consider the following code segment.

```
int sum = 0;
for (int i = 0, j = 0; i < 5; ++i)
{
    sum += i;
}
System.out.println(sum);
```

What will be displayed upon execution of this code segment?

- (A) 5
- (B) 10
- (C) 4
- (D) 7

16) Which of the following code segments can be used to determine whether number is an prime number?

```
(A) for (int i = 2; i <= number / 2; i++)
{
    if (number % i = 0)
    {
        return false;
    }
}
```

```
(B) for (int i = 2; i <= number / 2; i++)
{
    if (number / i == 0)
    {
        return false;
    }
}
```

```
(C) for (int i = 2; i <= number / 2; i++)
{
    if (number / i = 0)
    {
        return false;
    }
}
```

```
(D) for (int i = 2; i <= number / 2; i++)
{
    if (number % i == 0)
    {
        return false;
    }
}
```

18) Which of the following code segments prints out all the multiples of from 3 to 36?

```
(A) for (int n = 3; n < 36; n++)
{
    if (n % 3 == 0)
    {
        System.out.println( n );
    }
}
```

```
(B) for (int n = 3; n <= 36; n++)
{
    if (n % 3 == 0)
    {
        System.out.println(n);
    }
}
```

```
(C) for( int n = 3; n < 36; n--)
{
    if (n % 3 == 0)
    {
        System.out.println(n);
    }
}
```

```
(D) for (int n = 3; n <= 36; n--)
{
    if (n % 3 == 0)
    {
        System.out.println(n);
    }
}
```

19) Consider the following code segment.

```
int x = 1;
while (x <= 32)
{
    x = 2 * x;
    System.out.print(x + " ");
}
```

What will be displayed upon execution of this code segment?

- (A) 1 2 4 8 16 32 64
- (B) 2 4 8 16 32 64
- (C) 2 4 8 16 32
- (D) 1 2 4 8 16 32

20) Consider the following code segment.

```
int x = 5;
int y = 1;
while (x >= 0)
{
    x = x - 1;
    y = y * x;
    System.out.print(y + " ");
}
```

What will be displayed upon execution of this code segment?

- (A) 4 12 24 12 4 0
- (B) 4 12 24 24 0
- (C) 0 4 12 24 12 4 0
- (D) 4 12 24

24) Consider the following code segment.

```
for (int d = 1; d <= n; d++)
{
    if (n % d == 0)
    {
        System.out.println(d);
    }
}
```

What will be displayed upon execution of this code segment?

- (A) The quotients of n
- (B) The remainders of n
- (C) The remainders of d
- (D) The divisors of n

22) Consider the following code segment.

```
int years;
int principal = 1000;
double interest;
double rate = 0.5;
for (years = 0; years < 5; years++)
{
    interest = principal * rate;
    principal += interest;
    System.out.println(principal);
}
```

What will be displayed after 1 iteration of the for loop?

- (A) 1000
- (B) 500
- (C) 0
- (D) 1500

1) Which one of the following array declarations is incorrect?

- (A) `int[] a = {1, 2, 3, 4, 5};`
- (B) `int[] a = new int[5];`
- (C) `int[] a = {"1", "2", "3", "4", "5"};`
- (D) `int[] a = new int [5];`

2) What will be displayed upon execution of this code segment?

```
int[] myarr = new int[10];
for (int i = 0 ; i < myarr.length; i++)
{
    myarr[i] = i + 1;
}
for (int i = 0; i < myarr.length; i++)
{
    System.out.print(myarr[i]);
}
```

- (A) 0 1 2 3 4 5 6 7 8 9
- (B) 12345678910
- (C) 1 2 3 4 5 6 7 8 9 10
- (D) myarr 1 2 3 4 5 6 7 8 9 10

3) Consider the following code segment. Which of the following will occur upon execution of this code segment?

```
for (int j = 0; j > a.length; j++)  
{  
    System.out.println(a[j]);  
}
```

- (A) Compile time error
- (B) Run time error
- (C) No error
- (D) Infinity loop run

4) What will be displayed upon execution of this code segment?

```
int[] a = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
int sum = 0;  
for (int i = 0; i < a.length; i++)  
{  
    if (i % 2 == 0)  
    {  
        sum += a[i];  
    }  
}  
System.out.println("Sum: " + sum);
```

- (A) Sum:25
- (B) Sum:30
- (C) Sum:10
- (D) Sum:40

For Questions 12 - 14, refer to the following code.

```
public class Customer
{
    private String name;
    private String SSN;
    private int Age;

    public Customer(String name, String SSN, int Age)
    {
        this.name = name;
        this.SSN = SSN;
        this.Age = Age
    }
    public String getName()
    {
        return name;
    }
    public String getSSN()
    {
        return SSN;
    }
    public String getAge()
    {
        return Age;
    }
}
```

```
Customer[] Cust = new Customer[100];
```

12) Which of the following code segments to print the names of customers who are older greater than 50?

I.

```
for (int i = 0; i < Cust.length; i++)
{
    if (Cust[i].getAge() > 50)
    {
        System.out.println(Cust[i].getName());
    }
}
```

II.

```
String name = " ";  
for (int i = 0; i < Cust.length; i++)  
{  
    if (Cust[i].getAge() > 50)  
    {  
        name = Cust[i].getName();  
        System.out.println(name);  
    }  
}
```

III.

```
System.out.println(Cust[i].getName());
```

Which of the above code segment gives the correct output?

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

13) Which of the following code segments will print all of the details of the customer?

```
Customer[] Cust = new Customer[];
```

(A) `System.out.println(Cust.getName());`
`System.out.println(Cust.getSSN());`
`System.out.println(Cust.getAge());`

(B) `for (int i = 0; i < 100; i++)`
`{`
`System.out.println(Cust[i].getName());`
`System.out.println(Cust[i].getSSN());`
`System.out.println(Cust[i].getAge());`
`}`

(C) `for (i = 0; i < 100; i++)`
`{`
`System.out.println(Cust);`
`}`

(D) None of the above

14) Which of the following code segments will print the SSN and name of customers with "Edison" as their names?

- (A)

```
for (Customer c : Cust)
{
    if ((c.getName()).equals("Edson"))
    {
        System.out.println(c.getSSN() + "," + c.getName());
    }
}
```
- (B)

```
System.out.println(c.getSSN());
System.out.println(c.getName());
```
- (C)

```
for (int i = 0 ; i < Cust.length; i++)
{
    System.out.println(Cust.getName()+ " , " + Cust.getSSN());
}
```
- (D) None of the above

23) Which of the following code segments populates a two-dimensional with 0 as each element?

```
int[][] matr = new int[3][4];
```

(A)

```
for (int r = 0; r < 3; r++)
{
    for (int c = 0; c < 4; c++)
    {
        matr[r][c] = 0;
    }
}
```

(B)

```
for (int r = 0, c = 0; r < 3, c < 4; r++)
{
    matr[r][c]=0;
}
```

(C)

```
for ( int r = 0, c = 0; r < 3, c < 4; r++, c-- )
{
    matr[r][c] = 0;
}
```

(D) None of the above

24) What will be displayed upon execution of this code segment?

```
int[][] M = new int[4][4];  
for (int r = 0, c = 0; r < 4, c > 4; r++, c--)  
{  
    M[r][c] = 0;  
}
```

- (A) Compile time error
- (B) Run time error
- (C) No Error
- (D) None of the above

Chapter 7
25) What will be returned upon execution of `matrfun(m)`?

```
int[][] m = {{1, 2, 3}, {4, 5, 6}};
```

```
public int[][] matrfun(int[][] matr)
{
    int[][] result = new int[matr.length][matr[0].length];
    for(int r = 0; r < matr.length; r++)
    {
        for(int c = 0; c < matr[0].length; c++)
        {
            result[r][c] = matr[r][c] * 2;
        }
    }
    return result;
}
```

(A) 1 2 3
4 5 6

(B) 2 4 6
8 10 12

(C) 1 4 9
16 25 36

(D) None of the above

26) The following method replaces the diagonal elements in an array with 5.

```
public void diagonal(int[][] arr)
{
    // Method body
}
```

Which of the following code segments is the appropriate method body?

(A)

```
for (int r = 0; r < arr.length; r++)
{
    for (int c = 0; c < arr[0].length; c++)
    {
        System.out.println(arr[r][c]);
    }
}
```

(B)

```
for (int r = 0; r < arr.length; r++)
{
    for (int c = 0; c < arr[0].length; c++)
    {
        arr[r][c] = 5;
    }
}
```

(C)

```
for (int r = 0; r < arr.length; r++)
{
    for (int c = 0; c < arr[0].length; c++)
    {
        if (r == c)
            arr[r][c] = 5;
    }
}
```

(D) None of the above.

For Questions 27 - 28, consider the following code.

```
public class Student
{
    private int id;
    private int gpa;

    public Student(int id,int gpa)
    {
        this.id = id;
        this.gpa = gpa;
    }
    public int getId()
    {
        return id;
    }
    public int getGpa()
    {
        return gpa;
    }
}
```

27) The method `indexele()` prints out the element at `[4][5]` in an array.

```
public void indexele(Student[][] x)
{
    // Method body
}
```

Which of the following code segments is is the appropriate method body?

```
(A) for (int r = 0; r < x.length; r++)
{
    for (int c = 0; c < x[0].length; c++)
    {
        if (r == 4 && c == 5)
            System.out.println(x);
    }
}
```

```
(B) for (int r = 0; r < x.length; r++)  
    {  
        for (int c = 0; c < x[0].length; c++)  
        {  
            if(r == 4 && c == 5)  
                System.out.println(x[r][c].getId() + " " + x[r][c].getGpa());  
        }  
    }
```

```
(C) for (int r = 0; r < x.length; r++)  
    {  
        for (int c = 0; c < x[0].length; c++)  
        {  
            if(r == 4 && c == 5)  
                System.out.println(x.getId() + " " + x.getGpa());  
        }  
    }
```

(D) None of the above

28) The method `highestGpa()` finds the highest GPA.

```
public int highestGpa(Student [][]x)
{
    // Method body
}
```

Which of the following code segments is is the appropriate method body?

(A)

```
for (int r = 0; r < x.length; r++)
{
    for (int c = 0; c < x[0].length; c++)
    {
        int max_gpa = 0;
        max_gpa = x[r][c];
    }
}
```

(B)

```
for (int r = 0; r < x.length; r++)
{
    for (int c = 0; c < x[0].length; c++)
    {
        max_gpa = x[r][c];
    }
}
```

(C)

```
int max_gpa = 0;
for (int r = 0; r < x.length; r++)
{
    for(int c = 0; c < x[0].length; c++)
    {
        if(x[r][c].getGpa() > max_gpa)
        {
            max_gpa = x[r][c];
        }
    }
}
return max_gpa;
```

(D) None of the above

29) What will be returned upon execution of matrixfun(m, n);?

```
int[][] m = {{1, 2, 3}, {1, 2, 2}, {3, 2, 1}};  
int[][] n = {{1, 1, 1}, {1, 2, 1}, {1, 1, 1}};
```

```
public static int[][] matrixfun(int[][] a, int[][] b)  
{  
    int[][] c = new int[3][3];  
    for (int i = 0; i < 3; i++)  
    {  
        for (int j = 0; j < 3; j++)  
        {  
            for (int l = 0; l < 3; l++)  
            {  
                c[i][j] = c[i][j] + a[i][l] * b[l][j];  
            }  
        }  
    }  
    return c;  
}
```

(A) 2 4 6
4 5 6
4 5 6

(B) 2 4 6
4 5 7
4 5 8

(C) 6 8 6
5 7 5
6 8 6

(D) 4 4 6

4 5 6

4 5 6

11. 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());

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

```
/** Precondition: Array stuArr of Student is initialized.
 * @return Student with highest idNum
 */
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

Questions 13-15 refer to the Ticket and Transaction classes below.

```
public class Ticket
{
    private String row;
    private int seat;
    private double price;

    //constructor
    public Ticket(String aRow, int aSeat, double aPrice)
    {
        row = aRow;
        seat = aSeat;
        price = aPrice;
    }

    //accessors getRow(), getSeat(), and getPrice()
    ...
}

public class Transaction
{
    private int numTickets;
    private Ticket[] tickList;

    //constructor
    public Transaction(int numTicks)
    {
        numTickets = numTicks;
        tickList = new Ticket[numTicks];
        String theRow;
        int theSeat;
        double thePrice;
        for (int i = 0; i < numTicks; i++)
        {
            < read user input for theRow, theSeat, and thePrice >
            ...

            /* more code */
        }
    }

    /** @return total amount paid for this transaction */
    public double totalPaid()
    {
        double total = 0.0;
        /* code to calculate amount */
        return total;
    }
}
```

13. 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(theRow, theSeat, thePrice);
- (C) tickList[i] = new tickList(getRow(), getSeat(), getPrice());
- (D) tickList[i] = new tickList(theRow, theSeat, thePrice);
- (E) tickList[i] = new tickList(numTicks);

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

- (A) for (Ticket t : tickList)
total += t.price;
- (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.price;

15. Suppose it is necessary to keep a list of all ticket transactions. Assuming that there are NUMSALES 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];

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

```
/** Precondition:  
 * - arr is an array, arr.length = n.  
 * - arr[0]...arr[n-1] initialized with integers.  
 * 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.

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

17. Refer to method match below:

```
/** @param v an array of int sorted in increasing order
 * @param w an array of int sorted in increasing order
 * @param N the number of elements in array v
 * @param M the number of elements in array w
 * @return true if there is an integer k that occurs
 * in both arrays; otherwise returns false
 * 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].
 */
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) $v[0] \dots v[vIndex-1]$ and $w[0] \dots w[wIndex-1]$ contain no common value, $vIndex \leq N$ and $wIndex \leq M$.
- (B) $v[0] \dots v[vIndex]$ and $w[0] \dots w[wIndex]$ contain no common value, $vIndex \leq N$ and $wIndex \leq M$.
- (C) $v[0] \dots v[vIndex-1]$ and $w[0] \dots w[wIndex-1]$ contain no common value, $vIndex \leq N-1$ and $wIndex \leq M-1$.
- (D) $v[0] \dots v[vIndex]$ and $w[0] \dots w[wIndex]$ contain no common value, $vIndex \leq N-1$ and $wIndex \leq M-1$.
- (E) $v[0] \dots v[N-1]$ and $w[0] \dots w[M-1]$ contain no common value, $vIndex \leq N$ and $wIndex \leq M$.

18. Consider this class:

```
public class Book
{
    private String title;
    private String author;
    private boolean checkoutStatus;

    public Book(String bookTitle, String bookAuthor)
    {
        title = bookTitle;
        author = bookAuthor;
        checkoutStatus = false;
    }

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

    //Other methods are 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[] card;

    /** 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

32. Consider a class MatrixStuff that has a private instance variable:

```
private int[] [] mat;
```

Refer to method alter below that occurs in the MatrixStuff class. (The lines are numbered for reference.)

```
Line 1: /** @param mat the matrix initialized with integers
Line 2: * @param c the column to be removed
Line 3: * Postcondition:
Line 4: * - Column c has been removed.
Line 5: * - The last column is filled with zeros.
Line 6: */
Line 7: public void alter(int[] [] mat, int c)
Line 8: {
Line 9:     for (int i = 0; i < mat.length; i++)
Line 10:         for (int j = c; j < mat[0].length; j++)
Line 11:             mat[i][j] = mat[i][j+1];
Line 12:         //code to insert zeros in rightmost column
Line 13:         ...
Line 14: }
```

The intent of the method alter is to remove column c. Thus, if the input matrix mat is

```
2 6 8 9
1 5 4 3
0 7 3 2
```

the method call alter(mat, 1) should change mat to

```
2 8 9 0
1 4 3 0
0 3 2 0
```

The method does not work as intended. Which of the following changes will correct the problem?

I Change line 10 to

```
for (int j = c; j < mat[0].length - 1; j++)
```

and make no other changes.

II Change lines 10 and 11 to

```
for (int j = c + 1; j < mat[0].length; j++)
    mat[i][j-1] = mat[i][j];
```

and make no other changes.

III Change lines 10 and 11 to

```
for (int j = mat[0].length - 1; j > c; j--)
    mat[i][j-1] = mat[i][j];
```

and make no other changes.

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

33. This question refers to the following method:

```
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);
}
```

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

34. The method `changeNegs` below should replace every occurrence of a negative integer in its matrix parameter with 0.

```
/** @param mat the matrix
 * Precondition: mat is initialized with integers.
 * Postcondition: All negative values in mat replaced with 0.
 */
public static void changeNegs(int[] [] mat)
{
    /* code */
}
```

Which is correct replacement for `/* code */`?

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

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

```
III for (int[] row : mat)
    for (int element : row)
        if (element < 0)
            element = 0;
```

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

35. 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 Jan. 15
rainfall[12][25] is the amount of rain on Dec. 25
```

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 Feb. 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.
 * - Feb 29 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 (double[] month : rainfall)
 for (double rainAmt : month)
 total += rainAmt;
return total / 365;
```

- (A) None  
(B) I only  
(C) II only  
(D) III only  
(E) II and III only

36. This question is based on the Point class below:

```
public class Point
{
 /** The coordinates. */
 private int x;
 private int y;

 public Point (int xValue, int yValue)
 {
 x = xValue;
 y = yValue;
 }

 /** @return the x-coordinate of this point */
 public int getx()
 { return x; }

 /** @return the y-coordinate of this point */
 public int gety()
 { return y; }

 /** Set x and y to new_x and new_y. */
 public void setPoint(int new_x, int new_y)
 {
 x = new_x;
 y = new_y;
 }

 //Other methods are not shown.
}
```

The method changeNegs below takes a matrix of Point objects as parameter and replaces every Point that has at least one negative coordinate with the Point (0,0).

```
/** @param pointMat the matrix of points
 * Precondition: pointMat is initialized with Point objects.
 * Postcondition: Every point with at least one negative coordinate
 * has been changed to have both coordinates
 * equal to zero.
 */
public static void changeNegs (Point [][] pointMat)
{
 /* code */
}
```

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

```
I for (int r = 0; r < pointMat.length; r++)
 for (int c = 0; c < pointMat[r].length; c++)
 if (pointMat[r][c].getX() < 0
 || pointMat[r][c].getY() < 0)
 pointMat[r][c].setPoint(0, 0);
```

```
II for (int c = 0; c < pointMat[0].length; c++)
 for (int r = 0; r < pointMat.length; r++)
 if (pointMat[r][c].getX() < 0
 || pointMat[r][c].getY() < 0)
 pointMat[r][c].setPoint(0, 0);
```

```
III for (Point[] row : pointMat)
 for (Point p : row)
 if (p.getX() < 0 || p.getY() < 0)
 p.setPoint(0, 0);
```

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

37. A simple Tic-Tac-Toe board is a  $3 \times 3$  array filled with either X's, O's, or blanks. Here is a class for a game of Tic-Tac-Toe:

|   |   |   |
|---|---|---|
| X |   |   |
|   | O |   |
| X |   | O |

```
public class TicTacToe
{
 private String[] [] board;
 private static final int ROWS = 3;
 private static final int COLS = 3;

 /** Construct an empty board. */
 public TicTacToe()
 {
 board = new String[ROWS][COLS];
 for (int r = 0; r < ROWS; r++)
 for (int c = 0; c < COLS; c++)
 board[r][c] = " ";
 }

 /** @param r the row number
 * @param c the column number
 * @param symbol the symbol to be placed on board[r][c]
 * Precondition: The square board[r][c] is empty.
 * Postcondition: symbol placed in that square.
 */
 public void makeMove(int r, int c, String symbol)
 {
 board[r][c] = symbol;
 }

 /** Creates a string representation of the board, e.g.
 * |o |
 * |xx|
 * | o|
 * @return the string representation of board
 */
 public String toString()
 {
 String s = ""; //empty string
 /* more code */
 return s;
 }
}
```

Which segment represents a correct replacement for */\* more code \*/* for the `toString` method?

(A) 

```
for (int r = 0; r < ROWS; r++)
{
 for (int c = 0; c < COLS; c++)
 {
 s = s + "|";
 s = s + board[r][c];
 s = s + "|\\n";
 }
}
```