

Practice Exam #1

SECTION I

Time — 1 hour and 15 minutes

Number of questions — 40

Percent of total grade — 50

- What is printed when the following statement is executed?

```
System.out.println(17 / 5 % 3 + 17 * 5 % 3);
```

- (A) 1
- (B) 4
- (C) 9
- (D) 42
- (E) 42.5

- Consider the following method.

```
public void process(String s)
{
    s = s.substring(2, 3) + s.substring(1, 2) +
        s.substring(0, 1);
}
```

What is printed as a result of executing the following statements (in a method in the same class)?

```
String s = "ABCD";
process(s);
System.out.println(s);
```

- (A) ABCD
- (B) CBA
- (C) CDBCA
- (D) CDBCAB
- (E) IndexOutOfBoundsException

3. Suppose `m` is a two-dimensional array of size 4 by 4, with all its elements initialized to zeroes. What will be the values stored in `m` after `fill(m)` is called? The method `fill` is defined as follows:

```
public void fill(int[][] m)
{
    int n = m.length;
    for (int i = 1; i < n - 1; i++)
    {
        for (int j = 1; j < n - 1; j++)
            m[i][j] = 1;
    }
}
```

- (A) 0000
0000
0000
0000
- (B) 1100
1100
0000
0000
- (C) 0000
0110
0110
0000
- (D) 1110
1110
1110
0000
- (E) 1111
1111
1111
1111
4. Which of the following statements will result in a syntax error?
- (A) String x = "123";
(B) Integer x = "123";
(C) Object x = "123";
(D) String[] x = {"123"};
(E) All of the above will compile with no errors.

5. What is printed as a result of executing the following statements?

```
double x = 2.5, y = 1.99;
System.out.println((int)(x/y) + (int)(x*y));
```

- (A) 0
- (B) 3
- (C) 4
- (D) 4.0
- (E) 5

6. Which of the following expressions is true if and only if NOT all three variables a, b, and c have the same value?

- (A) a != b && b != c
- (B) a != b || b != c
- (C) a >= b && b >= c && c >= a
- (D) a > b || b > c || a > c
- (E) !(a == b || b == c || a == c)

7. What is the result when the following code segment is compiled and executed?

```
int m = 4, n = 5;
double d = Math.sqrt((m + n)/2);
System.out.println(d);
```

- (A) Syntax error “sqrt(double) in java.lang.Math cannot be applied to int”
- (B) 1.5 is displayed
- (C) 2.0 is displayed
- (D) 2.1213203435596424 is displayed
- (E) ArithmeticException

8. For which of the following pairs of values a and b does the expression

$$(a > 20 \&\& a < b) \mid\mid (a > 10 \&\& a > b)$$

evaluate to true?

- (A) 5 and 0
- (B) 5 and 10
- (C) 15 and 10
- (D) 15 and 20
- (E) None of the above

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

```
List<Integer> lst = new ArrayList<Integer>();
for (int k = 1; k <= 6; k++)
    lst.add(new Integer(k));
for (int k = 0; k < 3; k++)
{
    Integer i = lst.remove(k);
    lst.add(i);
}
for (Integer i : lst)
    System.out.print(i);
```

- (A) 123456
(B) 456123
(C) 456321
(D) 246135
(E) IndexOutOfBoundsException
10. Which of the following methods are equivalent (always return the same value for the same values of input parameters)?

I. public boolean fun(int a, int b, int c)
{

```
    if (a >= b)
        if (b >= c)
            return true;
        else
            return false;
    else
        return false;
}
```

II. public boolean fun(int a, int b, int c)
{

```
    if (a >= b && b >= c)
        return true;
    else
        return false;
}
```

III. public boolean fun(int a, int b, int c)
{

```
    return a >= b || b >= c;
}
```

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

11. Consider the following class.

```
public class Matrix
{
    private int[][] m;

    /** Initializes m to a square n by n array with all
     *  the elements on the diagonal m[k][k] equal to 0 and
     *  all other elements equal to 1
     */
    public Matrix(int n)
    {
        m = new int[n][n];

        < missing code >
    }

    < other constructors and methods not shown >
}
```

Which of the following could replace <missing code> in Matrix's constructor, so that it compiles with no errors and works as specified?

I. `for (int r = 0; r < n; r++)
 for (int c = 0; c < n; c++)
 m[r][c] = 1;
 for (int k = 0; k < n; k++)
 m[k][k] = 0;`

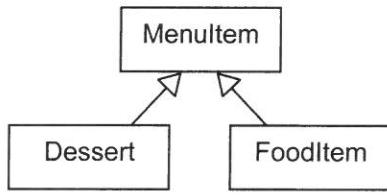
II. `for (int k = 0; k < n; k++)
 m[k][k] = 0;
 for (int r = 0; r < n; r++)
 for (int c = 0; c < n; c++)
 if (r != c)
 m[r][c] = 1;`

III. `for (int c = 0; c < n; c++)
 for (int r = 0; r < n; r++)
 if (r == c)
 m[r][c] = 0;
 else
 m[r][c] = 1;`

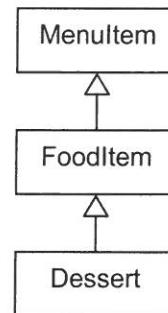
- (A) I only
- (B) II only
- (C) I and II only
- (D) II and III only
- (E) I, II, and III

12. Which of the following statements about developing a Java program is FALSE?
- (A) The main purpose of testing a Java program is to make sure it doesn't generate run-time exceptions.
 - (B) Hello.class, a Java class file obtained by compiling Hello.java under a Windows operating system can be executed on a Mac computer.
 - (C) The size of an int variable in Java is four bytes, the same under any operating system and computer model.
 - (D) At run time, int values are represented in the computer memory in the binary number system.
 - (E) Division of an integer value by zero in a Java program will cause a run-time ArithmeticException.
13. Amelia and Pat are working on a programming project for a restaurant. They are considering a hierarchy of classes that includes, among other classes, MenuItem, FoodItem, and Dessert. Pat, who likes to eat his dessert first, is proposing to make both FoodItem and Dessert direct subclasses of MenuItem; Amelia's proposal is to make FoodItem a subclass of MenuItem and Dessert a subclass of FoodItem.

Pat's design:



Amelia's design:



Which of the following is an advantage of Pat's design as compared to Amelia's?

- I. Both FoodItem and Dessert can reuse public methods of MenuItem.
 - II. Both FoodItem and Dessert type of object can be passed to a method that accepts a MenuItem as a parameter.
 - III. Pat's design better reflects the IS-A relationships between the three classes.
- (A) None of the three
 - (B) I only
 - (C) II only
 - (D) I and II only
 - (E) I, II, and III

14. Consider the following code segment.

```
int[] nums = new int[51];

for (int k = 0; k < nums.length; k++)
    nums[k] = 1;

for (int k = 3; k <= 50; k += 3)
    nums[k] = 0;

for (int k = 5; k <= 50; k += 5)
    nums[k] = 0;
```

How many elements in the array `nums` have the value 0 after this code has been executed?

- (A) 23
- (B) 25
- (C) 26
- (D) 27
- (E) 28

15. Consider the following method.

```
public int countSomething(int[] p)
{
    int count = 0;
    for (int i = 0; i < p.length; i++)
    {
        count++;
        int j = p[i];
        while (j != i)
        {
            j = p[j];
            count++;
        }
    }
    return count;
}
```

Given

```
int[] arr = {0, 2, 3, 1};
```

what will `countSomething(arr)` return?

- (A) 2
- (B) 3
- (C) 5
- (D) 10
- (E) 13

Questions 16-17 refer to the following classes.

```
public class Point
{
    private int x, y;

    public Point(int _x, int _y) { x = _x; y = _y; }
    public int getX() { return x; }
    public int getY() { return y; }
    public void move(int dx, int dy) { x += dx; y += dy; }
}

public class Polygon
{
    private ArrayList<Point> vertices;

    public Polygon() { vertices = new ArrayList<Point>(); }
    public void add(Point p) { vertices.add(p); }

    /** Returns the x-coordinate of the k-th vertex
     *  (counting from 0)
     */
    public int getX(int k)
    { return <missing expression>; }

    /** Moves every vertex of the polygon horizontally by dx
     *  and vertically by dy
     */
    public void move(int dx, int dy)
    { <missing code> }
}
```

16. Which of the following could replace <*missing expression*> in the `getX` method of the `Polygon` class?

- (A) `vertices[k].x`
- (B) `vertices.getX(k)`
- (C) `vertices.get(k).x`
- (D) `vertices.get(k).getX()`
- (E) `vertices.getX().get(k)`

17. Which of the following could replace <missing code> in Polygon's move method for it to work as specified?

- I.

```
for (int k = 0; k < vertices.size(); k++)
{
    Point p = vertices.get(k);
    p.x += dx;
    p.y += dy;
}
```
- II.

```
for (int k = 0; k < vertices.size(); k++)
    vertices.get(k).move(dx, dy);
```
- III.

```
for (Point p : vertices)
    p.move(dx, dy);
```

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

18. Suppose class C has a private int data field value:

```
public class C
{
    private int value;

    < other fields, constructors, and methods not shown >
}
```

Suppose we have a method

```
public static int compare(C x, C y)
{
    return x.value - y.value;
}
```

and we need to find a "home" for it: place it into some class. Where can we place this method so that it compiles with no errors?

- (A) Only into C
- (B) Only into C or any subclass of C
- (C) Only into C or any superclass of C
- (D) Into any class
- (E) This method will always cause a syntax error, no matter what class we place it in.

19. Classes `Salsa` and `Swing` implement an interface `Dance`. If both calls

```
    perform(new Salsa());
    perform(new Swing());
```

are valid, which of the following headers of the `perform` method(s) in a class `Dancer` will compile successfully?

- I. Two methods:

```
    public void perform(Salsa dance)
    public void perform(Swing dance)
```

- II. `public void perform(Dance dance)`

- III. `public void perform(Object dance)`

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

20. What are the contents of the array `nums` after the following code segment has been executed?

```
int[] nums = new int[8];
nums[0] = 0;
int n = 1;
while (n < nums.length)
{
    int k;
    for (k = n; k < 2*n; k++)
        nums[k] = nums[k-n] + 1;
    n = k;
}
```

- (A) 0 1 1 1 1 1 1 1
- (B) 0 1 0 1 0 1 0 1
- (C) 0 1 1 2 1 2 2 3
- (D) 0 1 2 3 1 2 3 4
- (E) 0 1 2 3 4 5 6 7

21. Consider the following method.

```
/** Rearranges the elements in words according to
 * the values stored in an integer array indices,
 * so that the element of words at index indices[k]
 * is moved to the element at index k.
 * Precondition: words.size() == indices.length
 */
public void permute(List<String> words, int[] indices)
{
    ArrayList<String> temp = new ArrayList<String>();
    < missing code >
}
```

For example, after executing the code segment

```
List<String> words = new ArrayList<String>();
words.add("I");
words.add("am");
words.add("Sam");
int[] indices = {2, 0, 1};
permute(words, indices);
```

words will become the list ["Sam", "I", "am"]. Which of the following code segments could replace <missing code> in the permute method?

- I.

```
for (String word : words)
    temp.add(word);
for (int k = 0; k < indices.length; k++)
    words.set(k, temp.get(indices[k]));
```
- II.

```
for (int j : indices)
    temp.add(words.get(j));
for (int k = 0; k < indices.length; k++)
    words.set(k, temp.get(k));
```
- III.

```
while (words.size() > 0)
    temp.add(words.remove(0));
for (int j : indices)
    words.add(temp.get(j));
```

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

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

```
ArrayList<String> digits = new ArrayList<String>();
for (int k = 0; k <= 9; k++)
    digits.add("") + k;

for (int k = 0; k <= 4; k++)
{
    String d1 = digits.remove(k);
    String d2 = digits.remove(k);
    digits.add(k, d1 + "+" + d2);
}
System.out.println(digits);
```

- (A) [0+1, 1+2, 2+3, 3+4, 4+5]
- (B) [0+1, 2+3, 4+5, 6+7, 8+9]
- (C) [0+1, 1+2, 2+3, 3+4, 5, 6, 7, 8, 9]
- (D) [0+1, 1+2, 2+3, 3+4, 4+5, 6, 7, 8, 9]
- (E) [0+0, 1+1, 2+2, 3+3, 4+4, 5, 6, 7, 8, 9]

23. Consider the following method.

```
/** Returns the number of zeroes in s.
 * Precondition: s.length() = 31; s holds several
 *                  0's followed by several 1's
 *                  (s can also be all 0's or all 1's)
 */
public int countZeroes(String s)
{
    int i = 0, j = 30;
    while (i <= j)
    {
        int k = (i + j) / 2;
        if (s.substring(k, k+1).equals("0"))
            i = k + 1;
        else
            j = k - 1;
    }
    return i;
}
```

How many iterations through the `while` loop will be made in the best and the worst case?

| | Best case | Worst case |
|-----|-----------|------------|
| (A) | 1 | 5 |
| (B) | 1 | 15 |
| (C) | 4 | 5 |
| (D) | 4 | 15 |
| (E) | 5 | 5 |

24. A database server processes queries based on their priority and time of arrival. The priority associated with each query is an integer in the range from 1 to 100. Two designs have been proposed:

Design 1: Keep all the queries in one list, sorted by their time of arrival.

Design 2: Keep the queries in 100 lists; each list holds all the queries of a given priority, sorted by their time of arrival.

Which of the following three operations will be more efficient with Design 1 than with Design 2?

- I. Process the earliest query of the highest priority
- II. Report how many queries have arrived within the last hour
- III. Create a single list of all queries sorted by priority and, within the same priority, by time of arrival.

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

25. Consider the following method.

```
private int swap(int a, int b)
{
    if (a < b)
    {
        b = a;
        a = b;
    }
    return b - a;
}
```

What are the values of the variables a, b, and c after the following statements are executed?

```
int a = 2, b = 5;
int c = swap(a, b);
```

- (A) 2, 5, 0
- (B) 2, 5, 3
- (C) 2, 5, -3
- (D) 2, 2, 0
- (E) 5, 2, 3

26. Consider the following class.

```
public class BankAccount
{
    private int balance;

    public BankAccount() { balance = 0; }
    public BankAccount(int amt) { balance = amt; }

    public int getBalance() { return balance; }
    public void makeDeposit(int amt) { balance += amt; }
}
```

What is the output when the following code segment (in a client class) is executed?

```
List<BankAccount> bank = new ArrayList<BankAccount>();
bank.add(new BankAccount());
bank.add(new BankAccount(5));
bank.add(new BankAccount(10));
bank.add(new BankAccount(15));
for (BankAccount customer : bank)
    customer.makeDeposit(10);
int total = 0;
for (BankAccount customer : bank)
    total += customer.getBalance();
System.out.println(total);
```

- (A) 0
- (B) 30
- (C) 40
- (D) 60
- (E) 70

27. The statement

```
System.out.println(Integer.MAX_VALUE);
```

prints 2147483647, which is equal to $2^{31} - 1$. What does the following statement print?

```
System.out.println(Integer.MAX_VALUE + 2);
```

- (A) 0
- (B) 2147483649
- (C) 2147483647.0
- (D) -2147483647
- (E) Nothing: it causes an ArithmeticException

28. Brad has derived his class from the library class JPanel. JPanel's paintComponent method displays a blank picture in a panel. Brad has redefined JPanel's paintComponent to display his own picture. Brad's class compiles with no errors, but when he runs the program, only a blank background is displayed. Which of the following hypotheses CANNOT be true in this situation?
- (A) Brad misspelled "paintComponent" in his method's name.
(B) Brad specified an incorrect return type for his paintComponent method.
(C) Brad chose the wrong type for a parameter in his paintComponent method.
(D) Brad specified two parameters for his paintComponent method, while JPanel's paintComponent takes only one parameter.
(E) Brad has a logic error in his paintComponent code that prevents it from generating the picture.
29. Suppose each pixel (picture element) in a digital image is represented by a 24-bit color value. How much memory does it take to store an uncompressed image of 2048 pixels by 1024 pixels?
- (A) 2 KB
(B) 2 MB
(C) 6 MB
(D) 6 GB
(E) 32 TB
30. Consider the following method.

```
public void printSomething(String s)
{
    int n = s.length();
    if (n < 1)
        return;
    String s1 = s.substring(1, n);
    printSomething(s1);
    System.out.println(s);
    printSomething(s1);
}
```

How many letters 'A' and how many letters total will be printed as a result of calling printSomething("ABCD")?

| | A's | Total |
|-----|-----|-------|
| (A) | 1 | 10 |
| (B) | 4 | 10 |
| (C) | 1 | 26 |
| (D) | 4 | 26 |
| (E) | 15 | 26 |

31. Consider the following method.

```
public int countSomething(int[] arr)
{
    int m = arr[0];
    int count = 1;
    for (int k = 1; k < arr.length; k++)
    {
        int a = arr[k];
        if (a > m)
        {
            m = a;
            count = 1;
        }
        else if (m == a)
            count++;
    }
    return count;
}
```

For which of the following arrays `countSomething` will return 3?

- (A) `int[] arr = {0, 1, 1, 1, 1};`
- (B) `int[] arr = {1, 6, 5, 4, 0};`
- (C) `int[] arr = {1, 0, 5, 6, 1};`
- (D) `int[] arr = {3, 2, 1, 0, 5};`
- (E) None of the above

32. The statement

```
Animal a = new Mammal("Elephant");
```

compiles with no errors. Which of the following situations will permit that?

- (A) `Mammal` is a class with a constructor that takes one parameter of the `String` type, and `Animal` is its subclass.
- (B) `Animal` is a class with a constructor that takes one parameter of the `String` type, `Mammal` is its subclass that has no constructors defined.
- (C) `Mammal` is a class with a constructor that takes one parameter of the `String` type, `Animal` is an interface, and `Mammal` implements `Animal`.
- (D) `Animal` has a public static data field `String Mammal`.
- (E) None of the above

33. What is printed when the following code segment is executed?

```
String[] xy = {"X", "Y"};
String[] yx = xy;
yx[0] = xy[1];
yx[1] = xy[0];
System.out.println(xy[0] + xy[1] + yx[0] + yx[1]);
```

- (A) XXXX
- (B) XYYX
- (C) XYXY
- (D) XYYY
- (E) YYYY

34. Consider the following interface TV and class MyTV.

```
public interface TV
{
    void tuneTo(String channel);
}

public class MyTV implements TV
{
    private ArrayList<String> myFavoriteChannels;

    public MyTV(ArrayList<String> channels)
    { /* implementation not shown */ }

    public void tuneTo(int k)
    { /* implementation not shown */ }

    public void tuneTo(int k, String name)
    { /* implementation not shown */ }
}
```

One of them has one or more errors and won't compile properly. Which of the following best describes the compiler errors reported for the code that is shown?

- (A) In the TV interface, the `tuneTo` method header is missing the keyword `public`
- (B) `MyTV` should be declared abstract; it does not define `tuneTo(String)`
- (C) `tuneTo` is defined more than once in `MyTV`
- (D) Cannot convert `int` to `String` in the `tuneTo` method in `MyTV`
- (E) Two errors: (1) `tuneTo` is defined more than once and (2) cannot convert `int` to `String` in the `tuneTo(int)` method in `MyTV`

Questions 35-36 refer to the following class Game and the incomplete class ChessGame.

```
public class Game {
    private String gameName;
    private List<String> players;

    public Game(String name)
    {
        gameName = name;
        players = new ArrayList<String>();
    }

    public Game(String name, String[] people)
    {
        gameName = name;
        players = new ArrayList<String>();
        for (String nm : people)
            players.add(nm);
    }

    public void addPlayer(String name) { players.add(name); }

    public String getPlayer(int k)
    {
        return players.get(k - 1);
    }

    public String toString()
    {
        return gameName + " game " + players.toString();
    }
}

public class ChessGame extends Game
{
    public ChessGame(String white, String black)
    {
        < missing code >
    }
}
```

35. Consider the following code segment in a Game's client class.

```
String[] players = {"Annette", "Bertrand",
                     "Claude", "Danielle"};
Game game = new Game("Bauernschnapsen", players);
System.out.println(<missing expression>);
```

Which of the following can replace <*missing expression*> so that the code results in printing "Annette"?

- (A) game.getPlayer(0)
- (B) game.getPlayer(1)
- (C) game.players.get(0)
- (D) game.players.get(1)
- (E) game.getPlayers().get(0)

36. Which of the following can replace <*missing code*> in ChessGame's constructor so that the statement

```
System.out.println(new ChessGame("Deep Blue",
                                  "Kasparov"));
```

prints

Chess game [Deep Blue, Kasparov]

- I. super("Chess", white, black);
- II. super("Chess");
 super.addPlayer(white);
 super.addPlayer(black);
- III. String[] players = {black, white};
 super("Chess", players);

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

37. Consider the following method:

```
public int[][] makeCounts(int n)
{
    int[][] counts = new int[3][n];
    counts[0][0] = 0;
    counts[1][0] = 0;
    counts[2][0] = 1;
    for (int k = 1; k < n; k++)
    {
        counts[0][k] = counts[0][k-1] + counts[1][k-1];
        counts[1][k] = counts[1][k-1] + counts[0][k-1] +
                      counts[2][k-1];
        counts[2][k] = counts[2][k-1] + counts[1][k-1];
    }
    return counts;
}
```

What values are in the array returned by `makeCounts(5)`?

(A)

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

(B)

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 |
| 0 | 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | 4 | 5 |

(C)

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 1 | 3 | 6 |
| 0 | 1 | 2 | 5 | 8 |
| 1 | 1 | 4 | 7 | 9 |

(D)

| | | | | |
|---|---|---|---|----|
| 0 | 0 | 1 | 3 | 8 |
| 0 | 1 | 2 | 5 | 12 |
| 1 | 1 | 2 | 4 | 9 |

(E)

| | | | | |
|---|---|---|---|----|
| 0 | 1 | 3 | 9 | 27 |
| 0 | 1 | 3 | 9 | 27 |
| 1 | 1 | 3 | 9 | 27 |

38. What is the output from the following code segment?

```
List<String> cities = new ArrayList<String>();
cities.add("Atlanta");
cities.add("Boston");
cities.add("Chicago");
for (String city : cities)
    city = city.substring(1);
System.out.println(cities);
```

- (A) [A, B, C]
- (B) [C, B, A]
- (C) [a, n, o]
- (D) [tlanta, oston, hicago]
- (E) [Atlanta, Boston, Chicago]

39. Consider the following two recursive versions of the method `choose(n, k)`.

Version 1

```
public static int choose(int n, int k)
{
    if (k == 0)
        return 1;
    else
        return choose(n, k-1) * (n-k+1)/k;
}
```

Version 2

```
public static int choose(int n, int k)
{
    if (k < 0 || k > n)
        return 0;
    else if (n == 0)
        return 1;
    else
        return choose(n-1, k-1) + choose(n-1, k);
}
```

When `choose(4, 2)` is called, how many times total, including the original call, will `choose` be called in each version?

| | Version 1 | Version 2 |
|-----|-----------|-----------|
| (A) | 2 | 7 |
| (B) | 2 | 19 |
| (C) | 3 | 7 |
| (D) | 3 | 19 |
| (E) | 3 | 27 |

40. Given two arrays of `double` values, sorted in ascending order, one with 100 elements, the other with 10 elements, how many comparisons will it take in an optimal algorithm to merge these arrays into one sorted array, in the best case and in the worst case?

| | Best case | Worst case |
|-----|-----------|------------|
| (A) | 10 | 109 |
| (B) | 50 | 110 |
| (C) | 100 | 110 |
| (D) | 109 | 999 |
| (E) | 100 | 1000 |



Answers and Solutions

Exam #1 ~ Multiple Choice

- | | | | |
|-------|-------|-------|-------|
| 1. A | 11. E | 21. E | 31. E |
| 2. A | 12. A | 22. B | 32. C |
| 3. C | 13. A | 23. E | 33. E |
| 4. B | 14. A | 24. B | 34. B |
| 5. E | 15. D | 25. A | 35. B |
| 6. B | 16. D | 26. E | 36. B |
| 7. C | 17. D | 27. D | 37. D |
| 8. C | 18. A | 28. B | 38. E |
| 9. D | 19. E | 29. C | 39. E |
| 10. A | 20. C | 30. C | 40. A |

Notes:

1. $17/5$ gives 3; $3 \% 3 = 0$; $85 \% 3 = 1$.
2. Strings are immutable; a method cannot change a `String` object.
3. The elements in the 2 by 2 square in the middle are set to 1.
4. Can't convert a `String` into an `Integer` — not allowed in Java.
5. $1 < x/y < 2$ and $4 < x*y < 5$; truncated to integers gives $1 + 4 = 5$.
6. It is easier to negate each of the expressions (using De Morgan's Laws when necessary) and check whether the result is equivalent to "all three values are the same." In Choice B, `!(a != b || b != c)` is the same as `(a == b && b == c)`, which is true if and only if `a == b` and `b == c` and `a == c`.
7. $(m + n)/2$ evaluates to 4.
8. Either way, `a > 10`. Try `a = 15` with `b = 10` or `b = 20`.
9. When the k -th element is removed from the list in the second `for` loop, the subsequent elements are shifted to the left, and their indices are decremented by one. As a result, this loop removes every other element, 1, 3, and 5 (with the original indices 0, 2, and 4). These values are added at the end of the list.
10. In I and II, `fun` returns `true` when `a >= b >= c`.
11. The code in Option I first sets all the elements to 1, then sets the elements on the diagonal to 0; the code in Option II first sets all the elements on the diagonal to 0, then all the elements not on the diagonal to 1; the code in Option III checks whether the element is on the diagonal or not and sets it accordingly. All three work.
12. Choice A is false: the main purpose of testing a program is to make sure it works as specified, not just that it doesn't crash.