

QUESTION 1

What is the value of AB_{16} when converted to binary?

- A. 10001001_2 B. 10011010_2 C. 10101011_2 D. 10111100_2 E. 11001101_2

QUESTION 2

How many *'s are output by the code to the right if x is initialized to 18?

- A. 18 B. 9 C. 7
D. 6 E. 5

```
int x;

// code to initialize x not shown

while (x > 0) {
    System.out.print('*');
    x = x / 2;
}
```

QUESTION 3

What is the running time of the while loop? Choose the most restrictive correct answer.

- A. $O(1)$ B. $O(\log x)$ C. $O(x)$
D. $O(x/2)$ E. $O(x^2)$

QUESTION 4

Which of these creates an object representing the book 2001, a science fiction book by Arthur Clarke?

- A. `Book b = new Book("2001", "Arthur Clarke", "science fiction");`
B. `Book b = new Book(2001, "Arthur Clarke", "science fiction");`
C. `Book b = Book("2001", "Arthur Clarke", "science fiction");`
D. `Book b = Book(2001, "Arthur Clarke", "science fiction");`
E. Both B and D

```
public class Book {

    public Book(String t, String a,
                  String g) {

        title = t;
        author = a;
        genre = g;
    }

    public String toString() {
        return <*1> + title + <*1> + ", a " +
            genre + " by " + author;
    }

    private String title, author, genre;
}
```

QUESTION 5

Which of these replaces <*1> in the code to the right to be a String containing only a double quotation mark?

- A. `"\""` B. `""` C. `\""`
D. `" "` E. `""\"`

QUESTION 6

What is the name of the static method that executes when the Java interpreter starts running a class?

- A. `main()` B. `start()` C. `execute()` D. `class()` E. `run()`

<p>QUESTION 7</p> <p>What is returned by <code>compute(-4)</code>?</p> <p>A. 0 B. "0"</p> <p>C. '0' D. 'a'</p> <p>E. An empty string</p>	<pre>public static String compute(int i) { String s = ""; while (i>0) { char c = (char) ('a' + i%10); s = s + c; i = i / 10; } return s; }</pre>
<p>QUESTION 8</p> <p>What is returned by <code>compute(12345)</code>?</p> <p>A. "abcde" B. "bcdef"</p> <p>C. "edcba" D. "fedcb"</p> <p>E. An empty string</p>	
<p>QUESTION 9</p> <p>What replaces <code><*1></code> in the code to the right to add one to <code>i</code>?</p> <p>A. <code>i++</code> B. <code>+i+</code></p> <p>C. <code>++i</code> D. Either A or C</p> <p>E. A, B, or C</p>	<pre>for (int i = 0; i < 10; <*1>) { // code not shown }</pre>
<p>QUESTION 10</p> <p>What is output by the code to the right?</p> <p>A. 01 B. falsetrue</p> <p>C. 10 D. truefalse</p> <p>E. falseh</p>	<pre>char c = 'h'; System.out.print(Character.isDigit(c)); System.out.print(Character.isLetter(c));</pre>
<p>QUESTION 11</p> <p>How many of the comparison operators are evaluated when <code>b</code> is calculated?</p> <p>A. 0 B. 1 C. 2</p> <p>D. 3 E. 4</p>	<pre>int x = 3, y = 4, z = 12; boolean b; b = ((x+y < z) (x*y < z)) && ((x*z < y) && (x+z > y));</pre>
<p>QUESTION 12</p> <p>What is output by the code to the right (where <code>_</code> represents a blank space)?</p> <p>A. 23 B. 20_3 C. 203_</p> <p>D. 02_3 E. 023_</p>	<pre>int x = 2, y = 3; System.out.printf("%02d%-2d", x, y);</pre>

QUESTION 13

What is the difference between `x` and `y`?

- A. Outside classes can access `y`, but not `x`
- B. Once `y` has been given a value it cannot be changed
- C. More storage space is used to hold `y` than `x`
- D. Each `Test` object has its own copy of `x`, but all `Test` objects share one copy of `y`
- E. A `throw` statement can throw `x` but not `y`

```
public class Test {

    public Test(int x) {
        this.x = x;
    }

    private int x;
    private static int y;
}
```

QUESTION 14

If you modified this class to initialize `y` to 5, how should it be done so that the initialization happens only once?

- A. Inside the constructor add the statement:
`y = 5;`
- B. Add a static block that has the statement:
`y = 5;`
- C. Change the declaration of `y`:
`private static int y = 5;`
- D. Either A or B
- E. Either B or C

QUESTION 15

What is returned by the static method call `makeString("001230")`?

- A. "AABBCA" B. "AABACB"
- C. "AAABBC" D. "ABAABC"
- E. A run-time error occurs when `i` is 4

```
public static String makeString(String s) {
    String out = "";
    for (int i=0; i<s.length(); ++i)
        switch(s.charAt(i)) {
            case '0': out = "A" + out;
                     break;
            case '1': out = "B" + out + "B";
                     break;
            case '2': out = out + "C";
                     break;
        }
    return out;
}
```

QUESTION 16

Which of these strings causes `makeString()` to return "ABACAB" when passed as a parameter?

- A. "00210"
- B. "ABACAB"
- C. "02010"
- D. None of these, but there is a string which gives the return value
- E. None of these, and no string gives the return value

QUESTION 17

What does the array below look like after it is passed to the static method `process()`?

1	3	-2	8	4	2
---	---	----	---	---	---

- A.

-2	3	-2	8	8	3
----	---	----	---	---	---
- B.

-2	-2	-2	8	8	8
----	----	----	---	---	---
- C.

-2	8	-2	8	8	-2
----	---	----	---	---	----
- D.

3	3	-2	8	3	3
---	---	----	---	---	---
- E.

-2	-2	-2	8	8	-2
----	----	----	---	---	----

```
public static void process(int[] intArray)
{
    int min = intArray[0];
    int max = intArray[0];
    for (int i=1; i<intArray.length; ++i) {
        if (intArray[i]<min)
            min = intArray[i];
        else if (intArray[i]>max)
            max = intArray[i];
    }
    int mid = (min + max) / 2;
    for (int i=0; i<intArray.length; ++i)
        if (intArray[i]<mid)
            intArray[i] = min;
        else
            intArray[i] = max;
}
```

QUESTION 18

What happens if `null` is passed as a parameter to `process()`?

- A. A `NullPointerException` is thrown by the first statement of `process()`
- B. An `ArrayIndexOutOfBoundsException` is thrown by the first statement of `process()`
- C. A compile-time error occurs
- D. A new array is created to process
- E. The method completes without errors

QUESTION 19

Which of these expressions correctly assigns the value in `x` to variable `y`?

- A. `y = x` B. `y = (byte)x`
- C. `y = (int)x` D. Both A and B
- E. Both A and C

```
int x = 10;
byte y;
```

QUESTION 20

Which of these expressions decrements `int x` and sets `int y` to the new value of `x`?

- A. `y.equal(-x)` B. `y = x-1` C. `y = x--` D. `y = --x` E. `y : x - 1`

<p>QUESTION 21</p> <p>What replaces <*1> in the code to the right so that "true" is printed for any class MyClass?</p> <p>A. Superclass B. super</p> <p>C. Object D. this</p> <p>E. null</p>	<pre>// Assume MyClass has a default constructor MyClass m = new MyClass(); if (m instanceof <*1>) System.out.print("true");</pre>
<p>QUESTION 22</p> <p>What replaces <*1> in the code to the right to tell whether data member elements is empty?</p> <p>A. elements</p> <p>B. elements.empty()</p> <p>C. elements.size()==0</p> <p>D. elements.isEmpty()</p> <p>E. elements != null</p>	<pre>public class Stack<E> { public Stack() { elements = new ArrayList<E>(); } public void push(E element) { elements.add(element); } public E pop() { return elements.remove(elements.size()-1); } public boolean isEmpty() { return <*1>; } private ArrayList<E> elements; public static void test() { <*2> s.push(15); s.push(17); s.push(35); System.out.print(s.pop()); System.out.print(s.pop()); System.out.print(s.pop()); } }</pre>
<p>QUESTION 23</p> <p>Which of these replaces <*2> in the code to the right as a valid declaration and initialization of a stack of integer values?</p> <p>A. Stack<Integer> s = new Stack<Integer>();</p> <p>B. Stack<Integer> s = new Stack<Integer>;</p> <p>C. Stack<int> s = new Stack<int>;</p> <p>D. Stack<int> s = new Stack();</p> <p>E. More than one of these</p>	
<p>QUESTION 24</p> <p>Assume <*1> and <*2> are filled in correctly. What is output by the test() static method?</p> <p>A. 151735 B. 351715</p> <p>C. 151515 D. 353535</p> <p>E. An exception is thrown</p>	
<p>QUESTION 25</p> <p>What is the largest positive number that can be stored in data type short?</p> <p>A. 32766 B. 32767 C. 32768 D. 32769 E. 32770</p>	

QUESTION 26

What does A look like after the static method call `find(A, 14)` where A is the array of Integer values below?

-4	0	12	16	19	25
----	---	----	----	----	----

- A.

-4	0	12	16	19	25
----	---	----	----	----	----
- B.

25	19	16	12	0	-4
----	----	----	----	---	----
- C.

14	14	14	14	14	14
----	----	----	----	----	----
- D.

-4	0	12	14	16	19	25
----	---	----	----	----	----	----
- E.

25	19	16	14	12	0	-4
----	----	----	----	----	---	----

```
public static boolean find(
    Comparable[] array,
    Comparable item) {
    if (array == null) return false;
    int front = 0, back = array.length;
    do {
        int mid = (front + back)/2;
        if (array[mid].compareTo(item)==0)
            return true;
        else if (array[mid].compareTo(item)<0)
            front = mid+1;
        else
            back = mid-1;
    } while (front<=back);
    return false;
}
```

QUESTION 27

What is returned by the static method call in the previous question?

- A. true B. false C. 3
- D. 14 E. An exception is thrown

QUESTION 28

Which of these loops removes all copies of the string "TEST" from the list `myList`?

- A.

```
while (iter.next()) {
    if (iter.equals("TEST"))
        iter.remove();
}
```
- B.

```
while (iter.hasNext()) {
    String s = iter.next();
    if (s.equals("TEST")) {
        iter.previous();
        iter.remove();
    }
}
```
- C.

```
while (iter.hasNext()) {
    String s = iter.next();
    if (s.equals("TEST"))
        iter.remove();
}
```
- D. Either A or C
- E. Either B or C

```
List<String> myList;
// code to construct and initialize myList
// not shown

ListIterator<String> iter =
    myList.listIterator();
```

QUESTION 29

Which of these replaces **<*1>** in the code to the right to initialize data member `board` to a two-dimensional array with a number of rows and columns equal to `size` and with all entries initialized to 0?

- A. `board = new int[size][size];`
- B. `board = new int[size, size];`
- C. `board = new int[size][];`
`board[0] = new int[size];`
- D. Either A or C
- E. None of these initialize the entries to 0

For the remaining questions, assume that **<*1>** has been filled in correctly.

QUESTION 30

What is output by the code below?

```
DominoGame g = new DominoGame(4);
g.makeMove(0,0);
g.makeMove(1,1);
g.makeMove(3,1);
g.displayBoard();
```

- | | |
|---|---|
| <ul style="list-style-type: none"> A. OO.. .X.. .X.. .OO. | <ul style="list-style-type: none"> B. XX.. .O.. .O.. .XX. |
| <ul style="list-style-type: none"> C. O..O OXXO | <ul style="list-style-type: none"> D. X..X XOOX |
- E. No output because an exception is thrown

QUESTION 31

Suppose a method is added to check whether the current player has any valid move by repeatedly calling `validMove()`. What is the minimum number of calls necessary to check all possible moves if the board is 4x4?

- | | | |
|---|---|---|
| <ul style="list-style-type: none"> A. 12 D. 8 | <ul style="list-style-type: none"> B. 16 E. 9 | <ul style="list-style-type: none"> C. 20 |
|---|---|---|

```
// Players cover a square grid with
// "dominos" (2x1 rectangles) until
// there is no place for a player
// to cover. One plays dominos
// horizontally and one vertically.
```

```
public class DominoGame {

    public DominoGame(int size) {
        <*1>
        player = 1;
    }

    public void makeMove(int row, int col) {
        int row2 = row, col2 = col;
        if (player == 1) col2++;
        else row2++;
        if (board[row][col] == 0 &&
            board[row2][col2] == 0)
            board[row][col] =
                board[row2][col2] = player;
        else
            throw new IllegalArgumentException(
                "Invalid move");
        player *= -1;
    }

    public void displayBoard() {
        for (int[] row : board) {
            for (int piece : row) {
                if (piece == 1)
                    System.out.print("X");
                else if (piece == -1)
                    System.out.print("O");
                else System.out.print(".");
            }
            System.out.println();
        }
    }

    public boolean validMove(int row,
                             int col) {
        int row2 = row, col2 = col;
        if (player == 1) col2++;
        else row2++;
        if (board[row][col] == 0 &&
            board[row2][col2] == 0)
            return true;
        else return false;
    }

    private int[][] board;
    private int player;
}
```

QUESTION 32

What sorting algorithm is implemented by the static method `sort()`?

- A. Bubble sort B. Shell sort
C. Insertion sort D. Quick sort
E. Merge sort

```
public static void sort(int[] array) {
    sort(array, 0, array.length);
}

public static void sort(int[] array,
                        int i, int j) {
    if (i+1 == j) return;
    sort(array, i, (i+j)/2);
    sort(array, (i+j)/2, j);
```

QUESTION 33

Suppose `int[] array` shown below is sorted by calling `sort(array)`. What does `array` look like at the point marked by the comment during the execution of `sort(array, 0, 6)`, that is, after the completion of the recursive calls `sort(array, 0, 3)` and `sort(array, 3, 6)`?

17	25	-8	99	3	10
----	----	----	----	---	----

- A.

-8	3	10	17	25	99
----	---	----	----	----	----

B.

17	25	-8	99	3	10
----	----	----	----	---	----

C.

-8	17	25	3	10	99
----	----	----	---	----	----

D.

3	-8	10	17	99	25
---	----	----	----	----	----

E.

3	10	-8	17	99	25
---	----	----	----	----	----

```
// see Question 33

int[] temp = new int[j-i];
int x = i;
int mid = (i+j)/2;
int y = mid;
int z = 0;
while (x<mid && y<j)
    if (array[x]<array[y])
        temp[z++] = array[x++];
    else
        temp[z++] = array[y++];
while (x<mid)
    temp[z++] = array[x++];
while (y<j)
    temp[z++] = array[y++];
for (z=0; z<temp.length; ++z)
    array[i+z] = temp[z];
}
```

QUESTION 34

What is the value of `x` after executing the code to the right?

- A. 0 B. 1 C. 10
D. 15 E. Infinite loop

```
int x = 0, y = 10;
while (y>0)
    for (int i=0; i<10; ++i) {
        if (x<i) { x+=i; break; }
        if (y == i) continue;
        --y;
    }
```

QUESTION 35

Suppose a user-defined class overrides the `equals()` method. Which of these statements is true?

- A. For `HashMap` to work correctly with objects of the new type as keys, the class must override `toString()` B. For `HashMap` to work correctly with objects of the new type as keys, the class must override `hashCode()`
C. For `TreeMap` to work correctly with objects of the new type as keys, the class must override `toString()` D. For `TreeMap` to work correctly with objects of the new type as keys, the class must override `hashCode()`
E. Both C and D

QUESTION 36

What is the value of `a.x` after executing this declaration?

```
A a = new A(10);
```

- A. 0 B. 5 C. 10
D. 15 E. 20

```
public class A {
    public A(int z) {
        int x = 5;
        y = z;
    }
    private int x, y;
}
```

QUESTION 37

What replaces `<*1>` in the code to the right to call the other `SuperClass` constructor?

- A. `Class` B. `SuperClass`
C. `this` D. `constructor`
E. `super`

```
public class SuperClass {
    public SuperClass() {
        <*1>(10);
    }
    public SuperClass(int x) {
        System.out.print(x);
    }
}
```

QUESTION 38

Assume `<*1>` is filled in correctly. What is output when a `SubClass` object is constructed?

- A. 10 B. 2510
C. 25 D. 1025
E. nothing

```
public class SubClass extends SuperClass {
    public SubClass() {
        System.out.print(25);
    }
}
```

QUESTION 39

Suppose `i` is initialized to refer to a file with integer values separated by colons. What replaces `<*1>` in the code to the right so that the code sums up all the numbers in `i`?

- A. `s.read(":")`
B. `s.split(":")`
C. `s.separate(":")`
D. `s.isWhiteSpace(":")`
E. `s.useDelimiter(":")`

```
InputStream i;
//code to initialize i not shown
Scanner s = new Scanner(i);
<*1>;
int sum = 0;
while (s.hasNextInt())
    sum += s.nextInt();
```

QUESTION 40

Which of these keywords begins a block of code that is executed whether or not the code in a prior `try` block throws an exception?

- A. `main` B. `catch` C. `do` D. `finally` E. `always`