

A+ Computer Science

M/C Written Test

General Directions:

- 1) DO NOT OPEN EXAM UNTIL TOLD TO DO SO.
- 2) **NO CALCULATORS of any kind may be used.**
- 3) You have 45 minutes to complete this contest. If you are in the process of actually writing an answer when the signal to stop is given, you may finish writing that answer.
- 4) Papers may not be turned in until forty-five minutes have elapsed. If you finish the test before the end of the allotted time, remain at your seat and retain your paper until told to do otherwise. You may use this time to check your answers.
- 5) All answers must be written on the answer sheet/Scantron card provided. Indicate your answers in the appropriate blanks provided on the answer sheet or on the Scantron card. Clean erasures are necessary for accurate Scantron grading.
- 6) You may place as many notations as you desire anywhere on the test paper except on the answer sheet or Scantron card which is reserved for answers only.
- 7) You may use additional scratch paper provided by the contest director.
- 8) All questions have ONE and only ONE correct (BEST) answer. There is a penalty for all incorrect answers. **All provided code segments are intended to be syntactically correct, unless otherwise stated (i.e. `error` is an answer choice). Ignore any typographical errors and assume any undefined variables are defined as used.**
- 9) A reference to commonly used Java classes is provided with the test and you may use this reference during the contest. You may detach the reference sheets from the test booklet but DO NOT DO SO UNTIL THE CONTEST BEGINS.
- 10) Assume that any necessary import statements for Standard Java 23 Packages and classes (e.g. `.lang`, `.util`, `System`, `Math`, `Double`, etc.) are included in any programs or code segments that refer to methods from these classes and/or packages.

Scoring:

- 1) All questions will receive 6 points if answered correctly; no points will be given or subtracted if unanswered; 2 points will be deducted for each incorrect answer.

For more Computer Science practice tests and materials,

go to www.apluscompsci.com

Standard Classes and Interfaces — Supplemental Reference

class java.lang.Object

- o boolean equals(Object other)
- o String toString()
- o int hashCode()

interface java.lang.Comparable<T>

- o int compareTo(T other)
Return value < 0 if this is less than other.
Return value = 0 if this is equal to other.
Return value > 0 if this is greater than other.

class java.lang.Integer implements

Comparable<Integer>

- o Integer(int value)
- o int intValue()
- o boolean equals(Object obj)
- o String toString()
- o int compareTo(Integer anotherInteger)
- o static int parseInt(String s)

class java.lang.Double implements

Comparable<Double>

- o Double(double value)
- o double doubleValue()
- o boolean equals(Object obj)
- o String toString()
- o int compareTo(Double anotherDouble)
- o static double parseDouble(String s)

class java.lang.String implements

Comparable<String>

- o int compareTo(String anotherString)
- o boolean equals(Object obj)
- o int length()
- o String substring(int begin, int end)
Returns the substring starting at index begin and ending at index (end - 1).
- o String substring(int begin)
Returns substring(from, length()).
- o int indexOf(String str)
Returns the index within this string of the first occurrence of str. Returns -1 if str is not found.
- o int indexOf(String str, int fromIndex)
Returns the index within this string of the first occurrence of str, starting the search at the specified index.. Returns -1 if str is not found.
- o charAt(int index)
- o int indexOf(int ch)
- o int indexOf(int ch, int fromIndex)
- o String toLowerCase()
- o String toUpperCase()
- o String[] split(String regex)
- o boolean matches(String regex)

class java.lang.Character

- o static boolean isDigit(char ch)
- o static boolean isLetter(char ch)
- o static boolean isLetterOrDigit(char ch)
- o static boolean isLowerCase(char ch)
- o static boolean isUpperCase(char ch)
- o static char toUpperCase(char ch)
- o static char toLowerCase(char ch)

class java.lang.Math

- o static int abs(int a)
- o static double abs(double a)
- o static double pow(double base, double exponent)
- o static double sqrt(double a)
- o static double ceil(double a)
- o static double floor(double a)
- o static double min(double a, double b)
- o static double max(double a, double b)
- o static int min(int a, int b)
- o static int max(int a, int b)
- o static long round(double a)
- o static double random()
Returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.

interface java.util.List<E>

- o boolean add(E e)
- o int size()
- o Iterator<E> iterator()
- o ListIterator<E> listIterator()
- o E get(int index)
- o E set(int index, E e)
Replaces the element at index with the object e.
- o void add(int index, E e)
Inserts the object e at position index, sliding elements at position index and higher to the right (adds 1 to their indices) and adjusts size.
- o E remove(int index)
Removes element from position index, sliding elements at position (index + 1) and higher to the left (subtracts 1 from their indices) and adjusts size.

class java.util.ArrayList<E> implements List<E>

class java.util.LinkedList<E> implements List<E>, Queue<E>

Methods in addition to the List methods:

- o void addFirst(E e)
- o void addLast(E e)
- o E getFirst()
- o E getLast()
- o E removeFirst()
- o E removeLast()

class java.util.Stack<E>

- o boolean isEmpty()
- o E peek()
- o E pop()
- o E push(E item)

interface java.util.Queue<E>

- o boolean add(E e)
- o boolean isEmpty()
- o E peek()
- o E remove()

class java.util.PriorityQueue<E>

- o boolean add(E e)
- o boolean isEmpty()
- o E peek()
- o E remove()

interface java.util.Set<E>

- o boolean add(E e)
- o boolean contains(Object obj)
- o boolean remove(Object obj)
- o int size()
- o Iterator<E> iterator()
- o boolean addAll(Collection<? extends E> c)
- o boolean removeAll(Collection<?> c)
- o boolean retainAll(Collection<?> c)

class java.util.HashSet<E> implements Set<E>

class java.util.TreeSet<E> implements Set<E>

interface java.util.Map<K,V>

- o Object put(K key, V value)
- o V get(Object key)
- o boolean containsKey(Object key)
- o int size()
- o Set<K> keySet()
- o Set<Map.Entry<K, V>> entrySet()

class java.util.HashMap<K,V> implements Map<K,V>

class java.util.TreeMap<K,V> implements Map<K,V>

interface java.util.Map.Entry<K,V>

- o K getKey()
- o V getValue()
- o V setValue(V value)

interface java.util.Iterator<E>

- o boolean hasNext()
- o E next()
- o void remove()

**interface java.util.ListIterator<E> extends
java.util.Iterator<E>**

Methods in addition to the Iterator methods:

- o void add(E e)
- o void set(E e)

class java.lang.Exception

- o Exception()
- o Exception(String message)

class java.util.Scanner

- o Scanner(InputStream source)
- o boolean hasNext()
- o boolean hasNextInt()
- o boolean hasNextDouble()
- o String next()
- o int nextInt()
- o double nextDouble()
- o String nextLine()
- o Scanner useDelimiter(String pattern)

Note: Correct responses are based on **Java SE Development Kit 23 (JDK 23)** from Oracle, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g., "error" is an answer choice) and any necessary Java SE 23 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used. **For all output statements, assume that the System class has been statically imported using: `import static java.lang.System.*`**

QUESTION 1

Which of the following is equivalent to the number 10101_5 ?

- A. 802_9 B. 462_{12} C. 363_{15} D. $28B_{16}$ E. None are equivalent.

QUESTION 2

What is output by the code to the right?

- A. 28 B. 45
C. 52 D. 0
E. There is no output due to an error.

```
out.println(33 / 7 * 5 + 12 * 6 / 9);
```

QUESTION 3

What is output by the code to the right (z's represent spaces)?

- A. 42|2c|false B. 42|2e|111010
C. 42|2d|true D. 42|2d|111000
E. There is no output due to an error.

```
out.printf("%o|%x|%b", 34, 45, 56);
```

QUESTION 4

What is output by the code to the right?

- A. eaterLra
B. oeaterLra
C. oeaterLara
D. oeaterL211
E. There is no output due to an error.

```
String g = "LakeTahoe";
String h = "CraterLake";
g += h.substring(2,7);
g += g.charAt(12) + g.charAt(9);
g = g.substring(7);
out.println(g);
```

QUESTION 5

What is output by the code to the right?

- A. True B. False C. false D. true
E. There is no output due to a syntax error.

```
boolean a = true;
boolean b = false;
a |= b && a ^ b;
b &= a ^ b || a;
out.println(a && b || a ^ b);
```

QUESTION 6

What is output by the code to the right?

- A. 7 B. 5 C. 6 D. 5.0
E. There is no output due to an error.

```
int g = 234;
double f = Math.cbrt(g);
out.println(Math.round(f));
```

QUESTION 7

What is the output by the code to the right?

- A. 24 B. 13
C. 23 D. 134
E. There is no output due to an error.

```
int g = 9;
if(g % 4 != 0)
    out.print(1);
else if(g % 3 == 0)
    out.print(2);
if(g / 2 > 2)
    out.print(3);
out.println(4);
```

<p>QUESTION 8</p> <p>What is the output by the code to the right?</p> <p>A. 170 B. 152 C. 168 D. 165 E. There is no output due to an error.</p>	<pre>int sum = 0; for(int y = 0; y < 10; y++) for(int x = 0; x < y; x++) sum += x + 1; out.println(sum);</pre>
<p>QUESTION 9</p> <p>What is output by the code to the right?</p> <p>A. 195 B. 188 C. 192 D. 181 E. There is no output due to an error.</p>	<pre>out.println(43 * 3 212 + 13^34);</pre>
<p>QUESTION 10</p> <p>What is output by the code to the right?</p> <p>A. 26 37 B. 29 36 C. 27 32 D. 32 29 E. There is no output due to an error.</p>	<pre>int[] arr = new int[10]; for(int y = 0; y < 10; y++) arr[y] = (10 - y) * (y + 2); arr[3] = arr[1]++ + arr[7]; arr[5] *= arr[6]++; Arrays.sort(arr); out.println(arr[3]+" "+arr[5]);</pre>
<p>QUESTION 11</p> <p>Which of the following imports is required so that the code to the right will compile without error?</p> <p>A. import java.util.*; B. import java.io.*; C. import java.lang.*; D. All of the above. E. A and B.</p>	<pre>Scanner file=new Scanner (new File("template.dat")); int times=file.nextInt(); file.nextLine(); while(times-->0) { out.println(file.nextLine()); }</pre>
<p>QUESTION 12</p> <p>How many asterisks are output by the code to the right?</p> <p>A. 5 B. 6 C. 7 D. 8 E. There is no output due to an infinite loop.</p>	<pre>long g = 212; while(g >= 5) { out.print("*"); g /= 5; g *= 3; }</pre>
<p>QUESTION 13</p> <p>What is the correct order of precedence for the operators to the right ?</p> <p>A. II, III, IV, I B. I, II, III, IV C. I, IV, III, II D. I, III, II, IV E. I, III, IV, II</p>	<p>I. ++ II. >= III. << IV. ==</p>
<p>QUESTION 14</p> <p>What is the output by the code to the right ?</p> <p>A. 4 B. 32 C. 76 D. 8 E. 16</p>	<pre>int[] arr = new int[] { Short.SIZE,Integer.BYTES, Long.BYTES,Byte.SIZE, Float.BYTES,Double.SIZE }; Arrays.sort(arr); int i = arr[1] + arr[3]; out.println(i + arr[5]);</pre>

QUESTION 15

Which of the following is equivalent to 321_8 ?

- A. 207_{10} B. 250_9 C. 415_7 D. $E1_{16}$ E. 155_{12}

QUESTION 16

What is output by the code to the right?

- A. [Hello, Friend]
 B. [Hello, Friend, Old]
 C. [Hello, Darkness, Friend, Old]
 D. [Hello, Friend, Old, Darkness]
 E. There is no output due to an error.

```
ArrayList<String> a;
a = new ArrayList<String>();
a.add("Hello");
a.add("Darkness");
a.add("My");
a.set(1, "Old");
a.add(1, "Friend");
a.remove(a.size() - 1);
out.println(a);
```

QUESTION 17

What is output by the code to the right?

- A. 11 B. 8
 C. 7 D. 9
 E. There is no output due to an error.

```
int g = 5, h = 15, n = 0;
while(g <= h) {
    g = g * 2 - n * 2;
    h += h / 2 + n * 6;
    n++;
}
out.println(n);
```

QUESTION 18

What is output by the code to the right?

- A. 0
 B. 1
 C. MAX
 D. There is no output due to a compile error.
 E. There is no output due to a runtime error.

```
int g = Integer.MAX_VALUE;
while (++g <= 0);
out.println(g==Integer.MAX_VALUE?"MAX":g);
```

QUESTION 19

What is output by the code to the right?

- A. 285. 134 e1211e
 B. f 134 e b205 s.
 C. 43100e 134 e103
 D. 01232.ne143100e
 E. There is no output due to an error.

```
String g = "One of the boys.";
for(int y = 0; y < 9; y++) {
    g += g.charAt(y);
    g = g.substring(y);
    g += g.substring(y, y*2);
    g += g.charAt(2) + g.charAt(3);
}
out.println(g.substring(5,20));
```

QUESTION 20

Which of the following is not defined as a static class in Java?

- A. Math B. Integer C. Arrays D. Collections E. None of the above.

QUESTION 21

Which data structure is implemented in the code to the right?

- A. Max Heap B. Red Black Tree
 C. Min Heap D. Priority Queue
 E. Redwood Tree

```
TreeSet<Integer> ts;
ts = new TreeSet<Integer>();
for(int y = 0; y < 15; y++)
    ts.add(y + (int) (Math.random() * 20));
```

QUESTION 22

What could replace the **<1*>** in the code to the right so that the constructor for class C works without error and the `str` variable of the parent class is initialized as "C"?

- A. Nothing is required.
- B. `super(n, "C")`
- C. `super(n)`
- D. A or B.
- E. All of the above.

QUESTION 23

What could replace the **<2*>** so that the `count` variable of class A is incremented by 1?

- A. `count++`
- B. `A.count++`
- C. `super.count++`
- D. A and B.
- E. All of the above.

QUESTION 24

Assuming **<1*>** and **<2*>** are filled in correctly, what is output by the line marked `//q24` in the code to the right?

- A. 8
- B. 5
- C. 9
- D. 4
- E. There is no output due to an error.

QUESTION 25

Assuming **<1*>** and **<2*>** are filled in correctly, what is output by the section marked `//q25` in the code to the right?

- A. 5 9 11 5
- B. 5 5 5 5
- C. 5 9 11 12
- D. 5 9 5 12
- E. There is no output due to an error.

QUESTION 26

Assuming **<1*>** and **<2*>** are filled in correctly, what is output by the section marked `//q26` in the code to the right?

- A. BBCCDD
- B. BBCCCC
- C. BBBBBD
- D. BBCC
- E. There is no output due to an error.

```
class A{
    private int num;
    static int count;

    public A() {
        <2*>;
        num = 5;
    }

    public int getNum() {
        return num;
    }
}

class B extends A{
    private int num;
    String str;

    public B(int n, String s) {
        <2*>;
        num = n;
        str = s;
    }

    public String getStr() {
        return str+str;
    }
}

class C extends B{
    public int i;

    public C(int n, int i) {
        <1*>;
        this.i = i;
        <2*>;
    }
}

//////////client code//////////
A a = new A();
B b = new B(9, "B");
B c = new C(11, 3);
A d = new B(12, "D");
out.println(A.count); //q24

//section q25
String q25 = "" + a.getNum();
q25 += " " + b.getNum() + " ";
q25 += c.getNum() + " " + d.getNum();
out.println(q25);

//section q26
String q26 = b.getStr();
q26 += c.getStr();
q26 += d.getStr();
out.println(q26);
```

QUESTION 27

What is output by the code to the right?

- A. true true
- B. true false
- C. false true
- D. false false
- E. There is no output due to an error.

```
String a = "Out.Println";
String b = "What It Do ?";
String reg = "\\w{2,3}.P[a-z]+";
out.print(a.matches(reg)+" ");
reg = "([A-Za-z]+ )*?";
out.println(b.matches(reg));
```

QUESTION 28

What is output by the code to the right?

- A. 6 P
- B. 2 Y
- C. 0 K
- D. 8 R
- E. There is no output due to an error.

```
char c1 = '0';
char c2 = 'A';
for(int y = 0; y < 30; y++) {
    c1 = (char)((c1 * 2) % 10 + '0');
    c2 = (char)((c2 + y * 5) % 26 + 'A');
}
out.println(c1+" "+c2);
```

QUESTION 29

If a Insertion sort algorithm takes 4 seconds to process an array of length 10000, how long will it take to process a list of length 100000?

- A. 400 seconds.
- B. 40 seconds.
- C. 16 seconds.
- D. 160 seconds.
- E. 200 seconds.

QUESTION 30

What is output by the line marked //q30 in the code to the right?

- A. 26
- B. 20
- C. 18
- D. 12
- E. There is no output due to an error.

```
public static int recur(int n) {
    if(n < 0)
        return 1;
    else if (n % 3 == 2)
        return 2 * recur(n - 3);
    else if (n % 4 != 2)
        return 4 + recur(n - 2);
    return 3 + 3 * recur(n - 1);
}
```

QUESTION 31

What is output by the line marked //q31 in the code to the right?

- A. 31
- B. 36
- C. 29
- D. 39
- E. There is no output due to an error.

```
////////////////client code////////////////
out.println(recur(9)); //q30
out.println(recur(16)); //q31
```


QUESTION 32

What could replace the **<1*>** in the code to the right so that the remove method removes the Node at index i in the Structure class?

- A. root
- B. get(i)
- C. new Node(i)
- D. A or B.
- E. All of the above.

QUESTION 33

What could replace the **<2*>** in the code to the right so that the Structure class works as intended?

- A. root.next = root;
- B. root.prev = root;
- C. add(root);
- D. A and B are both required.
- E. All of the above are required.

QUESTION 34

What is output by the line marked //q34 in the code to the right?

- A. One B. Two
- C. Three D. Four
- E. There is no output due to an error.

QUESTION 35

What is output by the line marked //q35 in the code to the right?

- A. One Two Four Five
- B. Five Four One Two
- C. One Three Four Five
- D. Five Four One Three
- E. There is no output due to an error.

QUESTION 36

Which data structure is implemented by the Structure class to the right?

- A. Red Black Tree
- B. Priority Queue
- C. Binary Search Tree
- D. Singly-Linked List
- E. Doubly-Linked List

```
class Node{
    Comparable val;
    Node prev, next;
    public Node(Comparable c) {
        val = c;
    }
    public String toString() {
        return ""+val;
    }
}
class Structure{
    Node root;
    public Structure(Node n) {
        root = n;
        <2*>
    }
    public Node get(int i) {
        Node cur = root;
        while(i-- > 0 && cur != root.prev)
            cur = cur.next;
        if(i > 0)
            return null;
        return cur;
    }
    public Node remove(int i) {
        Node rem = <1*>;
        rem.prev.next = rem.next;
        rem.next.prev = rem.prev;
        return rem;
    }
    public void add(Node n) {
        Node cur = root;
        while(cur != root.prev)
            cur = cur.next;
        cur.next = n;
        root.prev = n;
        n.next = root;
        n.prev = cur;
    }
    public String toString() {
        String fin = root + " ";
        Node cur = root;
        while(cur != root.prev) {
            cur = cur.next;
            fin += cur + " ";
        }
        return fin.trim();
    }
}
//////////client code//////////
Node n = new Node("One");
Structure s = new Structure(n);
s.add(new Node("Two"));
s.add(new Node("Three"));
s.add(new Node("Four"));
out.println(s.remove(2)); //q34
s.add(new Node("Five"));
out.println(s); //q35
```

QUESTION 37

What is the big O runtime of the code to the right?

- A. $O(N \log N)$
- B. $O(N^2 \log N)$
- C. $O(N)$
- D. $O(N^2)$
- E. $O(N^3)$

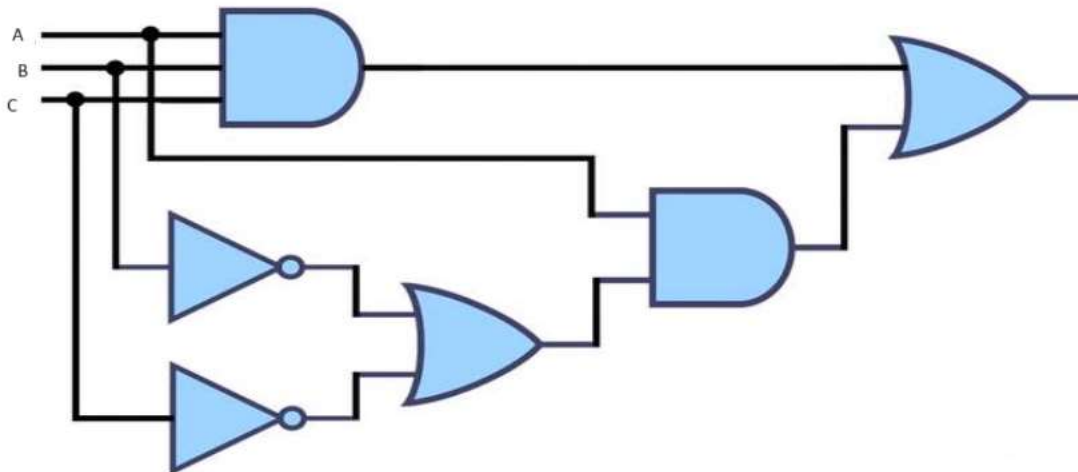
```
int n = 10, sum = 0;
for(int i = 0; i < n; i++)
    for(int ii = 1; ii <= n; ii *= 2)
        sum++;
out.println(sum);
```

QUESTION 38

What is output by the code to the right?

- A. 40
- B. 38
- C. 48
- D. 42
- E. There is no output due to an error.

QUESTION 39 What is the simplified boolean expression corresponding to the following boolean logic diagram?



QUESTION 40

What is the eight-bit two's complement representation of the following number?

-89_{10}