

University Interscholastic League

Computer Science Competition

Number 121 (District 1 - 2010)

General Directions (Please read carefully!):

- 1) DO NOT OPEN EXAM UNTIL TOLD TO DO SO.
- 2) **NO CALCULATOR OF ANY KIND MAY BE USED.**
- 3) There are 40 questions on this contest exam. 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 45 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. 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, but not on the answer sheet or Scantron card which are 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. Ignore any typographical errors and assume any undefined variables are defined as used.**
- 9) A reference to commonly used Java classes is provided at the end of the test, and you may use this reference sheet 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 packages and classes (e.g. `.util`, `ArrayList`, etc.) are included in any programs or code segments that refer to methods from these classes and 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 an incorrect answer.

QUESTION 1

What is the sum of DC_{16} and AC_{16} ?

- A. ED_{16} B. 2324_{16} C. 188_{16} D. $17C_{16}$ E. $1EF_{16}$

QUESTION 2

What is output by the code to the right?

- A. 21 B. 33 C. 17
D. 15 E. 14

```
int x = 3 * 5 + 6 / 2 - 1;
System.out.print(x);
```

QUESTION 3

What is output by the code to the right?

- A. 8 B. 14 C. 10
D. 12 E. 7

```
int mark = 0;
for(int i = -2; i < 5; i++)
    mark += 2;
System.out.print(mark);
```

QUESTION 4

What is output by the code to the right?

- A. s-vo B. Louin-Ahn C. s-von
D. s-von-Ah E. s-von-Ahn

```
String gamer = "Louis-von-Ahn";
String part = gamer.substring(4, 8);
System.out.print(part);
```

QUESTION 5

What is output by the code to the right?

- A. 9 B. 7 C. 6
D. 4 E. 5

```
int x = 2;
int[] vals = {12, 56, 34, x + 2, x, 3};
System.out.print(vals.length);
```

QUESTION 6

What is output by the code to the right?

- A. 6 B. 12 C. 4
D. 7 E. 5

```
int w = 10;
int z = 7;
z = w / 4 + z / 2;
System.out.print(z);
```

QUESTION 7

In the code to the right, how many of the 8 possible combinations of values for the variables a, b, and c will result in d being set to true?

- A. 1 B. 0 C. 7
D. 4 E. 8

```
boolean a, b, c;
//code to initialize a, b, and c

boolean d = !a || b || !c;
```

<p>QUESTION 8</p> <p>What is output by the code to the right?</p> <p>A. 3 B. 12 C. 2</p> <p>D. 23 E. 1</p>	<pre>boolean p = true; boolean q = false; if(p && q){ if(p q) System.out.print(1); else System.out.print(2); }else System.out.print(3);</pre>
<p>QUESTION 9</p> <p>What replaces <*1> in the code to the right so the Point3D class inherits the Point class?</p> <p>A. inherit B. inherits C. static</p> <p>D. extends E. implements</p>	<pre>public class Point{ private int x; private int y; public Point(int x, int y){ this.x = x; this.y = y; } public String toString(){ return x + "," + y; } }</pre>
<p>Assume <*1> is filled in correctly.</p>	<pre>public class Point3D <*1> Point{ private int z; public Point3D(int x, int y, int z){ super(x, y); this.z = z; } }</pre> <pre>//////////////////////////////////// // client code Point3D p3 = new Point3D(0, 1, 2); System.out.println(p3.toString());</pre>
<p>QUESTION 10</p> <p>What is output by the client code to the right?</p> <p>A. x,y B. 0,1 C. 1,2</p> <p>D. 0,0 E. 0,1,2</p>	
<p>QUESTION 11</p> <p>What is output by the code to the right?</p> <p>A. 31 500 B. 500 500 C. 62 500</p> <p>D. 8000 500 E. 62 62</p>	<pre>int bx = 500; int ax = bx >> 4; System.out.print(ax + " " + bx);</pre>
<p>QUESTION 12</p> <p>What is output by the code to the right?</p> <p>A. 11.0 B. 10 C. 10201</p> <p>D. 10.0 E. 10.04</p>	<pre>int st = 101; double res = Math.floor(Math.sqrt(st)); System.out.print(res);</pre>

<p>QUESTION 13</p> <p>What is output by the code to the right?</p> <p>A. C C++ Java</p> <p>B. C\tC++\tJava</p> <p>C. "C\tC++\tJava"</p> <p>D. CTC++TJava</p> <p>E. C</p>	<pre>String langs = "C\tC++\tJava"; System.out.print(langs);</pre>
<p>QUESTION 14</p> <p>What is output by the code to the right?</p> <p>A. 120500 B. +1205 C. 1205d</p> <p>D. 001205 E. 1205</p>	<pre>int value = 1205; System.out.printf("%06d", value);</pre>
<p>QUESTION 15</p> <p>What is returned by the method call validate(3, 10)?</p> <p>A. 0 B. null C. true</p> <p>D. 1 E. false</p>	<pre>public boolean validate(int x, int y){ boolean result = y > x && x >= 0; result = result && x <= 100 && y <= 100; result = result && y >= 0; return result; }</pre>
<p>QUESTION 16</p> <p>How many '*'s are output by the code to the right?</p> <p>A. 100 B. 10 C. 81</p> <p>D. 11 E. 20</p>	<pre>final int LIMIT = 10; for(int i = 0; i < LIMIT; i++) { for(int j = 0; j < LIMIT; j++) { System.out.print('*'); } }</pre>
<p>QUESTION 17</p> <p>What replaces <*1> in the code to the right to obtain the character at position i in the String lang?</p> <p>A. charAt(i) B. lang.substring(i,i)</p> <p>C. lang[i] D. lang.substring(i)</p> <p>E. lang.charAt(i)</p> <p>Assume <*1> is filled in correctly.</p>	<pre>String lang = "Python"; String changed = ""; for(int i = 0; i < lang.length(); i++){ char c = <*1>; changed = c + changed; } System.out.print(changed);</pre>
<p>QUESTION 18</p> <p>What is output by the code to the right?</p> <p>A. n B. Python C. Pytho</p> <p>D. nohtyP E. P</p>	

<p>QUESTION 19</p> <p>What is output by the code to the right?</p> <p>A. 10 B. 3 C. 2</p> <p>D. There is no output due to a syntax error.</p> <p>E. There is no output due to a runtime error.</p>	<pre>int result = 12; int[] values = {2, 4}; result /= values[0] + values[1]; System.out.print(result);</pre>
<p>QUESTION 20</p> <p>What is output by the code to the right?</p> <p>A. false false B. false true</p> <p>C. true false D. true true</p> <p>E. There is no output due to a NullPointerException.</p>	<pre>ArrayList<String> st1; st1 = new ArrayList<String>(); ArrayList<String> st2 = null; ArrayList<String> st3; st3 = new ArrayList<String>(); System.out.print(st1.equals(st2)); System.out.print(" " + st1.equals(st3));</pre>
<p>QUESTION 21</p> <p>What are the possible values the code to the right will output?</p> <p>A. -3, -2, -1 B. -3, 3</p> <p>C. 0, 1, 2, 3 D. -3, -2, -1, 0</p> <p>E. -3, -2, -1, 0, 1, 2, 3</p>	<pre>int min = -3; int r = 4; double init = Math.random(); int rand = (int)(init * r) + min; System.out.print(rand);</pre>
<p>QUESTION 22</p> <p>Which of the following is not a Java keyword?</p> <p>A. this B. short C. long D. super E. ifelse</p>	
<p>QUESTION 23</p> <p>What does method figure return if t is "aa" and wds is the following array:</p> <p>{"aaa", "aa", "AAA", "aa", "a", "aa"}</p> <p>A. 5 B. 6 C. 2</p> <p>D. -1 E. 1</p>	<pre>// pre: t != null public int figure(String[] wds, String t) { int start = wds.length - 1; for(int i = start; i >= 0; i--) { if(wds[i] != null && wds[i].equals(t)) return i; } return -1; }</pre>
<p>QUESTION 24</p> <p>Which searching algorithm does method figure implement?</p> <p>A. merge B. binary</p> <p>C. sequential D. interpolation</p> <p>E. radix</p>	

<p>QUESTION 25</p> <p>What is output by the code to the right?</p> <p>A. 7 B. 6 C. 5</p> <p>D. 4 E. 3</p>	<pre>int accum = 3; ++accum; accum++; accum += 1; accum = accum + 1; System.out.print(accum);</pre>
<p>QUESTION 26</p> <p>What replaces <*1> in the code to the right to advance the Iterator named <code>it</code> to get the next element in the iteration?</p> <p>A. <code>it.next()</code> B. <code>it.get()</code></p> <p>C. <code>list.get()</code> D. <code>E next()</code></p> <p>E. <code>it.hasNext<E>()</code></p>	
<p>Assume <*1> is filled in correctly.</p>	
<p>QUESTION 27</p> <p>Which of the following best describes what method <code>myst</code> does?</p> <p>A. Removes all Strings from <code>list</code> that were initially at odd positions.</p> <p>B. Always removes all elements from <code>list</code>.</p> <p>C. Removes Strings with an even length from <code>list</code>.</p> <p>D. Moves Strings with an even length to the end of <code>list</code>.</p> <p>E. Nothing. The method will never result in a change to <code>list</code>.</p>	<pre>// pre: no elements of list equal null public void myst(ArrayList<String> list) { Iterator<String> it = list.iterator(); while(it.hasNext()) { String temp = <*1>; if(temp.length() % 2 == 0) it.remove(); } }</pre>
<p>QUESTION 28</p> <p>What is output by the code to the right?</p> <p>A. <code>enilC</code></p> <p>B. <code>CCCCC</code></p> <p>C. <code>Cline</code></p> <p>D. There is no output due to a syntax error.</p> <p>E. There is no output due to a runtime error.</p>	<pre>ArrayList<Character> chars; chars = new ArrayList<Character>(); String name = "Cline"; for(int i = 0; i < name.length(); i++) chars.add(name.charAt(i)); for(char c : chars) System.out.print(c);</pre>
<p>QUESTION 29</p> <p>What is output by the code to the right?</p> <p>A. 12 -5 B. 3 10 C. -5 3</p> <p>D. 12 10 E. -5 -5</p>	<pre>int[] data = {12, -5, 4, 3, 10}; PriorityQueue<Integer> holder; holder = new PriorityQueue<Integer>(); for(int i : data) holder.add(i); System.out.print(holder.remove() + " "); System.out.print(holder.remove());</pre>

QUESTION 30

What is returned by method `instrumentedSort` if `list` is the following array?

`{-1, 5, 12, 14, -5, 20, 31}`

- A. 6 B. 8 C. 21
D. 4 E. 28

```
public int instrumentedSort(int[] list){
    int count = 0;
    for(int i = 1; i < list.length; i++) {
        int temp = list[i];
        int j = i;
        while( j > 0 && temp < list[j - 1]) {
            list[j] = list[j - 1];
            list[j - 1] = temp;
            j--;
            count++;
        }
    }
    return count;
}
```

QUESTION 31

Which sorting algorithm does method `instrumentedSort` implement?

- A. selection sort B. quicksort
C. heap sort D. merge sort
E. insertion sort

QUESTION 32

What is returned by the method call `explore(-1, 1, 1, mat)` if `mat` is the matrix shown below?

-1	-1	0	2	1	6
0	-1	-1	-1	-1	-1
2	2	-1	1	13	1
11	-1	13	13	0	20

- A. 1 B. 5 C. 8
D. There is no output due to a syntax error.
E. There is no output due to a runtime error.

```
public int explore(int tgt, int r, int c,
                  int[][] mat) {
    int total = 0;
    if(0 <= r && r < mat.length
        && 0 <= c && c < mat[0].length){

        int[] rd = {-1, -1, 1, 1};
        int[] cd = {-1, 1, -1, 1};
        if( mat[r][c] == tgt ) {
            mat[r][c]--;
            total++;
            for(int i = 0; i < 4; i++){
                int nr = r + rd[i];
                int nc = c + cd[i];
                total += explore(tgt, nr, nc, mat);
            }
        }
    }
    return total;
}
```

QUESTION 33

The following values are added one at a time, in the order shown to a binary search tree using the traditional insertion algorithm. How many nodes in the resulting tree have a depth of 3? The depth of a node is the number of links from the root node of the tree to that node. The root node has a depth of 0.

11, 30, 9, 16, 14, 4, 29, 34, 78, 58

- A. 0 B. 1 C. 3 D. 4 E. 9

QUESTION 34

What is returned by the method call `treat(3)`?

- A. 9 B. 6 C. 3
- D. There is no output due to a syntax error.
- E. There is no output due to a runtime error.

```
public int treat(int x){
    int total = 0;
    for(int i = 0; i < x; i++)
        for(int j = 0; j < x; j++)
            total++;

    for(int i = 0; i < x; i++){
        int total = 3;
        for(int j = 0; j < x; j++){
            total += 3;
        }
    }

    return total;
}
```

QUESTION 35

What is output by the code to the right?

- A. -3 B. -2 C. 0
- D. 3 E. 2

```
int num = -17;
int div = 5;
System.out.print( num % div );
```

QUESTION 36

What is output by the client code to the right?

- A. 10 100 50 0 B. 10 100 10 100
- C. 10 100 50 100 D. 10 10 10 10
- E. 50 100 50 100

```
public interface Movable{
    public int dx();
    public int dy();
}
```

QUESTION 37

Consider the following class:

```
public class Bike implements Movable{
    public int x;

    public Bike(){
        x = 1;
    }

    public int dx(){ return x; }
}
```

Which of the following best explains why the `Bike` class will not compile?

- A. The instance variable `x` may not be public.
- B. The `Bike` class cannot have any constructors.
- C. The `Bike` class must have an instance variable named `y`.
- D. The `Bike` class does not have a `dy` method as required by the `Movable` interface.
- E. The `Bike` class does not have a `toString` method.

```
public class Plane implements Movable{
    public int dx(){ return 100; }
    public int dy(){ return 10; }
}
```

```
public class Jet extends Plane{
    public int dy(){ return 50; };
}
```

```
// client code
Plane p1 = new Plane();
Plane p2 = new Jet();
System.out.print(p1.dy() + " " + p1.dx());
System.out.print(" ");
System.out.print(p2.dy() + " " + p2.dx());
```


QUESTION 38

What replaces **<*1>** in the code to the right to make the code block a constructor for the `Structure` class that can be accessed by all other classes?

- A. `public Structure`
- B. `public Structure<Integer>`
- C. `public Structure<Object>`
- D. `public Structure<E>`
- E. `Structure`

Assume **<*1>** is filled in correctly.

QUESTION 39

What is the Big O of the client method `demo` to the right if $N = \text{vals.length}$? Choose the most restrictive correct answer.

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

QUESTION 40

What type of data structure does the `Structure` class implement?

- A. an array based list
- B. a binary search tree
- C. a linked list
- D. a stack
- E. a heap

```
public class Structure<E> {
    private E data;
    private Structure<E> s;

    <*1>(E d) {
        data = d;
    }

    public void add(E d) {
        if(s == null)
            s = new Structure<E>(d);
        else
            s.add(d);
    }

    public E get(int p) {
        return get(p, 0);
    }

    private E get(int p, int c) {
        if(c == p)
            return data;
        else
            return s.get(p, c + 1);
    }

    public Structure<E> remove(int p) {
        return remove(p, 0);
    }

    private Structure<E> remove(int p,
                                int c) {
        if(p == c)
            return s;
        else
            s = s.remove(p, c + 1);
        return this;
    }
}

// client method
public Structure<Integer> demo(int[] vals) {
    Structure<Integer> s;
    s = new Structure<Integer>(vals[0]);
    for(int i = 1; i < vals.length; i++)
        s.add(vals[i]);
    return s;
}
```

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()

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

Methods in addition to the List methods:

- 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.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)

No Test Material on this Page.

Computer Science Answer Key

UIL District 1 - 2010

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

Notes:

The clause "Choose the most restrictive correct answer." is necessary because per the formal definition of Big O, an algorithm that is $O(N^2)$ is also $O(N^3)$, $O(N^4)$, and so forth.

34. The second declaration of the `int total` causes a syntax error.

35. `%` is often referred to as the modulus operator, but it is more accurate to call it the remainder operator. Given an `int x` and another `int y`, then $(x / y) * y + (x \% y)$ will equal `x`.