

# University Interscholastic League

## Computer Science Competition

Number 119 (Invitational A - 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  $1001_2$  and  $11_2$ ?

- A.  $1100_2$       B.  $1111_2$       C.  $1010_2$       D.  $111_2$       E.  $1011_2$

**QUESTION 2**

What is output by the code to the right?

- A. 2      B. 1      C. 0  
D. 6      E. 1.5

```
int x = 3000 / 10 / 100 * 2;
System.out.print(x);
```

**QUESTION 3**

What is output by the code to the right?

- A. 19      B. 10      C. 20  
D. 0      E. 1

```
int count = 0;
for(int i = 0; i < 20; i++)
    count++;
System.out.print(count);
```

**QUESTION 4**

What is output by the code to the right?

- A. Yu      B. oo      C. yu  
D. or      E. Yourdon

```
String nm = "Yourdon";
String part = "";
part = part + nm.charAt(1) + nm.charAt(3);
System.out.println(part);
```

**QUESTION 5**

What is output by the code to the right?

- A. 0      B. 4      C. 3  
D. null      E. 5

```
int[] nums = {5, 1, 7, 5, 5, 3, 5};
System.out.print(nums[5]);
```

**QUESTION 6**

What is output by the code to the right?

- A. 1      B. 42.5      C. 5  
D. 4.5      E. 5.0

```
int w = 10;
double c = w / 4 + 2.5;
System.out.print(c);
```

**QUESTION 7**

Which answer is logically equivalent to the following boolean expression, where  $p$  and  $q$  are boolean variables?

$!(p \parallel !q)$

- A.  $!q$       B.  $!p \ \&\& \ q$       C.  $!p \parallel q$       D.  $!p \ \&\& \ !q$       E.  $p$

<p><b>QUESTION 8</b></p> <p>What is output by the code to the right?</p> <p>A. 2                      B. 12                      C. 1</p> <p>D. 123                    E. 13</p>	<pre>int x = 12; int y = 12; if( x &lt; y )     System.out.print(1); if( x &lt;= y )     System.out.print(2); else     System.out.print(3);</pre>
<p><b>QUESTION 9</b></p> <p>What replaces <b>&lt;*1&gt;</b> in the code to the right to call the constructor in the <code>City</code> class with two parameters using <code>nm</code> as the first argument and <code>0</code> as the second?</p> <p>A. <code>super(nm, 0)</code></p> <p>B. <code>this(nm, 0)</code></p> <p>C. <code>this.City(nm, 0)</code></p> <p>D. <code>super.City(nm, 0)</code></p> <p>E. <code>City(nm, 0)</code></p>	<pre>public class City{      private String name;     private int pop;      public City(String nm){         &lt;*1&gt;;     }      public City(String nm, int p){         name = nm;         pop = p;     }      public String toString(){         return name + " " + pop;     } }</pre>
<p>Assume <b>&lt;*1&gt;</b> is filled in correctly.</p>	
<p><b>QUESTION 10</b></p> <p>What is output by the client code to the right?</p> <p>A. <code>waco 100,000</code></p> <p>B. <code>c1</code></p> <p>C. <code>Waco 0</code></p> <p>D. <code>WACO</code></p> <p>E. <code>Waco 100000</code></p>	<pre>//////////////////////////////////// // client code City c1 = new City("Waco", 100000); System.out.println( c1 );</pre>
<p><b>QUESTION 11</b></p> <p>What is output by the code to the right?</p> <p>A. <code>-11 -11</code>    B. <code>10 10</code>            C. <code>10 -1</code></p> <p>D. <code>10 -2</code>        E. <code>10 -11</code></p>	<pre>int bx = 10; int ax = ~bx; System.out.print(bx + " " + ax);</pre>
<p><b>QUESTION 12</b></p> <p>What is output by the code to the right?</p> <p>A. 32                      B. 32.0                      C. 25</p> <p>D. 25.0                    E. 10.0</p>	<pre>double res = Math.pow(5, 2); System.out.print(res);</pre>
<p><b>QUESTION 13</b></p> <p>What is output by the code to the right?</p> <p>A. <code>P    T</code>            B. <code>Pt</code>                      C. <code>PTT</code></p> <p>D. <code>P\tT</code>                E. <code>PtT</code></p>	<pre>String letters = "P\tT"; System.out.print(letters);</pre>

<p><b>QUESTION 14</b></p> <p>What is output by the code to the right?</p> <p>A. 19.5910000      B. 20.0</p> <p>C. 7.5                      D. 19.5910</p> <p>E. 19.59100</p>	<pre>double value = 19.591; System.out.printf("%7.5f", value);</pre>
<p><b>QUESTION 15</b></p> <p>What is returned by the method call <code>process(3)</code>?</p> <p>A. 9                      B. 8                      C. 0</p> <p>D. 3                      E. 5</p>	<pre>public int process(int x){     int y = x;     x++;     y--;     return x * y; }</pre>
<p><b>QUESTION 16</b></p> <p>How many '*'s are output by the code to the right?</p> <p>A. 20                      B. 50                      C. 6</p> <p>D. 15                      E. 30</p>	<pre>for(int r = 0; r &lt; 10; r++)     for(int c = 0; c &lt; 5; c++)         System.out.print('*');</pre>
<p><b>QUESTION 17</b></p> <p>What is output by the code to the right?</p> <p>A. cara                      B. bata                      C. :-):-(</p> <p>D. ::-)                      E. batd</p>	<pre>String garbage = ":car:-)bat:-(a!!d"; String arg1 = "[^a-zA-Z]+"; String[] parts = garbage.split(arg1); System.out.print( parts[1] + parts[3] );</pre>
<p><b>QUESTION 18</b></p> <p>What is output by the code to the right?</p> <p>A. 4                      B. 17                      C. 32</p> <p>D. 34                      E. 15</p>	<pre>int[] vals = {-2, 0, 7, 10, 12, 3, 2}; int total = 0; for(int i = 0; i &lt; vals.length; i++){     total += vals[i];     if(total &gt;= 10)         break; } System.out.print(total);</pre>
<p><b>QUESTION 19</b></p> <p>What is output by the code to the right?</p> <p>A. 91                      B. 17                      C. 'Q'</p> <p>D. '17'                      E. Q</p>	<pre>int offset = 3; char ch = 'N'; ch = (char)(ch + offset); System.out.print(ch);</pre>
<p><b>QUESTION 20</b></p> <p>What is output by the code to the right?</p> <p>A. ABD                      B. AAAAAA</p> <p>C. BCE                      D. BCDEFG</p> <p>E. There is no output due to a <code>StringIndexOutOfBoundsException</code>.</p>	<pre>String start = "ABCDEFGH"; String result = ""; for(int i = 1; i &lt; start.length(); i += i)     result += start.charAt(i); System.out.print(result);</pre>

**QUESTION 21**

Consider the class headers to the right. Assume all of the classes to the right have a default constructor. Which of the following statements will cause a syntax error?

- I. `Media m1 = new TVShow();`  
 II. `Media m2 = new SchoolPaper();`  
 III. `SchoolPaper p1 = new Paper();`
- A. I only      B. II only      C. III only  
 D. I and II only      E. I, II, and III

```
public class Media

public class Paper extends Media

public class SchoolPaper extends Paper

public class TVShow extends Media
```

**QUESTION 22**

What is output by the following client code?

```
Brick b1 = new Brick();
Brick b2 = new Brick();
b1.hit();
System.out.print( b1.isShowing() + " ");
System.out.print( b2.isShowing() );
```

- A. false true      B. true false  
 C. false false      D. true true  
 E. There is no output due to a runtime error caused by the client code.

```
public class Brick{
    private boolean hidden;

    public void hit(){
        hidden = true;
    }

    public boolean isShowing(){
        return !hidden;
    }
}
```

**QUESTION 23**

This question makes use of the `Brick` class from question 22. What is output by the following client code?

```
ToughBrick t1 = new ToughBrick();
t1.hit();
System.out.print( t1.isShowing() + " ");
t1.hit();
System.out.print( t1.isShowing() );
```

- A. false true      B. true false  
 C. false false      D. true true  
 E. There is no output due to a syntax error in the `ToughBrick` class.

```
public class ToughBrick extends Brick{
    private int reqHits;
    private int hits;

    public ToughBrick(){
        reqHits = 2;
    }

    public void hit(){
        hits++;
        if(hits == reqHits)
            hidden = true;
    }
}
```

**QUESTION 24**

Which of the following is not a Java keyword?

- A. `throw`      B. `switch`      C. `do`      D. `finally`      E. `range`

**QUESTION 25**

What is output by the code to the right?

- A. 3 2      B. 3 -2.6      C. -5 -2  
 D. -3 -2      E. 3 -2

```
int num = -13;
int div = 5;
System.out.print( num % div );
System.out.print( " " + (num / div) );
```

**QUESTION 26**

The quicksort algorithm sorts values in an array into ascending order based on the following algorithm:

pick the pivot

partition the elements of the array based on the pivot

quicksort the elements less than or equal to the pivot

quicksort the elements greater than the pivot

Given the following initial array:

{5, 8, 12, 10, -5, 7, 13}

Which of the following is a possible ordering of the elements of the array after the first partition assuming the middle element of the array (the value 10 in this case) is chosen as the first pivot?

- A. {-5, 8, 7, 5, 10, 12, 13}      B. {13, 5, 12, 10, -5, 7, 8}
- C. {10, 5, 8, -5, 7, 12, 13}      D. {12, 13, 5, -5, 8, 7, 10}
- E. {5, 8, 12, 10, -5, 7, 13}

**QUESTION 27**

Given an array of N distinct elements in random order, what is the Big O of the traditional implementation of the quicksort algorithm?

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

**QUESTION 28**

What is output by the code to the right?

- A. 3.0      B. 4      C. 4.0
- D. 3      E. 5

```
double aa = 2.4;
double bb = 1.6;
int x = (int)aa + (int)bb;
System.out.print(x);
```

**QUESTION 29**

What is output by the code to the right?

- A. false0      B. -3
- C. true0      D. 4
- E. There is no output due to an  
ArrayIndexOutOfBoundsException.

```
int[] small = {-3, 4, 1, 2};
int index = small[0];
if(index > 0 && index < small.length
    && small[index] > index){
    System.out.print( small[index] );
}
System.out.print(index);
```

**QUESTION 30**

What is output by the code to the right?

- A. "XA"      B. XAXAXAXA
- C. XAXAXA      D. XXXAAA
- E. "XAXAXA"

```
String st = "XA";
st = st + st + st;
System.out.print(st);
```

<p><b>QUESTION 31</b></p> <p>Method <code>max</code> to the right is suppose to return the max value in the array named <code>list</code> but it does not always work as intended. What must be changed to make the method work as intended?</p> <p>A. Change the segment <code>if( d &gt; max )</code> to <code>if( d &lt;= max)</code></p> <p>B. Change the segment <code>for(double d : list)</code> to <code>for(double d : list&lt;Double&gt;)</code></p> <p>C. Change the segment <code>for(double d : list)</code> to <code>for(double d = list[0] : list)</code></p> <p>D. Change the statement <code>return max;</code> to <code>return d;</code></p> <p>E. Change the statement <code>double max = 0.0;</code> to <code>double max = list[0];</code></p>	<pre>// pre: list.length &gt; 0 public double max(double[] list){     double max = 0.0;     for(double d : list)         if( d &gt; max )             max = d;     return max; }</pre>
<p><b>QUESTION 32</b></p> <p>What is output by the code to the right?</p> <p>A. <code>true[l, m, n]</code></p> <p>B. <code>true[l, m, m, n]</code></p> <p>C. <code>false[l, m, m, n]</code></p> <p>D. <code>null[l, m, n]</code></p> <p>E. <code>true[a, b, u]</code></p>	<pre>Set&lt;Character&gt; chs; chs = new TreeSet&lt;Character&gt;(); String tal = "milner"; for(int i = 0; i &lt; tal.length(); i++)     chs.add(tal.charAt(i));  ArrayList&lt;Character&gt; lst; lst = new ArrayList&lt;Character&gt;(); String ta2 = "manblum"; for(int i = 0; i &lt; ta2.length(); i++)     lst.add(ta2.charAt(i));  System.out.print(chs.retainAll(lst)); System.out.print(chs);</pre>
<p><b>QUESTION 33</b></p> <p>What is output by the code to the right?</p> <p>A. <code>true</code>      B. <code>25</code>      C. <code>9</code></p> <p>D. <code>15</code>      E. <code>false</code></p>	<pre>int b1 = 8; int b2 = 7; b2 = b1   b2 &amp; 17; System.out.print(b2);</pre>
<p><b>QUESTION 34</b></p> <p>What is output by the code to the right?</p> <p>A. <code>7 7</code>      B. <code>10 13</code>      C. <code>13 10</code></p> <p>D. <code>7 10</code>      E. <code>13 7</code></p>	<pre>PriorityQueue&lt;Integer&gt; pq; pq = new PriorityQueue&lt;Integer&gt;(); pq.add(13); pq.add(7); pq.add(7); pq.add(10); System.out.print( pq.remove() + " "); System.out.print( pq.remove() );</pre>

<p><b>QUESTION 35</b></p> <p>What is output by the code to the right?</p> <p>A. IB</p> <p>B. OASIB</p> <p>C. OASJLR</p> <p>D. GMLAIB</p> <p>E. There is no output due to a <code>NoSuchElementException</code>.</p>	<pre>String[] initials = {"GM", "OAS", "LA",                     "IB", "JLR"};  ArrayList&lt;String&gt; con; con = new ArrayList&lt;String&gt;(); for(String s : initials)     con.add(s);  Iterator&lt;String&gt; it = con.iterator(); while(it.hasNext())     if(it.next().length() == 2)         System.out.print(it.next());</pre>
<p><b>QUESTION 36</b></p> <p>Which of the following best describes what method <code>problem</code> returns?</p> <p>A. The maximum value in <code>mat</code>.</p> <p>B. The index of the row in <code>mat</code> closest to 0 that contains the minimum value in <code>mat</code>.</p> <p>C. The index of the last row in <code>mat</code> that contains a negative value.</p> <p>D. The index of the row in <code>mat</code> closest to 0 that contains the most negative values.</p> <p>E. The minimum value in <code>mat</code>.</p>	<pre>public int problem(double[][] mat) {     int res = 0;     int m = -1;     for(int i = 0; i &lt; mat.length; i++){         int var = 0;         for(int j = 0; j &lt; mat[i].length; j++){             if( mat[i][j] &lt; 0 )                 var++;             if( var &gt; m ){                 res = i;                 m = var;             }         }     }     return res; }</pre>
<p><b>QUESTION 37</b></p> <p>What is output by the code to the right?</p> <p>A. [a, a, A, B]    B. [A, B, a, a]</p> <p>C. [B, A, a, a]    D. [A, a, a, B]</p> <p>E. [a, A, B]</p>	<pre>char[] arr = {'a', 'B', 'A', 'a'}; Arrays.sort(arr); System.out.println( Arrays.toString(arr) );</pre>
<p><b>QUESTION 38</b></p> <p>What is output by the code to the right?</p> <p>A. null    B. true    C. false</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>String nm = "Wirth"; boolean result = nm instanceof ArrayList; System.out.println(result);</pre>



**QUESTION 39**

What is output by the following client code?

```
Structure<String> s;
s = new Structure<String>();
s.add("A");
s.add("B");
s.add("B");
s.add("AB");
s.add("A");
System.out.print(s.size());
```

- A. 3                      B. 2                      C. 5  
D. 10                    E. 6

**QUESTION 40**

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

- A. A stack              B. A set                      C. A heap  
D. A queue              E. A list

```
public class Structure<E>{

    private ArrayList<E> con;

    public Structure(){
        con = new ArrayList<E>();
    }

    public boolean add(E obj){
        boolean result = con.contains(obj);
        if( !result )
            con.add(obj);
        return result;
    }

    public boolean present(E obj){
        return con.contains(obj);
    }

    public int size(){
        return con.size();
    }

    public boolean remove(E obj){
        return con.remove(obj);
    }
}
```

## 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 Invitational A - 2010

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

### 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.

17. The first elements of `parts` is an empty `String` formed by the start of the `String` and the first colon.

23. Descendant classes do not have access to the private instance variables of ancestor classes.

31. The change is necessary for cases when all values in the array are less than 0.

38. The syntax error is "Incompatible conditional operand types `String` and `ArrayList`". It is impossible for variable of type `String` to ever refer to an object that is an `ArrayList` so the compiler rejects the expression.