

# University Interscholastic League

## Computer Science Competition

### Practice Test- 2008

General Directions (Please read carefully!):

- 1) DO NOT OPEN EXAM UNTIL TOLD TO DO SO.
- 2) **NO CALCULATORS 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. 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, 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  $A8_{16}$  and  $110111_2$ ?

- A.  $FF_{16}$       B.  $10111111_2$       C.  $DF_{16}$       D.  $99_{10}$       E.  $777_8$

**QUESTION 2**

What is output by the code to the right?

- A. 4      B. 14      C. 0  
D. 14      E. 5

```
int x = 14;
int y = 5;
System.out.print( x % y );
```

**QUESTION 3**

What is output by the code to the right?

- A. 10      B. 11      C. 9  
D. 0      E. 1

```
int amt = 0;
for(int i = 0; i <= 10; i++)
    amt++;
System.out.print( amt );
```

**QUESTION 4**

What is output by the code to the right?

- A. ajRed      B. ajRe      C. jRed  
D. jRedd      E. ajred

```
String s = "RajReddy";
System.out.print( s.substring(1,5) );
```

**QUESTION 5**

What is output by the code to the right?

- A. 0      B. 2  
C. 1      D. null  
E. There is no way to predict the output.

```
int[] vals = new int[5];
System.out.print( vals[2] );
```

**QUESTION 6**

What is output by the code to the right?

- A. 0.5      B. 2  
C. 2.0      D. 1.5  
E. There is no output due to a syntax error.

```
int x = 1;
double a = 2.5 - x;
System.out.print( a );
```

**QUESTION 7**

How many combinations of values for the boolean variables a, b, and c will result in d being set to true?

- A. 16      B. 0      C. 5  
D. 8      E. 3

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

boolean d = ( a && b || c );
```

<p><b>QUESTION 8</b></p> <p>What is output by the code to the right?</p> <p>A.    yesno            B.    noyes            C.    yes</p> <p>D.    no                E.    There is no output.</p>	<pre>int x = 5; if( x &lt; 5 )     System.out.print( "yes" ); else     System.out.print( "no" );</pre>
<p><b>QUESTION 9</b></p> <p>What replaces <b>&lt;*1&gt;</b> in the code to the right so that the instance variable <code>side</code> can only be accessed by the <code>Square</code> class?</p> <p>A.    public</p> <p>B.    private</p> <p>C.    protected</p> <p>D.    package</p> <p>E.    only</p>	<pre>public class Square{     &lt;*1&gt; int side;      public Square(int s){         side = s;     }      public void grow(){         side++;     }      public int getArea(){         return side * side;     } }</pre>
<p>Assume <b>&lt;*1&gt;</b> is filled in correctly.</p>	<pre>//////////////////////////////////// //client code Square sq = new Square(2); sq.grow(); System.out.print( sq.getArea() );</pre>
<p><b>QUESTION 10</b></p> <p>What is output by the client code to the right?</p> <p>A.    4</p> <p>B.    2</p> <p>C.    9</p> <p>D.    0</p> <p>E.    1</p>	
<p><b>QUESTION 11</b></p> <p>What is output by the code to the right?</p> <p>A.    7                    B.    -7                    C.    25</p> <p>D.    33                  E.    29</p>	<pre>int x = 13; int y = 20; System.out.println( y ^ x );</pre>
<p><b>QUESTION 12</b></p> <p>What is output by the code to the right?</p> <p>A.    1.0                  B.    1                      C.    2</p> <p>D.    0.0                  E.    NaN</p>	<pre>System.out.println( Math.pow( 2.0, 0.0 ) );</pre>
<p><b>QUESTION 13</b></p> <p>What is output by the code to the right?</p> <p>A.    word                B.    "word"                C.    \\word</p> <p>D.    \word               E.    \ord</p>	<pre>System.out.print("\\word");</pre>

<p><b>QUESTION 14</b></p> <p>What is output by the code to the right?</p> <p>A. 2.55      B. 2.56      C. 2.559</p> <p>D. 2      E. %4.2f 2.559</p>	<pre>System.out.printf("%4.2f", 2.559);</pre>
<p><b>QUESTION 15</b></p> <p>What is returned by the method call <code>sample(-5)</code>?</p> <p>A. 0      B. -10      C. -5</p> <p>D. 25      E. -25</p>	<pre>public static int sample(int x){     return x - x; }</pre>
<p><b>QUESTION 16</b></p> <p>What is output when method <code>use</code> is called?</p> <p>A. 10-142</p> <p>B. -12410</p> <p>C. 01234</p> <p>D. 24-110</p> <p>E. 4-1210</p>	<pre>public static void myst(int[] data){     int min, temp;     for(int i = 0; i &lt; data.length; i++){         min = 0;         for(int j = i + 1; j &lt; data.length; j++){             if( data[j] &lt; data[min] )                 min = j;         }         temp = data[min];         data[min] = data[i];         data[i] = temp;     } }</pre>
<p><b>QUESTION 17</b></p> <p>Method <code>myst</code> attempts to implement the selection sort algorithm, but has a logic error. Which of the following changes will correct the logic error?</p> <p>A. Replace <code>min = 0</code> with <code>min = data[i]</code>.</p> <p>B. Replace <code>j = i + 1</code> with <code>j = i</code>.</p> <p>C. Replace <code>min = 0</code> with <code>min = i</code>.</p> <p>D. Replace <code>temp = data[min]</code> with <code>temp = data[0]</code>.</p> <p>E. Replace <code>min = j</code> with <code>min = data[j]</code>.</p>	<pre>public static void use(){     int[] data = {10, -1, 4, 2};     myst(data);     for(int i : data)         System.out.print( i ); }</pre>
<p><b>QUESTION 18</b></p> <p>What is output by the code to the right?</p> <p>A. [4, 2]      B. [4]      C. [2]</p> <p>D. [2, 4]      E. [0, 2, 4]</p>	<pre>ArrayList&lt;Integer&gt; nums =     new ArrayList&lt;Integer&gt;(); nums.add(2); nums.add(4); System.out.println(nums);</pre>
<p><b>QUESTION 19</b></p> <p>What is returned by the method call <code>two(7)</code>?</p> <p>A. 1      B. 3      C. 5</p> <p>D. 6      E. 4</p>	<pre>public static int two(int x){     if( x &lt;= 2 )         return 1;     else         return 1 + two( x - 2 ); }</pre>

<p><b>QUESTION 20</b></p> <p>What is output by the code to the right?</p> <p>A. 17                      B. 18                      C. 1.7776</p> <p>D. 1                        E. 2</p>	<pre>double a = 1.7776; System.out.print( (int)( a * 10 ) );</pre>
<p><b>QUESTION 21</b></p> <p>What is output by the code to the right?</p> <p>A. 0                        B. 3                        C. 4</p> <p>D. t                        E. e_t</p>	<pre>String sample = "the_thirty_three"; String[] parts = sample.split("h"); System.out.print( parts.length );</pre>
<p><b>QUESTION 22</b></p> <p>What is output by the code to the right?</p> <p>A. baaba                  B. abaaba                  C. aaa</p> <p>D. aba                    E. abaa</p>	<pre>String w1 = "a"; String w2 = "ba"; System.out.println( w1 + w2 + w1 );</pre>
<p><b>QUESTION 23</b></p> <p>What is output by the code to the right?</p> <p>A. 1234                    B. 521</p> <p>C. 3632                    D. 2521</p> <p>E. There is no output due to a an ArrayOutOfBoundsException.</p>	<pre>int[] list = {2, 5, 2, 1}; for( int val : list )     System.out.print( val + 1 );</pre>
<p><b>QUESTION 24</b></p> <p>Which of the following is closest to the value method count will most likely return?</p> <p>A. 1.0 / n    B. 0                      C. n</p> <p>D. 0.5 * n    E. n * n</p>	<pre>public static int count(int n){     int ct = 0;     for(int i = 0; i &lt; n; i++){         if( Math.random() &lt; 0.5 )             ct++;     }     return ct; }</pre>
<p><b>QUESTION 25</b></p> <p>What is output by the code to the right?</p> <p>A. 8                        B. 6</p> <p>C. 13                      D. 12</p> <p>E. 11</p>	<pre>int count = 2; int[][] mat = new int[3][4];  for(int r = 0; r &lt; mat.length; r++){     for(int c = 0; c &lt; mat[0].length; c++){         mat[r][c] = count;         count++;     } }  System.out.print( mat[2][2] );</pre>

<p><b>QUESTION 26</b></p> <p>What is output by the code to the right?</p> <p>A. 13                      B. 16                      C. 4</p> <p>D. 24                      E. 21</p>	<pre>int x = 4; int y = 3; x *= y + 1; System.out.print( x );</pre>
<p><b>QUESTION 27</b></p> <p>What is output by the code to the right when given this input?</p> <p>value1 is 10 value2 is 12</p> <p>A. value1is</p> <p>B. value1 is 10value2 is 12</p> <p>C. value1 is 10 value2 is 12</p> <p>D. 10 12</p> <p>E. value1 is</p>	<pre>Scanner sc = new Scanner(System.in); System.out.print( sc.next() ); System.out.print( sc.next() );</pre>
<p><b>QUESTION 28</b></p> <p>What is output by the code to the right?</p> <p>A. 10                      B. 4                      C. 5</p> <p>D. 6                      E. 1</p>	<pre>int limit = 10; int testVar = 1; while( testVar &lt; limit ){     testVar++;     limit--; } System.out.print( limit );</pre>
<p><b>QUESTION 29</b></p> <p>What is output by the line marked //1 in the client code to the right?</p> <p>A. ic1                      B. 5                      C. obj@12x</p> <p>D. val: 0                      E. val: 5</p>	<pre>public class IntCell{     private int val;      public IntCell(int x){         val = x;     }      public void inc(){         val++;     }      public String toString(){         return "val: " + val;     } }</pre>
<p><b>QUESTION 30</b></p> <p>What is output by the line marked //2 in the client code to the right?</p> <p>A. val: 6                      B. val: 5                      C. val: 0</p> <p>D. 5                      E. 0</p>	<pre>//////////////////////////////////// //client code IntCell obj = new IntCell(5); System.out.println( obj ); //1 IntCell otherObj = obj; otherObj.inc(); otherObj = new IntCell(0); System.out.println( obj.toString() ); //2</pre>

**QUESTION 31**

Which of the following best describes what method `eval` returns?

- A. The minimum of the three parameters.
- B. The maximum of the three parameters.
- C. How many of the three parameters equal each other.
- D. The range of the three parameters.
- E. The sum of the three parameters.

```
private static int eval (int a, int b, int c) {
    int m;
    m = a;
    if (b < m)
        m = b;
    if (c < m)
        m = c;
    return m;
}
```

**QUESTION 32**

Which of the following replaces **<\*1>** in the code to the right to create a two dimensional array of `ints` with one more row than the number of characters in the `String s` and one more column than then number of characters in the `String t`?

- A. `int[s.length()][t.length()]`
- B. `new int[t.size()+1][s.size()+1]`
- C. `new int[i + 1][j + 1]`
- D. `new int[n + 1][m + 1]`
- E. More than one of these.

```
public static int comp (String s, String t) {
    int d[][];
    int n, m, i, j;
    char si, tj, cost;
```

```
    n = s.length ();
    m = t.length ();
    if (n == 0)
        return m;
    if (m == 0)
        return n;
```

```
    d = <*1>;
    for (i = 0; i <= n; i++)
        d[i][0] = i;
```

```
    for (j = 0; j <= m; j++)
        d[0][j] = j;
```

```
    for (i = 1; i <= n; i++) {
        si = s.charAt (i - 1);
        for (j = 1; j <= m; j++) {
            tj = t.charAt (j - 1);
```

```
            if (si == tj)
                cost = 0;
            else
                cost = 1;
```

```
            d[i][j] = eval(d[i-1][j]+1,
                          d[i][j-1]+1, d[i-1][j-1] + cost);
        }
    }
```

```
    return d[n][m];
}
```

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

**QUESTION 33**

What is returned by the method call `comp("uilcs", "uilcs")` ?

- A. 6
- B. 5
- C. 3
- D. 2
- E. 0

**QUESTION 34**

What is returned by the method call `comp("state", "stilte")` ?

- A. 2
- B. 3
- C. 4
- D. 5
- E. 6

**QUESTION 35**

Consider the Node and Structure classes to the right. What is output by the following code?

```
Structure s1 = new Structure();
System.out.println( s1.find() );
```

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

**QUESTION 36**

What is output by the following code?

```
int[] values = {2, 1, -1, 2, 5};
Structure s2 = new Structure();
for(int i : values )
    s2.add( i );
s2.show();
```

- A. 251-1            B. -1125            C. 2251-1
- D. -11225        E. 21-15

**QUESTION 37**

What is output by the following code?

```
int[] values3 = {2, 1, -1, 2, 5};
Structure s3 = new Structure();
for(int i : values3 )
    s3.add( i );
System.out.print( s3.find() );
```

- A. 3                      B. 2                      C. 1
- D. 0                     E. -1

**QUESTION 38**

What type of data structure does the Structure class implement?

- A. A binary search tree.
- B. A linked list.
- C. A min heap.
- D. A max heap.
- E. A hash table.

```
public class Node{
    public int val;
    public Node lc;
    public Node rc;
}
```

```
public class Structure{
    private Node myCon;

    public void add(int v){
        myCon = add(v, myCon);
    }
```

```
private Node add(int v, Node n){
    if( n == null ){
        n = new Node();
        n.val = v;
    }
    if( v < n.val )
        n.lc = add(v, n.lc);
    else if( v > n.val )
        n.rc = add( v, n.rc);
    return n;
}
```

```
public void show(){
    show(myCon);
}
```

```
private void show(Node n){
    if( n != null ){
        System.out.print(n.val);
        show(n.rc);
        show(n.lc);
    }
}
```

```
public int find(){
    return find(myCon);
}
```

```
private int find(Node n){
    if( n == null )
        return -1;
    return 1
        + Math.max( find(n.lc), find(n.rc) );
}
```



**QUESTION 39**

What is output by the code to the right when the method call `build(5)` is made?

- A. 543210      B. 43210      C. 01234  
D. 1234      E. 4321

```
public void show(LinkedList<Integer> list){  
    Iterator<Integer> it = list.iterator();  
    while( it.hasNext() )  
        System.out.print( it.next() );  
}
```

```
public void build(int n){  
    LinkedList<Integer> list = new  
        LinkedList<Integer>();  
  
    for(int i = 0; i < n; i++)  
        list.add( i );  
  
    show(list);  
}
```

**QUESTION 40**

What is the running time of method `show` given a `LinkedList` with  $N$  elements? Assume the `print` method is  $O(1)$ . Choose the most restrictive correct answer.

- A.  $O(N)$       B.  $O(1)$       C.  $O(\log N)$   
D.  $O(N^2)$       E.  $O(N^3)$

**No Material on this page.**

## 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 (to-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 the specified substring. 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 the specified substring, 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 x.
- o void add(int index, E e)  
Inserts x 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>**

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)

# Computer Science Answer Key

## UIL Practice Test 2008

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

### Notes:

See commentary sheet.

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.