

University Interscholastic League

Computer Science Competition

UTCS Invitational - 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	
<p>What does 11011_2 minus 111_2 equal?</p> <p>A. 10000_2 B. 10100_2 C. 11100_2 D. 10900_{10} E. 10_2</p>	
QUESTION 2	
<p>What is output by the code to the right?</p> <p>A. 7.0 B. 7.5 C. 8.5</p> <p>D. 9.5 E. 10.5</p>	<pre>double a = 2.5; double b = a; a++; a = a + b + a; System.out.print(a);</pre>
QUESTION 3	
<p>What is output by the code to the right?</p> <p>A. 2 B. 5 C. 6</p> <p>D. 11 E. 12</p>	<pre>int total = 0; for(int i = 1; i <= 11; i += 2) total += 2; System.out.print(total);</pre>
QUESTION 4	
<p>What is output by the code to the right?</p> <p>A. ydflyF B. ifi C. fis</p> <p>D. oyd E. ydf</p>	<pre>String sci = "Floyd"; String fi = "fly"; String last = sci + fi + sci; last = last.substring(3,6); System.out.print(last);</pre>
QUESTION 5	
<p>What is output by the code to the right?</p> <p>A. 5 B. 7 C. 11</p> <p>D. 19 E. null</p>	<pre>int x = 3; int[] init = {x, x, 2, x + 2, x * 2}; System.out.print(init.length);</pre>
QUESTION 6	
<p>What is output by the code to the right?</p> <p>A. 4 3 B. -30 2 C. -12 2</p> <p>D. 3 2 E. -12 6</p>	<pre>int w = 3; int z = 2; int v = w * -2 - 3 * z; System.out.print(v + " " + z);</pre>
QUESTION 7	
<p>What is output by the code to the right?</p> <p>A. true true B. true false</p> <p>C. false true D. false false</p> <p>E. true false true</p>	<pre>boolean p = true; boolean q = false; boolean r = true; System.out.print(!(p && q && !r) + " "); System.out.print(!(r && (p q) && !r));</pre>

<p>QUESTION 8</p> <p>What is output by the code to the right?</p> <p>A. 12 B. 23 C. 234</p> <p>D. 13 E. 34</p>	<pre>boolean p = true; int x = 10; if(p != false){ if(x != 5){ System.out.print(1); x = 5; } else System.out.print(2); if(x == 5) System.out.print(3); }else System.out.print(4);</pre>
<p>QUESTION 9</p> <p>What is output by the line marked line 1 in the client code to the right?</p> <p>A. 100 false B. false C. 0 0</p> <p>D. 100 true E. true</p>	<pre>public class Book{ private final int numPages; private boolean haveRead; public Book(int pages){ numPages = pages; } public String toString(){ return numPages + " " + haveRead; } }</pre>
<p>QUESTION 10</p> <p>What is output by the line marked line 2 in the client code to the right?</p> <p>A. true B. false C. 0</p> <p>D. 1 E. null</p>	<pre>//////////////////////////////////// // client code Book b1 = new Book(100); System.out.print(b1); // line 1 Book b2 = new Book(100); System.out.print(b1.equals(b2)); // line 2</pre>
<p>QUESTION 11</p> <p>What is output by the code to the right?</p> <p>A. 64 64 B. 0 64 C. 104 64</p> <p>D. 127 64 E. 128 0</p>	<pre>int ax = 40; int bx = 64; int cx = ax bx bx; System.out.print(cx + " " + bx);</pre>
<p>QUESTION 12</p> <p>What is output by the code to the right?</p> <p>A. -3.0 B. -2.4 C. -2.0</p> <p>D. 0.0 E. 3.0</p>	<pre>double st = -3.4; System.out.print(Math.ceil(st));</pre>

<p>QUESTION 13</p> <p>What is output by the code to the right?</p> <p>A. RedGreenBlueTeal</p> <p>B. RedGreen BlueTeal</p> <p>C. RedGreen\tBlueTeal</p> <p>D. Red Green Blue Teal</p> <p>E. RedGreen Blue Teal</p>	<pre>System.out.print("Red"); System.out.println("Green\tBlue"); System.out.print("Teal");</pre>
<p>QUESTION 14</p> <p>What is output by the code to the right?</p> <p>A. 19 B. 20.00 C. 19.591</p> <p>D. 20.06 E. 19.59</p>	<pre>double value = 19.591; System.out.printf("%5.2f", value);</pre>
<p>QUESTION 15</p> <p>What is returned by the method call <code>longhorn(2, 5)</code>?</p> <p>A. 0 B. 2 C. 10</p> <p>D. 13 E. 14</p>	<pre>public int longhorn(int y, int x){ x++; x = y + x + y; int z = y; z--; y++; return x + y + z; }</pre>
<p>QUESTION 16</p> <p>Which answer is logically equivalent to the following <code>boolean</code> expression, where <code>w, x, y, z</code> are <code>int</code> variables?</p> <p style="text-align: center;"><code>!(x == y) && !(w > z)</code></p> <p>A. <code>(x != y) && (w <= z)</code> B. <code>(x != y) && (w != z)</code> C. <code>(x > y) && (w < z)</code></p> <p>D. <code>(x != w) && (y != z)</code> E. <code>(x < y) (w >= z)</code></p>	
<p>QUESTION 17</p> <p>How many '*'s are output by the code to the right?</p> <p>A. 20 B. 30 C. 60</p> <p>D. 200 E. 220</p>	<pre>final int LIMIT = 10; for(int i = 0; i <= LIMIT; i++) for(int j = 0; j < LIMIT * 2; j++) System.out.print('*');</pre>
<p>QUESTION 18</p> <p>What is output by the code to the right?</p> <p>A. arCodd B. EC C. odd</p> <p>D. Edg E. dd</p>	<pre>String tw = "EdgarCodd"; String res = tw.substring(3); System.out.println(res);</pre>

<p>QUESTION 19</p> <p>What is output by the code to the right?</p> <p>A. 11 B. 12 C. 21 D. 22 E. 121</p>	<pre>int val = 3; boolean p = val * val * val * val > 100; boolean q = !p; if(!!p) System.out.print(1); else System.out.print(2); if(q !q) System.out.print(1); else System.out.print(2);</pre>
<p>QUESTION 20</p> <p>In the code to the right, how many of the 16 possible combinations of values for the variables a, b, c, and d will result in e being set to true?</p> <p>A. 0 B. 1 C. 3 D. 7 E. 15</p>	<pre>boolean a, b, c, d; // code to initialize a, b, c, and d boolean e = a && b c && d;</pre>
<p>QUESTION 21</p> <p>What is output by the client code to the right?</p> <p>A. [16, 9, 4, 1] B. [0, 1, 2, 3] C. [3, 2, 1, 0] D. [0, 0, 0, 0] E. [9, 4, 1 0]</p>	<pre>public void toy(int[] ls) { int val = 1; for(int i = ls.length - 1; i >= 0; i--) { ls[i] = val * val; val++; } } // client code int[] data = new int[4]; toy(data); System.out.print(Arrays.toString(data));</pre>
<p>QUESTION 22</p> <p>Which of the following is not a valid Java identifier?</p> <p>A. TV_TIME B. tvTime C. TVTIME D. TV345Time E. TV<Time></p>	
<p>QUESTION 23</p> <p>What are the possible values the code to the right will output?</p> <p>A. [-6, -3, 0, 3] B. [0, 3, 6, 9, 12] C. [-6, -3, 0, 3, 6] D. [0, 1, 2, 3] E. [0.0, 0.1, 0.2, 0.3, 0.4]</p>	<pre>double seed = Math.random(); int result = ((int)(seed * 4) - 2) * 3; System.out.print(result);</pre>
<p>QUESTION 24</p> <p>What is output by the code to the right?</p> <p>A. -231-2 B. iiii C. 0123 D. 2132 E. null null null null</p>	<pre>int[] simple = {-2, 3, 1, -2}; for(int i : simple) System.out.print(i);</pre>

QUESTION 25

What replaces **<*1>** in the code to the right to execute the body of the `while` loop if the element at index 5 in `od` is equal to 0?

- A. `od[5] == 0`
- B. `od[5]`
- C. `!od[5] == false`
- D. `!od[5]`
- E. `od[5] = 0 = true`

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

QUESTION 26

What is output by the code to the right?

- A. 0 B. 6 C. 8
- D. 32 E. 64

QUESTION 27

What is output by the client code to the right?

- A. `[-2, 3, -1, 1]` B. `[2, 4, 6, 8]`
- C. `[0, 0, 0, 0]` D. `[5, 7, 9, 11]`
- E. There is no output due to a syntax error.

QUESTION 28

What replaces **<*1>** in the method `tally` so that subsequent `case` statements in the `switch` block are not executed for a particular character in `evts`?

- A. `break` B. `goto` C. `try`
- D. `throws` E. `discontinue`

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

QUESTION 29

What is returned by the method call `tally("HHTFF")`?

- A. 0 B. 1 C. 4
- D. 9 E. 10

```
int[] od = new int[6];
int count = 0;
while( <*1> ) {
    count++;
    int index = 0;
    od[index]++;
    int cry = 0;
    do{
        od[index] += cry;
        cry = od[index] / 2;
        od[index] %= 2;
        index++;
    } while(cry > 0);
}
System.out.print(count);
```

```
public void play(int[] vals){
    vals = new int[] {2, 4, 6, 8};
    for(int i = 0; i < vals.length; i++)
        vals[i] += 3;
}

// client code
int[] ar = {-2, 3, -1, 1};
play(ar);
System.out.print(Arrays.toString(ar));
```

```
public int tally(String evts) {
    String code = "-HFT";
    int total = 0;
    for(int i = 0; i < evts.length(); i++) {
        int pos = code.indexOf(evts.charAt(i));
        switch(pos) {
            case 1: total++; <*1>;
            case 2: total += 2; <*1>;
            case 3: total += 3;
        }
    }
    return total;
}
```

<p>QUESTION 30</p> <p>What is output by the code to the right?</p> <p>A. 9 4 B. 9 3 C. 12 4</p> <p>D. 15 4 E. 12 3</p>	<pre>int hold = 3; int val = ++hold * 3; System.out.print(val + " " + hold);</pre>
<p>QUESTION 31</p> <p>What is output by the method call <code>sample("UTA")</code>?</p> <p>A. UendU</p> <p>B. UendA</p> <p>C. UUUendU</p> <p>D. end</p> <p>E. UUUendUTA</p>	<pre>public void sample(String s){ if(s.length() == 0) System.out.print("end"); else { System.out.print(s.charAt(0)); int len = s.length() - 1; sample(s.substring(0, len)); System.out.print(s.charAt(len)); } }</pre>
<p>QUESTION 32</p> <p>What replaces <code><*1></code> in the method <code>myst</code> to execute the body of the <code>while</code> loop if there are any more elements in the current iteration of <code>data</code>?</p> <p>A. <code>it.next()</code> B. <code>it < data.size()</code></p> <p>C. <code>i < it.size()</code> D. <code>it.hasNext()</code></p> <p>E. None of these are correct.</p>	<pre>public int myst(ArrayList<String> data) { Iterator<String> it = data.iterator(); int total = 0; while(<*1>) { if(it.next().length() < 10) total += it.next().length(); } return total; }</pre>
<p>Assume <code><*1></code> is filled in correctly.</p>	
<p>QUESTION 33</p> <p>What is output by the client code to the right?</p> <p>A. 1 B. 5 C. 10</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>// client code ArrayList<String> sm; sm = new ArrayList<String>(); sm.add("Moore"); System.out.print(myst(sm));</pre>
<p>QUESTION 34</p> <p>Assume method <code>sample(int[] data)</code> is $O(N^2)$ where $N = \text{data.length}$. When method <code>sample</code> is passed an array with <code>length = 1,000</code> it takes 1 second for method <code>sample</code> to complete. If method <code>sample</code> is then passed an array with <code>length = 6,000</code> what is the expected time it will take method <code>sample</code> to complete?</p> <p>A. 6 seconds B. 36 seconds C. 60 seconds D. 64 seconds E. 80 seconds</p>	
<p>QUESTION 35</p> <p>What is output by the code to the right?</p> <p>A. 2.5 B. 5.0 C. 7.5</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>double a = 2.5; for(int i = 0; i < 5; i++) a++; System.out.print(a);</pre>

QUESTION 36

The code to the right contains a syntax error. Which of the following best explains the reason for the syntax error?

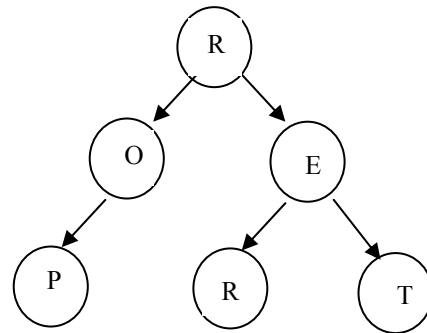
- A. `b3` is not a valid identifier.
- B. The `+` operator is not defined for the type `byte`.
- C. The `int` literals `3` and `10` may not be assigned to variables of type `byte`.
- D. `byte` is not a valid Java type.
- E. The expression `b1 + b2` evaluates to the type `int` and cannot be assigned to a `byte` variable.

```
byte b1 = 3;
byte b2 = 10;
byte b3 = b1 + b2;
```

QUESTION 37

What is the result of a post order traversal of the binary tree shown to the right?

- A. PORRET B. ROPTER C. PORTER
- D. TERROP E. ROPERT

**QUESTION 38**

Which of the following can replace `<*1>` in the code to the right as the type variable for the `Structure` class to indicate that the `Structure` class is generic?

- I. `AnyType`
 - II. `E`
 - III. `myThings`
- A. I only B. II only C. III only
 - D. I and II E. I, II, and III only

```
public class Structure<<*1>> {
    private ArrayList<<*1>> con;

    public Structure() {
        con = new ArrayList<<*1>>();
    }

    public void add(<*1> val) {
        con.add(0, val);
    }

    public <*1> access() {
        return con.get(0);
    }

    public <*1> remove() {
        return con.remove(0);
    }

    public boolean isEmpty() {
        return con.size() == 0;
    }
}
```

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

QUESTION 39

If a `Structure` object contains `N` elements what is the Big O of the `remove` method in the `Structure` class? Pick the most restrictive correct answer.

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

QUESTION 40

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

- A. a list B. a queue C. a hashtable
- D. a heap E. a stack

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)

Computer Science Answer Key

UTCS Invitational - 2010

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

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.