

University Interscholastic League

Computer Science Competition

UTCS Open - 2012

General Directions:

- 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 UTCS Open. 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 UTCS Openments 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 does 1001_2 minus 111001_2 equal?

- A. 110000_2 B. -110000_2 C. -101110_2 D. 1001_2 E. -1001_2

QUESTION 2

What is output by the code to the right?

- A. 10 B. 9 C. 8.75
D. 7 E. 3

```
int x = 7 / 2;
int y = x * 2 + x / 2;
System.out.print(y);
```

QUESTION 3

What is output by the code to the right?

- A. 38 B. 20 C. -2
D. -20 E. -38

```
int val = 0;
for(int i = 5; i < 24; i += 2)
    val -= 2;
System.out.print(val);
```

QUESTION 4

What is output by the code to the right?

- A. CCM
B. CASECOLTMALC
C. CASE_COLT+&\$*MALC
D. CASE-COLT+&\$8MALC
E. Case-Colt+&\$8Malc

```
String c = "Case-Colt+&$8Malc";
c = c.toUpperCase();
System.out.print(c);
```

QUESTION 5

What is output by the code to the right?

- A. 3 6 B. 12 3 C. 9 6
D. There is no output due to a syntax error.
E. There is no output due to a runtime error.

```
int[] st = {5, -1, 3, 6, -3};
st[4] += st[3] + st[st[3]];
System.out.print(st[4] + " " + st[3]);
```

QUESTION 6

What is output by the code to the right?

- A. 0 B. 2 C. 3
D. 17 E. 50

```
int x1 = 17;
int y1 = 50;
x1 %= y1;
System.out.print(x1);
```

QUESTION 7

How many combinations of values for the boolean variables *p*, *q*, and *r* will result in *s* being set to true?

- A. 2 B. 3 C. 5
D. 6 E. 7

```
boolean p, q, r;
//code to initialize p, q, and r

boolean s = p || q && !r;
```

<p>QUESTION 8</p> <p>What is output by the code to the right?</p> <p>A. 1 B. 13 C. 23</p> <p>D. 123 E. 2</p>	<pre>int[] st2 = {5, 1, 2, 3, 3}; if(st2.length > st2[1]) { System.out.print(1); } else System.out.print(2); System.out.print(3);</pre>
<p>QUESTION 9</p> <p>What is output by the statement in the client code to the right marked // line 1?</p> <p>A. true1 B. true0</p> <p>C. false1 D. false0</p> <p>E. There is no output due to a runtime error.</p>	<pre>public class School { private boolean isPrivate; private int numStudents; public void show() { numStudents = 10; isPrivate = true; System.out.print(this); } public String toString() { return "" + isPrivate + numStudents; } }</pre>
<p>QUESTION 10</p> <p>What is output by the statement in the client code to the right marked // line 2?</p> <p>A. this B. sc2</p> <p>C. true10 D. false0</p> <p>E. The line causes no visible output.</p>	<pre>// client code School sc1 = new School(); System.out.print(sc1 + " "); // line 1 School sc2 = new School(); sc2.show(); // line 2</pre>
<p>QUESTION 11</p> <p>What is output by the code to the right?</p> <p>A. 119 B. 100 C. 87</p> <p>D. 51 E. 32</p>	<pre>int m = 100; int n = 51; System.out.print(m ^ n);</pre>
<p>QUESTION 12</p> <p>What is output by the code to the right?</p> <p>A. 0 B. 4 C. 4.472</p> <p>D. 5 E. 20</p>	<pre>int m1 = (int) Math.sqrt(20); System.out.print(m1);</pre>
<p>QUESTION 13</p> <p>What is output by the code to the right?</p> <p>A. A\tBtC\tD</p> <p>B. A\BtC\D</p> <p>C. A B C D</p> <p>D. A BtC D</p> <p>E. A D</p>	<pre>System.out.print("A\tBtC\tD");</pre>

<p>QUESTION 14</p> <p>What is output by the code to the right?</p> <p>A. 35.0(35.0) B. (35.0)(35.0)</p> <p>C. 35.0-35.0 D. (35.0)(-35.0)</p> <p>E. 35.0-(35.0)</p>	<pre>double mon = 35; System.out.printf("%(4.1f", mon); System.out.printf("%(4.1f", -mon);</pre>
<p>QUESTION 15</p> <p>What is returned by the method call manip(0.5, -0.5)?</p> <p>A. -6.0 B. -3.75</p> <p>C. 0 D. 1.25</p> <p>E. There is no output due to a syntax error in method manip.</p>	<pre>public double manip(double a, double b) { a++; b--; ++a; return b * a; }</pre>
<p>QUESTION 16</p> <p>What is output by the code to the right?</p> <p>A. 55 B. 90 C. 99</p> <p>D. 110 E. 3628800</p>	<pre>String stars = ""; for(int i = 0; i < 10; i++) for(int j = i; j < 10; j++) stars += "*"; System.out.print(stars.length());</pre>
<p>QUESTION 17</p> <p>Which of the following Java expressions is equivalent to the formula to the right? F, m1, m2, and r are variables of type double. G is a constant of type double.</p> <p>A. F = G * m1 * m1 / r * r;</p> <p>B. F = Gm1m2/r/r;</p> <p>C. F = G * m1 * m2 / (r * r);</p> <p>D. F = G * m1 * m2 / r ^ 2;</p> <p>E. None of A through D is correct.</p>	$F = \frac{Gm_1m_2}{r^2}$
<p>QUESTION 18</p> <p>What is output by the code to the right?</p> <p>A. 2 5 B. 2 15 C. 4 15</p> <p>D. 3 3 E. 5 5</p>	<pre>int x2 = 12; int y2 = x2 + 2; x2 = x2 - 10; y2++; System.out.print(x2 + " " + y2);</pre>
<p>QUESTION 19</p> <p>Which answer is logically equivalent to the following boolean expression, where p and q are boolean variables?</p> <p>!(p q r)</p> <p>A. !p && q r B. !p !q !r C. !(p && q) && !(q && r)</p> <p>D. !p && !q && !r E. p && q && r</p>	

<p>QUESTION 20</p> <p>What is output by the code to the right?</p> <p>A. [7, 9, 3] B. [3, 7, 9]</p> <p>C. [7, 3, 9] D. [7, 9, 1]</p> <p>E. [7, 1, 9]</p>	<pre>ArrayList<Integer> list1; list1 = new ArrayList<Integer>(); list1.add(7); list1.add(9); list1.add(1, 3); list1.set(1, list1.set(2, list1.get(1))); System.out.print(list1);</pre>
<p>QUESTION 21</p> <p>What is output by the statement in the client code to the right marked // line 1?</p> <p>A. 6 B. 7 C. 8</p> <p>D. 9 E. 10</p>	<pre>public int sort(String[] s) { int c = 0; for(int i = 1; i < s.length; i++) { String temp = s[i]; int j = i; while(j > 0 && temp.compareTo(s[j - 1]) > 0) { c++; s[j] = s[j - 1]; s[j - 1] = temp; j--; } } return c; }</pre>
<p>QUESTION 22</p> <p>What is output by the statement in the client code to the right marked // line 2?</p> <p>A. [nt, smu, tam, tcu, ttu, ut, utd]</p> <p>B. [nt, smu, tam, tcu, ttu, utd, ut]</p> <p>C. [ut, tcu, tam, nt, ttu, smu, utd]</p> <p>D. [ut, utd, ttu, tcu, tam, smu, nt]</p> <p>E. [utd, ut, ttu, tcu, tam, smu, nt]</p>	<pre>// client code String[] scs = {"ut", "tcu", "tam", "nt", "ttu", "smu", "utd"}; System.out.print(sort(scs)); // line 1 String arr = Arrays.toString(scs); System.out.print(arr); // line 2</pre>
<p>QUESTION 23</p> <p>What sorting algorithm does the sort method implement?</p> <p>A. selection sort B. insertion sort C. radix sort</p> <p>D. heap sort E. bubble sort</p>	<pre>double[] d1 = {Integer.MAX_VALUE, Long.MAX_VALUE, Float.MAX_VALUE, Short.MAX_VALUE}; double[] d2 = {Long.MAX_VALUE, Float.MAX_VALUE, Double.MAX_VALUE, Byte.MAX_VALUE}; int c2 = 0; for(int i = 0; i < d1.length; i++) if(d1[i] > d2[i]) c2++; System.out.print(c2);</pre>
<p>QUESTION 24</p> <p>What is output by the code to the right?</p> <p>A. 0 B. 1 C. 2</p> <p>D. 3 E. 4</p>	<pre>double[] d1 = {Integer.MAX_VALUE, Long.MAX_VALUE, Float.MAX_VALUE, Short.MAX_VALUE}; double[] d2 = {Long.MAX_VALUE, Float.MAX_VALUE, Double.MAX_VALUE, Byte.MAX_VALUE}; int c2 = 0; for(int i = 0; i < d1.length; i++) if(d1[i] > d2[i]) c2++; System.out.print(c2);</pre>

QUESTION 25

What is output by the statement in the client code to the right marked // line 1?

- A. 0
- B. 1
- C. 2
- D. 5
- E. 7

QUESTION 26

What is output by the statement in the client code to the right marked // line 2?

- A. 0 180
- B. 0 190
- C. 180 190
- D. 190 190
- E. 190 200

QUESTION 27

What is output by the statement in the client code to the right marked // line 3?

- A. null
- B. 0
- C. There is no visible output.
- D. The output varies from one run of the program to the next.
- E. There is no output due to a runtime error.

QUESTION 28

Assume the following method is added to the Test class.

```
public static void show() {
    System.out.print(this);
}
```

What is output by the following client code?

```
Test t3 = new Test(50);
t3.bump();
Test.show();
```

- A. this
- B. t3
- C. 50
- D. 55
- E. There is no output due to a syntax error in method show.

```
public class Test {
    private static int mys = 0;
    private int score = -1;

    public Test() { mys++; }

    public Test(int s) {
        mys++;
        score = s;
    }

    public String toString() {
        return score + "";
    }

    public void bump() { this.score += 5; }

    public static int getMys() {
        return mys;
    }
}

// client code
Test[] ts = new Test[5];
ts[1] = new Test();
ts[3] = new Test(180);
System.out.print(Test.getMys()); // line 1

ts[1] = new Test();
ts[3] = new Test(180);
ts[1] = ts[3];
ts[3].bump();
ts[3].bump();
String tStr = ts[1] + " " + ts[3];
System.out.print(tStr); // line 2

System.out.print(ts[2]); // line 3
```

<p>QUESTION 29</p> <p>What is output by the code to the right?</p> <p>A. [5, 7, 2] 6</p> <p>B. [5, 7, 2] 3</p> <p>C. [2, 5, 7] 6</p> <p>D. [2, 5, 7, 7] 3</p> <p>E. The output varies from one run of the program to the next.</p>	<pre>int[] ds1 = {5, 7, 3, 3, 7, 2}; int[] ds2 = {4, 6, 1, 2, 5, 7}; Set s1 = new TreeSet(); Set s2 = new HashSet(); for(int i = 0; i < ds1.length; i++) { s1.add(ds1[i]); s2.add(ds2[i]); } s1.retainAll(s2); System.out.print(s1 + " " + s2.size());</pre>
<p>QUESTION 30</p> <p>Consider the following interface and classes to the left. All classes have a default constructor. Which of the following statements will compile without error?</p> <p>I. TimeStamp t1 = new StockTrade();</p> <p>II. StockTrade t2 = new SPut();</p> <p>III. ShortS t3 = new StockTrade();</p> <p>IV. Trade t4 = new ShortS();</p> <p>A. I only B. II only</p> <p>C. II and III only D. II and IV only</p> <p>E. I, II, and IV only</p>	<pre>public interface TimeStamp public interface Mg public abstract class Trade public class StockTrade extends Trade implements TimeStamp public class Call extends StockTrade public class Put extends StockTrade public class ShortS extends StockTrade public class SPut extends Put implements Mg</pre>
<p>QUESTION 31</p> <p>Consider the following interface and classes to the left. All classes have a default constructor. Which of the following statements will compile without error?</p> <p>I. TimeStamp t5 = new SPut();</p> <p>II. Mg t6 = new Put();</p> <p>III. TimeStamp t7 = new TimeStamp();</p> <p>IV. Object t8 = new Trade();</p> <p>A. I only B. IV only</p> <p>C. I and IV only D. II and III only</p> <p>E. I, II, III, and IV</p>	<pre>public class Call extends StockTrade public class Put extends StockTrade public class ShortS extends StockTrade public class SPut extends Put implements Mg</pre>
<p>QUESTION 32</p> <p>What is returned by method check(-1.5)?</p> <p>A. Infinity B. NaN C. 0</p> <p>D. There is no output due to a syntax error in method check.</p> <p>E. There is no output due to a runtime error.</p>	<pre>public double check(double a) { int x = 0; double b; a = a * b; a = a / x; return a; }</pre>

QUESTION 33

What is output by the statement in the client code to the right marked // line 1?

- A. 3 B. 4
C. 12 D. 16
E. The statement marked // line 1 does not produce any visible output.

```
public int algo(int x, int y) {
    int z = 0;
    while(y != 0) {
        int t = y;
        y = x % y;
        x = t;
        z += 4;
    }
    System.out.print(z);
    return x;
}

// client code
int res = algo(125, 1900); // line 1

System.out.print(res); // line 2
```

QUESTION 34

What is output by the statement in the client code to the right marked // line 2?

- A. 0 B. 1 C. 5
D. 25 E. 125

```
// client code
int res = algo(125, 1900); // line 1

System.out.print(res); // line 2
```

QUESTION 35

What is returned by method handle if t is the matrix shown below?

2	4	0	2	1	9
0	-1	5	4	0	-4
2	2	7	1	5	2
5	1	3	2	5	1
4	1	-1	3	1	4
10	4	-9	2	1	5

- A. "1 0" B. "3 5" C. "4 5"
D. "5 3" E. "5 4"

```
public String handle(int[][] t) {
    int c = 0;
    String res = "NONE";
    int m = Integer.MAX_VALUE;
    for(int x = 0; x < t.length; x++)
        for(int y = 0; y < t.length; y++) {
            int t1, t2;
            t1 = t2 = 0;
            for(int z=0; z<t.length; z++,c++) {
                t1 += t[x][z];
                t2 += t[z][y];
            }
            if(Math.abs(t1 - t2) < m) {
                m = Math.abs(t1 - t2);
                res = x + " " + y;
            }
        }
    System.out.print(c);
    return res;
}
```

QUESTION 36

What is printed by method handle t.length == 10, and all rows of t[x].length == 10 for all x such that 0 <= x < 10?

- A. 1000 B. 100
C. 10 D. 0 0
E. The output cannot be determined without knowing the elements in the matrix t.

QUESTION 37

Consider method `count` to the right. When `sc` is connected to a file named `t1.txt` and `set` is a `HashSet` the method takes 10 seconds to complete and outputs the following:

500000 25000

When `sc` is connected to the same file named `t1.txt` and `set` is a `TreeSet` the method takes 20 seconds to complete and output is the same.

What is the expected time for method `count` to complete when `sc` is connected to a file that produces the following output:

1500000 50000

when `set` is a `HashSet` and when `set` is a `TreeSet`?

	HashSet	TreeSet
A.	360 seconds	720 seconds
B.	60 seconds	132 seconds
C.	60 seconds	120 seconds
D.	20 seconds	60 seconds
E.	30 seconds	64 seconds

```
// pre: set != null, set.size() == 0
public void count(Scanner sc,
                  Set<String> set) {

    int tot = 0;
    while(sc.hasNext()) {
        tot++;
        String t = sc.next();
        set.add(t);
    }
    System.out.print(tot + " " + set.size());
}
```

QUESTION 38

What are the best case and worst case order (Big O) of method `create`? $N = d.length$. Pick the most restrictive correct set of answers.

	Best Case	Worst Case
A.	$O(1)$	$O(N^2)$
B.	$O(N)$	$O(N)$
C.	$O(N)$	$O(N^2)$
D.	$O(N^2)$	$O(N^2)$
E.	$O(N)$	$O(N^3)$

```
public ArrayList<Integer> create(int[] d) {
    ArrayList<Integer> res;
    res = new ArrayList<Integer>();
    for(int x : d)
        if(!res.contains(x))
            res.add(res.size() / 2, x);
    return res;
}
```

Go on to the next page.

QUESTION 39

Given the `Structure` class to the right what is output by the following client code?

```
int[] ds = {0, -2, 2, -2, 3, 4};
Structure<Integer> str;
str = new Structure<Integer>();
for(int xd : ds)
    str.add(xd);

for(int i = 0; i < str.size(); i++)
    System.out.print(str.remove() + " ");
```

- A. 4 3 -2 2 0 B. 0 -2 2 -2 3 4
 C. 4 3 -2 D. 4 3 -2 2 -2 0
 E. There is no output due to a syntax error in the client code.

QUESTION 40

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

- A. a set B. a stack C. a queue
 D. a max heap E. an array based list

```
public class Structure<E> {

    private List<E> con;

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

    public void add(E x) { con.add(0, x); }

    public E get() { return con.get(0); }

    public E remove() {return con.remove(0);}

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

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

    public String toString() {
        return con.toString();
    }
}
```

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)

Computer Science Answer Key

UIL UTCS Open 2012

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

28. `this` is undefined in static methods. `this` refers to the object that called or invoked the method, but static methods have no calling object.

32. Failure to initialize the variable `b` is the syntax error.

39. The stack is not emptied because the size is decreasing as the loop control variable `i` is increasing.