



University Interscholastic League Computer Science Competition

Number 149 (Invitational A - 2015)

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.
- 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.**

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.

Note: Correct responses are based on Java, **J2sdk v 1.7.25**, from Sun Microsystems, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (i. e. `error` is an answer choice) and any necessary Java 2 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used. **For all output statements, assume that the `System` class has been statically imported... `import static java.lang.System.*`;**

QUESTION 1

Which of these is NOT equivalent to $88_{16} - 206_8$?

- A. 2_{16} B. 2_8 C. 2_2 D. 2_{10} E. All are equivalent

QUESTION 2

What is the result of the expression to the right?

- A. -4 B. -1.7
C. -2 D. 3 E. 3.6

$23 / 10 \% 3 - 4$

QUESTION 3

What is output by the code to the right?

- A.
----*----*
9.5
B.
----*----*
9.50
C.
----*----*
9.5
D.
----*----*
9.50
E.
----*----*
9.50

```
System.out.println("----*----*");
System.out.printf("%5.2f", 9.5);
```

QUESTION 4

What is output by the code to the right?

- A. UILCOMPUTERSCONCE2015
B. UILCOMPUTERSCIENCE2015
C. UOLCOMPUTORSCONCO2015
D. UOLCOMPUTORSCOONCO2015
E. OUOOLOC000MOPUOT00ROSOC00NOC0020001050

```
String s = "UILCOMPUTERSCIENCE2015";
s = s.replaceAll("[IE]+", "O");
out.println(s);
```

QUESTION 5

What is output by the code to the right?

- A. true B. false

```
boolean p = true;
boolean q = false;
out.println(!(p||q));
```

QUESTION 6

What is output by the code to the right?

- A. 50 B. 10.0
C. 10 D. 10000000000
E. 50.0

```
int x = 100;
double y = 0.5;
out.println(Math.pow(x,y));
```

<p>QUESTION 7</p> <p>What is output by the code to the right?</p> <p>A. 34 B. 35.0 C. 34.4 D. 35</p> <p>E. There is no output due to an error.</p>	<pre>long g = 42; double d = 7.6; g -= d; out.printf("%d",g);</pre>
<p>QUESTION 8</p> <p>What is output by the code to the right?</p> <p>A. -4 B. -1 C. 4 D. 1</p> <p>E. There is no output due to an error.</p>	<pre>int a = 25; if (a%-6>0) out.println(a%-6); else out.println(a/-6);</pre>
<p>QUESTION 9</p> <p>What is output by the code to the right?</p> <p>A. 11 B. 5.2 C. 9 D. 8</p> <p>E. There is no output due to an error.</p>	<pre>int x = 100,y = 0; while ((x/=3)>0) y+=2; out.println(x+y);</pre>
<p>QUESTION 10</p> <p>What is output by the code to the right?</p> <p>A. 8 B. 2 C. 3 D. 7</p> <p>E. There is no output due to an error.</p>	<pre>int list[]={-5,-6,4,2,-3,7}; out.println(list[list[4]] +list.length);</pre>
<p>QUESTION 11</p> <p>Below is a value in a data file called "stuff.dat". 5.2</p> <p>In the code segment to the right, which choice is best for <statement 1> in order to retrieve the data for calculation purposes?</p> <p>A. double d = f.nextInt(); B. double d = f.nextDouble(); C. int n = f.nextDouble(); D. int n = f.nextInt(); E. All statements will work properly</p>	<pre>Scanner f = new Scanner(new File("stuff.dat")); <statement 1></pre>
<p>QUESTION 12</p> <p>What is output by the code to the right?</p> <p>A. 62 64 B. 62 32</p> <p>C. 126 64 D. 126 128</p> <p>E. There is no output due to an error.</p>	<pre>int a = 0, b = 1; do{ b*=2; a+=b; } while (b<50); out.println(a+" "+b);</pre>
<p>QUESTION 13</p> <p>To the right are three lines taken from the Java Order of Precedence chart. Which choice represents the correct order of precedence for these three lines?</p> <p>A. III, I, II B. I, II, III C. I, III, II</p> <p>D. III, II, I E. II, I, III</p>	<p>I. & II. && III. < > <= >= instanceof</p>
<p>QUESTION 14</p> <p>Which of the choices listed to the right represents the correct order from greatest to least of the bit storage capacity for the data types listed?</p> <p>A. I, III, IV, II, V B. V, IV, III, II, I</p> <p>C. V, II, III, IV, I D. I, II III, IV, V</p> <p>E. None of these</p>	<p>I. char II. double III. float IV. int V. long</p>

<p>QUESTION 15</p> <p>What is output by the code to the right?</p> <p>A. 0 1 2 2 4 6 8</p> <p>B. 0 1 2 0 1 2</p> <p>C. 0 1 2 2 4 6</p> <p>D. There is no output due to a compile error.</p> <p>E. There is no output due to a runtime error.</p>	<pre>Integer [] list={0,1,2}; ArrayList<Integer> aList = new ArrayList<Integer>(); aList.add(2);aList.add(4); aList.add(6);aList.add(8); for(Integer x:list) out.print(x+" "); list = aList.toArray(list); for(Integer x:list) out.print(x+" ");</pre>
<p>QUESTION 16</p> <p>Using the mergeSort code to the right, what is output by the client code below?</p> <pre>int[] list = {5,7,3,9,4,6}; mergeSort(list); outputList(list);</pre> <p>A. 9 7 6 5 4 3</p> <p>B. 3 4 5 6 7 9</p> <p>C. 5 7 3 9 4 6</p> <p>D. 6 4 9 3 7 5</p> <p>E. not possible to determine</p>	<pre>public static void mergeSort(int[] list){ int n = list.length; int[] temp = new int[n]; //<doc 1> mergeSortHelper(list, 0, n - 1, temp); } private static void mergeSortHelper(int[] list, int front, int back, int[] temp) { //<statement 1> if (front < back){ //<doc 2><statement 2> int mid = (front + back) / 2; //<doc 3> mergeSortHelper(list, front, mid, temp); //<doc 4> mergeSortHelper(list, mid + 1, back, temp); //<doc 5> merge(list, front, mid, back, temp); } }</pre>
<p>QUESTION 17</p> <p>In the code to the right, which of the lines below the five indicated <statements> needs to be altered in order to sort a list in descending order?</p> <p>A. <statement 1></p> <p>B. <statement 2></p> <p>C. <statement 3></p> <p>D. <statement 4></p> <p>E. <statement 5></p>	<pre>private static void merge(int[] list, int front, int mid, int back, int[] temp){ int i = front; int j = mid + 1; int k = front; //<doc 6><statement 3> while (i <= mid && j <= back){ <statement 4> if (list[i] < list[j]){ temp[k] = list[i]; i++; } else{ temp[k] = list[j]; j++; } k++; } //<doc 7><statement 5> while (i <= mid){ temp[k] = list[i]; k++; i++; } //<doc 8> while (j <= back) { temp[k] = list[j]; j++; k++; } //<doc 9> for(int x=front;x<=back;x++) list[x]=temp[x]; }</pre>
<p>QUESTION 18</p> <p>There are nine <doc> comments in the code to the right, explaining the purpose of the code just below each one. Which of the choices below is NOT correct relating to these <doc> statements?</p> <p>A. <doc 4> merge sort call for last half of list</p> <p>B. <doc 1> initial merge sort call</p> <p>C. <doc 2> find the middle position of the current list</p> <p>D. <doc 7> clean up remaining second half elements, if any</p> <p>E. <doc 9> transfer elements from temporary list to original list</p>	<pre>private static void merge(int[] list, int front, int mid, int back, int[] temp){ int i = front; int j = mid + 1; int k = front; //<doc 6><statement 3> while (i <= mid && j <= back){ <statement 4> if (list[i] < list[j]){ temp[k] = list[i]; i++; } else{ temp[k] = list[j]; j++; } k++; } //<doc 7><statement 5> while (i <= mid){ temp[k] = list[i]; k++; i++; } //<doc 8> while (j <= back) { temp[k] = list[j]; j++; k++; } //<doc 9> for(int x=front;x<=back;x++) list[x]=temp[x]; }</pre>
<p>QUESTION 19</p> <p>What is the least restrictive running time for the worst case scenario for merge sort algorithm?</p> <p>A. O(1) B. O(log N) C. O(N^2)</p> <p>D. O(N) E. O(N log N)</p>	<pre>public static void outputList(int[]list){ for(int x=0;x<list.length;x++) out.print(list[x]+" "); out.println(); }</pre>

QUESTION 20

The two's complement system is all about representing negative numbers in binary. For example, the positive value 72 in 8-bit binary is **01001000**. To find the binary representation for -72 using two's complement, you use this easy conversion process. Start from the right and keep all zeroes the same until you reach the first 1 digit. Keep that 1 the same also, and flip everything else, with an 8-bit binary result of **10111000** for -72. With that in mind, which of the following choices represents the decimal equivalent of the two's complement binary value **10010101**?

- A. -109 B. -106 C. -107 D. -105 E. -108

QUESTION 21

What is output by the code to the right?

- A. 11 B. 59 C. 59.5 D. 11.5
E. There is no output due to an error.

```
int x = 31;
char a = 48;
double d = 19.5;
out.println(x-d+a);
```

QUESTION 22

What is output by the client code to the right?

- A.
Power = electric
Power = electric
Power = electric
B.
Power = motor
Power = motor
Power = electric
C.
Power = motor
Power = electric
Power = electric
D.
Power = motor
Power = motor
Power = motor
E. None of these

```
class Vehicle
{
    public String power = "motor";
    public void view()
    {
        out.println("Power = "
                    + power);
    }
}
class Car extends Vehicle
{
    public String power = "electric";
    public Car(String power)
    {
        this.power = power;
    }
    public void view()
    {
        out.println("Power = "
                    + power);
    }
}
Vehicle car = new Car("electric");
//statement 1
out.println("Power = " + car.power);
//statement 2
out.println("Power = " +
((Car)car).power);
//statement 3
car.view();
```

QUESTION 23

Which choice best describes statement 1 in the client code to the right?

- A. downcasting B. supercasting
C. early binding D. subcasting
E. late binding

QUESTION 24

Which choice best describes statement 2 in the client code to the right?

- A. downcasting B. supercasting
C. early binding D. subcasting
E. late binding

QUESTION 25

Which choice best describes statement 3 in the client code to the right?

- A. downcasting B. supercasting
C. early binding D. subcasting
E. late binding

<p>QUESTION 26</p> <p>What is output by the code to the right?</p> <p>A. HAPPY*N*Y*A*!* B. HAPPY N* Y*A*! C. HAPPY*N***Y*A*!* D. HAPPY*N*Y*A*! E. There is no output due to an error</p>	<pre>String s = "HAPPY NEW YEAR!"; args=s.split("[B-E R-W]+"); for(String t:args) out.print(t+"*");</pre>
<p>QUESTION 27</p> <p>What is output by the code to the right?</p> <p>A. 012340240304 B. 02134024030405 C. 012340123401234012340123401234 D. 0123402403042 E. There is no output due to an error.</p>	<pre>char[]list={'0','1','2','3','4'}; for(int x=1;x<=list.length;x++) for(char a:list) if(a%x==0) out.print(a);</pre>
<p>QUESTION 28</p> <p>What is output by the client code below?</p> <pre>//client code ListNode ln = new ListNode(); ln = new ListNode(3,ln); ln = new ListNode(5,ln); ListNode m = ln; while(m!=null) { out.print(m.val+" "); m=m.next; }</pre> <p>A. 0 0 0 B. 5 3 0 C. null null null D. 5 3 E. 0 3 5</p>	<pre>class ListNode { public ListNode() { val = 0; next = null; } public ListNode(int v, ListNode n) { val = v; next = n; } public int val; public ListNode next; }</pre>
<p>QUESTION 29</p> <p>What is output by the code to the right?</p> <p>A. 3 4 4.0 B. 4 3 4.0 C. 4 3 5.6 D. 3 4 5.6 E. There is no output due to an error.</p>	<pre>double [][] dubs = {{7.3,4.5,2.7}, {3.4,5.6,7.8}, {1.2,7.3,4.0}, {5.2,3.6,4.9}}; out.println(dubs.length+" "+ dubs[3].length + " "+dubs[2][2]);</pre>
<p>QUESTION 30</p> <p>What is output by the client code 1 to the right?</p> <p>A. 3 1 5 B. 1 2 4 C. 1 5 3 D. 0 5 3 E. There is no output due to an error.</p>	<pre>public static int mystNum(int x, int y, int z) { x=x/y+z; y=x/y+z; z=x/y+z; return x/y+z; }</pre> <pre>//client code 1 int x=8,y=2,z=-1; out.print(mystNum(x,y,z)+" "); x=10;y=5;z=3; out.print(mystNum(x,y,z)+" "); x=10;y=1;z=3; out.println(mystNum(x,y,z));</pre>
<p>QUESTION 31</p> <p>What is output by the client code 2 to the right?</p> <p>A. 3 5 -5 B. -3 5 -5 C. 4 2 8 D. -3 5 5 E. There is no output due to an error.</p>	<pre>//client code 2 int x=5,y=2,z=-1; out.print(mystNum(x,y,z)+" "); x=8;y=5;z=3; out.print(mystNum(x,y,z)+" "); x=9;y=3;z=-1; out.println(mystNum(x,y,z));</pre>

QUESTION 32

Using the generic stack pseudocode to the right, what was the last value popped, and which item is left at the top of the stack ?

- A. 7 3 B. 3 7
C. 7 4 D. 4 7
E. None of these

```
Push 9
Push 3
Push 5
Pop x
Push 4
Pop x
Pop x
Push 7
```

QUESTION 33

How many ordered triples make this boolean expression true?

$$\overline{A+B+C}$$

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

QUESTION 34

Infix notation is the kind normally used in algebraic expressions, such as $3 + 5 * 6$, where the operators are between the operands. However, there is also prefix notation, where the operators are before the operands, such as $+ 3 * 5 6$, and postfix notation, operators after operands, like this: $3 5 6 * +$. Notice carefully that the operands never move around: only the operators change places.

Here is another example: the infix expression $6 * 7 + 9 - 8 * 2$ translates the prefix expression $- + * 6 7 9 * 8 2$, and $6 7 * 9 + 8 2 * -$ for postfix.

Given these examples to examine and study carefully, which of the infix expressions below matches the postfix expression shown?

$3 9 * 6 - 5 2 ^ +$

- A. $3 - 9 * 6 + 5 ^ 2$ B. $3 * 9 + 6 - 5 ^ 2$ C. $3 * 9 - 6 + 5 ^ 2$
D. $9 * 6 - 5 ^ 2 + 3$ E. None of these

QUESTION 35

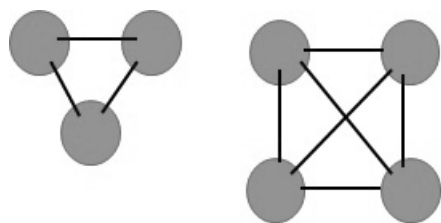
What is output by the code to the right?

- A. 0 B. 00000014
C. 00000000 D. 14.00000
E. 14.0

```
byte c = 100;
c>>=7;
out.println(Integer.toBinaryString(c));
```

QUESTION 36

Below are two complete graphs. The three node graph has three edges, and the four node graph has 6 edges. How many edges would a six-node complete graph have?



- A. 8 B. 12 C. 15 D. 9 E. None of these

QUESTION 37

What is output by the code to the right?

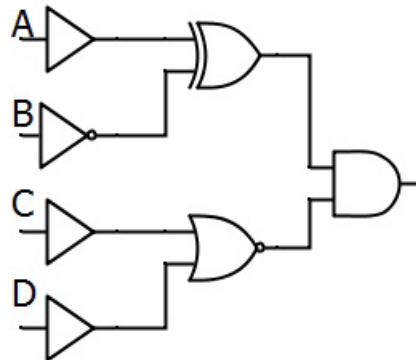
- A. [1.2, 3.4, 5.6, 9.4][1.1, 1.2, 3.4, 5.6, 9.4]
- B. [1.2, 5.6, 3.4, 9.4][1.1, 1.2, 3.4, 9.4, 5.6]
- C. [1.2, 3.4, 5.6, 9.4][1.2, 3.4, 5.6, 9.4, 1.1]
- D. [3.4, 5.6, 1.2, 9.4][3.4, 5.6, 1.2, 9.4, 1.1]
- E. None of these

```
double[]list = {3.4, 5.6, 1.2, 9.4};
ArrayList<Double>dList= new
    ArrayList<Double>();
for(double d:list)
    dList.add(d);
PriorityQueue<Double> pqd = new
    PriorityQueue<Double>(dList);
out.print(pqd);
pqd.add(1.1);
out.println(pqd);
```

QUESTION 38

Which of the following logical statements is represented by the digital electronics diagram shown?

- A. $A \oplus \bar{B} * \bar{C} + D$
- B. $(A + \bar{B}) * \bar{C} \oplus D$
- C. $A \oplus \bar{B} + \bar{C} * D$
- D. $A * \bar{B} + \bar{C} \oplus D$
- E. None of these



QUESTION 39

Find $f(20)$ according to the recursive function definition shown below.

$f(20) =$

$f(x) = f(x-4) + 1$ when $x > 5$
 $f(x) = 2$ otherwise

QUESTION 40

Simplify this expression to have only two operators and one NOT. The allowable operators include AND(*), OR(+), XOR(\oplus), and NOT (over bar).

$$\overline{(A + B * C)} * \overline{(A * (B * C))}$$

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)
- o static int parseInt(String s, int radix)

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)

```