

University Interscholastic League

Computer Science Competition

Number 120 (Invitational B - 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 AC_{16} and 4_{16} ?

- A. ACE_{16} B. 100_{10} C. 10110101_2 D. 10110000_2 E. 11_2

QUESTION 2

What is output by the code to the right?

- A. 0 B. 2 C. 5
D. 10 E. 22

```
int y = 5 % (10 + 12) / 2;
System.out.print(y);
```

QUESTION 3

What is output by the code to the right?

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

```
int tally = 0;
for(int i = 5; i > 1; i--)
    tally++;
System.out.print(tally);
```

QUESTION 4

What is output by the code to the right?

- A. LARRYWALLPERL
B. LARRYWALL--PERL
C. LarryWall--Perl
D. LARRYWALL__PERL
E. LARRYWALL++PERL

```
String cser = "LarryWall--Perl";
cser = cser.toUpperCase();
System.out.print(cser);
```

QUESTION 5

What is output by the code to the right?

- A. null B. 0 C. 6
D. 5 E. 1

```
int[] ts = new int[5];
ts[1] = ts[2] + ts[4];
System.out.print(ts[1]);
```

QUESTION 6

What is output by the code to the right?

- A. 36 B. 22 C. 33
D. 54 E. 20

```
int w = 3;
int z = 2;
z *= (z + 1) * w + z;
System.out.print(z);
```

QUESTION 7

What is output by the code to the right?

- A. false true B. true false
C. true true D. false false
E. false true false true

```
boolean p = false;
boolean q = true;
System.out.print( (p && q) + " ");
System.out.print( p || q );
```

<p>QUESTION 8</p> <p>What is output by the code to the right?</p> <p>A. 2 B. 1 C. 13</p> <p>D. 3 E. 23</p>	<pre>boolean r = true; boolean s = true; if(r ^ s) System.out.print(1); else if(r s) System.out.print(2); else System.out.print(3);</pre>
<p>QUESTION 9</p> <p>What replaces <*1> in the code to the right so that the instance variable <code>price</code> is only accessible in the <code>Rental</code> class?</p> <p>A. <code>final</code> B. <code>public</code> C. <code>static</code></p> <p>D. <code>private</code> E. <code>implements</code></p>	<pre>public class Rental{ <*1> int price; public Rental(int p){ price = p; } public Rental(){ price = 10; } public int getPrice(){ return price; } }</pre>
<p>Assume <*1> is filled in correctly.</p>	<pre>public class Movie extends Rental { public String toString(){ return "movie:" + getPrice(); } } //////////////////////////////////// // client code Movie m = new Movie(); System.out.println(m);</pre>
<p>QUESTION 5</p> <p>What is output by the client code to the right?</p> <p>A. <code>Rental@</code></p> <p>B. <code>Movie@</code></p> <p>C. <code>movie:</code></p> <p>D. <code>10</code></p> <p>E. <code>movie:10</code></p>	
<p>QUESTION 11</p> <p>What is output by the code to the right?</p> <p>A. 63 B. 64 C. 127</p> <p>D. 1023 E. 2147483647</p>	<pre>int bx = 63; int ax = 64; int cx = ax bx; System.out.print(cx);</pre>
<p>QUESTION 12</p> <p>What is output by the code to the right?</p> <p>A. 3 B. 9 C. 2.99999</p> <p>D. 81 E. 3.0</p>	<pre>double res = Math.sqrt(9); System.out.print(res);</pre>
<p>QUESTION 13</p> <p>What is output by the code to the right?</p> <p>A. 1 2 3 B. "1""2""3" C. 3</p> <p>D. One E. 123</p> <p>Two</p> <p>Three</p>	<pre>System.out.print("1"); System.out.print("2"); System.out.print("3");</pre>

<p>QUESTION 14</p> <p>What is output by the code to the right?</p> <p>A. 00150 B. "%05d" C. +150</p> <p>D. 150.00 E. 150</p>	<pre>int value = 150; int len = (" " + value).length() + 2; String fr = "%0" + len + "d"; System.out.printf(fr, value);</pre>
<p>QUESTION 15</p> <p>What is returned by the method call <code>repeat(-3)</code>?</p> <p>A. -12 B. 2 C. 0</p> <p>D. -6 E. -4</p>	<pre>public int repeat(int val){ int res = 0; if(val >= 0) res = 2; else res = val + repeat(val + 1) + res; return res; }</pre>
<p>QUESTION 16</p> <p>How many '*'s are output by the code to the right?</p> <p>A. 30 B. 10 C. 20</p> <p>D. 18 E. 22</p>	<pre>int lim = 10; for(int i = 0; i < lim; i++) System.out.print('*'); lim *= 2; for(int i = 0; i < lim; i++) System.out.print('*');</pre>
<p>QUESTION 17</p> <p>What is output by the code to the right?</p> <p>A. false false B. true true</p> <p>C. false true D. true false</p> <p>E. 1 1</p>	<pre>String a1 = "DOG"; String a2 = "CAT"; String a3 = "BEAR"; boolean b1 = a1.compareTo(a2) < 0; boolean b2 = a2.compareTo(a3) > 0; System.out.print(b1 + " " + b2);</pre>
<p>QUESTION 18</p> <p>What is output by the code to the right?</p> <p>A. 8 B. 6</p> <p>C. 9 D. 7</p> <p>E. There is no output due to a syntax error.</p>	<pre>final int LIMIT; int x = 3; int y = 2; LIMIT = x * y; System.out.print(LIMIT + x);</pre>
<p>QUESTION 19</p> <p>What is output by the code to the right?</p> <p>A. true</p> <p>B. false</p> <p>C. LO</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>ArrayList<String> list1; list1 = new ArrayList<String>(); ArrayList<String> list2; list2 = new ArrayList<String>(); list1.add("LO"); list2.add(list1.get(0)); System.out.print(list1 == list2);</pre>

QUESTION 20

Consider the following problem description. In the game Monopoly there are spaces on the board that represent properties that players can own. There are various behaviors and data common to all types of properties. Railroads are a kind of property with behaviors that are unique to railroads. Utilities are another kind of property with behaviors that are unique to utilities. Which of the following set of class headers best represents the relationships between properties, railroads, and utilities?

- A. `public class Property`
`public class Railroad`
`public class Utility`
- B. `public class Utility`
`public class Railroad extends Utility`
`public class Property extends Railroad`
- C. `public class Railroad`
`public class Utility`
`public class Property extends Railroad, Utility`
- D. `public class Property`
`public class Railroad extends Property`
`public class Utility extends Property`
- E. `public class Property`
`public class Railroad extends Property`
`public class Utility extends Railroad`

QUESTION 21

What is output by the code to the right?

- A. 0123
- B. 0123445566778899
- C. 0123456789
- D. 456789
- E. 012344

```
for(int i = 0; i < 10; i++){
    System.out.print(i);
    if( i > 3 ){
        System.out.print(i);
        break;
    }
}
```

QUESTION 22

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

- A. 1 B. 4 C. 7
- D. 5 E. 3

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

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

QUESTION 23

Which of the following is not a valid Java identifier?

- A. `_length` B. `prop386Time` C. `3times` D. `MAGIC_NUM` E. `finalClass`

<p>QUESTION 24</p> <p>The intent of the code to the right is to print out the values of (N!) as N goes from 1 to 30. However the code does not work as intended. Which of the following best explains why the code does not correctly print out all values of (N!) as N goes from 1 to 30?</p> <p>A. The variable <code>fact</code> needs to be initialized to 0.</p> <p>B. The computation of factorial is wrong. The code segment <code>fact = fact * i;</code> must be changed to <code>fact = fact + i;</code></p> <p>C. The range of the data type <code>long</code> is not adequate to represent N! except for small values of N.</p> <p>D. The expression <code>fact * i</code> evaluates to a double not a <code>long</code>.</p> <p>E. <code>long</code>s cannot be concatenated with <code>Strings</code>.</p>	<pre>long fact = 1; for(int i = 1; i <= 30; i++){ fact = fact * i; System.out.println(i + " " + fact); }</pre>
<p>QUESTION 25</p> <p>What is output by the code to the right when <code>nums</code> is the following array?</p> <p>{2, 1, 2, 3, 3, 4, 3}</p> <p>A. 0</p> <p>B. -4</p> <p>C. 8</p> <p>D. 4</p> <p>E. 22</p>	<pre>public void process(int[] nums){ int total = 0; for(int i : nums){ switch(i) { case 1 : total++; break; case 2 : total -= 1; break; case 3 : total *= 2; break; default: total += 2; } } System.out.print(total); }</pre>
<p>QUESTION 26</p> <p>What is output by the code to the right?</p> <p>A. 5 B. 8 C. 1</p> <p>D. 0 E. 7</p>	<pre>String nm; nm = " Charles,William,Bach"; int index = nm.indexOf(','); String part; part = nm.substring(index).substring(5); System.out.print(part.length());</pre>
<p>QUESTION 27</p> <p>What is returned by the method call <code>tough(5)</code>?</p> <p>A. -25 B. -11 C. -45</p> <p>D. 20 E. -13</p>	<pre>public int tough(int n){ if(n <= 0) return -3; else return -n + tough(n - 2) + tough(n - 4); }</pre>
<p>QUESTION 28</p> <p>What is output by the code to the right?</p> <p>A. false true B. true true</p> <p>C. true false D. false false</p> <p>E. true false true</p>	<pre>char ans = 'Y'; boolean b1, b2; b1 = ans != 'Y' ans != 'y'; b2 = ans > 'A'; System.out.print(b1 + " " + b2);</pre>

QUESTION 29

What is output by the code to the right when method gamma is called?

- A. 2 2
- B. 2 4
- C. 2 0
- D. 0 2
- E. 4 4

```
public int alpha(int x, int y){
    x *= 2;
    y -= 2;
    return x - y;
}
```

```
public int beta(int z){
    z++;
    z = alpha(z, z / 2);
    return z;
}
```

QUESTION 30

What is output by the code to the right when method delta is called?

- A. 2
- B. 0
- C. 7
- D. 11
- E. 8

```
public void gamma(){
    int x = 2;
    System.out.print( alpha(x, x + 2) );
    System.out.print( " " + x );
}
```

```
public void delta(){
    int x = beta(3);
    System.out.print( x );
}
```

QUESTION 31

Consider method instrumentedSort to the right. If nums.length equals 10, what will method instrumentedSort return?

- A. 45
- B. 55
- C. 10
- D. 3628800
- E. The output cannot be determined unless the code is actually run.

```
public int instrumentedSort(int[] nums){
    int comps = 0;
    int lim1 = nums.length - 1;
    int lim2 = nums.length;
    for(int i = 0; i < lim1; i++){
        int min = i;
        for(int j = i + 1; j < lim2; j++){
            comps++;
            if( nums[j] < nums[min] )
                min = j;
        }
        int temp = nums[i];
        nums[i] = nums[min];
        nums[min] = temp;
    }
    return comps;
}
```

QUESTION 32

Which sorting algorithm does the method instrumentedSort implement?

- A. Quicksort
- B. Radix sort
- C. Merge sort
- D. Insertion sort
- E. Selection sort

QUESTION 33

The following values are inserted one at a time into a binary search tree using the traditional insertion algorithm. How many leaves are in the resulting tree?

13, 0, -5, 5, 10, 7, 15, 21

- A. 1
- B. 8
- C. 0
- D. 3
- E. 6

QUESTION 34

Which of the following can be sent as an argument to method `showAll` so that the elements in `c` are guaranteed to be printed out in ascending order?

- A. `HashSet<Integer>`
- B. `LinkedList<Integer>`
- C. `Stack<Integer>`
- D. `ArrayList<Integer>`
- E. `TreeSet<Integer>`

```
public void showAll(Collection<Integer> c){
    for(Integer val : c)
        System.out.println( val );
}
```

QUESTION 35

What is returned by method `op` if `mat` is the matrix shown below?

2	3	2	12	-3	-6
2	-1	-5	9	3	0
-3	7	7	-2	-1	3
11	5	3	3	5	10

- A. 4
- B. 11
- C. 7
- D. 41
- E. 21

```
public int op(int[][] mat){
    int total = 0;
    for(int r = 0; r < mat.length; r += 2){
        final int LIM = mat[r].length;
        for(int c = r; c < LIM; c += 2){
            total += mat[r][c];
        }
    }
    return total;
}
```

QUESTION 36

The intent of method `freq` to the right is to create and return a `map` with a tally of the number of times each letter occurs in the file that the `Scanner sc` is connected to, but it does not work as intended. What must be changed to make the method work as intended?

- A. Change the segment `result.put((char)('A' + i), 0)` to `result.put(0)`
- B. Change the boolean expression `Character.isLetter(ch)` to `ch != null`
- C. Change the segment `x++` to `result.put(ch, x + 1)`
- D. Change the boolean expression `Character.isLetter(ch)` to `Character.isLetter(ch) == true`
- E. Change the statement `int x = result.get(ch)` to `int x = result.get(i)`

```
public Map<Character, Integer> freq(
    Scanner sc){
    Map<Character, Integer> result =
        new TreeMap<Character, Integer>();

    for(int i = 0; i < 26; i++){
        result.put( (char)('A' + i), 0);
    }

    while( sc.hasNextLine() ){
        String line = sc.nextLine();
        for(int i = 0; i < line.length(); i++){
            char ch = line.charAt(i);
            if( Character.isLetter(ch) ){
                ch = Character.toUpperCase(ch);
                int x = result.get(ch);
                x++;
            }
        }
    }

    return result;
}
```


QUESTION 37

Assume method `sample(int[] data)` is $O(N^2)$ where $N = \text{data.length}$. When method `sample` is passed an array with `length = 4,000` it takes 4 seconds for method `sample` to complete. If method `sample` is then passed an array with `length = 6,000` what is the expected time it will take method `sample` to complete?

- A. 4 seconds B. 6 seconds C. 9 seconds D. 32 seconds E. 16 seconds

QUESTION 38

What is output by the following client code?

```
int[] data = { 2, 9, 9, 7, 9, 2 };
Structure<Integer> st1;
st1 = new Structure<Integer>();

for(int i : data)
    st1.add(i);
```

```
st1.remove();
st1.remove();
System.out.print( st1.remove() + " "
    + st1.remove() );
```

- A. 7 9 B. 9 9 C. 2 9
D. 9 2 E. 9 7

```
public class Structure<E>{

    private ArrayList<E> con;

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

    public void add(E obj){
        con.add(obj);
    }

    public E access(){
        return con.get(con.size() - 1);
    }

    public E remove(){
        return con.remove(con.size() - 1);
    }

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

    /*
    public Structure<E> myst(){
        Structure<E> result
            = new Structure<E>();
        result.con = this.con;
        return result;
    }
    */
}
```

QUESTION 39

Assume method `myst` is uncommented. What is output by the following client code?

```
int[] data = { 2, 9, 9, 7, 9, 2 };
Structure<Integer> st2;
st2 = new Structure<Integer>();

for(int i : data)
    st2.add(i);
st2.add( st2.access() );

Structure<Integer> st3 = st2.myst();
st2.remove();
st2.remove();
System.out.print( st3.remove() + " " +
    st3.remove() );
```

- A. 7 9 B. 9 7 C. 2 9
D. There is no output due to a syntax error in method `myst`.
E. There is no output due to a runtime error caused by method `myst`.

QUESTION 40

What type of data structure does the `Structure` class implement assuming method `myst` is commented out?

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

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 B - 2010

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