University Interscholastic League

Computer Science Competition

Number 111 (Regional - 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. 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 101100_2 and $C09_{16}$?

A. $C39_{16}$ B. $C41_{16}$ C. 3026_8 D. 6065_8 E. 5101_8

QUESTION 2

What is output by the code to the right?

A. 2 B. 1 C. 0

D. 21 E. 10

```
int x = 21;
int y = 10;
System.out.print( y % x );
```

QUESTION 3

What is output by the code to the right?

A. 0 B. 15 C. 1

D. -15 E. 16

```
int total = 0;
for(int i = 15; i >= 0; i--)
  total++;

System.out.print( total );
```

QUESTION 4

What is output by the code to the right?

A. bess_z_16

B. bessz

C. bess

D. bess_z_!^

E. e

```
String name = "BeSS_Z_16";
name = name.toLowerCase();
System.out.println( name );
```

QUESTION 5

What is output by the code to the right?

A. false B. true C. null

D. -1 E. The output cannot be known until runtime.

```
boolean[] used = new boolean[5];
boolean result = used[2] || used[3];
System.out.print( result );
```

QUESTION 6

What is output by the code to the right?

A. 9 B. 6.25 C. 9.25

D. 7.75 E. 9.0

```
double a = 2.25;
a++;
double b = 1.25 + a * 2;
System.out.print( b );
```

QUESTION 7

Which answer is logically equivalent to the following Boolean expression? `p` and `q` are `boolean` variables.

```
!( !p && q )
```

A. `!!p && !q` B. `p || q` C. `true` D. `p && q` E. `p || !q`

QUESTION 8

What is output by the code to the right?

A. 5 B. 4 C. 3

D. 2 E. 1

```
int r = 3, s = 6;
if( s > r && r > 0 ){
  r++;
  if( s > 5 )
    r++;
}
else
    r--;
System.out.print( r );
```

QUESTION 9

What replaces **<*1>** in the code to the right so that method `won` increments the instance variable `wins` by 1?

A. `++wins`

B. `won()`

C. `this.wins++`

D. `wins.inc()`

E. More than one of these.

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

```
public class Record{
  private int wins;
  private int losses;

  public void won(){
    <*1>;
  }

  public void lost(){
    losses++;
  }

  public double ave(){
    int total = wins + losses;
    return (double)wins / total;
  }
}

////////////////////////////////////////
// client code
Record rec = new Record();
rec.lost();
System.out.print( rec.ave() );
```

QUESTION 10

What is output by the client code to the right?

A. `0` B. `NaN` C. `0.0`

D. There is no output due to a syntax error in the client code.

E. There is no output due to a runtime error that occurs when the client code is executed.

QUESTION 11

What is output by the code to the right?

A. `127` B. `1` C. `15000`

D. `120` E. `153`

```
int m = 15;
int n = 3;
m = m << n;
System.out.println( m );
```

QUESTION 12

What is output by the code to the right?

A. `-3.0` B. `0` C. `-2`

D. `-3` E. `-2.0`

```
double val = -2.5;
System.out.print( Math.floor(val) );
```

QUESTION 13

What is output by the code to the right?

A. `ed`
`laz`

B. `ed laz`

C. `ed`

D. `"ed\nlaz"`

E. `ed\nlaz`

```
String ch = "ed\nlaz";
System.out.print( ch );
```

QUESTION 14

What is output by the code to the right?

A. `1.8` B. `1.800` C. `1.789`

D. `1` E. `2.0`

```
String format = "%3.1f";
double v = 1.789;
System.out.printf( format, v );
```

QUESTION 15

What is returned by the method call `example(3, 8)`?

A. `-5` B. `-6` C. `6`

D. `4` E. `5`

```
public static int example(int x, int y){
  y -= x;
  x--;
  return y++;
}
```

QUESTION 16

What is output by the code to the right if the value `4` is entered at the call to `key.nextInt()`?

A. `4`

B. `12`

C. The code successfully completes execution with no output.

D. There is no output due to a syntax error.

E. There is no output due to an `ArrayIndexOutOfBoundsException`.

```
int[] temps = {7, 13, 16, 12};
Scanner key = new Scanner( System.in );
int d = key.nextInt();

if( d > 0 || d < temps.length )
  System.out.print( temps[d] );
```

QUESTION 17

How many `*`'s are output by the code to the right?

A. `5` B. `24` C. `36`

D. `10` E. `25`

```
for(int i = 1; i < 6; i++){
  for(int j = 1; j < 6; j++){
    if( i != j )
      System.out.print( "*" );
    else
      break;
  }
}
```

QUESTION 18

What is returned by the method call `self(11)`?

A. `5` B. `11` C. `6`

D. `1` E. `0`

```
public static int self(int n){
  if( n <= 0 )
    return 0;
  else
    return 1 + self( n - 2 );
}
```

QUESTION 19

What is output by the code to the right?

A. `[Z, MM, A]` B. `[A, MM, Z]`

C. `[A, Z, MM]` D. `[Z, A, MM]`

E. There is no output due to a syntax error.

```
ArrayList<String> letters;
letters = new ArrayList<String>();
letters.add("Z");
letters.add("A");
letters.add("MM");
Collections.sort( letters );
System.out.println( letters );
```

QUESTION 20

What replaces **<*1>** in the code to the right to indicate the `Lunch` class inherits the `Meal` class?

A. `inherits`

B. `extends`

C. `implements`

D. `isa`

E. `sub`

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

QUESTION 21

What is output by the line marked `Line 1` in the client code to the right?

A. `0`

B. `2`

C. `6`

D. `3`

E. `5`

QUESTION 22

What is output by the line marked `Line 2` in the client code to the right?

A. `14`

B. `7`

C. `7.5`

D. There is no output due to a syntax error in that section of client code.

E. There is no output due to a `ClassCastException`.

```
public class Meal{

  private int price;

  public Meal(int p){
    price = p;
  }

  public int getPrice(){
    return price;
  }

  public void inc(){
    price++;
  }
}

public class Lunch <*1> Meal{

  private boolean hasDrink;

  public Lunch(int p, boolean d){
    super( p );
    hasDrink = d;
  }

  public int getPrice(){
    int result = super.getPrice() / 2;
    if( hasDrink )
      result++;
    return result;
  }
}

////////////////////////////////////////
// client code
Meal sat = new Meal(5);
sat.inc();
System.out.print( sat.getPrice() ); //Line 1


Meal sun = new Lunch(13, true);
System.out.print( sun.getPrice() ); //Line 2
```

QUESTION 23

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

A. `2`

B. `-1`

C. `3`

D. `4`

E. `0`

```
public static void first(int[] data){
  data = new int[4];
}

public static void second(){
  int[] list = {2, 3};
  first( list );
  System.out.print( list[1] );
}
```

QUESTION 24

What is output by the code to the right?

A. `[8, 12, 10]` B. `[8, 10, 12]`

C. `[12, 10, 8]` D. `[10, 8]`

E. `[10, 12, 8]`

```
LinkedList<Integer> scores;
scores = new LinkedList<Integer>();
scores.addLast( 12 );
scores.addFirst( 10 );
scores.add( 8 );
System.out.println( scores );
```

QUESTION 25

What is output by the code to the right?

A. `5` B. `10` C. `9`

D. There is no output due to a syntax error.

E. There is no output due to an `ArrayIndexOutOfBoundsException`.

```
char[][] table = new char[10][5];
System.out.print( table[8].length );
```

QUESTION 26

Method `sort` to the right attempts to implement the insertion sort algorithm, but the method has one or more syntax errors. Which of the following best describes how to correct the syntax error(s)?

A. The `int` variable `j` and the `String` variable `cur` must both be declared before the `for` loop.

B. The statement
`vals[j + 1] = vals[j--];`
must be replaced with the following statements:

```
vals[j + 1] = vals[j];
j--;
```

C. The statement `String cur = vals[i];` must be replaced with the following statement:

```
String cur = new String( vals[i] );
```

D. The line
`while( j >= 0 && cur < vals[j] ){`
must be replaced with the following:

```
while( j >= 0
  && cur.compareTo( vals[j] ) < 0 ){
```

E. More than one of these.

```
// sort into ascending value
// using the insertion sort algorithm

public static int sort(String[] vals){
  int count = 0;

  for(int i = 1; i < vals.length; i++){
    int j = i - 1;
    String cur = vals[i];

    while( j >= 0 && cur < vals[j] ){
      vals[j + 1] = vals[j--];
      count++;
    }

    vals[j + 1] = cur;
  }
  return count;
}
```

Assume method `sort` has been corrected.

QUESTION 27

What is returned by method `sort` if `vals` initially contains the following `String`s?

`{"S", "O", "M", "K", "D", "B", "A"}`

A. `0` B. `18` C. `7`

D. `21` E. `6`

QUESTION 28

What is output by the client code to the right?

A. `true`

B. `false`

C. `null`

D. There is no output due to a syntax error.

E. There is no output due to an `IndexOutOfBoundsException`.

```
public static boolean same(
                    ArrayList<Object> ls){
  Object last = ls.get( ls.size() - 1 );
  return ls.get(0).equals( last );
}

////////////////////////////////////////
// client code
ArrayList<String> initials;
initials = new ArrayList<String>();
initials.add("DT");
initials.add("TD");
initials.add("dt");
System.out.println( same( initials ) );
```

QUESTION 29

What is output by the code to the right assuming method `handle` is sent a `Scanner` object that is connected to a file that contains the following data?

```
.5  1  0.1
+0.5 1G
0.5 1.5
```

A. `out5.1`

B. `not2.1`

C. `not0.0`

D. `not0.50.5`

E. `37.1`

```
/* nextDouble() throws an
   InputMisMatchException when the next
   input token cannot be translated into
   valid double value.

   nextDouble() throws a
   NoSuchElementException if the input is
   exhausted.
*/

public static void handle(Scanner s){
  double total = 0.0;
  try{
    for(int i = 0; i < 10; i++)
      total += s.nextDouble();
    System.out.print( total );
  }
  catch(InputMismatchException e1){
    System.out.print("not");
  }
  catch(NoSuchElementException e2){
    System.out.print("out");
  }
  System.out.print( total );
}
```

QUESTION 30

What is output by method `figure` if `input` initially contains the following `Strings`?

`{"Java", "C", "Eiffel", "C++"}`

A. `[Java, Eiffel]`

B. `[Java, Eiffel, C++]`

C. `[C, C++]`

D. `[Java, C]`

E. There is no output due to an `IllegalStateException`.

```
public static void figure
                  (ArrayList<String> input){
  Iterator<String> it = input.iterator();
  while( it.hasNext() ){
    if( it.next().length() <= 3 )
      it.remove();
  }

  System.out.println( input );
}
```

QUESTION 31

If the `ArrayList nums` contains N items what is the running time of method `removeAll`? Choose the most restrictive correct answer.

A. O(N) B. $O(N^2)$ C. $O(N^3)$

D. $O(N^{1/2})$ E. O(NlogN)

```
public static void removeAll
                  (ArrayList<Integer> nums){
  int i;
  for(i = nums.size() - 1; i >= 0; i--){
    nums.remove(i);
  }
}
```

QUESTION 32

Which of these method calls returns `5`?

A. `pieces("1cs2cs3cs4cs5cs_easy")`

B. `pieces("cs1cs2cscs315sc")`

C. `pieces("apcs_uilcs_ibcs_hscs_")`

D. Both A and B

E. Both B and C

```
public static int pieces(String st){
  String[] result = st.split("cs");
  return result.length;
}
```

QUESTION 33

Given the following measurements, what is the most likely running time for method `sample(int[] data)` where N is equal to `data.length`? Choose the most restrictive correct answer.

Value of N	Time for method `sample` to complete
1,000	1 second
4,000	64 seconds
16,000	4096 seconds

A. O(NlogN) B. $O(N^3)$ C. $O(N^4)$ D. $O(2^N)$ E. O(N!)

QUESTION 34

What is returned by the method call `find(mat, -2)` where `mat` is the 2D array below?

0	4	8	5	8	5	2	-2
9	3	8	1	8	5	1	2
9	9	7	7	5	5	5	3
-4	5	3	7	3	3	2	-2
0	0	0	0	0	0	0	0
1	0	0	6	0	0	-3	3
2	1	-1	3	-1	2	-2	5
2	4	-2	2	-1	-1	7	0

A. `0` B. `1` C. `2`

D. `4` E. `8`

```
//pre: mat is a square 2d array
public static int find(int[][] mat, int t){
  final int L = mat.length - 1;
  int res = 0;
  boolean ok = true;
  int c = 0;
  int i;

  while( c < mat.length ){
    ok = mat[c][L] >= t && mat[L][c] >= t;
    if( ok ){
      i = 1;
      while( ok && i <= L ){
        ok = mat[c][i] <= mat[c][i-1];
        ok =ok && mat[i][c] <= mat[i-1][c];
        i++;
      }
      res = ok ? res + 1 : res;
    }
    c++;
  }
  return res;
}
```

QUESTION 35

Consider the following class that attempts to implement the `Structure` interface shown to the right.

```
public class SimpleStructure
                    implements Structure{
  private Comparable comp;

  public Structure add(Comparable obj){
    comp = obj;
  }
}
```

Which of the following is true about the `SimpleStructure` class?

A. The class compiles with no errors.

B. The class does not compile because it does not implement the `toString` method.

C. The class does not compile because the `add` method does not return a `Structure`.

D. The class does not compile because it does not have a constructor.

E. More than one of these.

QUESTION 36

What is output by the client code to the right?

A. `75056`

B. `05567`

C. `7650`

D. `0567`

E. There is no output.

QUESTION 37

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

A. A linked list

B. A binary search tree

C. A min heap

D. A max heap

E. A stack

```
public interface Structure{
  public Structure add(Comparable obj);
  public String toString();
}

public class EmptyStructure
                      implements Structure{
  public Structure add(Comparable obj){
    return new NEStructure( obj );
  }

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

public class NEStructure
                      implements Structure{
  private Comparable data;
  private Structure left;
  private Structure right;

  public NEStructure(Comparable obj){
    data = obj;
    left = new EmptyStructure();
    right = new EmptyStructure();
  }

  public Structure add(Comparable obj){
    int val = obj.compareTo( data );
    if( val < 0 )
      left = left.add( obj );
    else if( val > 0 )
      right = right.add( obj );
    return this;
  }

  public String toString(){
    return right.toString() + data +
                            left.toString();
  }
}
////////////////////////////////////////
// client code
Structure s1 = new EmptyStructure();
s1 = s1.add(7);
s1 = s1.add(5);
s1 = s1.add(0);
s1 = s1.add(5);
s1 = s1.add(6);
System.out.println( s1 );
```

QUESTION 38

What is output by the client code to the right?

A. `A`

B. `B`

C. `null`

D. There is no output due to a syntax error in the client code.

E. There is no output due to a `NullPointerException`.

```
public class Node{
  public Object data;
  public Node p, n;

  public Node(Object obj){
    data = obj;
  }
}
////////////////////////////////////////
// client code
Node a = new Node("A");
Node b = new Node("B");
a.p = a;
a.n = b;
b.p = a.p.p;
b.n = b.p.n.p;
System.out.println( b.n.data );
```

QUESTION 39

What type of variables must `one` and `two` be so that the running time of method `countMatches` is O(N) when it returns `0`? The variables `one` and `two` both contain N distinct elements, but they do not share any elements in common.

	one	two
A.	`ArrayList`	`HashSet`
B.	`LinkedList`	`ArrayList`
C.	`ArrayList`	`ArrayList`
D.	`ArrayList`	`TreeSet`
E.	`LinkedList`	`HashSet`

```
public int countMatches(List<Integer> one,
                        Collection<Integer> two){

  int totalMatches = 0;
  for(int i = 0; i < one.size(); i++)
    if( two.contains( one.get( i ) ) )
      totalMatches++;

  return totalMatches;
}
```

QUESTION 40

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

A. `0`

B. `6`

C. `null`

D. There is no output due to a syntax error.

E. There is no output due to an `ArrayIndexOutOfBoundsException`.

```
public static void poorlyWritten(int[] data){
  int total = 0;
  try{
    int i = 0;
    while( true ){
      total += data[i];
      i++;
    }
  }
  catch(ArrayIndexOutOfBoundsException e){
  }
  System.out.print( total );
}

public static void doExample(){
  int[] data = {1, 2, 3};
  poorlyWritten( data );
}
```

class java.lang.Object
- boolean equals(Object other)
- String toString()
- int hashCode()

interface java.lang.Comparable<T>
- 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>
- Integer(int value)
- int intValue()
- boolean equals(Object obj)
- String toString()
- int compareTo(Integer anotherInteger)
- static int parseInt(String s)

class java.lang.Double implements Comparable<Double>
- Double(double value)
- double doubleValue()
- boolean equals(Object obj)
- String toString()
- int compareTo(Double anotherDouble)
- static double parseDouble(String s)

class java.lang.String implements Comparable<String>
- int compareTo(String anotherString)
- boolean equals(Object obj)
- int length()
- String substring(int begin, int end)
  Returns the substring starting at index begin and ending at index (end - 1).
- String substring(int begin)
  Returns substring(from, length()).
- int indexOf(String str)
  Returns the index within this string of the first occurrence of str. Returns -1 if str is not found.
- 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.
- charAt(int index)
- int indexOf(int ch)
- int indexOf(int ch, int fromIndex)
- String toLowerCase()
- String toUpperCase()
- String[] split(String regex)
- boolean matches(String regex)

class java.lang.Character
- static boolean isDigit(char ch)
- static boolean isLetter(char ch)
- static boolean isLetterOrDigit(char ch)
- static boolean isLowerCase(char ch)
- static boolean isUpperCase(char ch)
- static char toUpperCase(char ch)
- static char toLowerCase(char ch)

class java.lang.Math
- static int abs(int a)
- static double abs(double a)
- static double pow(double base, double exponent)
- static double sqrt(double a)
- static double ceil(double a)
- static double floor(double a)
- static double min(double a, double b)
- static double max(double a, double b)
- static int min(int a, in b)
- static int max(int a, int b)
- static long round(double a)
- 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>
- boolean add(E e)
- int size()
- Iterator<E> iterator()
- ListIterator<E> listIterator()

class java.util.ArrayList<E> implements List<E>
Methods in addition to the List methods:
- E get(int index)
- E set(int index, E e)
  Replaces the element at index with the object e.
- 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.
- 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:
- void addFirst(E e)
- void addLast(E e)
- E getFirst()
- E getLast()
- E removeFirst()
- E removeLast()

class java.util.Stack<E>
- boolean isEmpty()
- E peek()
- E pop()
- E push(E item)

interface java.util.Queue<E>
- boolean add(E e)
- boolean isEmpty()
- E peek()
- E remove()

class java.util.PriorityQueue<E>
- boolean add(E e)
- boolean isEmpty()
- E peek()
- E remove()

interface java.util.Set<E>
- boolean add(E e)
- boolean contains(Object obj)
- boolean remove(Object obj)
- int size()
- Iterator<E> iterator()
- boolean addAll(Collection<?> extends E> c)
- boolean removeAll(Collection<?> c)
- 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>
- Object put(K key, V value)
- V get(Object key)
- boolean containsKey(Object key)
- int size()
- Set<K> keySet()
- 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>
- K getKey()
- V getValue()
- V setValue(V value)

interface java.util.Iterator<E>
- boolean hasNext()
- E next()
- void remove()

interface java.util.ListIterator<E> extends java.util.Iterator<E>

Methods in addition to the `Iterator` methods:
- void add(E e)
- void set(E e)

class java.lang.Exception
- Exception()
- Exception(String message)

class java.util.Scanner
- Scanner(InputStream source)
- boolean hasNext()
- boolean hasNextInt()
- boolean hasNextDouble()
- String next()
- int nextInt()
- double nextDouble()
- String nextLine()
- Scanner useDelimiter(String pattern)

Computer Science Answer Key
UIL Regional 2008

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

9. E. Answers A and C are both correct.

28. D. The parameter for the method same is ArrayList<Object>. The argument must also be ArrayList<Object>. A subtype of the parameterized type is not allowed unless wildcard syntax is used on the parameter. The following type on the parameter would correct the problem: ArrayList<? extends Object>

31. A. The ArrayList remove(int) method is average case O(N), but when removing an item from the end of the list the method is O(1). The code always removes from the end of the list.

33. B. The data indicates $O(N^3)$. If an algorithm is $O(N^3)$ and the amount of data is doubled the time is expected to increase by a factor of 8. If the amount of data is doubled again time should increase by another factor of 8.Thus when the data is quadrupled time is expected to increase by a factor of 64. (8 times 8)

35. C. `SimpleStructure` inherits `toString` from `Object` so it does not have to re-implement it.