University Interscholastic League

Computer Science Competition

Number 133 (District 1 - 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 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 757_8 and 540_8?

A. 1317_8 B. 1317_{10} C. 1297_{10} D. 217_8 E. 1517_8

QUESTION 2

What is output by the code to the right?

A. 1.0 B. 1.75 C. 2.15

D. 2.55 E. 2.75

```
double a = 3.5;
double b = 4 / 5 + a / 2;
System.out.print(b);
```

QUESTION 3

What is output by the code to the right?

A. 20 B. 40 C. 42

D. 60 E. 63

```
int val = 0;
for(int i = 0; i < 20; i++){
  val++;
  val += 2;
}
System.out.print(val);
```

QUESTION 4

What is output by the code to the right?

A. ibe B. v205 C. v_l

D. Be E. iBe

```
String c1 = "Tyler_Brownsville";
String c2 = "" + c1.charAt(12);
c2 += c1.charAt(5) + "" + c1.charAt(15);
System.out.print(c2);
```

QUESTION 5

What is output by the code to the right?

A. false false 6 B. false true 5

C. true false 5 D. true true 6

E. The output will vary from one run of the program to the next.

```
boolean[] st = new boolean[5];
System.out.print(st[3] + " " + !st[0]);
System.out.print(" " + st.length);
```

QUESTION 6

What is output by the code to the right?

A. -5.0 5.99 B. -10.0 5.99

C. -10 5.0 D. -5.0 5.0

E. 10.0 5.0

```
double a1 = 5.99;
double b1 = -1.25;
double c1 = ((int) a1) * ((int) b1);
System.out.print(c1 + " " + a1);
```

QUESTION 7

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

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

A. `!p || q` B. `!p && q` C. `!p && !q`

D. `p || q` E. `p || !q`

QUESTION 8

What is output by the code to the right?

A. `1` B. `2` C. `13`

D. `23` E. `123`

```
int x2 = 10;
if(x2 * x2 > x2 + x2)
  if(x2 > 0)
    System.out.print(1);
  else
    System.out.print(2);
System.out.print(3);
```

QUESTION 9

Which of the following can replace **<*1>** so that the class `College` to the right compiles without syntax error?

A. `super(s)` B. `this(s)`

C. `mascot = s` D. `School(s)`

E. More than one of A through D is correct.

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

QUESTION 10

What is output by the client code to the right?

A. `Raiders` B. `!!`

C. `!!Raiders` D. `Raiders!!`

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

```
public class School {
  private String mascot;

  public School(String m) {mascot = m;}

  public void rah() {mascot += "!";}

  public String toString() {return mascot;}
}

public class College extends School {
  public College(String s) { <*1>; }
}

// client code
College tt = new College("Raiders");
tt.rah();
tt.rah();
System.out.print(tt);
```

QUESTION 11

What is output by the code to the right?

A. `282475249` B. `10000000` C. `15`

D. `17` E. `13`

```
int m = 7;
int n = 10;
System.out.print(m ^ n);
```

QUESTION 12

What is output by the code to the right?

A. `3125` B. `625` C. `625.0`

D. `256` E. `125`

```
int m1 = 4;
int n1 = (int) Math.pow(m1 + 1, m1 - 1);
System.out.print(n1);
```

QUESTION 13

What is output by the code to the right?

A. `1\23\4` B. `1  23  4`

C. `1t23t4` D. `1  2`

E. `1  2  3  4`

```
System.out.print("1\t2");
System.out.print("3\t4");
```

QUESTION 14

What is output by the code to the right?

A. `0045.70` B. `045.7` C. `45.70`

D. `+45.7` E. `045.70`

```
double tk = 45.7;
System.out.printf("%05.2f", tk);
```

QUESTION 15

What is returned by the method call `pick("universityTexas")` ?

A. `nvriyea` B. `unvi`

C. `niet` D. `unvis`

E. There is no output due to an infinite loop.

```
public String pick(String n) {
  String result = "";
  int lim = n.length();
  for(int i = 1; i <= lim; i *= 2)
    result += n.charAt(i);
  return result;
}
```

QUESTION 16

What is output by the code to the right?

A. `3` B. `10` C. `30`

D. `54` E. `75`

```
String stars = "";
for(int i = 0; i < 5; i++)
  for(int j = 0; j < 5; j++)
    stars += "***";
System.out.print(stars.length());
```

QUESTION 17

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

A. `null` B. `200` C. `10 20`

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

E. The output will vary from one run of the program to the next.

QUESTION 18

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

A. `false false`

B. `false true`

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

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

E. The output will vary from one run of the program to the next.

```
public class GasTank {
  private int cap;
  private int mpg;

  public GasTank(int c, int m) {
    cap = c;
    mpg = m;
  }

  public int range() {
    return cap * mpg;
  }
}

// client code
GasTank g1 = new GasTank(10, 20);
System.out.print(g1); // line 1

GasTank g2 = new GasTank(20, 15);
GasTank g3 = new GasTank(20, 15);
boolean b1 = g2 == g3;
boolean b2 = g3.equals(g2);
System.out.print(b1 + " " + b2); // line 2
```

QUESTION 19

Which of the following Java expressions is equivalent to the formula to the right? `a` and `t` are variables of type `double`.

A. `0.5 * a * t * t`

B. `a * t ^ 2 / 2`

C. `a * t * t * 2`

D. `(a * t * t) / 0.5`

E. `a * t ^^ 2 * 0.5`

$$\frac{1}{2}at^2$$

QUESTION 20

What is output by the code to the right?

A. `0` B. `3` C. `5`

D. `15` E. `null`

```
int[][] table = new int[5][3];
System.out.print(table[0].length);
```

QUESTION 21

What is output by the code to the right?

A. `1.0` B. `-1.0` C. `-2.0`

D. `2` E. `2.0`

```
double a3 = 17.02 * -0.1;
System.out.print(Math.floor(a3));
```

QUESTION 22

What is output by the code to the right?

A. `c84k c84k` B. `C*$K c84k`

C. `c*$K c*$K` D. `C84K c84k`

E. `c84k C84K`

```
String an1 = "C84K";
String an2 = an1.toLowerCase();
System.out.print(an1 + " " + an2);
```

QUESTION 23

What is output by the code to the right?

A. `[C, X, D]` B. `[X, X, D]`

C. `[C, X, X]` D. `[C, X, C]`

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

```
List<String> list;
list = new ArrayList<String>();
list.add("C");
list.add("D");
list.add(1, "X");
list.set(2, list.get(1));
System.out.print(list);
```

QUESTION 24

What is output by the client code to the right?

A. `48 6 24`

B. `48 12 24`

C. `48 6 12`

D. `24 6 24`

E. `48 6 48`

```
public int mystery(int x, int y) {
  x *= 2;
  y /= 3;
  return x * y;
}

// client code
int x = 6;
int y = 12;
System.out.print(mystery(x, y));
y = mystery(y, x);
System.out.print(" " + x + " " + y);
```

QUESTION 25

Which of the following can replace **`<*1>`** in the following line of code so that it compiles without error?

```
double <*1> = 0.0;
```

A. `37_` B. `Big` C. `final` D. `int` E. None of A through D are correct.

QUESTION 26

Given method `calc` to the right what is output by the following client code?

```
int[] data3 = {12, 14, -10, 4, 5, -10};
System.out.print(calc(data3));
```

A. `0` B. `2` C. `5`

D. `10` E. `26`

QUESTION 27

Given method `calc` to the right what is output by the following client code?

```
int[] data4 = {3, 1, 11, -3, 15, 1};
System.out.print(calc(data4));
```

A. `NaN` B. `0` C. `4`

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

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

```
public int calc(int[] list) {
  int total = 0;
  int count = 0;
  for(int i = 0; i < list.length; i++)
    if(list[i] % 2 == 0) {
      total += list[i];
      count++;
    }
  return total / count;
}
```

QUESTION 28

What is output by the code to the right?

A. `[A, B, aa, a]` B. `[aa, a, A, B]`

C. `[a, A, aa, B]` D. `[A, B, a, aa]`

E. `[A, a, aa, B]`

```
String[] sts = {"A", "a", "aa", "B"};
Arrays.sort(sts);
System.out.print(Arrays.toString(sts));
```

QUESTION 29

What is output by the code to the right?

A. `03025` B. `01234` C. `12345`

D. `10` E. `250`

```
int[] data = {0, 3, 0, 2, 5};
String all = "";
for(int xv : data)
  all += xv;
System.out.print(all);
```

QUESTION 30

What is output by the code to the right?

A. `{-4=aa, 0=A, 3=A,B, 5=null}`

B. `{-4=aa, 3=B, 5=null}`

C. `{-4=aa, 0=A, 3=B, 5=null}`

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

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

```
TreeMap<Integer, String> samp;
samp = new TreeMap<Integer, String>();
samp.put(3, "A");
samp.put(0, "A");
samp.put(3, "B");
samp.put(-4, "aa");
samp.put(5, samp.get(-1));
System.out.print(samp);
```

QUESTION 31

A method is $O(N^3)$. When N = 200,000 the method takes 3 seconds to complete. What is the expected runtime for the method when N = 800,000?

A. 12 seconds B. 48 seconds C. 192 seconds D. 384 seconds E. 576 seconds

QUESTION 32

Given method `sort` to the right what is output when the following client code is executed.

```
int[] us = {12, 17, 5, -5, 15, 0, -3};
sort(us);
```

A. `[-5, 5, 12, 15, 17, 0, -3]`

B. `[-5, 0, -3, 5, 12, 15, 17]`

C. `[-5, -3, 0, 5, 12, 15, 17]`

D. `[12, 17, 5, -5, 15, 0, -3]`

E. None of A through D are correct.

```
public void sort(int[] lt) {
  int temp, j;
  for(int i = 1; i < lt.length; i++) {
    temp = lt[i];
    j = i;
    while( j > 0 && temp < lt[j - 1]) {
      lt[j] = lt[j - 1];
      lt[j - 1] = temp;
      j--;
    }
    if(i == 4)
     System.out.print(Arrays.toString(lt));
  }
}
```

QUESTION 33

Which sorting algorithm does method `sort` implement?

A. radix sort B. selection sort

C. insertion sort D. heap sort

E. None of A through D are correct.

QUESTION 34

Given method `gen` to the right what is output by the following client code?

```
ArrayList w = new ArrayList();
gen(w, 0, "eerily");
System.out.print(w.size());
```

A. `0` B. `14`

C. `64` D. `128`

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

```
public void gen(ArrayList pats,
                      int pos, String cur) {
  if(pos == cur.length())
    pats.add(cur);
  else {
    gen(pats, pos + 1, cur);
    String temp = cur.substring(0, pos) +
              'e' + cur.substring(pos + 1);
    gen(pats, pos + 1, temp);
  }
}
```

QUESTION 35

What is output by the code to the right?

A. `35.216` B. `16.804` C. `4.402`

D. `2.201` E. There is no output due to a syntax error.

```
double a5 = 8.804;
double b5 = a5 >> 2;
System.out.print(a5 + " " + b5);
```

QUESTION 36

Method `min` to the right contains a logic error. Which of the following changes is required so that method `min` always meets its post condition?

A. Change `int min = 0` to `int min = lt[0]`.

B. Change every occurrence of the identifier `min` to `m`.

C. Change `int min = 0` to `int min = Integer.MAX_VALUE`.

D. Change `int i` to `int x`.

E. More than one of A through D is correct.

```
// pre: lt != null, lt.length > 0
// post: return the minimum value in lt
public int min(int lt[]) {
  int min = 0;
  for(int i : lt)
    min = min > i ? i : min;
  return min;
}
```

QUESTION 37

What is returned by method `h(t, 1, 0, -10)` if `t` is the matrix shown below?

1	4	0	22	25	6
0	-1	5	20	47	50
2	3	7	17	10	52
11	5	13	15	4	20

A. 5 B. 8 C. 12

D. 14 E. 2000

```
public int h(int[][] t, int i, int j,
                                    int d) {
  if(i == -1 || i == t.length)
    return 1000;
  if(j == t[0].length)
    return 0;
  if(t[i][j] <= d)
    return 2000;
  int m1 = 1 + h(t, i - 1, j, t[i][j]);
  int m2 = 2 + h(t, i, j + 1, t[i][j]);
  int m3 = 1 + h(t, i + 1, j, t[i][j]);
  return Math.min(Math.min(m1, m2), m3);
}
```

QUESTION 38

Given class `N` below, what is output by the client code to the right?

```
public class N {
  public Object d;
  public N n;
  public N(Object d1, N n1) {
    d = d1; n = n1; }
}
```

A. 401 B. 302 C. 321

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

E. The output will vary from one run of the program to the next.

```
// client code
N n1 = new N(0, null);
n1.n = new N(1, new N(2, new N(3, n1)));
N n2 = new N(n1, new N(4, n1.n));
n2.n.n = n1.n.n.n;
N t = n2;
for(int i = 1; i < 11; i++) {
  if(i % 3 == 0)
    System.out.print(t.d);
  t = t.n;
}
```

QUESTION 39

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

```
Structure st = new Structure();
st.add(24);
st.add(36);
st.add(24);
st.add(8);
while(!st.isEmpty())
  System.out.print(st.remove());
```

A. 36824 B. 82436

C. 8243624 D. 2436248

E. The code runs without error, but there is no output.

```
public class Structure<E> {
  private ArrayList<E> con;

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

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

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

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

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

QUESTION 40

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

A. a graph B. a heap C. a stack

D. a queue E. a list

Standard Classes and Interfaces — Supplemental Reference

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()`
- `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.ArrayList<E> implements List<E>

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 District 1 2012

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

Notes:

17. The GasTank inherits the toString method from the Object class, but the behavior of Object's toString is unpredictable.

27. An `ArithmeticException` occurs due to a divide by 0.

35. The `>>` operator may not be applied to `doubles`.

36. Choice A and C are both correct.