University Interscholastic League

Computer Science Competition

Number 127 (District 1 - 2011)

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 756_8 and 36_8?

A. 1112_8 B. 702_8 C. 1114_8 D. 1014_8 E. 1102_{16}

QUESTION 2

What is output by the code to the right?

A. 0 B. 2 C. 4

D. 5 E. 248

```
int x = 744;
int y = x / 10 / 7 / 2;
System.out.print(y);
```

QUESTION 3

What is output by the code to the right?

A. 0 B. 5 C. 8

D. 32 E. 1024

```
int total = 1;
for(int i = 0; i < 10; i++){
  total *= 2;
  i++;
}
System.out.print(total);
```

QUESTION 4

What is output by the code to the right?

A. acke_acke B. acker_Thacker

C. ack_Thacker D. Tha_Thacker

E. Thacker_Thacker

```
String nm = "Thacker";
String pt = nm.substring(2, 5);
System.out.print(pt + "_" + nm);
```

QUESTION 5

What is output by the code to the right?

A. 8 3 B. 8 9

C. 9 3 D. 9 6

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

```
int x2 = 3;
double[] points = new double[x2 * 2 + x2];
System.out.print(points.length + " " + x2);
```

QUESTION 6

What is output by the code to the right?

A. 10 B. 50 C. 101

D. 201 E. 601

```
int x3 = 50;
int y3 = 10;
int z3 = x3 * y3 / 5 + 101;
System.out.print(z3);
```

QUESTION 7

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

A. 7 B. 5 C. 4

D. 1 E. 0

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

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

QUESTION 8

What is output by the code to the right?

A. 4

B. 1

C. 12

D. 13

E. 14

```
int x4 = 10;
int y4 = 5;
if(x4 > y4 * 2){
  System.out.print(1);
  if(x4 % y4 != 0)
    System.out.print(2);
  else
    System.out.print(3);
}
else
  System.out.print(4);
```

QUESTION 9

Which of the following can replace **<*1>** in the code to the right so that the `move` method in the `Snail` class returns the value stored in the constant named `SOUTH` from the `Critter` class?

I. `Critter.SOUTH`
II. `SOUTH`
III. `Snail.SOUTH`

A. I only B. III only C. I and II only

D. II and III only E. I, II, and III

```
public class Critter{
  public static final int NORTH = 0;
  public static final int EAST = 1;
  public static final int SOUTH = 2;
  public static final int WEST = 3;

  private int dir;

  public Critter(){  dir = EAST;  }

  public int move(){
    dir = dir + 3;
    if(dir > WEST)
      dir -= 4;
    return dir;
  }
}

public class Snail extends Critter{
  public int move(){ return <*1>; }
}

// client code
Critter c1 = new Critter();
c1.move();
c1.move();
System.out.println(c1.move());
```

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

QUESTION 10

What is output by the client code to the right?

A. 0

B. 1

C. 2

D. 3

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

QUESTION 11

What is output by the code to the right?

A. `true` B. `0` C. `64`

D. `130` E. `194`

```
int m = 130;
int n = 64;
System.out.print(n | m);
```

QUESTION 12

What is output by the code to the right?

A. `10` B. `25.0` C. `25`

D. `32` E. `32.0`

```
System.out.print(Math.pow(2, 5));
```

QUESTION 13

What is output by the code to the right?

A. `10\5\2` B. `1` C. `1052`

D. `0` E. `10\\5\\2`

```
System.out.print("10\\5\\2");
```

QUESTION 14

What is output by the code to the right?

A. `8.314` B. `8.31` C. `4.2`

D. `+8.31` E. `+8.3`

```
System.out.printf("%+4.2f", 8.314);
```

QUESTION 15

What is returned by the method call `eval(-5, 2)`?

A. `12` B. `13` C. `15`

D. `20` E. `-9`

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

QUESTION 16

What is output by the code to the right?

A. `5` B. `16` C. `20`

D. `24` E. `36`

```
String res = "";
for(int i = 0; i < 5; i++)
  res += "*";
for(int i = 0; i < 2; i++)
  res = res + res;
System.out.print(res.length());
```

QUESTION 17

What is output by the code to the right?

A. `false false` B. `false true`

C. `true false` D. `true true`

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

```
String str = "12";
Integer num = new Integer(12);
Double val = new Double(12.0);
System.out.print(str.equals(num));
System.out.print(" " + num.equals(val));
```

QUESTION 18

What is output by client code to the right?

A. `-1` B. `0` C. `1`

D. `x` E. `Demo@61de33`

QUESTION 19

If the method named `q19` is uncommented, the `Demo` class will no longer compile. Which of the following best explains why uncommenting method `q19` results in a syntax error?

A. A static method may not call a non-static method as shown.

B. `q19` is not a valid method name.

C. Static methods cannot return values.

D. The method call must be `this.toString()`.

E. The method call must be `Demo.toString()`.

```
public class Demo {
  private int x;

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

  /*
  public static String q19() {
    return toString();
  }
  */

}

// client code
Demo d = new Demo();
System.out.print(d.toString());
```

QUESTION 20

Which of the following can replace **<*1>** in the code to the right so that the class compiles without error?

A. `ValuePair` B. `2Values`

C. `<E>Values` D. `Values <2String>`

E. None of A, B, C, or D are correct.

```
public class <*1> {
  public int value;
  public String name;
}
```

QUESTION 21

What is output by the code to the right?

A. `10` B. `19`

C. `21` D. `100`

E. `1000`

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

QUESTION 22

Which of the following best explains the syntax error in the code segment to the right?

A. The only valid parameter for the `Scanner` constructor is `System.in`.

B. `DATA` is not a valid file name. It must have an extension such as `.txt`.

C. `file_name` is not a valid identifier.

D. The `Scanner` constructor throws a checked exception that must be handled.

E. `sc` is not a valid identifier.

```
String file_name = "DATA";
File f = new File(file_name);
Scanner sc = new Scanner(f);
```

QUESTION 23

What is output by the code to the right?

A. `[A, BA, C]` B. `[BA, C]`

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

E. `[A, B, C]`

```
ArrayList<String> non;
non = new ArrayList<String>();
non.add("A");
non.add("B");
non.add(1, "BA");
non.set(2, "C");
System.out.println(non);
```

QUESTION 24

What is output by method `some` to the right if `sc` is connected to a file that contains the following data and uses the default delimiters?

```
821  56.67  37  15
71   36
aaa  bbb
```

A. `821aaa` B. `56.6771` C. `82171`

D. `56.6736` E. `82156.67371571`

```
public void some(Scanner sc) {
  System.out.print(sc.next());
  sc.nextDouble();
  sc.nextInt();
  sc.nextLine();
  System.out.print(sc.next());
}
```

QUESTION 25

Which of the following can replace **<*1>** in the code to the right so that the output is 8?

I. `x5 %= 97`
II. `x5 = x5 >> 4`
III. `x5 /= 9`

A. I only B. II only

C. III only D. I and II only

E. II and III only

```
int x5 = 80;
<*1>;
System.out.print(x5);
```

QUESTION 26

What is output by the code to the right?

A. `[0.201, 0.2, 0.2, 0.01, 0.01]`

B. `[.1, .1, .2, .201, .2]`

C. `[0.01, 0.2, 0.201]`

D. `[0.01, 0.01, 0.2, 0.2, 0.201]`

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

```
double[] ds = {.01, 0.01, .2, .201, .20};
Arrays.sort(ds);
System.out.print(Arrays.toString(ds));
```

QUESTION 27

What is output by the client code to the right?

A. `[0]` B. `[8]` C. `[0, 3]`

D. `[9]` E. `[]`

QUESTION 28

Which search algorithm does method `srch` use?

A. insertion B. selection C. merge

D. sequential E. binary

```
public ArrayList<Integer> srch(char[] cs,
                                char tgt) {
  ArrayList<Integer> res;
  res = new ArrayList<Integer>();
  for(int i = 0; i < cs.length; i++)
    if(cs[i] == tgt)
      res.add(i);
  return res;
}

// client code
String u = "engelbart";
char t = 'e';
System.out.print(srch(u.toCharArray(), t));
```

QUESTION 29

What is output by the code to the right?

A. `i` B. `b7` C. `B`

D. `72` E. `h`

```
char let = 'b';
System.out.print((char) (let + 7));
```

QUESTION 30

What is output by the code to the right?

A. `f.*n f.n` B. `false false`

C. `false true` D. `true false`

E. `true true`

```
String winner = "frances e. allen";
boolean b1 = winner.matches("f.*n");
boolean b2 = winner.matches("f.n");
System.out.print(b1 + " " + b2);
```

QUESTION 31

What is output by the code to the right?

A. `5`

B. `3`

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

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

E. There is no output due to an infinite loop that occurs when the code is run.

```
ArrayList<Object> aList;
aList = new ArrayList<Object>();
aList.add("cat");
aList.add(aList);
aList.add(12);

int size = 0;
Iterator<Object> it = aList.iterator();
while(it.hasNext()) {
  it.next();
  size++;
}
System.out.print(size);
```

QUESTION 32

What is output by the code to the right?

A. `[35][35]`

B. `true[5, 35]`

C. `[5, 35][5, 35]`

D. `[5, 5][35]`

E. `true[35]`

```
TreeSet<Integer> ts1;
ts1 = new TreeSet<Integer>();
TreeSet<Integer> ts2;
ts2 = new TreeSet<Integer>();
ts1.add(5);
ts1.add(35);
ts1.add(5);
ts1.add(5);
ts2.add(5);
ts2.add(5);
System.out.print(ts1.removeAll(ts2));
System.out.print(ts1);
```

QUESTION 33

If the `LinkedList dt` contains N elements, what is the worst case Big O of method `total`? Pick the most restrictive correct answer.

A. O(N) B. O(NlogN) C. $O(N^{3/2})$

D. $O(N^2)$ E. O(N!)

```
public int total(LinkedList<String> dt) {
  int result = 0;
  Iterator<String> it = dt.iterator();
  while(it.hasNext()) {
    String temp = it.next();
    if(temp != null )
      result += temp.length();
  }
  return result;
}
```

QUESTION 34

What is returned by the method call `sig(new int[] {15, 15}, 0, 15)`?

A. `225` B. `30` C. `15`

D. `4` E. `2`

QUESTION 35

What is returned by the method call `sig(new int[] {5,10,5,5,5,10}, 0, 15)`?

A. `4` B. `8` C. `10`

D. `12` E. `32`

```
public int sig(int[] vs, int p, int t) {
  if(p == vs.length && t == 0)
    return 1;
  else if(p == vs.length)
    return 0;
  else
    return sig(vs, p + 1, t) +
           sig(vs, p + 1, t - vs[p]);
}
```

QUESTION 36

What is output by the code to the right?

A. `true true`

B. `true false`

C. `false true`

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

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

```
List<Integer> m1 = new
ArrayList<Integer>();
boolean p1 = m1 instanceof List<Integer>;
boolean p2;
p2 = m1 instanceof ArrayList<Integer>;
System.out.print(p1 + " " + p2);
```

QUESTION 37

Method `sort` uses the traditional merge sort algorithm to sort an array of `int`s. It takes method `sort` 2 seconds to sort an array with 1,000,000 elements all equal to a single value. What is the expected time for method `sort` to sort an array with 2,000,000 elements all equal to a single value?

A. 4.4 seconds B. 8 seconds C. 8.8 seconds D. 16 seconds E. 64 seconds

QUESTION 38

What is output by the client code to the right?

A. `AAB` B. `BAC` C. `BAA`

D. `ABC` E. `ABA`

QUESTION 39

If a `Structure` contains N elements, what is the Big O of the `remove` method in the `Structure` class? Pick the most restrictive correct answer.

A. O(1) B. O(logN) C. O(N)

D. O(NlogN) E. $O(N^2)$

QUESTION 40

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

A. a queue B. a stack C. a min heap

D. a hash table E. a binary search tree

```
public class Structure<E> {

  private ArrayList<E> con;

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

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

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

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

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

// client code
Structure<Character> stEx;
stEx = new Structure<Character>();
String someData = "ABACBAAB";
int start = someData.length() - 1;
for(int i = start; i >= 0; i--)
  stEx.add(someData.charAt(i));
String out = "";
out += stEx.remove();
out += stEx.remove();
out += stEx.remove();
System.out.print(out);
```

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 2011

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

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.

33. The traditional implementation of mergesort is O(NlogN). For an O(NlogN) algorithm the time will a little more than double when the amount of data double. The data set in the question, an array with all values equal, is $O(N^2)$ for the traditional quicksort algorithm

37. The `instanceof` operator may not be used to perform checks against parameterized data types.

39. Removing the first element of an `ArrayList` is O(N) because all elements are shifted one spot towards the beginning of the array.