University Interscholastic League

Computer Science Competition

Number 131 (Invitational A - 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 does ABC_{16} minus $1FF_{16}$ equal?

A. 853_{16} B. 953_{16} C. $8BD_{16}$ D. $8CD_{16}$ E. $9CD_{16}$

QUESTION 2

What is output by the code to the right?

A. 4 B. 6.67 C. 15

D. 15.3333 E. 24

```
int x = 3;
int y = 10 / x + x * 4;
System.out.print(y);
```

QUESTION 3

What is output by the code to the right?

A. 5 B. 12 C. 13

D. 25 E. 50

```
int val = 0;
int limit = 25;
for(int i = 1; i < limit; i += 2)
  val++;
System.out.print(val);
```

QUESTION 4

What is output by the code to the right?

A. BoBo2 B. BoBoBo C. BoBob

D. BoBoB E. 'BoBo2'

```
String name = "Bo";
name = name + name + 2;
System.out.print(name);
```

QUESTION 5

What is output by the code to the right?

A. 0 B. 0.0

C. 1.0 D. 4

E. The output will vary from one execution of the code to the next.

```
double[] list = new double[6];
System.out.print(list[4]);
```

QUESTION 6

What is output by the code to the right?

A. 1 B. 2 C. 8

D. 12 E. 16

```
int x1 = 2;
int y1 = x1 * x1 * x1 * x1;
System.out.print(y1);
```

QUESTION 7

What is output by the code to the right?

A. false false B. true false

C. false true D. true true

E. 0 1

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

QUESTION 8

What is output by the code to the right?

A. 24 B. 23 C. 13

D. 14 E. 1

```
String n2 = "126547";
if(n2.indexOf('a') != -1)
   System.out.print(1);
else
   System.out.print(2);
if(n2.length() > 6)
  System.out.print(3);
else
  System.out.print(4);
```

QUESTION 9

What replaces **<*1>** in the code to the right so that the output of the client code to the right is `go Longhorns`?

A. `mascot`

B. `School.mascot`

C. `String mascot`

D. `toString()`

E. `this.mascot`

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

QUESTION 10

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

```
School sc2 = new School();
System.out.print(sc2);
```

A. `go null`

B. `go`

C. `"go mascot"`

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

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

```
public class School {
  private String mascot;

  public School(String mascot) {
    <*1> = mascot;
  }

  public String toString() {
    return "go " + mascot;
  }
}

// client code
School sc = new School("Longhorns");
System.out.print(sc.toString());
```

QUESTION 11

What is output by the code to the right?

A. true B. false C. 6

D. 111 E. 117

```
int m = 47;
int n = 70;
System.out.print(m | n);
```

QUESTION 12

What is the largest value that can be output by the code to the right?

A. 50 B. 55 C. 100

D. 110 E. 150

```
int tot = 0;
for(int i = 0; i < 10; i++) {
  int temp = (int) (Math.random() * 11);
  tot += temp - 5;
}
System.out.print(tot);
```

QUESTION 13

What is output by the code to the right?

A.
```
cat
dog
ape
```

B.
```
cat dog
    ape
```

C.
```
cat
dog ape
```

D.
```
apecatdog
```

E.
```
catdogape
```

```
System.out.print("cat");
System.out.print("dog");
System.out.println("ape");
```

QUESTION 14

What is output by the code to the right?

A. `a4` B. `14.26730` C. `14.2673`

D. `14.2672` E. `14.267299`

```
double a4 = 14.267299;
System.out.printf("%7.4f", a4);
```

QUESTION 15

What is returned by the method call `eval(5, 3)` ?

A. `12` B. `32` C. `36`

D. `40` E. `4096`

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

QUESTION 16

What is output by the code to the right?

A. `20` B. `22`

C. `45` D. `200`

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

```
String stars = "";
for(int i = 0; i < 10; i++)
  stars += "*";
for(int i = 0; i < 10; i++)
  stars += "*";
System.out.println(stars.length());
```

QUESTION 17

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

A. `(-b + Math.sqrt(b ^^ 2 - 4 * a * c)) / (2 * a)`

B. `(-b +- Math.sqrt(b ** 2 - 4 * a * c)) / (2 * a)`

C. `(-b + (b * b - 4 * a * c) ^ 0.5) / (2 * a)`

D. `-(b + Math.sqrt(b * b - 4 * a * c)) / 2a`

E. `(-b + Math.sqrt(b * b - 4 * a * c)) / (2 * a)`

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

QUESTION 18

What is output by the code to the right?

A. `10` B. `12` C. `14`

D. `15` E. `17`

```
String garbage = "1000\\100\"\t+5";
System.out.print(garbage.length());
```

QUESTION 19

What is output by the code to the right?

A. `5` B. `7` C. `35`

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

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

```
final int rate;
int sts = 7;
int profit = 5;
rate = sts * profit;
System.out.print(rate);
```

QUESTION 20

Which answer is logically equivalent to the following `boolean` expression, where `x`, `y`, and `z` are `int` variables?

```
(x != y) || !(y >= z)
```

A. `(x != y) && !(y >= z)` B. `(x == y) && (y == z)` C. `!((x == y) || (y >= z))`

D. `(x != z) && (y == z)` E. `!((x == y) && (y >= z))`

QUESTION 21

Method `Total` to the right will not compile due to a syntax error. Which of the following best explains the syntax error in method `Total`?

A. `Total` is not a legal method name.

B. The keyword `static` must be removed from the method header.

C. The variable `res` is not initialized.

D. A `char` may not be added to a variable of type `int`.

E. The `for` loop must have a set of braces, `{}`.

```
public static int Total(String st) {
  int res;
  for(int i = 0; i < st.length(); i++)
    res += st.charAt(i);
  return res;
}
```

QUESTION 22

What is output by the code to the right?

A. `0` B. `null`

C. The code runs, but there is no output.

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

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

```
String[] names = new String[4];
System.out.print(names[2].length());
```

QUESTION 23

What is output by the code to the right?

A. `0` B. `1.0` C. `1`

D. `1.946465` E. `2`

```
double a5 = 3.89293;
a5 /= 2;
System.out.print( (int) a5 );
```

QUESTION 24

Which of the following could replace **`<*1>`** so that the following line of code compiles without syntax error?

```
int <*1> = 15;
```

A. `x5` B. `5_x` C. `_5` D. `x-5` E. More than one of A through D is correct.

QUESTION 25

What is output by the code to the right?

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

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

E. `['C', 'A', 'B']`

```
ArrayList<Character> grades;
grades = new ArrayList<Character>();
grades.add('A');
grades.add('A');
grades.add(1, 'B');
grades.add(0, 'C');
grades.remove(1);
System.out.println(grades);
```

QUESTION 26

Method `get` to the right contains a logic error. Which of the following will occur when the method call `get("aaaa")` is made?

A. The program will crash due to a `StackOverError`.

B. Nothing. An infinite loops occurs.

C. The program will crash due to a `NullPointerException`.

D. The program will crash due to an `OutOfMemoryError`.

E. The program will crash due to an `IndexOutOfBoundsException`.

```
public String get(String st) {
  String r = "";
  for(int i = 0; i < st.length(); i *= 2)
    r = r + st.charAt(i) + r;
  return r;
}
```

QUESTION 27

What is output by the code to the right?

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

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

E. `false true true`

```
int v1 = 15;
int v2 = 30;
System.out.print((v1 >= v2) + " ");
System.out.print(v1 > 0 && v2 % v1 == 0);
```

QUESTION 28

What replaces **<*1>** in method `check` to the right so that `diff` is incremented if the element at index `i` in `a` does not equal the element at index `i` in `b`?

A. `a[i] != b[i]`

B. `!(a[i].equals(b[i]))`

C. `a.get(i) != b.get(i)`

D. `a[i].compareTo(b[i]) != 0`

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

```
public int check(int[] a, int[] b) {
  int diff = 0;
  for(int i = 0; i < a.length; i++)
    if( <*1> )
      diff++;
  return diff;
}
```

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

QUESTION 29

What is output by the client code to the right?

A. `0` B. `2` C. `3`

D. `12` E. `14`

```
// client code
int[] h1 = {5, -2, 4, 10, 45};
int[] h2 = {5, 2, -4, 10, 45};
System.out.print(check(h1, h2));
```

QUESTION 30

What is output by the code to the right?

A. `10000000` B. `640`

C. `0.15625` D. `0`

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

```
int bw = 10;
bw = bw >> 6;
System.out.print(bw);
```

QUESTION 31

An array with 1,000,000 distinct `int`s in random order is passed to a method that uses the heapsort algorithm, it takes 4 seconds for the method to complete. What is the expected time for the method to complete when sorting an array with 4,000,000 distinct `int`s in random order?

A. 1 second B. 4 seconds C. 8 seconds D. 17.6 seconds E. 64 seconds

QUESTION 32

Which of the following replaces **<*1>** in the code to the right so that the body of the `if` statement is executed if the element at position `j - 1` in `vs` is greater than the element at position `j`?

A. `vs.get(j-1).compareTo(vs.get(j)) > 0`

B. `vs[j-1].compareTo(vs[j]) > 0`

C. `vs.get(j-1).compareTo(vs.get(j))`

D. `vs.get(j-1) <= => vs.get(j)`

E. None of answers A through D are correct.

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

QUESTION 33

Which of the following replaces **<*2>** in the code to swap the elements at positions `j - 1` and `j` in `vs`?

A. `vs.set(j, vs.get(j - 1))`

B. `vs.set(j - 1, vs.remove(j))`

C. `vs.set(j, vs.set(j-1, vs.get(j)))`

D. `vs.set(j - 1, vs.get(j + 1))`

E. None of answers A through D are correct.

Assume **<*2>** is filled in correctly.

QUESTION 34

Which sorting algorithm does method `sort` implement?

A. radix sort B. insertion sort

C. selection sort D. quick sort

E. None of answers A through D are correct.

```
public void sort(ArrayList<Integer> vs) {
  for(int i = 0; i < vs.size(); i++)
    for(int j = 1; j < vs.size(); j++)
      if( <*1> )
        <*2>;
}
```

QUESTION 35

What is returned by the method call `add("aaaa")`?

A. 32 B. 80 C. 164

D. 200 E. 228

```
public int add(String s) {
  if(s.length() > 20)
    return s.length();
  else
    return add(s + s) + add(s + s + s);
}
```

QUESTION 36

What is returned by method `handle` if `t` is the matrix shown below?

1	4	0	2	1	6
0	-1	5	4	0	-4
2	2	7	1	13	2
10	5	13	13	4	20
1	4	2	1	3	2
0	-6	-5	3	-4	5

A. -88 B. -42 C. -11

D. -10 E. 8

```
public int handle(int[][] t) {
  int res = 0;
  for(int i = 0; i < t.length; i++) {
    int t1, t2;
    t1 = t2 = 0;
    for(int j = i; j < t.length; j++) {
      t1 += t[i][j];
      t2 += t[j][i];
    }
    if(t1 == t2)
      res += t1;
    else
      res -= t2;
  }
  return res;
}
```

QUESTION 37

What is output by the code to the right?

A. 13 17 B. -10 -10 50 50 -10 -20

C. 17 13 D. 50 50

E. -20 -10 50 50 -10 -10

```
Stack<Integer> st = new Stack<Integer>();
int[] data = {13, 17, -20, 50, -10};
for(int i : data)
  if(i % 5 == 0) {
    st.push(i);
    st.push(i > 0 ? i : -10);
  }
while(!st.isEmpty())
  System.out.print(st.pop() + " ");
```

GO ON TO THE NEXT PAGE.

QUESTION 38

What is output by the following client code?

```
String sch = "texasutamtechstate";
Structure<Character> st1;
st1 = new Structure<Character>();
for(int i = 0; i < sch.length(); i++)
  st1.add(sch.charAt(i));
String temp1 = st1.toString();
String[] res;
res = temp1.split("[\\s,\\[\\]]+");
for(String s3 : res)
    System.out.print(s3);
```

A. `texasutamtechstate`

B. `texasumch`

C. `hcmusaxet`

D. The program runs without error, but there is no output.

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

QUESTION 39

Given a `Structure` that contains N `Integer`s what is the order (Big O) of the add method for a value that is not already present in the `Structure`? Pick the most restrictive correct answer.

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

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

QUESTION 40

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

A. a graph B. a set

C. a stack D. an array based list

E. a linked list

```
public class Structure<E> {

  private ArrayList<E> con;

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

  public void add(E obj) {
    if(!con.contains(obj))
      con.add(0, obj);
  }

  public boolean present(E obj) {
    return con.contains(obj);
  }

  public String toString() {
    return con.toString();
  }

  public boolean remove(E obj) {
    return con.remove(obj);
  }
}
```

No Test Material on This Page

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 Invitational A 2012

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

10. With the addition of a non-default constructor, the built in default constructor is no longer available.

24. Choices A and C are both correct.

34. The sorting algorithm used is the bubble sort.