University Interscholastic League

Computer Science Competition

Number 123 (Regional - 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 does $\texttt{FB}_{16}$ minus $\texttt{11000011}_2$ equal?

A. $\texttt{1A}_{16}$ B. $\texttt{11001}_2$ C. $\texttt{37}_{10}$ D. $\texttt{111001}_2$ E. $\texttt{56}_{10}$

QUESTION 2

What is output by the code to the right?

A. `7.0` B. `6.25` C. `5.0`

D. `1.25` E. `7`

```
double a = 2.5;
double b = a * 5 / 2;
System.out.print(b);
```

QUESTION 3

What is output by the code to the right?

A. `10` B. `12` C. `20`

D. `22` E. `30`

```
int result = 0;
for(int i = -5; i <= 5; i++){
  result += 2;
}
System.out.print(result);
```

QUESTION 4

What is output by the code to the right?

A. `s s` B. `s Wilkes` C. `kes`

D. `s` E. `kes Wilkes`

```
String per = "Wilkes";
String st = per.substring(3).substring(2);
System.out.print( st + " " + per );
```

QUESTION 5

What is output by the code to the right?

A. `39 39` B. `42 41` C. `47 46`

D. `41 41` E. `40 39`

```
int[] his = {47, 42, 37, 40, 42};
his[0] = his[4];
his[4]--;
System.out.print(his[0] + " " + his[4]);
```

QUESTION 6

What is output by the code to the right?

A. `7 3` B. `3 7` C. `0 0`

D. `7 0` E. `3 3`

```
int w = 7;
int z = 3 % w;
w = z;
z = w;
System.out.print( w + " " + z );
```

QUESTION 7

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

```
!((x >= y) && (w < z))
```

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

D `!(x == y) || !(w == z)` E. `(x < y) || (w >= z)`

QUESTION 8

What is output by the code to the right?

A. `12` B. `23` C. `13`

D. `123` E. `3`

```
boolean p = true;
boolean q = false;
boolean r = p ^ q;
if( !p && r )
  System.out.print(1);
else
  System.out.print(2);
if( p && !q && r )
  System.out.print(3);
```

QUESTION 9

What replaces **`<*1>`** in the code to the right to indicate that `MAX_SCORE` is a class constant that is accessible in all other classes?

A. `public static final`

B. `static final`

C. `public final`

D. `public static`

E. `public class final`

```
public class Score{
  <*1> int MAX_SCORE = 100;
  private int score;

  public Score(int sc){
    score = sc;
  }

  public int getScore(){
    return score;
  }

  public String toString(){
    return getScore() + " points";
  }
}

public class CurvedScore extends Score{
  private int added;

  public CurvedScore(int sc, int ad){
    super(sc);
    added = ad;
  }

  public int getScore(){
    return super.getScore() + added;
  }
}
```

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

QUESTION 10

What is output by the following client code?

```
CurvedScore cs = new CurvedScore(75, 5);
System.out.println(cs);
```

A. `75 points`

B. `80 points`

C. `0 points`

D. `100 points`

E. `5 points`

QUESTION 11

What is output by the code to the right?

A. `30` B. `31` C. `61`

D. `51` E. `11111`

```
int dx = 30 | 21 & 10;
System.out.print(dx);
```

QUESTION 12

What is the maximum possible number of `'*'`s the code to the right will print when run?

A. `1` B. `4` C. `5`

D. `6` E. `2147483647`

```
double limit = Math.random() * 5;
for(int i = 0; i <= limit; i++)
  System.out.print('*');
```

QUESTION 13

What is output by the code to the right?

A. `mas"Miners"`

B. `Owls\"Blaze`
`Miners`

C. `Owls"BlazeMiners`

D. `OwlsBlaze`
`Miners`

E. `"Owls"BlazeMiners"`

```
String mas = "Owls\"Blaze";
System.out.print(mas);
System.out.print("Miners");
```

QUESTION 14

What is output by the code to the right? ƀ indicates a blank space.

A. `+5.00` B. ƀƀ`5.00` C. `(5)`

D. `(5.00)` E. ƀƀƀƀ`5`

```
double value = -2.5 * 2;
System.out.printf("%(5.2f", value);
```

QUESTION 15

What is returned by the method call `tough(3)`?

A. `77` B. `43` C. `32`

D. `8` E. `2`

```
public int tough(int x){
  if(x < 0)
    return 2;
  else
    return x + tough(x - 1) + tough(x - 1);
}
```

QUESTION 16

What is output by the code to the right?

A. `21` B. `20`

C. `10` D. `5`

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

```
int result = 0;
int i = 20;
while( i > 0 ){
  result++;
  i /= 2;
}
System.out.print(result);
```

QUESTION 17

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

A. `1002` B. `250` C. `125`

D. `20` E. `10`

```
int limit = 5;
for(int i = 0; i < limit; i++)
  for(int j = 0; j < limit; j++)
    for(int k = 0; k < limit * 2; k++)
      System.out.print('*');
```

QUESTION 18

What is output by the code to the right?

A.
```
CPU
    RAMNEC
```

B.
```
CPU     RAM
NEC
```

C.
```
CPU
     RAM
NEC
```

D.
```
CPU
RAM     NEC
```

E.
```
CPURAMNEC
```

```
System.out.print("CPU");
System.out.println("\tRAM");
System.out.print("NEC");
```

QUESTION 19

What is output by the code to the right?

A. `false` B. `true` C. `null`

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

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

```
int[] list1 = {2, 4, 6};
int[] list2 = {2, 4, 5};
list2[2] = list1[2];
System.out.print(list1 == list2);
```

QUESTION 20

What is output by the code to the right?

A. `3628800` B. `55` C. `45`

D. `11` E. `0`

```
int limit = 10;
int total = 0;
for(int i = 1; i <= limit; i++)
  total += i;
System.out.print(total);
```

QUESTION 21

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

A. `234` B. `243` C. `2415`

D. `2154` E. `1524`

```
public int one(int x){
  System.out.print(x);
  x *= 2;
  return x;
}

public void two(){
  System.out.print( one(2) + 3 + one(4) );
}
```

QUESTION 22

What is returned by the method call `toy(3)`?

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

D. `4` E. `6`

```
public int toy(int y){
  ++y;
  y++;
  return y++;
}
```

QUESTION 23

What is output by the code to the right?

A. `12` B. `4` C. `7`

D. `11` E. `10`

```
String junk;
junk = "DELL_640_IBM_360_HP_2020_DEC";
String[] parts = junk.split("\\d+");
System.out.print(parts.length);
```

QUESTION 24

What is output by the code to the right?

A. `13` B. `5` C. `3`

D. `1` E. `0`

```
String name = "william_KAHAN";
int count = 0;
for(int i = 0; i < name.length(); i++) {
  char ch = name.charAt(i);
  if( ch == 'a' && ch == 'i' )
    count++;
}
System.out.print(count);
```

QUESTION 25

The `Coord` class to the right will not compile due to a syntax error. Which of the following best describes the syntax error that is present?

A. Instance variables such as `x` and `y` cannot be declared final.

B. The instance variables `x` and `y` must be assigned a value in the line of code where they are declared.

C. The constructor may not have parameters that use the same identifier as instance variables.

D. The `Coord` class does not have a `toString` method.

E. The keyword `this` is not defined in `static` methods.

```
public class Coord {
  private final int x;
  private final int y;

  public Coord(int x, int y) {
    this.x = x;
    this.y = y;
  }

  public static void print(){
    System.out.print(this.toString());
  }
}
```

QUESTION 26

Given methods `sort` and `swap` to the right, what is output by the following client code?

```
int[] data = {2, -5, 10, -5, 3};
sort(data, 0, 4);
System.out.print(Arrays.toString(data));
```

A. `[2, -5, 10, -5, 3]`

B. `[-5, -5, 2, 3, 10]`

C. `[-5, 2, 3, 10]`

D. `[10, 3, 2, -5, -5]`

E. `[10, 3, 2, -5]`

```
public void sort(int[] list, int a, int b){
  if(a < b) {
    int p = (a + b) / 2;
    swap(list, p, a);
    p = list[a];
    int i, j = a;
    for(i = a + 1; i <= b; i++ ){
      if(list[i] > p) {
        j++;
        swap(list, i, j);
      }
    }
    swap (list, a, j);
    sort( list, a, j - 1 );
    sort( list, j + 1, b );
  }
}

public void swap(int[] list, int a, int b){
  int t = list[a];
  list[a] = list[b];
  list[b] = t;
}
```

QUESTION 27

Which sorting algorithm do the methods `sort` and `swap` implement?

A. radix sort B. quicksort

C. insertion sort D. merge sort

E. selection sort

QUESTION 28

What is output by the code to the right?

A. `1` B. `520` C. `521`

D. The output cannot be determined due to overflowing the `int` data type.

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

```
System.out.print(521 ^ 520);
```

QUESTION 29

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

I. `new Iterator<Integer>(col)`
II. `col.iterator()`
III. `col.listIterator()`

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

D. II and III   E. I and III

```
public int myst(ArrayList<Integer> col) {
  int total = 0;
  Iterator<Integer> it = <*1>;
  while(it.hasNext())
    total += it.next();
  return total;
}

public void demo(){
  ArrayList<Integer> list;
  list = new ArrayList<Integer>();
  list.add(10);
  list.add(0, 5);
  System.out.print(myst(list));
}
```

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

QUESTION 30

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

A. `10`   B. `5`   C. `510`

D. `105`   E. `15`

QUESTION 31

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

I. `4`
II. `new Integer(4)`
III. `4.0`

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

D. I and II   E. II and III

```
Map<Integer,String> map;
map = new TreeMap<Integer,String>();

map.put(<*1>, "CBS");
map.put(<*1>, "FOX");

System.out.print(map.size());
boolean b = map.keySet().remove(<*1>);
System.out.print(b);
```

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

QUESTION 32

What is output by the code to the right?

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

D. `2true0`   E. `2true`

QUESTION 33

Assume method `sample(int[] data)` is O(N) where N = `data.length`. When method `sample` is passed an array with `length` = 2,000 it takes 1 second for method `sample` to complete. If method `sample` is then passed an array with `length` = 18,000 what is the expected time it will take method `sample` to complete?

A. 18 seconds   B. 27 seconds   C. 9 seconds   D. 36 seconds   E. 729 seconds

QUESTION 34

Which of the following is not a Java keyword?

A. `string`   B. `null`   C. `finally`   D. `throws`   E. `do`

QUESTION 35

If **<*1>** in method `make` is replaced with the following what is the Big O of method `make`? `vals` contains N distinct values. Pick the most restrictive correct set of answers.

	`TreeSet`	`HashSet`
A.	O(N^2)	O(N^2)
B.	O(logN)	O(1)
C.	O(N^2)	O(N)
D.	O(1)	O(logN)
E.	O(NlogN)	O(N)

```
public Set<Double> make(double[] vals) {
  Set<Double> result = new <*1><Double>();
  for(double d : vals)
    result.add(d);
  return result;
}
```

QUESTION 36

Given the `Point` and `Point3D` classes to the right what is output by the following client code?

```
Point3D p1 = new Point3D();
System.out.print(p1);
```

A. `-5:5:3`

B. `0:0:3`

C. `0:0:0`

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 37

Given the `Point` and `Point3D` classes to the right, what is output by the following client code?

```
Point p2 = new Point3D(1, 2, 3);
p2.inc();
System.out.print(p2.toString());
```

A. `2:3:3`

B. `2:2:3`

C. `1:2:3`

D. `1:2:4`

E. `2:3:4`

```
public class Point {
  private int x, y;

  public Point() { x = -5; y = 5; }

  public Point(int xx, int yy) {
    x = xx;
    y = yy;
  }

  public void move() {  x = y; }

  public void inc() {
    x++;
    y++;
  }

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

public class Point3D extends Point {
  private int z;

  public Point3D() { this(3); }

  public Point3D(int zz) { z = zz; }

  public Point3D(int x, int y, int zz) {
    super(x, y);
    z = zz;
  }

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

  public String toString() {
    return super.toString() + ":" + z;
  }
}
```

QUESTION 38

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

```
Struct<String> str1;
str1 = new Struct<String>();
str1.add("a");
str1.add("m");
str1.add("s");
str1.add("c");
System.out.println( str1.checkMid() );
```

A. `a`

B. `c`

C. `s`

D. `m`

E. There is no output.

QUESTION 39

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

```
Struct<String> str2;
str2 = new Struct<String>();
str2.add("S");
str2.add("C");
str2.add("T");
str2.add("U");
System.out.println( str2.remove() );
```

A. `T`

B. `C`

C. `U`

D. `S`

E. `true`

QUESTION 40

What kind of data structure does the `Struct` class implement?

A. A min heap

B. A stack

C. A max heap

D. A set

E. A binary search tree

```
public class Struct <E extends Comparable>{
  private ArrayList<E> con;

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

  public void add(E item) {
    con.add(item);
    int i = con.size() - 1;
    while ( (i != 0) &&
     (con.get(p(i)).compareTo(item) < 0)) {

      con.set(i, con.get(p(i)));
      con.set(p(i), item);
      i = p(i);
    }
  }

  public E remove() {
    E it = con.get(0);
    con.set(0, con.remove(con.size()-1));
    int i = 0;
    while( l(i) < con.size() ) {
      int le = l(i), ri = r(i);
      int si;
      if( ri >= con.size() )
        si = le;
      else if ( ch(le, ri) > 0)
        si = le;
      else
        si = ri;
      if ( ch(i, si) < 0) {
        E temp = con.get(i);
        con.set(i, con.get(si));
        con.set(si, temp);
        i = si;
      }
      else
        i = con.size();
    }
    return it;
  }

  private int ch(int x, int y) {
    E fi = con.get(x);
    return fi.compareTo(con.get(y));
  }

  private int l(int i) { return 2 * i + 1;}
  private int r(int i) { return 2 * i + 2;}
  private int p(int i) { return (i-1) / 2;}

  public E checkMid(){
    return con.get( con.size() / 2 );
  }
}
```

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 xing 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`, xing 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)`

No Test Material on this Page.

Computer Science Answer Key
UIL Regional - 2010

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

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. `cs` is a `CurvedScore` object so the call to `getScore` in `toString` results in a call to the `getScore` method in the `CurvedScore` class.

11. The `&` operator has a higher precedence than the `|` operator. `21 & 10` is evaluated first resulting in `0`. `30 | 0` evaluates to `30`.

12. The `Math.random()` method "returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0." Thus it is not possible limit will ever equal 5.0.