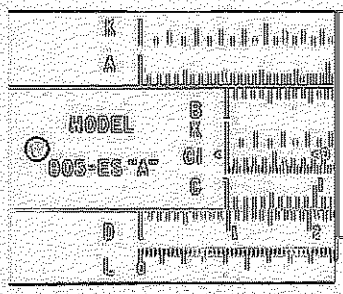
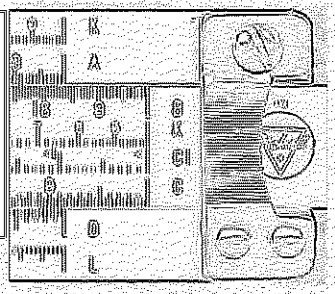
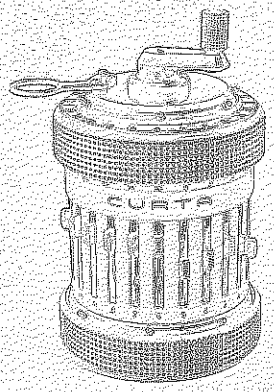
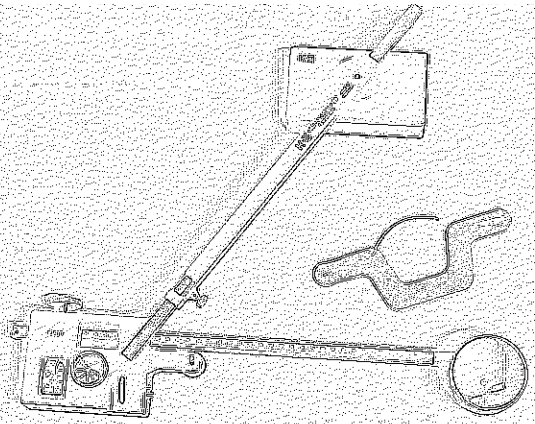


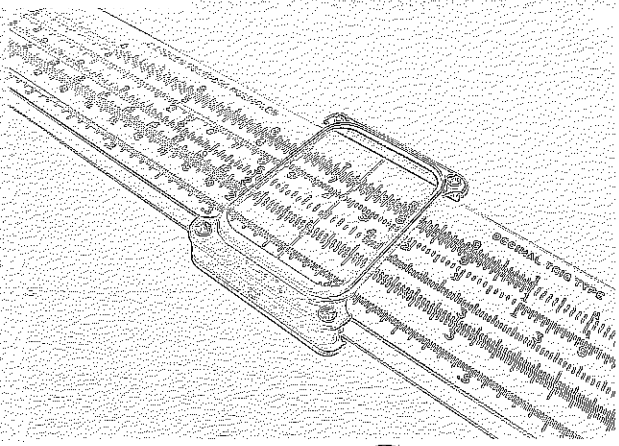
Texas Competitive Mathematics
Web - http://www.texasmath.org
Fax - (866) 606-3535
E-Mail - webmaster@texasmath.org

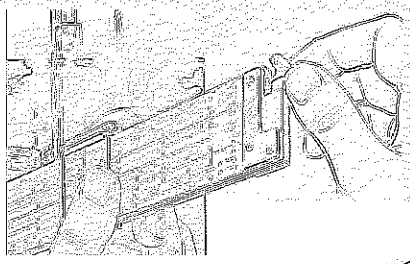


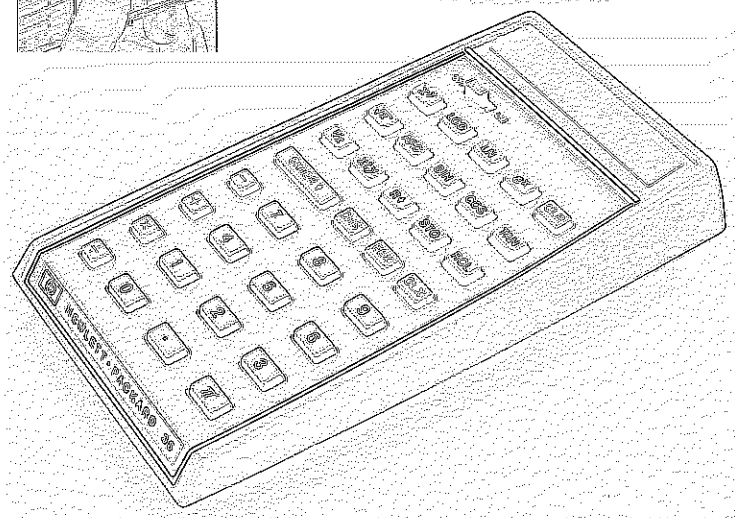
2008 UIL Computer Science District 1
(14 pages)

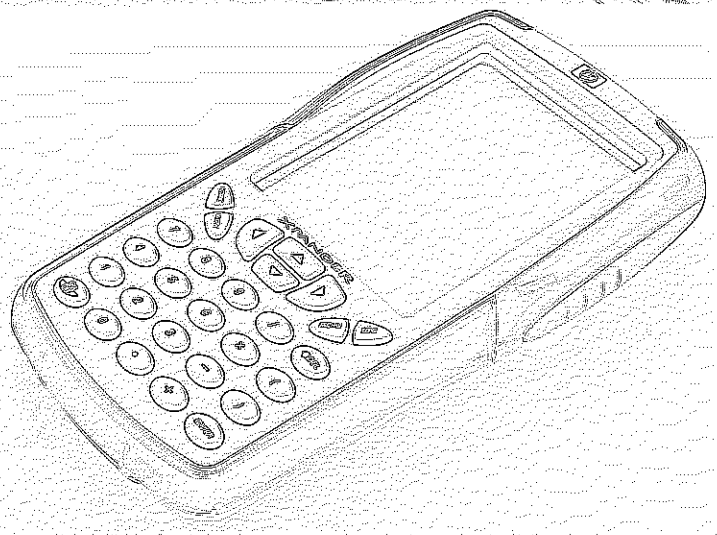


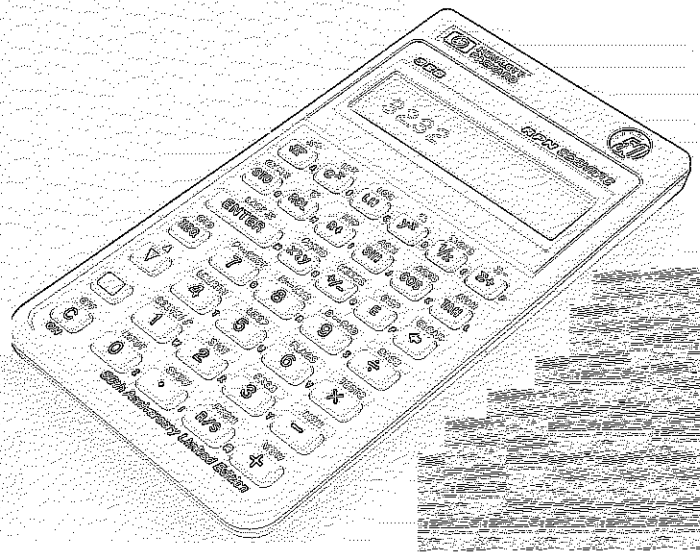


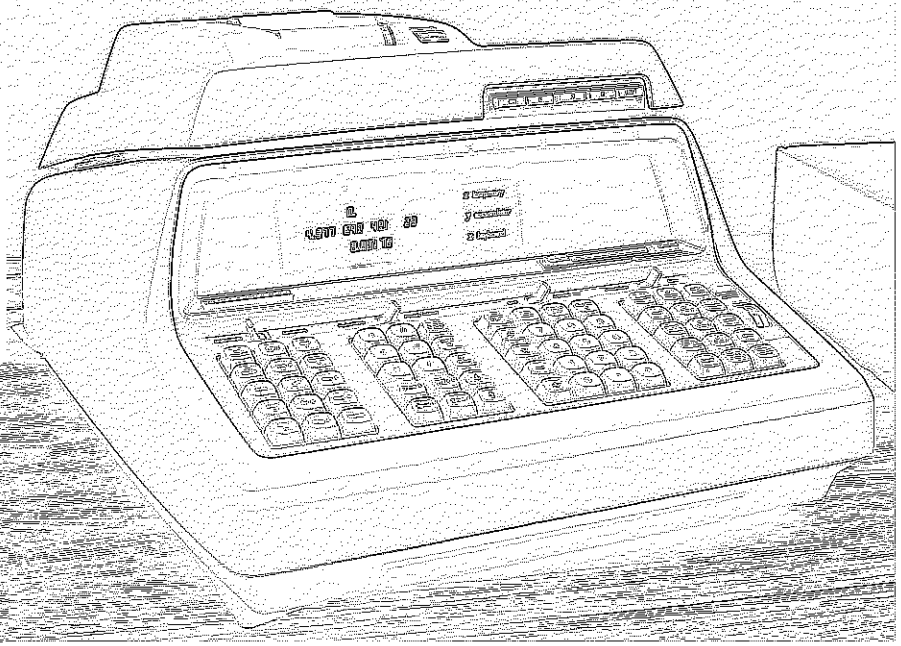












University Interscholastic League

Computer Science Competition

Number 109 (District 1 - 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. You may 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 123_{16} and DEF_{16}?

A. 1022_{16} B. $F12_{16}$ C. $EF0_{16}$ D. 1000_{16} E. $FF1_{16}$

QUESTION 2

What is output by the code to the right?

A. 2.0 B. 2.75 C. 3.0

D. 2 E. 3

```
int x = 11;
int y = 4;
System.out.print( x / y );
```

QUESTION 3

What is output by the code to the right?

A. 26 B. 25 C. 0

D. 1 E. 325

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

QUESTION 4

What is output by the code to the right?

A. esort B. Merg C. s

D. Merge E. sort

```
String alg = "Mergesort";
String sub = alg.substring( 4 );
System.out.print( sub );
```

QUESTION 5

What is output by the code to the right?

A. 9 B. 1 C. 3

D. 6 E. 0

```
int[] nums = new int[5];
nums[1] = 3;
nums[3] = 3 + nums[1] * 2;
System.out.print( nums[1] );
```

QUESTION 6

What is output by the code to the right?

A. 5 B. 5.0 C. 5.5

D. 6.0 E. 4.0

```
double b = 1.5;
b = b * 2 + 5 / 2;
System.out.println( b );
```

QUESTION 7

What is output by the code to the right?

A. false false B. false true

C. true false D. true true

E. true false true

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

QUESTION 8

What is output by the code to the right?

A. 3 B. 4 C. 2

D. 6 E. 5

```
int z = 3;
if( z <= 3 )
  z++;
if( z > 3 )
  z++;
else
  z--;
System.out.print( z );
```

QUESTION 9

How may constructors does the class `Book` have?

A. 1

B. 2

C. 3

D. 4

E. 5

QUESTION 10

What is output by the client code to the right?

A. `false`

B. `true`

C. `Blink`

D. `200`

E. `Blink:true`

```
public class Book{
  private String title;
  private int pages;

  public Book(String t){
    title = t;
  }

  public Book(String t, int len){
    title = t;
    pages = len;
  }

  public boolean longBook(){
    return pages > 250;
  }
}

////////////////////////////////////////
// client code
Book bk = new Book("Blink", 200);
System.out.print( bk.longBook() );
```

QUESTION 11

What is output by the code to the right?

A. 32 B. 16 C. 48

D. 0 E. 1

```
int f = 48;
int g = 16;
System.out.print( f | g );
```

QUESTION 12

What is output by the code to the right?

A. 3 B. 3.5 C. 4

D. -3 E. -3.0

```
double org = -3.5;
System.out.print( Math.abs(org) );
```

QUESTION 13

What is output by the code to the right?

A. `A "hard class"`

B. `A hard class`

C. `A "hard class`

D. `A "hard" class`

E. `a hard class`

```
System.out.print("A \"hard class");
```

QUESTION 14

What is output by the code to the right?

A. `+C` B. `12.0` C. `+12`

D. `-12` E. `012`

```
System.out.printf( "%+3d", 12 );
```

QUESTION 15

What is returned by the method call `example(2)`?

A. `5` B. `1` C. `2`

D. `4` E. `3`

```
public static int example(int x){
  x++;
  return x + x - 1;
}
```

QUESTION 16

What is output by the code to the right?

A. `983` B. `..` C. `8`

D. `5` E. `gh:`

```
String garbage = "851..983gh:23(10";
String[] data = garbage.split("\\D+");
System.out.print( data[1] );
```

QUESTION 17

What is output by the code to the right?

A. `3` B. `4` C. `0`

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

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

```
String car = "Ford";
Object obj = car;
System.out.print( ((String)obj).length() );
```

QUESTION 18

What is output by the code to the right?

A. `null` B. `10` C. `0`

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

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

```
ArrayList<String> classes;
classes = new ArrayList<String>();
System.out.print( classes.size() );
```

QUESTION 19

What is output by the code to the right?

A. -9

B. 30

C. 20

D. 10

E. 9

```
String first = "rock";
String second = "roll";
int result = first.compareTo( second );

if( result < 0 )
  System.out.print( 10 );
else if( result > 0 )
  System.out.print( 20 );
else
  System.out.print( 30 );
```

QUESTION 20

What replaces **<*1>** in the code to the right to set the new `Soda` object's `size` instance variable to the value of the parameter `sz`?

A. `size = sz`

B. `super.size = sz`

C. `super(sz)`

D. `this.size = sz`

E. More than one of these.

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

QUESTION 21

What is output by the client code to the right?

A. 0

B. 1

C. 1240

D. true

E. false

```
public class Drink{
  private int size;

  public Drink(int sz){
    size = sz;
  }

  public void sip(){
    size--;
  }
}

public class Soda extends Drink{
  private int caffeine;

  public Soda(int sz, int caf){
    <*1>;
    caffeine = caf;
  }
}

////////////////////////////////////////
// client code
Soda coke = new Soda(12, 40);
Soda pepsi = new Soda(12, 40);
System.out.print(coke == pepsi );
```

QUESTION 22

What is output by the code to the right when given this input?

```
2 2.1 3
```

A. 2 B. 4 C. 5

D. 7 E. 8

```
Scanner sc = new Scanner(System.in);
int sum = 0;
while( sc.hasNextInt() )
  sum += sc.nextInt();
System.out.print( sum );
```

QUESTION 23

What is output by the code to the right?

A. `false false`

B. `false true`

C. `true false`

D. `true true`

E. `true`

```
String name = "bob";
System.out.print( name.matches( "bo" ) );
System.out.print(" ");
System.out.print( name.matches( "b." ) );
```

QUESTION 24

What is output by the code to the right?

A. `[0, 2, 3]` B. `[3, 0, 1]`

C. `[0, 1]` D. `[0, 1, 0, 3]`

E. `[3, 2, 0]`

```
ArrayList<Integer> ar;
ar = new ArrayList<Integer>();
ar.add(3);
ar.add(0);
ar.add(0);
ar.set(1, 2);
System.out.print( ar.toString() );
```

QUESTION 25

Which of the following are valid identifiers in Java?

I. `2far`
II. `twoFar`
III. `TWO_FAR`

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

QUESTION 26

What is output by the code to the right?

A. `4`

B. `3`

C. `0`

D. `1`

E. `2`

```
String vals = "abac";
int total = 0;
char ch;

for( int i = 0; i < vals.length(); i++){
  ch = vals.charAt(i);
  switch ( ch ) {
    case 'a' : total++; break;
    case 'b' : total *= 2; break;
    default  : total--;
  }
}

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

QUESTION 27

What is returned by the method call `loopy("", 5)` ?

A. `531-1` B. `531` C. `-1135`

D. `135` E. `135531`

```
public static String loopy(String s, int x){
  if( x <= 0 )
    return s;
  else
    return x + s + loopy(s, x - 2);
}
```

QUESTION 28

Which sorting algorithm does the method `sort` implement?

A. Insertion sort B. Quick sort

C. Merge sort D. Heap sort

E. Selection sort

QUESTION 29

Assume the array `list` initially contains these elements:

`[9, 4, 11, 7, 5, 13]`

What are the contents of the array `list` at the point marked `//line A` when the variable `i` equals `4`?

A. `[4, 7, 9, 11, 5, 13]`

B. `[5, 4, 7, 9, 11, 13]`

C. `[13, 11, 9, 7, 5, 4]`

D. `[11, 9, 7, 4, 5, 13]`

E. `[9, 4, 11, 7, 5, 13]`

```
public void swap(int[] data, int i, int j){
  int t = data[i];
  data[i] = data[j];
  data[j] = t;
}

public void sort(int[] list){
  int j;
  int temp;
  boolean done;

  for(int i = 1; i < list.length; i++){
    temp = list[i];
    j = i;
    done = temp <= list[j - 1];

    // line A

    while( !done ){
      swap(list, j, j - 1);
      j--;
      done = j == 0 || temp <= list[j - 1];
    }
  }
}
```

QUESTION 30

What is returned by the method call `one(2)`?

A. `3` B. `2` C. `9`

D. `4` E. `6`

QUESTION 31

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

A. `9` B. `24` C. `16`

D. `17` E. `26`

```
public static int one(int x){
  x++;
  return x + x;
}

public static int one(int x, int y){
  x--;
  y++;
  return x * y;
}

public static int two(int x){
  return one(x) + one(x, x);
}
```

QUESTION 32

What are the elements in the `Set` named `hold` after the code to the right executes?

A. `[-1, 1, -1, 2]`

B. `[-1]`

C. `[3, 5]`

D. `[2, -1]`

E. The `Set` named `one` is empty after the code to the right executes.

```
int[] data = {-1, 1, -1, 2};
Set<Integer> hold = new HashSet<Integer>();
for(int i : data)
  hold.add( i );

Set<Integer> otherHold;
otherHold = new TreeSet<Integer>();
for(int i : data)
  otherHold.add( i + 2 );

hold.removeAll(otherHold);
```

QUESTION 33

What replaces **<*1>** in the code to the right so that the variable `con` is made to refer to the same array as the variable `temp`?

A. `con = temp`

B. `con.equals(temp)`

C. `con = new E[temp]`

D. `con = new Object[temp.length]`

E. More than one of these.

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

QUESTION 34

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

A. `5`

B. `4`

C. `32`

D. `8`

E. `15`

QUESTION 35

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

A. A binary search tree

B. A list

C. A set

D. A heap

E. A queue

```
public class Structure<E>{

  private E[] con;
  private int size;

  public Structure(){
    size = 0;
    resize(1);
  }

  public void add(E obj){
    if( size == con.length )
      resize( size * 2 );
    con[size] = obj;
    size++;
  }

  public E get(int pos){
    return con[pos];
  }

  public int cap(){
    return con.length;
  }

  private void resize(int len){
    E[] temp = (E[])(new Object[len]);
    for(int i = 0; i < size; i++)
      temp[i] = con[i];
    <*1>;
  }
}

////////////////////////////////////////
// client code
public static void structDemo(){
  Structure<Integer> s;
  s = new Structure<Integer>();
  for(int i = 1; i <= 5; i++)
    s.add( i );
  System.out.print( s.cap() );
}
```

QUESTION 36

In the code to the right, what kind of `Collection` must `col` be so that its elements are always printed out in ascending order?

A. `HashSet` B. `TreeSet`

C. `LinkedList` D. `ArrayList`

E. `Stack`

```
public void show(Collection<Integer> col){
  for(int i : col)
    System.out.println( i );
}
```

QUESTION 37

What is output by the code to the right?

A. 6

B. 5

C. 9

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

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

```
ArrayList<String> list;
list = new ArrayList<String>();
list.add("AA");
list.add("BB");
list.add("A");

ListIterator<String> it = list.listIterator();
while( it.hasNext() )
  if( it.next().length() > 1 )
    it.add( "CC" );

System.out.print( list.size() );
```

QUESTION 38

What is the running time of method `sample`? Assume N equals `x`. Choose the most restrictive correct answer.

A. $O(N^2)$ B. O(1) C. O(N!)

D. O(NlogN) E. O(logN)

```
public static int sample(int x){
  int total = 0;
  for(int i = 1; i <= x; i++){
    for(int j = 1; j <= x; j *= 2){
      total++;
    }
  }
  return total;
}
```

QUESTION 39

What replaces **<*1>** in the code to the right to set the elements at index `i` in the arrays `guess` and `ans` to `-1`?

A. `guess[i], ans[i] = -1;`

B. `guess[i] = -1; ans[i] = guess[i];`

C. `guess[i] *= -1; ans[i] *= -1;`

D. `guess[i] = ans[i] = -1;`

E. More than one of these.

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

QUESTION 40

What is output by the following client code?

```
int[] g = {2, 12, 1, 4, 1, 2};
int[] a = {1, 11, 2, 4, 12, 12};
System.out.print( eval(g, a) );
```

A. 0_0

B. 1_2

C. 1_6

D. 1_7

E. 1_12

```
public static String eval(int[] guess,
                          int[] ans){
  int w = 0;
  int b = 0;
  boolean go;
  int j;

  for(int i = 0; i < guess.length; i++){
    if( guess[i] == ans[i] ){
      b++;
      <*1>
    }
  }

  for(int i = 0; i < guess.length; i++){
    go = guess[i] != -1;
    j = i + 1;
    while( go && j < ans.length){
      if( guess[i] == ans[j] ){
        go = false;
        w++;
        guess[i] = -1;
        ans[j] = -1;
      }
      j++;
    }
  }

  return b + "_" + w;
}
```

No 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()`

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 District 1 2008

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

Notes:

6. The expression `5 / 2` evaluates to `2`. Since both operands are `int`s the `/` is integer division.

9. The automatic default constructor is lost if a class has any explicit constructors.

20. The instance variable size is declared `private` in class `Drink` so it cannot be accessed in any other class, not even child classes.

39. Options B and D are both correct.

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.