

# UIL COMPUTER SCIENCE WRITTEN TEST

# 2016 DISTRICT 2

**APRIL 1-6, 2016**

## General Directions (Please read carefully!)

---

1. DO NOT OPEN THE EXAM UNTIL TOLD TO DO SO.
2. There are 40 questions on this contest exam. You will have 45 minutes to complete this contest.
3. All answers must be legibly written on the answer sheet provided. Indicate your answers in the appropriate blanks provided on the answer sheet. Clean erasures are necessary for accurate grading.
4. You may write on the test packet or any additional scratch paper provided by the contest director, but NOT on the answer sheet, which is reserved for answers only.
5. All questions have ONE and only ONE correct answer. There is a 2-point penalty for all incorrect answers.
6. Tests 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 test until told to do otherwise. You may use this time to check your answers.
7. If you are in the process of actually writing an answer when the signal to stop is given, you may finish writing that answer.
8. All provided code segments are intended to be syntactically correct, unless otherwise stated. You may also assume that any undefined variables are defined as used.
9. A reference to many commonly used Java classes is provided with the test, and you may use this reference sheet during the contest. AFTER THE CONTEST BEGINS, you may detach the reference sheet from the test booklet if you wish.
10. Assume that any necessary import statements for standard Java SE packages and classes (e.g., `java.util`, `System`, etc.) are included in any programs or code segments that refer to methods from these classes and packages.
11. NO CALCULATORS of any kind may be used during this contest.

## Scoring

---

1. Correct answers will receive **6 points**.
2. Incorrect answers will lose **2 points**.
3. Unanswered questions will neither receive nor lose any points.
4. In the event of a tie, the student with the highest percentage of attempted questions correct shall win the tie.

# STANDARD CLASSES AND INTERFACES – SUPPLEMENTAL REFERENCE

## package java.lang

```
class Object
    boolean equals(Object anotherObject)
    String toString()
    int hashCode()

interface Comparable<T>
    int compareTo(T anotherObject)
        Returns a value < 0 if this is less than anotherObject.
        Returns a value = 0 if this is equal to anotherObject.
        Returns a value > 0 if this is greater than anotherObject.

class Integer implements Comparable<Integer>
    Integer(int value)
    int intValue()
    boolean equals(Object anotherObject)
    String toString()
    String toString(int i, int radix)
    int compareTo(Integer anotherInteger)
    static int parseInt(String s)

class Double implements Comparable<Double>
    Double(double value)
    double doubleValue()
    boolean equals(Object anotherObject)
    String toString()
    int compareTo(Double anotherDouble)
    static double parseDouble(String s)

class String implements Comparable<String>
    int compareTo(String anotherString)
    boolean equals(Object anotherObject)
    int length()
    String substring(int begin)
        Returns substring(from, length()).
    String substring(int begin, int end)
        Returns the substring from index begin through index (end - 1).
    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 fromIndex. Returns -1 if str is not found.
    int indexOf(int ch)
    int indexOf(int ch, int fromIndex)
    char charAt(int index)
    String toLowerCase()
    String toUpperCase()
    String[] split(String regex)
    boolean matches(String regex)
    String replaceAll(String regex, String str)

class 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 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, int b)
    static int max(int a, int b)
    static long round(double a)
    static double random()
        Returns a double greater than or equal to 0.0 and less than 1.0.
```

## package java.util

```
interface List<E>
class ArrayList<E> implements List<E>
    boolean add(E item)
    int size()
    Iterator<E> iterator()
    ListIterator<E> listIterator()
    E get(int index)
    E set(int index, E item)
    void add(int index, E item)
    E remove(int index)

class LinkedList<E> implements List<E>, Queue<E>
    void addFirst(E item)
    void addLast(E item)
    E getFirst()
    E getLast()
    E removeFirst()
    E removeLast()

class Stack<E>
    boolean isEmpty()
    E peek()
    E pop()
    E push(E item)

interface Queue<E>
class PriorityQueue<E>
    boolean add(E item)
    boolean isEmpty()
    E peek()
    E remove()

interface Set<E>
class HashSet<E> implements Set<E>
class TreeSet<E> implements Set<E>
    boolean add(E item)
    boolean contains(Object item)
    boolean remove(Object item)
    int size()
    Iterator<E> iterator()
    boolean addAll(Collection<? extends E> c)
    boolean removeAll(Collection<?> c)
    boolean retainAll(Collection<?> c)

interface Map<K,V>
class HashMap<K,V> implements Map<K,V>
class TreeMap<K,V> implements 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()

interface Iterator<E>
    boolean hasNext()
    E next()
    void remove()

interface ListIterator<E> extends Iterator<E>
    void add(E item)
    void set(E item)

class Scanner
    Scanner(InputStream source)
    Scanner(String str)
    boolean hasNext()
    boolean hasNextInt()
    boolean hasNextDouble()
    String next()
    int nextInt()
    double nextDouble()
    String nextLine()
    Scanner useDelimiter(String regex)
```

# UIL COMPUTER SCIENCE WRITTEN TEST – 2016 DISTRICT

**Note:** Correct responses are based on **Java SE Development Kit 8 (JDK 8)** from Sun Microsystems, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g., "error" is an answer choice) and any necessary Java SE 8 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used. **For all output statements, assume that the System class has been statically imported using:**

```
import static java.lang.System.*;
```

## Question 1.

Which of the following is equivalent to  $110011_2 + 10111_2$ ?

- A)  $100100_2$       B)  $2202_3$       C)  $111_8$       D)  $148_{10}$       E)  $4B_{16}$

## Question 2.

What is the output of the code segment to the right?

- A) 0      B) 1      C) 3      D) 4      E) 5

```
int x = 20;
int y = 15;
out.println(x % (x - y));
```

## Question 3.

What is the output of the code segment to the right?

- A) UILCS      B) 8573767883      C) uilcs  
D) U      E) 85  
I      73  
L      76  
C      78  
S      83

```
int[] uil = {'U', 'I', 'L', 'C', 'S'};
for (int i : uil)
    out.println((char) i);
```

## Question 4.

What is the output of the code segment to the right?

- A) bral En      B) bral Enhanc      C) ebral En  
D) bral Enh      E) ebral Enhan

```
String hobbes = "Cerebral Enhance-o-tron";
out.println(hobbes.substring(4, 11));
```

## Question 5.

Which of the following Boolean expressions is equivalent to the truth table for output X, as shown to the right?

- A)  $(R \mid\mid Q) \ \&\& \ !P$   
B)  $(R \ \&\& \ Q) \ \mid\mid \ !P$   
C)  $!(R \ \&\& \ Q) \ \mid\mid \ P$   
D)  $R \ \&\& \ (Q \ \mid\mid \ !P)$   
E)  $!(R \ \mid\mid \ Q) \ \&\& \ P$

P	Q	R	X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

## Question 6.

Which of the following outputs can never be produced by the code segment to the right?

- A) 6      B) 7      C) 11      D) 16      E) 17

```
int ran = 7;
int dumb = 11;
int num = (int)(ran + dumb * Math.random());
out.println(num);
```

## Question 7.

What is the output of the code segment to the right?

- A) -16      B) -14      C) 14      D) 16  
E) No output due to an error.

```
int alfa = 25;
int bravo = alfa * -3;
int charlie = bravo + 11;
out.println(charlie %= alfa);
```

**Question 8.**

What is the output of the code segment to the right if the value of test is initialized as follows?

```
int test = 4;
```

- A) 4  
 B) four.seven.eight.8  
 C) four.7  
 D) four.eight.7  
 E) seven.four.eight.8

```
switch(test) {
    case 7: out.print("seven.");
            test++;
    case 4: out.print("four.");
            test += 3;
    case 8: out.print("eight.");
}
out.println(test);
```

**Question 9.**

What is the output of the code segment to the right?

- A) ###  
 B) #####  
 C) #####  
 D) #####  
 E) The code segment prints an infinite string of # characters.

```
int pound = 4;
do {
    out.print("#");
    pound *= pound;
} while (pound < 50000);
```

**Question 10.**

What is printed by the following invocation of the crunch() method from a client class?

```
int[] id = {109, 105, 99, 107, 101, 108};
out.println( crunch(id) );
```

- A) 104    B) 107    C) 214    D) 314    E) 421

```
public static int crunch(int[] a) {
    int b = a[a.length / 2];
    int c = a[a.length / 2];
    for (int d : a) {
        if (d < b) b += d;
        if (d > c) c += d;
    }
    return (b + c) / 2;
}
```

**Question 11.**

What is the output of the code segment to the right if the user enters the following line of console input?

```
11 9 1 6 3 QUIT
```

- A) 10 -1 2 4 -1    B) -10 2 8 -5 3  
 C) 10 -2 -8 5 -3    D) -10 -19 -20 -26 -29  
 E) -10 -8 0 -5 -2

```
Scanner parse = new Scanner(System.in);
int dog = 1;
while (parse.hasNextInt()) {
    int cat = parse.nextInt();
    out.print(dog - cat + " ");
    dog = cat;
}
```

**Question 12.**

What is the output of the code segment to the right?

- A) 465    B) 945    C) 1905    D) 3840    E) 8415

```
int sum = 0;
for (int i = 15; i < 500; i *= 2)
    sum += i;
out.println(sum);
```

**Question 13.**

What is the output of the code segment to the right?

- A) 0    B) 11    C) 24    D) 27    E) 31

```
out.println(28 ^ 77 >> 3 | 15 & 27);
```

**Question 14.**

Which of the following abstract data types would be the most optimal choice for implementing a waiting list for backordered inventory in which customers' names are stored and removed from the list in a first-come, first-served manner?

- A) ArrayList    B) Queue    C) TreeMap    D) HashSet    E) Stack

**Question 15.**

What is the output of the code segment to the right?

- A) [0, 4, 8, 12, 16, 20, 24, 28, 32, 36]  
 B) [0, 4, 4, 12, 8, 20, 12, 28, 16, 36]  
 C) [4, 8, 16, 12, 16, 20, 24, 28, 32, 36]  
 D) [0, 2, 8, 6, 16, 10, 24, 14, 32, 18]  
 E) [0, 4, 2, 12, 5, 20, 8, 28, 10, 36]

```
List<Long> longs = new ArrayList<Long>();
for (int i = 0; i < 10; i++) {
    longs.add((long) i * 4);
    if (i % 2 == 0)
        longs.set(i / 3, (long) i * 2);
}
out.println(longs);
```

**Question 16.**

Which of the following strings does NOT match the regular expression shown to the right?

- A) 010101      B) 011001011      C) 1  
D) 10010111      E) 0001011

$$0^*1(0+1+)^*$$
**Question 17.**

What is the output of the code segment to the right?

- A) cdenu11  
B) feedbba  
C) bdeenu11  
D) febb  
E) No output due to an error.

```
Stack<Object> stk = new Stack<>();
String signal = "abbbccdeeeefgg";
for (char c : signal.toCharArray())
    stk.push(c);
while (!stk.isEmpty()) {
    if (stk.pop().equals(stk.pop()))
        out.print(stk.peek());
}
```

**Question 18.**

What is printed by the following invocation of the shuffle() method from a client class?

```
out.println(shuffle("Randomization"));
```

- A) iinoatzdmoanR      B) ontzaiomnRadi  
C) Rnaomdztaonii      D) aadiimnnooRtz  
E) idaRnmoiaztno

```
public static String shuffle(String s) {
    if (s.length() == 0) return "";
    int i = s.length() / 2;
    char c = s.charAt(i);
    String s1 = s.substring(0, i);
    String s2 = s.substring(i+1);
    return shuffle(s1) + shuffle(s2) + c;
}
```

**Question 19.**

Which of the following algorithms is implemented by the code() method to the right?

- A) Binary Search      B) Insertion Sort  
C) Selection Sort      D) Quicksort  
E) Merge Sort

```
static int code(char[] a, int b) {
    int n = a.length;
    if (n > 0) {
        int d = n / 2 + 1;
        if (a[d - 1] > b)
            return code(help(a, 0, d - 1), b);
        else if (a[d - 1] < b)
            return d + code(help(a, d, n), b);
        return d - 1;
    }
    return 0;
}
```

**Question 20.**

What is the expected runtime performance for the code() method in the worst case? Choose the most restrictive answer.

- A)  $O(N)$       B)  $O(N * \log_2 N)$       C)  $O(\log_2 N)$   
D)  $O(N^2)$       E) Indeterminate

```
static char[] help(char[] a, int x, int y){
    char[] b = new char[y - x];
    for (int i = x; i < y; i++)
        b[i-x] = a[i];
    return b;
}
```

**Question 21.**

What is the output of line <#1> in the **Client Code** to the right?

- A) -1      B) 0      C) 6      D) 12      E) 15

**Question 22.**

What is the output of line <#2> in the **Client Code** to the right?

- A) eeimn  
B) [t, e, r, , s]  
C) [e, e, e, i, m]  
D) ter s  
E) [e, e, i, m, n]

**Client Code**

```
String event = "computer science";
char[] data = event.toCharArray();
out.println(code(data, 'e')); //<#1>

char[] sub = help(data, 5, 10);
String s = Arrays.toString(sub);
out.println(s); //<#2>
```

**Question 23.**

What is the output of the code segment to the right?

- A) 13      B) 23      C) 2a      D) 31  
E) No output due to an error.

```
out.println(Integer.toString(36, 13));
```

<b>Question 24.</b> What is the output of the code segment to the right? <b>A) 0      B) 1      C) 2      D) 4      E) 7</b>	<pre>String unit = "rubber baby buggy bumpers"; String knife = "b.*b"; String[] slices = unit.split(knife); out.println(slices.length);</pre>																																				
<b>Question 25.</b> What is the output of the code segment to the right? <b>A) one fish      B) 2 fish      C) Red fish</b> <b>D) Blue fish      E) No output due to an error.</b>	<pre>String[] label = {"one fish", "2 fish",                   "Red fish", "Blue fish"}; if (label[0].compareTo(label[1]) &gt; 0)     if (label[2].compareTo(label[3]) &lt; 0)         out.println(label[2]);     else         out.println(label[3]); else if (label[1].compareTo(label[2]) &lt; 0)     out.println(label[1]); else     out.println(label[0]);</pre>																																				
<b>Question 26.</b> What is printed by the following invocation of the filter() method from a client class? <pre>out.println( filter(8) );</pre> <b>A) 0      B) 123      C) 13      D) 23      E) 3</b>	<pre>public static int filter(int stop) {     int i = 0;     int code = 0;     int[] a = {7, 1, 3, 4, 9, 8, 2, 5, 0, 9};     try {         while (i &lt; stop &amp;&amp; a[i] != 0)             a[i] = a[i++] / a[i];     } catch (ArithmeticException e) {         code = code * 10 + 1;     } catch (Exception e) {         code = code * 10 + 2;     } finally {         code = code * 10 + 3;     }     return code; }</pre>																																				
<b>Question 27.</b> What is printed by the following invocation of the filter() method from a client class? <pre>out.println( filter(3) );</pre> <b>A) 0      B) 123      C) 13      D) 23      E) 3</b>																																					
<b>Question 28.</b> What is the output of the code segment to the right? <b>A) [ .42 ]      B) [ .24 ]</b> [ .62.63 ]      [ .26.36 ] [ .82.83.84 ]      [ .28.38.48 ] <b>C) [ .6 ]      D) [ ]</b> [ .8.9 ]      [ .26 ] [ .10.11.12 ]      [ .28.38 ] <b>E) [ .6.8.9.10.11.12 ]</b>	<pre>for (int one = 4; one &lt; 10; one += 2) {     out.print("[");     for (int ten = 2; ten &lt;= one / 2; ten++)         out.print("." + ten + one);     out.println("]"); }</pre>																																				
<b>Question 29.</b> Given the truth table to the right with inputs P, Q, and R, which of the following is a valid statement about output X? <b>A) X will always be 1 whenever R is 1.</b> <b>B) X will always be 0 whenever Q is 0.</b> <b>C) X will always be 0 whenever exactly 2 inputs are both 1.</b> <b>D) X will always be 0 whenever the values of P and R are different from each other.</b> <b>E) More than one of these statements is valid.</b>	<table><tr><th>P</th><th>Q</th><th>R</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	P	Q	R	X	0	0	0	0	0	0	1	0	0	1	0	1	0	1	1	0	1	0	0	0	1	0	1	1	1	1	0	0	1	1	1	1
P	Q	R	X																																		
0	0	0	0																																		
0	0	1	0																																		
0	1	0	1																																		
0	1	1	0																																		
1	0	0	0																																		
1	0	1	1																																		
1	1	0	0																																		
1	1	1	1																																		
<b>Question 30.</b> What is the output of the code segment to the right? <b>A) 1      B) 3      C) 8      D) 10      E) 124</b>	<pre>out.println(2 ^ 3   6 - 4);</pre>																																				

**Question 31.**

Given the class and interface definitions to the right, what is the output of the following code segment?

```
Employee bart = new Slacker();
bart.paycheck(4);
bart.paycheck(2);
out.println(bart);
```

- A) \$45.0 Try harder. (0.75)
- B) \$75.0 Good job! (1.25)
- C) \$75.0 Good job! (0.75)
- D) \$60.0 Thank you! (1.00)
- E) No output due to an error.

```
public interface Employee {
    public double bonus = 1.00;
    public double paycheck(int hours);
}
```

```
public class Worker implements Employee {
    public double bonus = 1.25;
    public double payRate = 10.00;
    public double wages;
```

```
    public double paycheck(int hours) {
        double pay = hours * payRate;
        wages += pay * bonus;
        return pay;
    }
```

```
    private String comment() {
        if (bonus < 1.0) {
            return "Try harder.";
        }
        else if (bonus > 1.0) {
            return "Good job!";
        }
        else {
            return "Thank you.";
        }
    }
```

```
    public String toString() {
        return "$" + wages + " " + comment();
    }
}
```

```
public class Slacker extends Worker {
    public double bonus = 0.75;

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

**Question 32.**

Given the class and interface definitions to the right, what is the output of the following code segment?

```
Employee lisa = new Worker();
lisa.paycheck(4);
lisa.paycheck(2);
out.println(lisa);
```

- A) \$75.0 Good job!
- B) \$60.0 Thank you.
- C) \$25.0 Good job!
- D) \$45.0 Try harder.
- E) No output due to an error.

**Question 33.**

Given the class and interface definitions to the right, what is the output of the following code segment?

```
Employee maggie = new Employee();
maggie.paycheck(4);
maggie.paycheck(2);
out.println(maggie);
```

- A) \$45 Try harder.
- B) \$75.0 Good job!
- C) \$60.0 Thank you.
- D) \$25.0 Good job!
- E) No output due to an error.

```
Map<String, String> map = new TreeMap<>();
map.put("Dopey", "Doc");
map.put("Sleepy", "Sneezy");
map.put("Grumpy", "Bashful");
map.put("Happy", "Doc");
map.put("Grumpy", "Dopey");
out.println(map.get("Dopey") + " " +
    map.get("Doc") + " " +
    map.get("Grumpy"));
```

**Question 34.**

What is the output of the code segment to the right?

- A) Doc null Bashful
- B) Grumpy Happy Dopey
- C) Doc null [Bashful, Dopey]
- D) Grumpy [Dopey, Happy] null
- E) Doc null Dopey

**Question 35.**

What is the output of the code segment to the right?

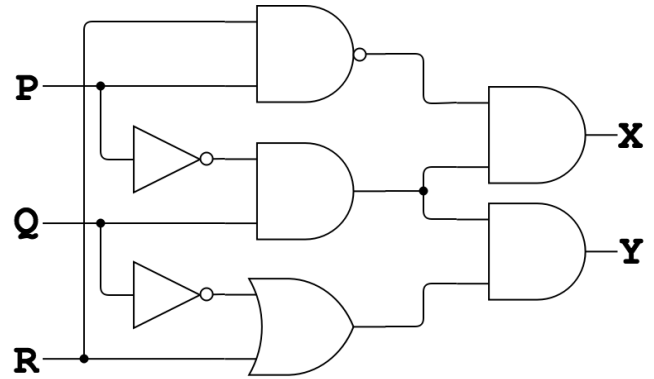
- A) [2, 6, 7]  
 B) [1, 2, 3, 4]  
 C) [5, 6, 9]  
 D) [1, 5, 8, 10]  
 E) No output due to an error.

```
int[][] grid = { {1, 2, 3, 4}, {5, 6, 7},
                 {8, 9}, {10} };
for (int i = 0; i < grid.length; i++)
    for (int j = 0; j < grid[i].length; j++)
        grid[j][i] = grid[i][j];
out.println(Arrays.toString(grid[1]));
```

**Question 36.**

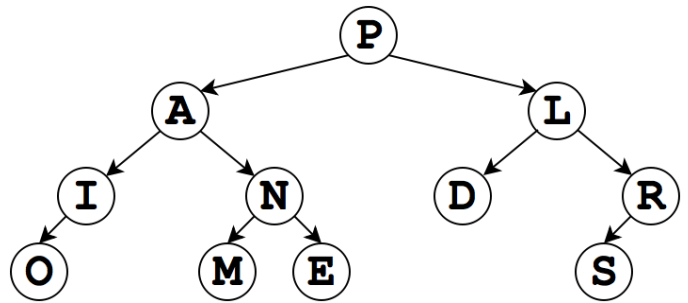
Which of the following set of inputs for the logic diagram to the right will result in a true outputs for both X and Y?

- A) P = false; Q = false; R = true  
 B) P = false; Q = true; R = false  
 C) P = false; Q = true; R = true  
 D) P = true; Q = false; R = true  
 E) P = true; Q = true; R = false

**Question 37.**

Which of the following is a pre-order traversal of the binary tree to the right?

- A) PAIONMELDRS  
 B) POIAMNEDLSR  
 C) PALINDROMES  
 D) OIAMNEPDLRS  
 E) OIMENADSRLP

**Question 38.**

What is the 8-bit, 2's complement representation of  $-111_{10}$ ?

- A)  $10000111_2$       B)  $10010001_2$       C)  $10010000_2$       D)  $-00000111_2$       E)  $01101111_2$

**Question 39.**

What is the prefix notation (Polish notation) for the arithmetic expression to the right?

Write your answer on the answer sheet.

$$(2 + 8) / 3$$

**Question 40.**

Write a simplified, Boolean expression that is equivalent to the expression to the right. Your answer should include as few logical operators as possible.

Write your answer on the answer sheet.

$$(X * Y + Z) * (X + Z)$$

★ DOUBLE-CHECK YOUR ANSWERS ★