UIL COMPUTER SCIENCE WRITTEN TEST

2016 REGION

APRIL 21-23, 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 REGION

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 $3D_{16}$ * 13_8?

A) 101001111_2 **B)** 22131_4 **C)** 1327_8 **D)** 761_{10} **E)** $29F_{16}$

Question 2.

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

A) 2 **B)** 5 **C)** 5.0 **D)** 6 **E)** 6.4

```
double m = 0.4 + 1.2 * 8;
out.println(m / 2);
```

Question 3.

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

A) (002016) **B)** (00002016) **C)** (-0002016)

D) -0002016 **E)** -00002016

```
int year = 2016;
out.printf("%(08d", -year);
```

Question 4.

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

A) .0100100 ..100100 .0100100

B) .0.00.00 .... .0.00.00

C) .0100100 1.100100 10100100

D) 0.00.00

E) .0.00.00 1.1..1.. 10100100

```
String mixed = "10100100";
out.print(mixed.replace("1", "."));
String zeros = mixed;
String ones = mixed.replace("0", ".");
out.println(" " + ones + " " + zeros);
```

Question 5.

Which of the following is equivalent to the Boolean expression on the right assuming that w, x, y, and z have been initialized with integer values?

A) w <= x || y != z

B) w >= x || y == z

C) w > x && y == z

D) w > x || y == z

E) !(w <= x) && !(y != z)

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

Question 6.

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

A) 0.111

B) 0.125

C) 6.000

D) 8.000

E) 9.000

```
double raw = -10.0 / 4;
double floor = Math.abs(Math.floor(raw));
double ceil = Math.abs(Math.ceil(raw));
out.printf("%.3f", Math.pow(floor, ceil));
```

Question 7.

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

A) 83 **B)** 236 **C)** 362 **D)** 623 **E)** 632

```
int hund = 236;
int ten = hund / 10;
int one = hund % 10;
out.println(ten + 10 * one);
```

Question 8.

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

```
int iffy = 12345;
```

A) A **B)** B **C)** C **D)** D **E)** E

```
if (iffy / 9 > 1000)
  if (iffy * 4 > 50000)
    out.print("A");
  else
    out.print("B");
else
  if (iffy % 3 == 0)
    if (iffy % 5 == 0)
      out.print("C");
    else
      out.print("D");
  else
    out.print("E");
```

Question 9.

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

A) 124862480 **B)** 12486248

C) 124862486 **D)** 1248624–80

E) The code segment prints an infinite string of digits.

```
byte digits = 1;
do {
  out.print(digits % 10);
  digits *= 2;
} while (digits < 128);
```

Question 10.

What is the return value of the following invocation of the `get()` method from a client class?

```
int[] q = {7, 1, 3, 4, 9, 8, 2, 5, 0, 9};
out.println( get(q, 7, 1) );
```

A) –1 **B)** 2 **C)** 3 **D)** 5 **E)** 8

```
public int get(int[] x, int y, int z) {
  int w = -1;
  for (int i = y - 1; i >= z; i--) {
    if (x[i - 1] > x[i + 1]) w = x[i + 1];
  }
  return w;
}
```

Question 11.

Assuming that the text file, `seuss.txt`, contains the values shown to the right, what is the output of this code segment?

A) 1

B) 2

C) 3

D) 4

E) No output due to an error.

seuss.txt

```
One fish
Two fish
Red fish
Blue fish
```

```
List<String> fish = new ArrayList<>();
Scanner fin = new Scanner("seuss.txt");
while (fin.hasNextLine()) {
  fish.add(fin.nextLine());
}
out.println(fish.size());
```

Question 12.

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

A) 1.0 **B)** 3.5 **C)** 4.0

D) 7.875 **E)** 8.0

```
double[] eight = new double[8];
double octo = 0;
for (int i = 0; i < eight.length; i++)
  eight[i] = i / 8.0;
for (double ocho : eight)
  octo += ocho;
out.println(octo);
```

Question 13.

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

A) –90 **B)** –30 **C)** 80 **D)** 85 **E)** 124

```
int me = 5;
int you = 24;
int us = 3;
out.print(me - you / us * me + you * me);
```

Question 14.

Which of the following Java classes does NOT implement the `Comparable` interface?

A) Random **B)** String **C)** Boolean **D)** File **E)** Integer

Question 15.

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

A) 1:1 **B)** 0:3 **C)** 1:3 **D)** 2:3

E) No output due to an error.

```
List<List<Object>> all = new ArrayList<>();
List<Object> some = new ArrayList<>();
some.add(all);
some.add(all.size());
all.add(some);
some.add(some.size());
out.println(all.size() + ":" + some.size());
```

Question 16.

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

A) `false null` **B)** `null null`

C) `false false` **D)** `false true`

E) `true null`

```
boolean[] bool = new boolean[10];
Boolean[] Bool = new Boolean[10];
out.println(bool[2] + " " + Bool[2]);
```

Question 17.

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

A) `-2147483648` **B)** `-1`

C) `0` **D)** `1`

E) `2147483647`

```
int max = Integer.MAX_VALUE;
int min = Integer.MIN_VALUE;
int sum = (-max) + (-min);
out.println(sum);
```

Question 18.

Which of the following could replace **<#1>** in the code segment to the right to initialize `posneg` to a value of either `-1` or `1`?

A) `(int)(Math.random() * 4 - 1)`

B) `(int)(Math.random() * 2) - 1`

C) `(int)(Math.random() * 2) * 2 - 1`

D) `(int)(Math.random() * 2) * -1`

E) More than one of these.

```
int posneg = <#1>;
```

Question 19.

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

A) `[]`

B) `[, -, , --, , , ----]`

C) `[.-., --.....----.]`

D) `[.-., ., --., .., ---, -.]`

E) `[, -, --, ----]`

```
String R = ".-.";
String E = ".";
String G = "--.";
String I = "..";
String O = "---";
String N = "-.";
String morse = R + E + G + I + O + N;
String[] dashes = morse.split(E);
out.println(Arrays.toString(dashes));
```

Question 20.

What return value is printed after the following invocation of the `find()` method from a client class?

```
int[] bits = {1, 0, 1, 0, 1, 0, 0, 1};
out.println( find(bits, 0) );
```

A) `-1` **B)** `1` **C)** `3` **D)** `5` **E)** `6`

```
public int find(int[] data, int item) {
  int i = -1;
  for (int j = 0; j < data.length; j++) {
    if (data[j] == item)
      i = j;
  }
  return i;
}
```

Question 21.

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

A) `2nd first FOURTH Third`

B) `2nd FOURTH Third first`

C) `FOURTH Third first 2nd`

D) `first 2nd Third FOURTH`

E) `first FOURTH Third 2nd`

```
Queue<String> queue = new PriorityQueue<>();
queue.add("first");
queue.add("2nd");
queue.add("Third");
queue.add("FOURTH");
while (!queue.isEmpty()) {
  out.print(queue.remove() + " ");
}
```

Question 22.

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

A) `2` **B)** `15` **C)** `33` **D)** `62` **E)** `124`

```
byte scan = 31;
scan <<= scan / 15;
out.println(scan);
```

Question 23.

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

A) `15` **B)** `37` **C)** `52` **D)** `77` **E)** `112`

```
out.println(Integer.parseInt("52", 15));
```

Question 24.

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

A) 9 **B)** 14 **C)** 15 **D)** 21 **E)** 36

Question 25.

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

A) [E, G, I, O, N, R]

B) [N, O, I, G, E, R]

C) [E, G, I, N, O, R]

D) [R, E, G, I, O, N]

E) [R, O, N, I, G, E]

Question 26.

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

A) Sequential Search **B)** Merge Sort

C) Insertion Sort **D)** Selection Sort

E) Quicksort

Question 27.

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

A) $O(\log_2 N)$ **B)** $O(N)$ **C)** $O(N * \log_2 N)$

D) $O(N^2)$ **E)** Indeterminate

```
static int process(List<String> a) {
  int n = 0;
  for (int i = 0; i < a.size(); i++) {
    n += help(a, i);
  }
  return n;
}

static int help(List<String> a, int i) {
  String c = a.get(i);
  int n = i - 1;
  while (n >= 0) {
    if (a.get(n).compareTo(c) > 0) break;
    n--;
  }
  a.add(n + 1, a.remove(i));
  return i - n - 1;
}
```

Client Code

```
String str = "REGION";
List<String> c = new ArrayList<>();
for (int i=0; i<str.length(); i++)
  c.add(str.substring(i, i+1));
out.println(process(c));       //<#1>
out.println(c);                //<#2>
```

Question 28.

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

A) UIL **B)** uil2016

C) 2016 Regional **D)** (2016)

E) uil2016regional

```
([a-z]*[0-9]+)*.+[^0-9]
```

Question 29.

What return value is printed after the following invocation of the `hash()` method from a client class?

```
out.println(hash("abcdefghij", 3));
```

A) fgc **B)** ghd

C) hid **D)** ije

E) No output due to an error.

Question 30.

What return value is printed after the following invocation of the `hash()` method from a client class?

```
out.println(hash("1234567890", 2));
```

A) 673 **B)** 784

C) 894052 **D)** 90563

E) No output due to an error.

```
public String hash(String src, int n) {
  if (n > src.length()) return "";
  String a = hash(src, n * 2);
  String b = hash(src, n * 2 + 1);
  return a + b + src.substring(n, n + 1);
}
```

Question 31.

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

A) −8531 **B)** −1 **C)** 0 **D)** 8531 **E)** 57005

```
int dead = 0xdead;
int alive = ~dead;
int wanted = dead ^ alive;
out.println(wanted);
```

Question 32.

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

A) 2T 3T 4T 5T 6T 7T 1T

B) 1T 2T 3T 4T 5T 6T 7T

C) 1T

D) 7T 6T 5T 4T 3T 2T 1T

E) 1T 7T 6T 5T 4T 3T 2T

Question 33.

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

A) 1T

B) 2T 3T 4T 5T 6T 7T 1T

C) 1T 7T 6T 3T 2T

D) 2T 3T 6T 7T 1T

E) 1T 2T 3T 4T 5T 6T 7T

Question 34.

What is the output of line **<#3>** in the **Client Code** to the right?

A) 2H 3H 6T 7T 1T

B) 1T 7T 6H 3H 2T

C) 2T 3H 6H 7T 1T

D) 1T 2T 3H 4H 5T 6T 7T

E) 1T 7T 6H 5T 4T 3H 2T

Question 35.

What is the output of line **<#4>** in the **Client Code** to the right?

A) 4T 5T

B) 4H 5T

C) 2T 3H 4T 5T 6H 7T 1T

D) 1T 2T 3H 4H 5T 6T 7T

E) 2T 3H 6H 7T 1T

Question 36.

What type of data structure does the `Disc` class to the right model?

A) Stack

B) Linked List

C) Hash Table

D) Queue

E) Priority Queue

```
public class Disc {
  boolean state;
  private Disc east, west;
  private int i;
  static int n;

  public Disc() { i = ++n; }

  public Disc(Disc w, Disc e) {
    this();
    bind(w, this);
    bind(this, e);
  }

  public static void bind(Disc w, Disc e) {
    w.east = e;
    e.west = w;
  }

  public boolean flip() {
    if (state && east != null)
      east.state = !east.state;
    if (!state && west != null)
      west.state = !west.state;
    return state = !state;
  }

  public Disc get(int x) {
    Disc d = this;
    while (d.east != null) d = d.east;
    do { d = d.west; }
    while (d != null && d.i != x);
    return d;
  }

  public String toString() {
    String s = "";
    Disc d = this;
    while (d.east != null) d = d.east;
    do {
      s += d.i + (d.state?"H ":"T ");
      d = d.west;
    } while (d != null);
    return s;
  }
}
```

Client Code

```
Disc base = new Disc();
Disc disc = new Disc();
Disc.bind(base, disc);
for (int i = 0; i < 5; i++)
  disc = new Disc(base, disc);
out.println(base);              //<#1>
Disc dFour = base.get(4);
Disc.bind(base.get(6), disc.get(3));
out.println(base);              //<#2>
base.get(3).flip();
out.println(disc);              //<#3>
out.println(dFour);             //<#4>
```

Question 37.

Consider the adjacency matrix to the right that describes a connected graph of 7 nodes. A "0" in any cell indicates that there is no direct connection between two nodes. A "1" indicates that there is a path from the corresponding node for the row to the corresponding node for the column. How many unique paths are there from node A to node D that visit each node at most once per path?

A) 1

B) 2

C) 3

D) 4

E) 5

	A	B	C	D	E	F	G
A	0	1	0	0	0	1	1
B	1	0	1	0	0	0	0
C	0	0	0	0	0	0	1
D	1	1	0	1	0	0	0
E	0	0	1	0	1	0	0
F	0	0	0	0	1	0	0
G	0	1	0	1	1	1	0

Question 38.

Which of the following is the equivalent Reverse Polish Notation (RPN) of the arithmetic expression to the right?

A) `3 7 2 / + 4 5 * –`

B) `5 * 4 – 2 / 7 + 3`

C) `– + * 3 / 4 5 7 2`

D) `– + 3 / 7 2 * 4 5`

E) `3 7 2 / 4 5 * + –`

```
3 + 7 / 2 – 4 * 5
```

Question 39.

Write a simplified, Boolean expression to describe output X, given inputs A, B, C, and D, as shown in the truth table to the right, where 0 denotes `false` and 1 denotes `true`. Your answer should use as few logical operators as possible.

Write your answer on the answer sheet.

A	B	C	D	**X**
0	0	0	0	**0**
0	0	0	1	**0**
0	0	1	0	**0**
0	0	1	1	**0**
0	1	0	0	**0**
0	1	0	1	**1**
0	1	1	0	**0**
0	1	1	1	**1**
1	0	0	0	**0**
1	0	0	1	**0**
1	0	1	0	**1**
1	0	1	1	**1**
1	1	0	0	**0**
1	1	0	1	**1**
1	1	1	0	**1**
1	1	1	1	**1**

Question 40.

Given the two 8-bit, signed, 2's complement binary representations in the expression to the right, what is the decimal value of the 8-bit, signed, 2's complement binary representation that results from evaluating the expression?

Write your answer on the answer sheet.

$$11011101_2 + 01011101_2$$

★ DOUBLE-CHECK YOUR ANSWERS ★