UIL COMPUTER SCIENCE WRITTEN TEST

2018 INVITATIONAL B

FEBRUARY/MARCH 2018

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(begin, 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)
```

Note: Correct responses are based on **Java SE Development Kit 8 (JDK 8)** from Oracle, 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 hexadecimal values is equivalent to 11001000_2?

A) 51   **B)** A8   **C)** FF   **D)** 64   **E)** C8

Question 2.

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

A) 78   **B)** -93   **C)** 111   **D)** 213   **E)** -52

```
out.println(15+((19-11)*10)-17);
```

Question 3.

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

A) `importstatic`
   `tpublicclass`

B) `importstatic`
   `publicclass`

C) `importstatic`
   `        publicclass`

D) `import`
   `static`
   `        public`
   `class`

E) `importstatic`
   `    public`
   `class`

```
out.print("import");
out.print("static\n\tpublic");
out.println("class");
```

Question 4.

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

A) `andtigers`

B) `lions and tigers`

C) `lionsandt`

D) `lionsandtigers`

E) `lionstigers`

```
String s="lions";
String t="tigers";
String u=s.concat("and"+t);
out.print(u);
```

Question 5.

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

A) `true`   **B)** `false`

```
boolean a=true,b=false,c;
c=!(a&&!(b||true));
out.print(a&&b||c);
```

Question 6.

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

A) 3   **B)** 4   **C)** 5   **D)** 4.0

E) There is no output. Will not compile due to a type mismatch.

```
int x;
double y=3.8712;
x=Math.floor(Math.round(y));
out.println(x);
```

Question 7.

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

A) 20   **B)** 19   **C)** 19.0   **D)** 19.35

E) There is no output. Will not compile due to a type mismatch.

```
int w,x=8;
double y=4.25,z=7.60;
w=x+(int)(y+z);
out.print(w);
```

Question 8.

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

A) `penny`

B) `nickel`

C) `penny nickel dime quarter`

D) `dime`

E) `dime quarter`

```
String s="JAVA",t="java";
if(s.equals(t))
        out.print("penny ");
else if(s.equals("java"))
        out.print("nickel ");
else if(t.equalsIgnoreCase(s))
        out.print("dime ");
else if(t.equals("java"))
        out.print("quarter");
```

Question 9.

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

A) `1 4 7 9`

B) `1 4 7`

C) `1 2 3 4 5 6 7 8 9`

D) `1 3 6 9`

E) `4 7 9`

```
for(int i=1;i<=9;i+=3)
        out.print(i+" ");
```

Question 10.

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

A) [4, 2, 0, 1, 4, 3]

B) [0, 5, 0, 1, 4, 4]

C) [2, 5, 2, 1, 4, 4]

D) [2, 4, 2, 1, 4, 3]

E) No output. Throws an ArrayIndexOutOfBoundsException .

```
int[] list= {2,5,0,1,4,3};
list[2]=list[0];
list[5]=list[2];
list[list[1]]=list[4];
out.print(Arrays.toString(list));
```

Question 11.

The code segment on the right is intended to read and print the values in `datafile.dat` all on one line. If `datafile.dat` contains the following values all on one line as shown here:

        5 9 1 7 3 4 6 2 3 8

which of the following <u>cannot</u> replace **<code>** in the code segment? Assume that all necessary classes have been imported correctly and that the main method throws an IOException.

A) `s.next()`

B) `s.nextLine()`

C) `s.nextInt()`

D) `s.nextDouble()`

E) All of the above <u>can</u> replace **<code>**.

```
File f=new File("datafile.dat");
Scanner s=new Scanner(f);
while(s.hasNext())
        out.print(<code>+" ");
```

Question 12.

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

A) 71

B) 61

C) 57

D) 70

E) 45

```
int m=10,n=1,p=1;
while(m>0) {
        p=p+n*m;
        n++;
        m-=2;
}
out.print(p);
```

Question 13.

In any given expression, which of the following operations would be done first?

    A) <=   **B)** `type cast`   **C)** *   **D)** ==   **E)** | (bitwise OR)

Question 14.

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

    A) 8   **B)** 16   **C)** 32   **D)** 64   **E)** 256

```
out.print(Short.SIZE);
```

Question 15.

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

    A) [2, 0, 1]

    B) [1, 2, 3]

    C) [2, 1, 0]

    D) [3, 2, 1]

    E) [0, 1, 2]

```
ArrayList<Integer> a=new
ArrayList<Integer>();
a.add(3);
a.add(2);
a.add(1);
a.add(0);
a.remove(0);
out.print(a);
```

Question 16.

Which of the following best describes the method shown on the right?

    A) Returns the sum of the values in the diagonal rows in array m.

    B) Returns the largest value in array m.

    C) Returns the sum of all of the values in array m .

    D) Returns the sum of the largest value in each of the rows in array m.

    E) Returns the smallest value in array m.

```
public static int method(int[][] m) {
int a=0;
for(int x=0;x<m.length;x++) {
    int r=0;
    for(int y=0;y<m[x].length;y++)
        if(m[x][y]>r)
            r=m[x][y];
    a+=r;
    }
return a;
}
```

Question 17.

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

    A) [3.14, 0.58, 2.72, 0.92, 1.2]

    B) [0.58, 3.14, 2.72, 0.92, 1.2]

    C) [3.14, 0.58, 1.2]

    D) [3.14, 0.92, 1.2]

    E) [3.14, 0.58, 0.92, 1.2]

```
List<Double> list=new LinkedList<Double>();
list.add(3.14);
list.add(2.72);
list.add(0.92);
list.set(1,0.58);
list.add(1.20);
list.get(2);
out.print(list);
```

Question 18.

What is the output of the code segment shown on the right?

    A) 3   **B)** 4   **C)** 5   **D)** 6   **E)** 7

```
String s="Anteaters hate aardvarks";
String[] t=s.split("a");
out.println(t.length);
```

Question 19.

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

    A) 0   **B)** 1   **C)** 90   **D)** 91

    E) There is no output due to an error.

```
long e=90,f=91,g;
g=f%2>e/90?e%2:f/90;
out.print(g);
```

Question 20.

Which of the following must replace **<code #1>** in the binary search method shown on the right?

    A) `(front+back)/2`

    B) `front+back`

    C) `(back-front)/2`

    D) `back/2`

    E) `mid/2`

Question 21.

Which of the following must replace **<code #2>** in the binary search method shown on the right?

    A) `front`

    B) `back`

    C) `mid`

    D) `1`

    E) No additional code is required.

Question 22.

Assuming that **<code #1>** and **<code #2>** have been filled in correctly, what is the greatest possible number of comparisons that will be required within the `while` loop, if `list` contains 1024 elements?

    A) 10

    B) 1024

    C) 100

    D) 32

    E) 512

```
//Use the following code to answer questions 20,
//21 and 22.

/*list is a String array sorted in ascending
order*/

public static int search(String[] list,String s)
{
  int front=0,back=list.length-1;
  int mid=<code #1>;
  while(front<=back) {
    if(list[mid].equals(s))
      return <code #2>;
    else {
      if(list[mid].compareTo(s)<0)
        front=mid+1;
      else
        back=mid-1;
      mid=<code #1>;
    }
  }
  return -1;
}
```

```
//Use the class shown here to answer questions 23 - 27.

import java.util.ArrayList;

public class DataStructure<E> {

      private ArrayList<E> list=new ArrayList<E>();

      public int size() {
            return list.size();
      }

      public E next() {
            return list.get(size()-1);
      }

      public void add(E item) {
            list.add(item);
      }

      public E remove() {
            return list.remove(size()-1);
      }

      public boolean isEmpty() {
            return list.isEmpty();
      }

      public String toString() {
            return list.toString();
      }
}
```

Which of the following can replace **?????** in the code segment shown on the right and ensure that the remainder of the code will compile and execute correctly?

    A) `ArrayList`

    B) `Object`

    C) `String`

    D) A, B, and C

    E) B and C

What is printed by **line #1** in the code segment to the right?

    A) `snow`

    B) `hail`

    C) `rain`

    D) `dust`

    E) `wind`

What is the output of the while loop shown between **comment #1** and **#2** in the code segment to the right?

    A) `sleet rain snow wind`

    B) `sleet wind snow rain`

    C) `sleet hail wind snow rain`

    D) `rain snow wind hail sleet`

    E) `rain snow wind sleet`

What is printed by **line #2** in the code segment to the right?

    A) `0`

    B) `1`

    C) `2`

    D) `3`

    E) `4`

```
DataStructure<?????> ds=new
DataStructure<?????>();
ds.add("rain");
ds.add("snow");
ds.add("wind");
ds.add("hail");
ds.add("dust");
ds.remove();
out.println(ds.next());//line #1
ds.add("sleet");
//comment #1
while(!ds.isEmpty())
    out.print(ds.remove()+" ");
//comment #2
out.println("\n"+ds.size());//line #2
```

Which of the following data structures does the class DataStructure, shown above, implement?

    A) set   **B)** linked list   **C)** priority queue   **D)** stack   **E)** queue

Which of the following is not a valid identifier?

    A) List   **B)** total_weight   **C)** FINAL   **D)** public   **E)** closed

Which of the following must replace **<code>** in the class Student shown on the right so that the code segment shown between the comments will function as a constructor?

A) String

B) void

C) int

D) static

E) No additional code is required.

Assuming that **<code>** has been filled in correctly, which of the following, if present in client code, will instantiate a Student object for a student named Larry?

A) `Student s1=new Student("Larry");`

B) `Student s1="Larry";`

C) `Student.name="Larry";`

D) `Student s1=get_name("Larry");`

E) `name=get_name("Larry");`

Assuming `s1` and `s2` are both `Student` objects constructed with the name "Larry", what is printed by this client code?

```
s1.add_grade(95);s1.add_grade(85);
s2.add_grade(95);s2.add_grade(85);
out.println(s1==s2);
```

A) `true`   B) `false`   C) `0`   D) `1`

E) `s1` and `s2` cannot be compared because the `Student` class does not contain a `compareTo` method.

```
public class Student {

    public String get_name() {
        return name;
    }

    public double average() {
        return total/numGrades;
    }

    public void add_grade(int g) {
        total+=g;
        numGrades++;
    }
    //comment
    public <code> Student(String name) {
        this.name=name;
    }
    //comment

    private String name;
    private int numGrades;
    private double total;

}
```

What is printed by the code segment shown on the right?

A) `12 2 6`

B) `7 0 10`

C) `10 1 7`

D) `7 1 8`

E) `12 1 6`

```
int x=25,y=12,z=2;
while(y>1) {
    x-=Math.min(y, z);
    y=x/z;
    z++;
}
out.println(x+" "+y+" "+z);
```

The output shown on the right was produced by printing the contents of an array each time a particular sort made a pass over that array. Which of the following sort routines was it?

A) quick sort

B) merge sort

C) bubble sort

D) selection sort

E) insertion sort

```
[5, 3, 1, 0, 2, 4] Original List
[0, 3, 1, 5, 2, 4]
[0, 1, 3, 5, 2, 4]
[0, 1, 2, 5, 3, 4]
[0, 1, 2, 3, 5, 4]
[0, 1, 2, 3, 4, 5]
[0, 1, 2, 3, 4, 5] Sorted List
```

Question 34.

What is the output of the code segment shown on the right?

A) 0   **B)** 1   **C)** 2   **D)** 3   **E)** 4

```
byte i=2,j=3,k=4;
out.print(i<<j>>k);
```

Question 35.

To use the `Collections.sort()` method all elements within the collection to be sorted must implement the _____ interface?

A) `Arrays`

B) `Collections`

C) `ArrayList`

D) `List`

E) `Comparable`

Question 36.

Which of the following Boolean expressions will produce the truth table shown on the right?

A) $A * B + A$

B) $A * B \oplus A$

C) $A + B \oplus A$

D) $A \oplus B \oplus A$

E) $A + B * A$

A	B	
T	T	T
T	F	T
F	T	T
F	F	F

Question 37.

Each element within a linked list data structure is stored within a _____?

A) array   **B)** edge   **C)** identifier   **D)** class   **E)** node
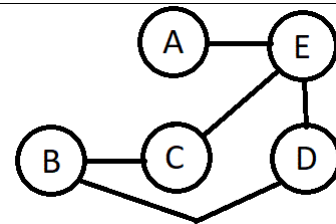
Question 38.

Which of the following is the correct infix representation of the postfix expression shown on the right? (The operands are 15, 4, 2, 3, and 5)

A) 15-4*2/(3+5)

B) 15-4*2/3+5

C) 15*4+2/3-5

D) 2*3+4/5-15

E) (3+5)/4*2-15

```
15 4 2 * 3 5 + / -
```

Question 39.

How many pairs of vertices in the graph shown on the right are adjacent?

Question 40.

What is the decimal equivalent of the 8 bit binary two's complement value 10111000?