

# UIL COMPUTER SCIENCE WRITTEN TEST

# 2019 DISTRICT

**MARCH 2019**

## 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)
```

# UIL COMPUTER SCIENCE WRITTEN TEST – 2019 DISTRICT

**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 is the sum of  $10011000_2$  and  $01010011_2$ ?

- A)  $10101011_2$       B)  $11101011_2$       C)  $11001011_2$       D)  $11011011_2$       E)  $11101010_2$

## Question 2.

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

- A) 39      B) 33      C) 35      D) 0      E) 4

```
out.print(15/8%3+8*4);
```

## Question 3.

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

- A) Mary  
"Pop\spins  
Re\turns"  
B) "Mary Pop\spinsRe\turns"  
C) Mary"Pop\spinsRe      urns"  
D) Mary"Pop\spinsRe\turns"  
E) There is no output. \s is an invalid escape sequence character.

```
out.print("Mary");
out.print("\\"Pop\\spins");
out.print("Re\\turns\\");
```

## Question 4.

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

- A) -verse      B) verse      C) -vers  
D) spider      E) spider-

```
String str="spider-verse";
out.print(str.substring(6).trim());
```

## Question 5.

Which of the lines on the right will print false?

- A) line #1  
B) line #2  
C) line #3  
D) line #4  
E) More than one of the above.

```
boolean a=true,b=true;
out.println(a&&b); //line #1
out.println(a||b); //line #2
out.println(a^b); //line #3
out.println(!a); //line #4
```

## Question 6.

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

- A) 15      B) 15.0      C) 16      D) 16.0      E) 15.99

```
double x=15.98;
out.println(Math.round(x));
```

## Question 7.

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

- A) 5.587      B) 3.254      C) 3.587      D) 3.0      E) 4

```
double d=1.254;
int i=6;
long lg=14;
out.println(lg/i+d);
```

<p><b>Question 8.</b></p> <p>What is the output of the code segment to the right?</p> <p>A) Far Cry Mario Kart B) Call Of Duty Mario Kart C) Grand Theft Auto D) Mario Kart E) Grand Theft Auto Mario Kart</p>	<pre>boolean a=false,b=true,c=false; if(a!=b&amp;&amp;b==c)     out.print("Far Cry "); else if(a  b&amp;&amp;c)     out.print("Call Of Duty "); else     out.print("Grand Theft Auto "); out.print("Mario Kart");</pre>
<p><b>Question 9.</b></p> <p>What is the output of the code segment shown on the right?</p> <p>A) 87654321 B) 987654321 C) 98765432 D) 8765432 E) 876543210</p>	<pre>int e=9; do {     e--;     out.print(e); }while(e&gt;1);</pre>
<p><b>Question 10.</b></p> <p>What is the output of the code segment to the right?</p> <p>A) [4, 1.5, 8.0, 0, 5.25] B) [4.0, 1.5, 8.0, 0.0, 5.25] C) [4.0, 0.0, 0.0, 0.0, 5.25] D) [0, 4, 1.5, 8.0, 5.25] E) There is no output due to an error.</p>	<pre>double[] d=new double[5]; int []i= {1,4,3,0,2}; d[4]=5.25;d[1]=1.5; d[i[4]]=8;d[i[2]]=i[3]; d[0]=4; out.print(Arrays.toString(d));</pre>
<p><b>Question 11.</b></p> <p>What is printed by the code segment shown on the right if the following strings are contained in datafile.dat?</p> <p>dog cat bird snake turtle mouse rabbit fish</p> <p>Assume that all necessary classes have been imported and that the main method throws an IOException.</p> <p>A) dog cat bird snake turtle mouse B) dog cat bird snake turtle C) cat snake mouse D) dog bird turtle E) cat snake mouse rabbit</p>	<pre>Scanner file=new Scanner(new File("datafile.dat")); while(!file.next().equals("rabbit"))     out.print(file.next()+" ");</pre>
<p><b>Question 12.</b></p> <p>What is the output of the code segment to the right?</p> <p>A) 20.0 B) 13.5 C) 25.0 D) 18.5 E) 19.25</p>	<pre>double e=0.0; for(double d=3.5;d&lt;6.5;d+=0.75)     e+=d; out.print(e);</pre>

**Question 13.**

What is printed by the line of code on the right?

- A) true
- B) false
- C) 5
- D) 3
- E) 7

```
out.print(3&2+2|3);
```

**Question 14.**

What is the output of this line of code?

```
out.println(Math.max(Short.BYTES, Byte.BYTES));
```

- A) 1
- B) 2
- C) 4
- D) 8
- E) 16

**Question 15.**

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

- A) [0, 3, 2]
- B) [2, 1, 2]
- C) [2, 3, 2]
- D) [2, 0, 3]
- E) [2, 0, 2]

```
ArrayList<String> list=new
ArrayList<String>();
list.add("2");list.add("0");
list.add("3");list.add("1");
list.add("2");
list.remove(2);list.remove("1");
out.print(list);
```

**Question 16.**

What is printed by this client code? UILClass is shown on the right.

```
UILClass uil=new UILClass("hello",1);
```

- A) hello
- B) null
- C) hello 1
- D) 1
- E) Nothing is printed.

```
public class UILClass {
    String s;
    int i;
    public UILClass(String s, int i) {
        System.out.print(this.s);
        this.s = s;
        this.i = i;
    }
    public String toString() {
        return s+" "+i;
    }
}
```

**Question 17.**

What is the output of the code segment shown here?

```
String str="325-978-1400";
out.print(str.matches(".{3}-\\w+-\\S+")+" ");
out.print(str.matches("325.978.1400")+" ");
out.print(str.matches(".*"));
```

- A) true true true
- B) true false true
- C) true true false
- D) false true false
- E) false false false

**//Use the classes Birthdate and Friend to answer questions 18 - 22.**

```
public class Birthdate {

    private int day,month,year;

    public Birthdate(int m, int d, int y) {
        super(); //line #1
        day = d;
        month = m;
        year = y;
    }
    public int getDay() {<code 1> day;}
    public int getMonth() {<code 1> month;}
    public int getYear() {<code 1> year;}
}

public class Friend {

    public String first,last;
    public Birthdate bd;

    public Friend(String first, String last, Birthdate bd) {
        <code 2>.first = first;
        <code 2>.last = last;
        <code 2>.bd = bd;
    }
    //Parameters m, d and y represent the month, day and year of the
//current date.
    public int age(int m, int d, int y) {
        <missing code segment>
    }
}
```

**Question 18.**

Which of the following best describes the code marked as **line #1** in the `Birthdate` class?

- A) It is a call to the `Friend` class constructor.
- B) It is the default constructor for the `Birthdate` class.
- C) It is a call to the `Object` class constructor.
- D) It causes the `Birthdate` class to extend the `Friend` class.
- E) It causes the `Friend` class to implement the `Birthdate` class.

**Question 19.**

Which of the following should replace **<code 1>** in the `Birthdate` class to ensure the class will compile and execute correctly?

- A) `static`
- B) `int`
- C) `this`
- D) `super`
- E) `return`

**Question 20.**

Which of the following should replace **<code 2>** in the `Friend` class to ensure the class will compile and execute correctly?

- A) `String`
- B) `super`
- C) `Friend`
- D) `this`
- E) `final`

**Question 21.**

Which of the following code segments can replace the **<missing code segment>** in the `Friend` class and ensure that the `age` method will return a friend's correct age in years?

<p><b>A.</b></p> <pre>int age=0; if(m&lt;bd.getMonth()) age=(y-1)-bd.getYear(); else if(m&gt;bd.getMonth()) age=y-bd.getYear(); else if(d&lt;bd.getDay()) age=(y-1)-bd.getYear(); else if(d&gt;=bd.getDay()) age=y-bd.getYear();</pre>	<p><b>B.</b></p> <pre>int age=0; if(m&lt;bd.getMonth()) age=(y-1)-bd.getYear(); else if(m&gt;bd.getMonth()) age=y-bd.getYear(); else if(d&lt;bd.getDay()) age=(y-1)-bd.getYear(); else if(d&gt;=bd.getDay()) age=y-bd.getYear(); return age;</pre>
<p><b>C.</b></p> <pre>int age=0; if(m&lt;bd.getMonth()) age=bd.getYear()-(y-1); else if(m&gt;bd.getMonth()) age=bd.getYear()-y; else if(d&lt;bd.getDay()) age=bd.getYear()-(y-1); else if(d&gt;=bd.getDay()) age=bd.getYear()-y; return age;</pre>	<p><b>D.</b></p> <pre>int age=0; if(m&lt;bd.month) age=(y-1)-bd.year; else if(m&gt;bd.month) age=y-bd.year; else if(d&lt;bd.day) age=(y-1)-bd.year; else if(d&gt;=bd.day) age=y-bd.year; return age;</pre>
<p><b>E.</b> More than one of the above.</p>	

**Question 22.**

Which of the following client code statements will correctly instantiate a `Friend` object for Bill Smith born on June 4, 1984?

- A) `myFriend=Friend("Bill","Smith",new Birthdate("June 4, 1984"));`
- B) `myFriend=Friend("Bill","Smith",new Birthdate(6,4,1984));`
- C) `Friend myFriend=new Friend(Bill,Smith,new Birthdate(6,4,1984));`
- D) `Friend myFriend=new Friend("Bill","Smith",Birthdate(6,4,1984));`
- E) `Friend myFriend=new Friend("Bill","Smith",new Birthdate(6,4,1984));`

**Question 23.**

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

**A)** [0, 1, 6, 9]  
[1, 2, 5, 8]  
[0, 3, 3, 4]  
[2, 4, 5, 7]

**B)** [0, 1, 0, 2]  
[1, 2, 3, 4]  
[6, 5, 3, 5]  
[9, 8, 4, 7]

**C)** [1, 0, 0, 1]  
[2, 4, 2, 3]  
[3, 5, 5, 4]  
[8, 7, 6, 9]

**D)** [0, 0, 1, 1]  
[2, 2, 3, 3]  
[4, 4, 5, 5]  
[6, 7, 8, 9]

**E)** No output due to an `ArrayIndexOutOfBoundsException`.

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

**Question 24.**

Which of the following is **NOT** a correct implementation of the insertion sort algorithm?

**A)**

```
public static void sort(int[] a) {
    for(int i=1;i<a.length;i++) {
        int ce=a[i];
        int k;
        for(k=i-1;k>=0 && a[k]>ce;k--)
            a[k+1]=a[k];
        a[k+1]=ce;
    }
}
```

**B)**

```
public static void sort(int[] a) {
    int i=1;
    do {
        int ce=a[i];
        int k=i-1;
        while(k>=0&&a[k]>ce) {
            a[k+1]=a[k];
            k--;
        }
        a[k+1]=ce;
        i++;
    }while(i<a.length);
}
```

**C)**

```
public static void sort(int[] list) {
    int x,y;
    for(int i=0;i<list.length;i++) {
        x=list[i];
        y=i;
        for(int j=i+1;j<list.length;j++) {
            if(list[j]<x) {
                x=list[j];
                y=j;
            }
        }
        list[y]=list[i];
        list[i]=x;
    }
}
```

**D)**

```
public static void sort(int[] a) {
    int x=1;
    while(x<a.length) {
        int y=x-1;
        int item=a[x];
        for(;y>=0&&a[y]>item;y--)
            a[y+1]=a[y];
        a[y+1]=item;
        x++;
    }
}
```

**E)** More than one of the above.



**Question 25.**

What is the run time complexity (Big O value) for an ascending insertion sort of an array that contains n elements if the array is already sorted in ascending order?

- A)  $O(1)$
- B)  $O(n)$
- C)  $O(n^2)$
- D)  $O(\log n)$
- E)  $O(n \log n)$

**Question 26.**

Which of the following should replace **<code>** in the segment shown on the right to ensure that all of the strings stored in *s* are printed?

- A) `s.hasNext()`
- B) `s.isEmpty()`
- C) `forEach()`
- D) `!s.isEmpty()`
- E) `s.next()`

```
//Use the following to answer 26 and 27
Stack<String> s=new Stack<String>();
s.push("socks");s.push("shoes");s.pop();
s.push("shirt");s.push("pants");s.pop();
s.push("socks");s.push("coat");s.pop();
while(<code>)
    out.print(s.pop()+" ");
```

**Question 27.**

What is the output of the code segment shown on the right once **<code>** has been filled in correctly?

- A) socks shirt socks
- B) shoes pants coat
- C) socks pants socks
- D) shoes shirt coat
- E) shoes pants socks

**Question 28.**

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

- A) 20
- B) 15
- C) 5
- D) 25
- E) 19

```
public static int mtd(int n) {
    if(n==0)
        return 5;
    else
        return n+mtd(n-1);
}
//client code
int sum=mtd(5);
out.print(sum);
```

**Question 29.**

What is printed by the line of code on the right?

- A) 2
- B) 16
- C) 10
- D) 8
- E) 10000

```
out.println(Integer.toString(16, 2));
```

**Question 30.**

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

- A) int
- B) double
- C) final
- D) method
- E) No additional code is required.

```
public static <code> mtd(int i,int j)
{
    double e=1.0;
    while(i<j) {
        if(i%2==0)
            e*=j;
        else
            e+=i;
        i++;
    }
    return e;
}
```

**Question 31.**

If **<code>** has been filled in correctly, what is the output of this line of client code?

```
System.out.print(mtd(3,6));
```

- A) 29
- B) 60.0
- C) 60
- D) 29.0
- E) 23.0

**Question 32.**

Which of the following data types cannot be assigned to a variable that has been declared as an Object data type?

- A) String
- B) Stack
- C) int
- D) Integer
- E) None of the above.

**Question 33.**

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

- A) 45
- B) 34
- C) 46
- D) 25
- E) 40

```
int t=0;
for(int x=1;x<=10;x++) {
    switch(x) {
        case 3:
        case 6:
        case 9:t+=x;break;
        case 5:
        case 7:
        case 10:t+=2;break;
        case 1:
        case 4:
        case 8:continue;
        default: t++;
    }
    t+=3;
}
out.print(t);
```

**Question 34.**

All methods in the Math class \_\_\_\_\_.

- A) are static
- B) are final
- C) return double
- D) are void
- E) have a constructor

**Question 35.**

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

- A) 9 3 0
- B) 9 3 1
- C) 9 4 0
- D) Error. Will not compile.
- E) Error. Throws an exception.

```
int x=8,y=5,z=1;
if(x>y/--z) {
    ++x;
    y-=2;
}
out.print(x+" "+y+" "+z);
```

**Question 36.**

Consider the partially implemented class `Combo` shown on the right. Which of the following statements will not compile and execute?

- A) `Combo<Integer,Double> combo1= new Combo<Integer,Double>(3,3.5);`
- B) `Combo<Integer,Double> combo2= new Combo<>(5,8.2);`
- C) `Combo<String,String> combo3= new Combo<String,String>("cold","hot");`
- D) `Combo<String,int> combo4= new Combo<String,int>("Hey!",5);`
- E) More than one of the above.

```
public class Combo<A,B> {
    private A one;
    private B two;
    public Combo(A o,B t) {
        one=o;
        two=t;
    }
}
```

**Question 37.**

Which of the following is the 8-bit two's complement binary equivalent of -51?

- A) 11001100    B) 11001101    C) 00110100    D) 11001110    E) 11111011

**Question 38.**

If A is true and B is false, what is the value of this Boolean expression?

$$A * B + \bar{A} * \bar{B}$$

- A) true
- B) false

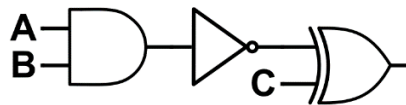
**Question 39.**

How many vertices are within the graph represented by the adjacency matrix shown on the right? Write your answer in the blank provided on the answer document.

```
int[][] adjmat= {{0,1,0,1,0},{1,0,1,0,0},
{0,1,0,1,0},{1,0,1,0,1},{0,0,0,1,0}};
```

**Question 40.**

Write a Java expression (do not use generic notation) in the blank provided on the answer document that is equivalent to the digital electronics diagram shown here.



# ★ ANSWER KEY – CONFIDENTIAL ★

## UIL COMPUTER SCIENCE – 2019 DISTRICT

Questions (+6 points for each correct answer, -2 points for each incorrect answer)

- |                  |                  |                       |                                   |
|------------------|------------------|-----------------------|-----------------------------------|
| 1) <u>  B  </u>  | 11) <u>  C  </u> | 21) <u>  B  </u>      | 31) <u>  D  </u>                  |
| 2) <u>  B  </u>  | 12) <u>  D  </u> | 22) <u>  E  </u>      | 32) <u>  E  </u>                  |
| 3) <u>  D  </u>  | 13) <u>  D  </u> | 23) <u>  C  </u>      | 33) <u>  C  </u>                  |
| 4) <u>  A  </u>  | 14) <u>  B  </u> | 24) <u>  C or E  </u> | 34) <u>  A  </u>                  |
| 5) <u>  E  </u>  | 15) <u>  E  </u> | 25) <u>  B  </u>      | 35) <u>  E  </u>                  |
| 6) <u>  C  </u>  | 16) <u>  B  </u> | 26) <u>  D  </u>      | 36) <u>  D  </u>                  |
| 7) <u>  B  </u>  | 17) <u>  A  </u> | 27) <u>  A  </u>      | 37) <u>  B  </u>                  |
| 8) <u>  E  </u>  | 18) <u>  C  </u> | 28) <u>  A  </u>      | 38) <u>  B  </u>                  |
| 9) <u>  A  </u>  | 19) <u>  E  </u> | 29) <u>  E  </u>      | *39) <u>  5  </u>                 |
| 10) <u>  B  </u> | 20) <u>  D  </u> | 30) <u>  B  </u>      | *40) <u>  !(A&amp;&amp;B)^C  </u> |

\* See "Explanation" section below for alternate, acceptable answers.

**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.

This page intentionally left blank

Explanations:

1.	B	<table><tr><td></td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>+</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td></td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr></table>		1	0	0	1	1	0	0	0	+	0	1	0	1	0	0	1	1		1	1	1	0	1	0	1	1
	1	0	0	1	1	0	0	0																					
+	0	1	0	1	0	0	1	1																					
	1	1	1	0	1	0	1	1																					
2.	B	15/8%3+8*4 = 1%3+8*4 = 1+8*4 = 1+32 =33																											
3.	D	All three statements use print and there are no newline escapes \n so everything is on one line. \' prints a single quote. \\ prints a backslash.																											
4.	A	<table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td></tr><tr><td>s</td><td>p</td><td>i</td><td>d</td><td>e</td><td>r</td><td>-</td><td>v</td><td>e</td><td>r</td><td>s</td><td>e</td></tr></table> <p>substring(6) begins at subscript 6 and continues to the end of the string. trim() removes any whitespace from either end of a string. In this case it has no effect.</p>	0	1	2	3	4	5	6	7	8	9	10	11	s	p	i	d	e	r	-	v	e	r	s	e			
0	1	2	3	4	5	6	7	8	9	10	11																		
s	p	i	d	e	r	-	v	e	r	s	e																		
5.	E	Both line #3 and line #4 will print false.																											
6.	C	Rounds to the nearest integer and returns a long.																											
7.	B	14/6+1.254 = 2+1.254 = 3.254																											
8.	E	The Boolean expressions for both the if statements evaluate to false leading to the else statement. Mario Kart would be printed in all cases.																											
9.	A	e is decremented before the first print so the output begins at 8. e=2 begins the last iteration of the loop where e is then decremented to 1 which is the last value printed.																											
10.	B	<table><tr><td>index</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>d</td><td>4.0</td><td>1.5</td><td>8.0</td><td>0.0</td><td>5.25</td></tr><tr><td>i</td><td>1</td><td>4</td><td>3</td><td>0</td><td>2</td></tr></table> <p>All int values are promoted to double.</p>	index	0	1	2	3	4	d	4.0	1.5	8.0	0.0	5.25	i	1	4	3	0	2									
index	0	1	2	3	4																								
d	4.0	1.5	8.0	0.0	5.25																								
i	1	4	3	0	2																								
11.	C	The file pointer is moved with every call to next() so every other word is printed until "rabbit" is encountered.																											
12.	D	<table><tr><td>d</td><td>e</td></tr><tr><td></td><td>0.0</td></tr><tr><td>3.5</td><td>3.5</td></tr><tr><td>4.25</td><td>7.75</td></tr><tr><td>5.0</td><td>12.75</td></tr><tr><td>5.75</td><td>18.5</td></tr><tr><td>6.5</td><td></td></tr></table>	d	e		0.0	3.5	3.5	4.25	7.75	5.0	12.75	5.75	18.5	6.5														
d	e																												
	0.0																												
3.5	3.5																												
4.25	7.75																												
5.0	12.75																												
5.75	18.5																												
6.5																													
13.	D	$3\&2+2 3 = 3\&4 3 = 0 3 = 3$ <table><tr><td></td><td>011</td></tr><tr><td>and</td><td>100</td></tr><tr><td></td><td>000</td></tr><tr><td>or</td><td>011</td></tr><tr><td></td><td>011</td></tr></table>		011	and	100		000	or	011		011																	
	011																												
and	100																												
	000																												
or	011																												
	011																												
14.	B	Short.BYTES = 2 and Byte.BYTES = 1. Math.max(2,1) is 2																											
15.	E	list.remove(2) removes the element at index 2. In this case "3". list.remove("1") removes the String "1".																											
16.	B	String s is a class field and is initialized to null then printed before being assigned a value in the parameter.																											
17.	A	<p>The regular expression in the second line will match any string that contains any character exactly 3 times followed by a dash then any word character one or more times followed by a dash then any non-whitespace character one or more times.</p> <p>In the third line matches() will return true if str contains 325 followed by any character, 978 followed by any character then 1400.</p> <p>The fourth line will match any character zero or more times.</p>																											
18.	C	super() is a call to the parent class constructor. Birthdate extends Object by default.																											
19.	E	Methods must return a value matching the return type of the method.																											
20.	D	The reserved word this designates the fields within the object rather than the parameters of the same name.																											
21.	B	<p>A. Does not return a value.</p> <p>C. Incorrect logic. Subtraction is reversed.</p> <p>D. Makes references to private fields.</p>																											

22.	E	A. Incorrect call to Birthdate constructor. B. No type for the myFriend object. C. Incorrect call to the Friend constructor. D. Missing key word new for the call to the Birthdate constructor.																
23.	C	Sorts each column in ascending order.																
24.	C	Choice C is an implementation of a selection sort.																
25.	B	If the array is already in ascending order that would be the best case scenario for an insertion sort.																
26.	D	isEmpty() returns true as long as the stack contains any elements.																
27.	A	push socks	socks															
		push shoes	shoes	socks														
		pop	socks															
		push shirt	shirt	socks														
		push pants	pants	shirt	socks													
		pop	shirt	socks														
		push socks	socks	shirt	socks													
		push coat	coat	socks	shirt	socks												
		pop	socks	shirt	socks													
28.	A	5+4+3+2+1+5=20																
29.	E	Integer.toString(x,y) returns x shown in base y. In this case 16 as a binary number. $16_{10} = 10000_2$ .																
30.	B	The method mtd returns the variable e. e is of type double therefore the return type of the method must also be double.																
31.	D	i=3 j=6 e=4.0 i=4 j=6 e=24.0 i=5 j=6 e=29.0																
32.	E	Object is the cosmic super class! Any type data including primitives can be assigned to a variable of type Object. Primitive data types are autoboxed into wrapper class objects.																
33.	C	x	1	2		3		4	5		6		7		8	9		10
		t	0	1	4	7	10		12	15	21	24	26	29		38	41	43
34.	A	Must be static to facilitate a call to a method using the class name i.e. Math.max(1,2). Some but not all return double.																
35.	E	z is decremented to 0 before the division is performed in the expression $x > y / --z$ . The code segment will compile but a division by 0 exception is thrown when it is executed.																
36.	D	Primitive data types cannot serve as parameter types for generic classes.																
37.	B	51 decimal is 00110011 binary. Take the complement to get 11001100. Add one and the result is 11001101.																
38.	B	This expression simplifies to XNOR. $!(True \wedge False) = !True = False$																
39.	5	Each row in the adjacency matrix corresponds to a vertex.																
40.	!(A&&B)^C	Also accept $!(A \& B)^C$ . Do not accept if written using generic notation.																

# UIL COMPUTER SCIENCE WRITTEN TEST

**Questions** (+6 points for each correct answer, -2 points for each incorrect answer)

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

## FOR ADMINISTRATIVE USE ONLY

# Right:	×	6 pts	=	
# Wrong:	×	-2 pts	=	
# Skipped:	×	0 pts	=	0

	Score	Initials
Judge #1:	<input type="text"/>	<input type="text"/>
Judge #2:	<input type="text"/>	<input type="text"/>
Judge #3:	<input type="text"/>	<input type="text"/>