# UIL COMPUTER SCIENCE WRITTEN TEST

# 2021 DISTRICT

## MARCH 2021

## 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)
```

**Package `java.util.function`**

**Interface BiConsumer<T,U>**
  void **accept**(T t, U u)

**Interface BiFunction<T,U,R>**
  R **apply**(T t, U u)

**Interface BiPredicate<T,U>**
  boolean **test**(T t, U u)

**Interface Consumer<T>**
  void **accept**(T t)

**Interface Function<T,R>**
  R **apply**(T t)

**Interface Predicate<T>**
  boolean **test**(T t)

**Interface Supplier<T>**
  T **get**()

**Note:** Correct responses are based on **Java SE Development Kit 14 (JDK 14)** 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 14 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 NOT equal to $AC_{16}$ - $81_{16}$?

**A)** $3B_{16}$      **B)** $43_{10}$      **C)** $101011_2$      **D)** $53_8$      **E)** All are equal.

---

**Question 2.**

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

**A)** 21    **B)** 43    **C)** 52    **D)** 67    **E)** 42

```java
out.print(80 - -12 - 50 / 2);
```

---

**Question 3.**

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

**A)** `InvAInvB`
`District`

**B)** `InvA`
`InvBDistrict`

**C)** `InvA`
`InvB`
`District`

**D)** `InvA`
`InvB`

`District`

**E)** `InvAInvBDistrict`

```java
out.println("InvA");
out.print("InvB");
out.println("District");
```

---

**Question 4.**

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

**A)** `true`      **B)** `3`      **C)** `6`

**D)** `false`      **E)** `32`

```java
out.print("greenday".compareTo("Greenapple"));
```

---

**Question 5.**

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

**A)** `true`

**B)** `false`

```java
boolean yes = true;
boolean no = false;
boolean maybe = true;
out.print(!yes && !no ^ maybe);
```

---

**Question 6.**

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

**A)** 3    **B)** 3.0    **C)** 4.0    **D)** 4    **E)** 3.1

```java
out.print(Math.ceil(Math.PI));
```

---

**Question 7.**

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

**A)** 152.5    **B)** 57.5    **C)** -477.5    **D)** 477.5

**E)** There is no output due to an error.

```java
int i = -5;
double d = 9.5;
char c = 'i';
out.print(c - d * i);
```

---

**Question 8.**

Which of the following can replace **<code>** in the code segment shown on the right and cause the output to be "Option 1"?

**A)** `x < 30 || x % 2 == 0`

**B)** `x > 30 && x % 2 != 0`

**C)** `!(x < 30) && !(x % 2 == 0)`

**D)** `x > 30 || x % 2 != 0`

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

```
int x = 50;
if(<code>)
    out.print("Option 1");
else
    out.print("Option 2");
```

**Question 9.**

How many @'s are printed by the code shown to the right?

**A)** 55    **B)** 56    **C)** 18    **D)** 19    **E)** 20

```
for(int x = 93; x > 38; x -= 3)
    out.print("@");
```

**Question 10.**

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

**A)** `[4, 1, 2, 6, 4, 5]`

**B)** `[4, 5, 2, 6, 4, 5]`

**C)** `[5, 1, 2, 6, 4, 5]`

**D)** `[5, 1, 6, 4, 0, 5]`

**E)** There is no output due to an error.

```
int []nums = {4,1,2,3,0,5};
nums[3] = 6;
nums[4] = nums[0];
nums[0] = nums[nums[nums.length - 1]];
out.print(Arrays.toString(nums));
```

```
    public static void main(String[] args) throws IOException
    {
        Scanner f = new Scanner(new File("data.dat"));
        String s = "";
        int t = 0;
        while(f.hasNext())
            {
                s += f.next();
                t++;
            }
        out.print(t);
        f.close();
    }
```

**Question 11.**

Consider the main method shown above. What is the output if the file `data.dat` contains: ***dog,cats,bird,turtle***? You may assume that all necessary classes have been imported.

**A)** 1

**B)** 0

**C)** 4

**D)** 7

**E)** There is no output due to an error.

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

A) 32 6

B) 32 5

C) 16 5

D) 16 6

E) 64 6

```
int x, y = 32, z = 0;
for(x = 1; x < y; x += x)
    z++;
out.print(x + " " + z);
```

What is the correct order of operations for the operators listed on the right?

A) I II III

B) II III I

C) III II I

D) II I III

E) III I II

I. >> right shift

II. % remainder

III. ~ bitwise not

Which of the following represents the output of the code segment shown on the right?

A)  1

B)  8

C)  16

D)  127

E)  128

```
int b = Math.max(Byte.BYTES, Byte.SIZE);
int c = Math.max(b, Byte.MAX_VALUE);
out.println(c);
```

```
int[] i = {4,9,7,2,8};
ArrayList<Integer> nums = new
ArrayList<Integer>();
for(int x:i) nums.add(x);
nums.add(2);
nums.add(2, 5);
nums.set(1, 6);
nums.remove(nums.size() - 2);
nums.get(1);
out.print(nums);
```

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

A) [2, 6, 4, 5, 9, 7, 8]

B) [2, 6, 5, 9, 7, 8]

C) [4, 6, 9, 5, 7, 2, 2]

D) [4, 5, 7, 2, 2]

E) [4, 6, 5, 7, 2, 2]

How many ordered pairs make this Boolean expression false?

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

$$\overline{A} + B$$

Which of the following statements will print "A"?

A) `out.print("March".length() > Math.round(Math.PI) : "A" ? "B");`

B) `out.print("March".length() > Math.round(Math.PI) ? "A" : "B");`

C) `out.print("March".length() < Math.round(Math.PI) ? "A" : "B");`

D) `out.print("A" : "B" ? "March".length() > Math.round(Math.PI));`

E) `out.print("A" ? "B" : "March".length() < Math.round(Math.PI));`

**Question 18.**

Which of the following is the *best estimate* of the output of the code segment shown on the right?

**A)** 19

**B)** 18

**C)** 10000

**D)** 9999

**E)** 1

```
Random r = new Random();
Set<Integer> s = new TreeSet<Integer>();
for(int x=1;x<=10000;x++)
    s.add((int)(r.nextDouble() * 19 + 18));
System.out.print(s.size());
```

**Question 19.**

How many instance variables have been declared within the class `District`?

**A)** 0

**B)** 2

**C)** 3

**D)** 4

**E)** 6

```
//Use the following to answer questions 19,
//20, 21 and 22.

public class District {

  private String w;
  private int x;
  public double y;
  boolean z;

  public District() {}

  public District(String w, int x, double y,
                  boolean z)
    {
     super();
     this.w = w;
     this.x = x;
     this.y = y;
     this.z = z;
    }

public String getW() {return w;}

public void setY(double value) {y = value;}

  public String toString() {
    return w + " " + x + " " + y + " " + z;
  }
}
```

**Question 20.**

Which of the following is the best description of the statement `super(); ?`

**A)** The statement is a call to the District class default constructor.

**B)** The statement enables the use of the keyword this in the assignment statements that follow.

**C)** The statement is a call to the Object class constructor.

**D)** The statement overrides the parent class constructor.

**E)** The statement is required in all constructor implementations.

**Question 21.**

Which of the following statements when found in client code outside of the District class will not compile and execute given this line of code that instantiates a District object d?

`District d = new District("hello", 8, 3.14, true);`

**A)** `System.out.print(d);`

**B)** `System.out.print(d.toString());`

**C)** `d.setY(5);`

**D)** `double y = d.y;`

**E)** `int x = d.x;`

## Question 22.

If the following client code is not within the District class, what is the output?

```
District d1 = new District();
District d2 = new
            District("hello",5,3.14,false);
out.print((d1 instanceof Object) + " ");
out.print(d2 instanceof Object);
```

**A)** true true

**B)** true false

**C)** false true

**D)** false false

**E)** There is no output due to an error.

## Question 23.

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

**A)** 54

**B)** 80

**C)** 66

**D)** 28

**E)** This segment throws an ArrayIndexOutOfBoundsException

```
int[][] mat = new int[5][5];
int m = 0;
for(int r = 0; r < mat.length; r++)
        for(int c = 0; c < mat[r].length; c++)
                mat[r][c] = r + c;
for(int r = 2; r < mat.length; r++)
        for(int c = 2; c < mat[r].length; c++)
                m += mat[r][c];
out.println(m);
```

## Question 24.

Which of the following must replace **<missing code>** to instantiate a Queue object that will store String objects?

**A)** List<>()

**B)** List<String>()

**C)** Queue<String>()

**D)** LinkedList<String>()

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

```
//Use the following code segment
//to answer questions 24, 25 and 26.

Queue<String> q = new <missing code>;
q.add("monday");
q.add("tuesday");
q.offer("wednesday");
q.add("thursday");
q.add("friday");
q.add(q.peek());
q.add(q.remove());
out.println(q);
Stack<String> s = new Stack<String>();
while(!q.isEmpty())
    s.add(q.poll());
out.println(s.peek()); //line #1
out.println(q.peek()); //line #2
```

## Question 25.

What is the output of **line #1** shown on the right?

**A)** monday

**B)** tuesday

**C)** wednesday

**D)** thursday

**E)** friday

## Question 26.

What is the output of **line #2** shown on the right?

**A)** monday

**B)** tuesday

**C)** friday

**D)** null

**E)** There is no output due to an error.

Which of the following is the decimal equivalent of the 8-bit binary two's complement number $10101111_2$?

  **A)** -25

  **B)** -81

  **C)** -103

  **D)** -175

  **E)** -38

```
//The method sort is intended to implement an ascending insertion sort. Use this code to
//answer questions 28, 29, 30 and 31.

public static void insertionSort(int[] list)
   {
   for(int x = 1; x < list.length; x++)
     {
     int y = <code 1>;
     int z = x;
     while(<code 2>)
       {
       list[z] = list[z - 1];
       z--;
      }
     list[z] = y;
     /*comment*/
     }
}
```

Which of the following must replace **<code 1>** to ensure that the method will compile, execute and sort `list` in ascending order?

  **A)** `list.length`

  **B)** `x + 1`

  **C)** `list[x]`

  **D)** `0`

  **E)** `list[0]`

Which of the following must replace **<code 2>** to ensure that the method will compile, execute and sort `list` in ascending order?

  **A)** `z > 0`

  **B)** `z > 0 && y < list[z]`

  **C)** `z < 0 && y > list[z - 1]`

  **D)** `z > 0 || y < list[z - 1]`

  **E)** `z > 0 && y < list[z - 1]`

**Question 30.**

Given the client code shown on the right and assume that **<code 1>** and **<code 2>** have been filled in correctly, which of the following is the order of the elements in list when x is equal to 4 and execution has reached the **/*comment*/**?

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

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

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

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

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

```
int[] list = {6,7,1,8,2,9,0,5,4,3};
insertionSort(list);
```

**Question 31.**

If it takes `insertionSort` 2 seconds to sort 3 million elements, what is the best estimate of how long will it take to sort 9 million elements?

A) 18 seconds

B) 9 seconds

C) 3 seconds

D) 6 seconds

E) 36 seconds

**Question 32.**

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

A) abc8defghijkl4m

B) abcdefghijklm

C) 84

D) abc8de5fghi1j2kl4m

E) abcde5fghij2klm

```
Scanner s = new
Scanner("abc8de5fghi1j2kl4m");
s.useDelimiter("[\\d && [^25]]");
while(s.hasNext())
    out.print(s.next());
```

**Question 33.**

Which of the answer choices must replace **<missing code>** in the code segment shown here to ensure that the code segment will compile, run and print "Fail"?

```
BiPredicate<Boolean,Boolean> x = (e,f) -> !(e || f);
if(<missing code>)
   System.out.print("Pass");
else
   System.out.print("Fail");
```

A) test(true, false)

B) BiPredicate.test(true, false)

C) x.test()

D) x.test(true, false)

E) x.test(boolean e, boolean f)

```java
//Use class UILString to answer questions
// 34 and 35

public class UILString implements Comparable/*missing code*/{

     private String str;

       public UILString(String str){
         this.str = str;
       }

       public int compareTo(UILString uil) {
         String s = uil.toString();
         String y,z;
         boolean x = true;
         if(s.length() > str.length()) {
           y = s;
           z = str;
           x = false;
         }
         else {
           y = str;
           z = s;
         }
         int d = y.length() - z.length();
         for(int i = y.length() - 1; i >= d; i--)
           if(y.charAt(i) != z.charAt(i - d))
                 if(x)
                         return y.charAt(i) - z.charAt(i - d);
                 else
                         return z.charAt(i - d) - y.charAt(i);
         if(d == 0) return 0;
         return str.length() < s.length() ? -1 : 1;
         }

     public String toString() {
           return str;
     }
}
```

**Question 34.**

Which of the following may replace **/*missing code*/** in the class shown above?

**A)** <>

**B)** <Object>

**C)** <UILString>

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

**E)** No additional code is required.

**Question 35.**

Assume that **/*code*/** has been filled in correctly. What is the output of the client code shown on the right?

**A)** [aaob, aob, bob, rma, tat, zma]

**B)** [rma, zma, aob, aaob, bob, tat]

**C)** [zma, rma, aob, aaob, bob, tat]

**D)** [zma, rma, aaob, aob, bob, tat]

**E)** [aob, aaob, bob, rma, tat, zma]

```java
UILString[] list = new UILString[6];
list[0] = new UILString("rma");
list[1] = new UILString("tat");
list[2] = new UILString("zma");
list[3] = new UILString("bob");
list[4] = new UILString("aob");
list[5] = new UILString("aaob");
Arrays.sort(list);
System.out.print(Arrays.toString(list));
```

**Question 36.**

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

A) 35487

B) 35487.0

C) 35,487.0

D) There is no output because the segment will not compile.

E) There is no output because the segment throws a NumberFormatException.

```
String s = "35487";
out.print(Double.parseDouble(s));
```

**Question 37.**

If an F is inserted into the binary search tree shown on the right, which of the following best describes where it will be placed?

A) As the root node

B) As the right node of D

C) As the right node of B

D) As the left node of N

E) As the right node of T
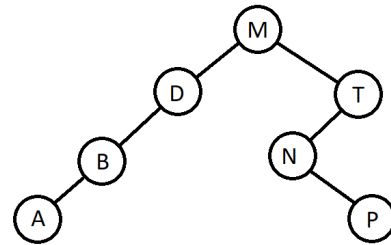


**Question 38.**

Which of the following would be returned by this call to the method f?

        f(462,1071)

A) 231

B) 3

C) 21

D) 2

E) 48

```
public static int f(int a,int b) {
  if(b == 0)
    return a;
  else
    return f(b, a % b);
}
```

**Question 39.**

What would be returned by this call to method.

        method(8);

Write your answer in the blank provided on the answer document.

```
public static int method(int n) {
  int x = 1, y = 1, z;
  if(n == 1 || n == 2)
    return 1;
  else
    for(int r = 3; r <= n; r++ ) {
      z = x + y;
      x = y;
      y = z;
    }
  return y;
}
```

**Question 40.**

If a binary tree contains 255 nodes, what is the least number of levels (including the root) that tree might have? Write your answer in the blank provided on the answer document.

Conference_____  Contestant Number_____

# 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:** |  |  |
| **Judge #2:** |  |  |
| **Judge #3:** |  |  |