

UIL COMPUTER SCIENCE WRITTEN TEST

2021 STATE

APRIL/MAY 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)
```

STANDARD CLASSES AND INTERFACES – SUPPLEMENTAL REFERENCE

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()
```

UIL COMPUTER SCIENCE WRITTEN TEST – 2021 STATE

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.*;`**

<p>Question 1.</p> <p>What is the product of 1011_2 and 1100_2?</p> <p>A) 17_{16} B) AC_{16} C) 132_{16} D) 84_{16} E) $A4_{16}$</p>	
<p>Question 2.</p> <p>What is the output of the statement shown on the right?</p> <p>A) 19 B) 20 C) 24 D) 13 E) -20</p>	<pre>out.print(8 + 5 - -4 / 2 * 3);</pre>
<p>Question 3.</p> <p>What is the output of the line of code shown on the right? <i>Asterisks indicate spaces.</i></p> <p>A) hello*****goodbye*** B) *****hello***goodbye C) *****goodbye D) hellogoodbye*** E) There is no output due to an error.</p>	<pre>out.printf("%10.0s%10s", "hello", "goodbye");</pre>
<p>Question 4.</p> <p>What is the output of the code segment to the right?</p> <p>A) ZustinSZnZntonioZoo B) ZustinSanAntonioZoo C) ZustinSanZntonioZoo D) AustinSanAntonioaoo E) AustinSZnAntonioZoo</p>	<pre>String str = "AustinSanAntonioZoo"; out.print(str.replace('a', 'Z'));</pre>
<p>Question 5.</p> <p>Which of the following is the output of the code segment shown on the right?</p> <p>A) true B) false</p>	<pre>boolean a = true, b = false; boolean c = true, d = false; boolean e = !(a && d) !(c ^ b); out.print(e);</pre>
<p>Question 6.</p> <p>What is the output of the code segment to the right?</p> <p>A) -2.0 B) 2.0 C) -3.0 D) 3.0 E) There is no output due to an error.</p>	<pre>out.print(Math.floor(-Math.cbrt(-9)));</pre>
<p>Question 7.</p> <p>What is the output of the code segment to the right?</p> <p>A) 44.5 B) 44 C) 45.5 D) 45.0 E) 44.0</p>	<pre>int m = 35, n = 18; double x = 19, y = 9.25; double r = m + x / n * y; r = (int)(r + 0.5); out.print(r);</pre>

<p>Question 8.</p> <p>What is the output of the code segment to the right?</p> <p>A) 19 B) 4 C) 17 D) 18 E) 26</p>	<pre>int x = 2147483647; int y = 0; while(x > 0) { switch(x % 10) { case 1: case 2: case 3: y++;break; case 4: case 5: case 6: y+=3;break; case 7: y*=2;break; default: y/=2; } x/=10; } out.print(y);</pre>
<p>Question 9.</p> <p>Which of the following code segments will print exactly 10 question marks?</p>	
<p>A.</p> <pre>for(int x = 1; x < 10; x++) out.print("?");</pre>	<p>B.</p> <pre>int e = 10; while(e <= 100) { out.print("?"); e++; }</pre>
<p>C.</p> <pre>int e = 0; while(e <= 100) { out.print("?"); e = e + 10; }</pre>	<p>D.</p> <pre>for(int r = 50; r >= 10; r-=5) out.print("?");</pre>
<p>E.</p> <pre>int x = 0; do { out.print("?"); x++; }while(x < 10);</pre>	
<p>Question 10.</p> <p>What is the output of the code segment to the right?</p> <p>A) secneicsretupmoc B) sciencescomputer C) csocmipeuntceers D) retupmocsecneics E) There is no output due to an error.</p>	<pre>String s = "computersciences"; char[] c = s.toCharArray(); int a = c.length / 2; for(int i = 0; i < a; i++) { char b = c[i]; c[i] = c[i + a]; c[i + a] = b; } for(char d:c) out.print(d);</pre>

<p>Question 11.</p> <p>Which of the following throws declarations is <u>required</u> for the main method to ensure that the class shown on the right will compile and execute as intended? <i>Assume that all necessary import statements are present.</i></p> <ul style="list-style-type: none"> A) throws FileNotFoundException B) throws Throwable C) throws IOException D) throws Exception E) This class does not require a throws declaration. 	<pre>public class Q11 { public static void main(String[] args) { File file = new File("data.dat"); try { Scanner scr = new Scanner(file); while(scr.hasNext()) out.print(scr.next()); scr.close(); } catch (FileNotFoundException e) { e.printStackTrace(); } } }</pre>
<p>Question 12.</p> <p>What is the output of the code segment to the right?</p> <ul style="list-style-type: none"> A) 2.5 B) 2.0 C) 3.0 D) 6.0 E) There is no output due to an error. 	<pre>double x = 5.0, y = 0.0; for(;x < 6.0;) { y += 0.5; x += 0.2; } out.print(y);</pre>
<p>Question 13.</p> <p>Which of the following represents the output of the line of code shown on the right?</p> <ul style="list-style-type: none"> A) 6 B) 8 C) 3 D) 2 E) No output. Throws an ArithmeticException. 	<pre>out.print(18 / 3 & 4 + 5 << 1);</pre>
<p>Question 14.</p> <p>Which of the following statements is true about the code segment shown on the right?</p> <ul style="list-style-type: none"> A) The coded segment will compile and execute and will print zero. B) The code segment will compile and execute but will print a non-zero value that is not the product of a and b. C) The code segment will compile and execute and will print the product of a and b. D) The code segment will compile, execute and print the product of a and b only if the cast to byte is removed. E) The code segment will not compile and execute as shown. 	<pre>byte a = 15; byte b = 10; byte c = (byte) (a * b); out.print(c);</pre>

<p>Question 15.</p> <p>What is the output of the code segment to the right?</p> <p>A) [orange, purple, yellow] B) [purple, yellow] C) [blue, red, purple, yellow] D) [red, purple, yellow] E) [blue, purple, yellow]</p>	<pre>ArrayList<String> list = new ArrayList<String>(); list.add("blue");list.add("green"); list.add("red");list.add("orange"); list.add("purple");list.add("yellow"); int x = 1; while(list.contains("orange")) { list.remove(x); x++; } out.print(list);</pre>
<p>Question 16.</p> <p>Which of the following must replace <code> in the method sum shown on the right?</p> <p>A) int...nums B) int[] nums C) int nums D) int:nums E) int nums{}</p>	<pre>public static void main(String[] args) { out.println(sum(5,2,4,7,3)); } public static int sum(<code>) { int t = 0; for(int x:nums) t+=x; return t; }</pre>
<p>Question 17.</p> <p>Shown on the right is an array in its original state and in its sorted state after a radix sort has been applied to the array. The algorithm uses base 10 and starts with the least significant digit. The state of the array after each pass of the algorithm is not shown. Which of the following represents the state of the array after the pass marked by question marks and the //comment?</p> <p>A) 7, 158, 3, 215, 1, 20, 45, 1254, 2111 B) 1, 3, 7, 2111, 158, 215, 20, 45, 1254 C) 1, 3, 158, 2111, 215, 7, 20, 45, 1254 D) 1, 1254, 158, 215, 2111, 20, 3, 7, 45, E) 1, 3, 7, 2111, 215, 20, 45, 1254, 158</p>	<pre>[7, 2111, 158, 3, 215, 1, 20, 45, 1254] [/ / / / / / / /] [?, ?, ?, ?, ?, ?, ?, ?]//comment [/ / / / / / / /] [1, 3, 7, 20, 45, 158, 215, 1254, 2111]</pre>

Question 18.

Which of the following methods returns a new 2D array that has all the original values in `m` and an additional row that contains the total of each of the columns in the original array `m`? `m` may or may not have the same number of rows as it does columns, however, each of the rows in `m` will always contain the same number of elements.

A.

```
public static int[][] method(int[][] m){
    int[][] t = new int[m.length + 1][m[0].length];
    int[] s = new int[m[0].length];
    for(int i = 0; i < m[i].length; i++) {
        int p = 0;
        for(int j = 0; j < m.length; j++)
            p += m[j][i];
        s[i] += p;
    }
    t[t.length - 1] = s;
    return t;
}
```

B.

```
public static int[][] method(int[][] m){
    int[][] t = new int[m.length + 1][m[0].length];
    int[] s = new int[m[0].length];
    for(int i = 0; i < m[0].length; i++) {
        int p = 0;
        for(int j = 0; j < m.length; j++)
            p += m[j][i];
        s[i] += p;
    }
    t[t.length - 1] = s;
    return t;
}
```

C.

```
public static int[][] method(int[][] m){
    int[][] t = Arrays.copyOf(m, m.length + 1);
    int[] s = new int[m[0].length];
    for(int i = 0; i < m[0].length; i++)
        for(int j = 0; j < m.length; j++)
            s[i] += m[j][i];
    t[t.length - 1] = s;
    return t;
}
```

D.

```
public static int[][] method(int[][] m){
    int[][] t = Arrays.copyOf(m, m.length + 1);
    int[] s = new int[m.length];
    for(int i = 0; i < m.length; i++) {
        int p = 0;
        for(int j = 0; j < m.length; j++)
            p += m[j][i];
        s[i] += p;
    }
    t[t.length - 1] = s;
    return t;
}
```

E.

```
public static int[][] method(int[][] m){
    int[][] t = Arrays.copyOf(m, m.length + 1);
    int[] s = new int[m[0].length];
    for(int i = 0; i < m[0].length; i++) {
        int p = 0;
        for(int j = 0; j < m.length; j++)
            p += m[i][j];
        s[i] += p;
    }
    t[t.length - 1] = s;
    return t;
}
```

Question 19.

Which of the following statements is NOT true about a protected data member of a class?

- A)** It may be accessed from within the class.
- B)** It may be accessed from within the package that contains the class.
- C)** It may be accessed from a subclass of the class in a different package.
- D)** It may be accessed from a different class within a different package.
- E)** None of the above. All of the statements are true about protected data members.

<p>Question 20.</p> <p>Which of the following must replace <code 1> in interface B shown on the right?</p> <ul style="list-style-type: none"> A) class B) interface C) implements D) extends E) super 	<pre>public interface A { public String cats(); }</pre>
<p>Question 21.</p> <p>Which of the following can replace <code 2> in class C shown on the right?</p> <ul style="list-style-type: none"> A) implements A implements B B) implements A extends B C) implements A, B D) extends A implements B E) extends A extends B 	<pre>public interface B <code 1> A { public String cats(int i, String s); public double dogs(double a, double b); } public class C <code 2>{ private int m; public C(int m) { this.m = m; } public String cats(int i, String s) { return s.substring(i); } public double dogs(double a, double b) { return a + b + m; } public String cats() { return "Cats!"; } public static double birds(int x) { return Math.pow(x, 2); } public boolean equals(Object obj) { <missing code> return this.m < o.m; } }</pre>
<p>Question 22.</p> <p>Which of the following is the only line of code that can replace <missing code> in the equals method in class C?</p> <ul style="list-style-type: none"> A) C o = (C) obj; B) C o = obj; C) A o = (C) obj; D) C o = (B) obj; E) Object o = obj; 	
<p>Question 23.</p> <p>Consider the client code shown below. Which line contains an error if all missing code within A, B and C has been filled in correctly?</p> <pre>A a = new C(1); B b = new C(2); C c = new C(3); out.println(a.cats());//line #1 out.println(b.cats(3, "turtles"));//line #2 out.println(c.birds(3));//line #3 out.println(a.cats(2, "snakes"));//line #4 out.println(b.dogs(3.5, 2.75));//line #5</pre> <ul style="list-style-type: none"> A) line #1 B) line #2 C) line #3 D) line #4 E) line #5 	

Question 24.

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

- A) $A \ \&\& \ !(B \ \&\& \ !C)$
- B) $A \ ^ \ !(B \ \&\& \ !C)$
- C) $A \ ^ \ (!B \ || \ C)$
- D) $!(A \ || \ B \ \&\& \ C)$
- E) More than one of the above

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

Question 25.

```
public static double method(double x) {
    double m = x / 2;
    double n = 0.0;
    while(true) {
        n = (m + x / m) / 2.0;
        if((int)(m * 1000 + 0.5) / 1000.0 == (int)(n * 1000 + 0.5) / 1000.0)
            break;
        m = n;
    }
    return (int)(n * 100 + 0.5) / 100.0;
}
```

Which of the following best describes what the method shown above will return?

- A) $\log_{10}(x)$ rounded to four decimal places.
- B) \sqrt{x} rounded to two decimal places.
- C) $\log_2(x)$ rounded to two decimal places.
- D) x^2 rounded to four decimal places.
- E) The average of the values between 0.0 and x rounded to two decimal places.

Question 26.

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

- A) 225
- B) 150
- C) 300
- D) 1
- E) 90

```
int r = 0;
for(int x = 1; x <= 30; x += 2) {
    if(x % 3 == 0)
        continue;
    r += x;
}
out.print(r);
```

Question 27.

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

- A) 70
- B) 280
- C) 440
- D) 350
- E) 4

```
int count = 0;
int x = 1;
do {
    int y = 1;
    while(y <= 7) {
        for(int z = 1; z <= 4; z++)
            count++;
        y++;
    }
    x++;
}while(x <= 10);
out.print(count);
```

<p>Question 28.</p> <p>Which of the following must replace <code> in the code segment shown on the right?</p> <p>A) <code>m.keySet()</code> B) <code>m.values()</code> C) <code>m.entrySet()</code> D) <code>m.size()</code> E) <code>m.hashCode()</code></p>	<pre>Map<Character,Integer> m = new TreeMap<>(); String s = "acclimatization"; for(int i = 0; i < s.length(); i++) m.put(s.charAt(i), i); Set<Character> z = <code>; for(char i:z) out.print(m.get(i) + " ");</pre>
<p>Question 29.</p> <p>Once <code> has been filled in correctly, what is the output of the code segment shown on the right?</p> <p>A) a c l i m t z o n B) a c i l m n o t z C) 10 2 12 3 5 14 13 11 9 D) a a a c c i i i l m n o t t z E) 0 1 4 3 5 14 13 7 9</p>	<pre>public static void main(String[] args) { int[] i = {0,1,2,3,4}; out.println(Arrays.toString(method(i))); out.println(Arrays.toString(i)); } public static int[] method(int[] j) { int[] k = {1,3,5,7,9}; Arrays.fill(j, 1, 2, 5); int[] m = j; j = k; k = m; return k; }</pre>
<p>Question 30.</p> <p>What is the output of the main method shown on the right?</p> <p>A) [1, 3, 5, 7, 9] [5, 5, 2, 3, 4] B) [1, 3, 5, 7, 9] [0, 1, 2, 3, 4] C) [0, 5, 2, 3, 4] [0, 1, 2, 3, 4] D) [0, 5, 2, 3, 4] [0, 5, 2, 3, 4] E) [1, 2, 5, 3, 4] [1, 2, 5, 3, 4]</p>	<pre>public static void main(String[] args) { int[] i = {0,1,2,3,4}; out.println(Arrays.toString(method(i))); out.println(Arrays.toString(i)); } public static int[] method(int[] j) { int[] k = {1,3,5,7,9}; Arrays.fill(j, 1, 2, 5); int[] m = j; j = k; k = m; return k; }</pre>
<p>Question 31.</p> <p>Which of the following represents the output of the code segment shown on the right?</p> <p>A) coencter B) scputce C) coenceer D) sciputerence E) coencmputer</p>	<pre>char[] c = {'c','o','m','p','u','t','e','r'}; "science".getChars(3, 6, c, 2); for(char ch:c) out.print(ch);</pre>
<p>Question 32.</p> <p>What is the output of the code segment shown on the right?</p> <p>A) 100.0 B) 10.0 C) 5.0 D) 12.0 E) 2.0</p>	<pre>double x = 6.0,y = 8.0; double z = Math.hypot(x, y); out.print(z);</pre>

```

//Use the following interface and classes to answer question #33
public interface Checker {
    boolean check(State s);
}
public class State {

    private String number;
    private String color;
    private int value;

    public State(String number, String color, int value) {
        this.number = number;
        this.color = color;
        this.value = value;
    }
    public String getName() {return number;}
    public String getColor() {return color;}
    public int getWeight() {return value;}

    public String toString() {return number + " " + color + " " + value;}

    public void print(State st, Checker si) {
        if(si.check(st))
            System.out.println(this.toString());
    }
}

import java.util.*;
public class State_Tester {

    public static void main(String[] args) {
        ArrayList<State> list = new ArrayList<>();
        list.add(new State("one","blue",35));
        list.add(new State("two","red",5));
        list.add(new State("three","green",35));
        list.add(new State("four","blue",15));
        list.add(new State("five","red",35));
        list.add(new State("six","green",25));
        for(State s:list) {
/*1*/ s.print(s, new Checkit());
/*2*/ s.print(s,new Checker() {public boolean check(State s)
                                {return s.getColor().equals("blue")&& s.getWeight() < 35;}});
/*3*/ s.print(s, (State st) -> st.getColor().equals("blue") && st.getWeight() < 35);
/*4*/ s.print(s, st -> {return st.getColor().equals("blue")
                        && st.getWeight() < 35;});
        }
    }

    public static class Checkit implements Checker{
        public boolean check(State s) {
            return s.getColor().equals("blue") && s.getWeight() < 35;
        }
    }
}

```

Question 33.

The main method in the class State_Tester shown above should print the State object that is blue and has a value less than 35 four different times. Which of the lines marked with comments will NOT compile and execute as intended?

- A) /*line 1*/ B) /*line 2*/ C) /*line 3*/ D) /*line 4*/
E) None of the above. Each of the lines will compile and execute as intended.

//Use the classes Toy and ToyBox to answer questions 34, 35, 36 and 37.

```
public class Toy {
    public String toy;
    public Toy nextToy;

    public Toy(String t) {
        toy = t;
        nextToy = null;
    }
}
```

```
public class ToyBox {

    private Toy top, bottom;

    public ToyBox() {
        top = null;
        bottom = null;
    }

    public void add(String toy) {
        Toy t = new Toy(toy);
        if(top == null) {
            top = t;
            bottom = t;
        }
        else {
            <code 1>
            <code 2>
        }
    }

    public Toy remove(String toy) {
        Toy t = top;
        Toy p = top;
        if(top.toy.equals(toy)) {
            Toy d = top;
            top = t.nextToy;
            bottom = null;
            d.nextToy = null;
            return d;
        }
        while(t.nextToy != null) {
            if(t.toy.equals(toy)) {
                p.nextToy = t.nextToy;
                t.nextToy = null;
                return t;
            }
            <missing code>
            p = t;
            t = t.nextToy;
        }
        return null;
    }
}
```

Question 34.

The add method within the ToyBox class shown above should add a toy as the last element within a ToyBox object. Which of the following should replace <code 1> and <code 2> ?

- A) bottom.nextToy = t; t = bottom;
- B) bottom.toy = toy; bottom = t;
- C) bottom = t; bottom.nextToy = t;
- D) bottom.nextToy = t; bottom = t;
- E) t = bottom.nextToy; bottom = null;

Question 35.

The `remove` method should remove and return the element whose `toy` field matches the `toy` formal parameter or return `null` if the element is not present. The `<missing code>` should handle removal if the last element is the one to be removed. Which of the following code segments can replace `<missing code>`?

A) <pre>if(t == bottom) if(bottom.toy.equals(toy)) { Toy d = bottom; bottom = t; bottom.nextToy = null; return d; }</pre>	B) <pre>if(bottom.toy.equals(toy)) if(t.nextToy == bottom){ Toy d = bottom; bottom = t; bottom.nextToy = null; return d; }</pre>	C) <pre>if(t.nextToy == bottom) if(bottom.toy.equals(toy)) { Toy d = bottom; bottom = t; bottom.nextToy = null; return d; }</pre>
D) <pre>if(t.nextToy == bottom) if(bottom.toy.equals(toy)) { Toy d = bottom; bottom = t; bottom.nextToy = d; return d; }</pre>	E) <pre>if(t.nextToy == bottom) if(bottom.toy.equals(toy)) { bottom = t; bottom.nextToy = null; return bottom; }</pre>	

Question 36.

Which of the following `toString` methods can be added to the `ToyBox` class so that the string that is returned contains a list of all the elements in the object separated by a comma and a space except there should be no comma after the last element?

A. <pre>public String toString() { String s = ""; Toy t = top; while(t != null) { if(t.nextToy != null) s += t.toy + ", "; else s += t.toy; t = t.nextToy; } return s; }</pre>	B. <pre>public String toString() { String s = ""; Toy t = top; while(t != null) { if(t.nextToy != null) s += t.toy + ", "; else s += t.toy; } return s; }</pre>	C. <pre>public String toString() { String s = ""; Toy t = top; while(t != null) { if(t.nextToy.nextToy != null) s += t.toy + ", "; else s += t.toy; t = t.nextToy; } return s; }</pre>
D. <pre>public String toString() { String s = ""; Toy t = top; while(t.nextToy != null) { if(t.nextToy != null) s += t.toy + ", "; else s += t.toy; t = t.nextToy; } return s; }</pre>	E. <pre>public String toString() { String s = ""; Toy t = top; while(t != null) { if(t.nextToy != null) s += t.toy + ", "; else s += t.toy; t = t.nextToy; } return t; }</pre>	

Question 37.

Once completely and correctly implemented which of the following data structures is implemented by the class `ToyBox`?

- A) singly linked list
- B) doubly linked list
- C) stack
- D) queue
- E) set

Question 38.

Which of the following represents the simplest form of the Boolean expression shown on the right?

- A) $B * A + C$
- B) $A + B + C$
- C) $A + B * A + C$
- D) $A * B + A * C$
- E) $B + A * C$

$$A * B + A * (B + C) + B * (B + C)$$

Question 39.

Determine the output of the main method shown on the right and write your answer in the blank provided on the answer document.

```
public static void main(String[] args) {
    out.print(method1("193"));
}

public static int method1(String s) {
    int x = 0;
    for(int i = 0; i < s.length(); i++) {
        char c = s.charAt(i);
        x = x * 16 + method2(c);
    }
    return x;
}

public static int method2(char c) {
    if(c >= 65 && c <= 70)
        return 10 + c - 65;
    else
        return c - 48;
}
```

Question 40.

Determine the output of the main method shown on the right and write your answer in the blank provided on the answer document.

```
public static void main(String[] args)
{
    out.print(method(416,23));
}

public static int method(int i, int j)
{
    if(i < j)
        return 0;
    else
        return 1 + method(i - j, j);
}
```

★ ANSWER KEY – CONFIDENTIAL ★

UIL COMPUTER SCIENCE – 2021 STATE

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

- | | | | |
|------------------|------------------|------------------|---------------------|
| 1) <u> D </u> | 11) <u> E </u> | 21) <u> C </u> | 31) <u> A </u> |
| 2) <u> A </u> | 12) <u> A </u> | 22) <u> A </u> | 32) <u> B </u> |
| 3) <u> C </u> | 13) <u> D </u> | 23) <u> D </u> | 33) <u> E </u> |
| 4) <u> E </u> | 14) <u> B </u> | 24) <u> E </u> | 34) <u> D </u> |
| 5) <u> A </u> | 15) <u> C </u> | 25) <u> B </u> | 35) <u> C </u> |
| 6) <u> B </u> | 16) <u> A </u> | 26) <u> B </u> | 36) <u> A </u> |
| 7) <u> D </u> | 17) <u> E </u> | 27) <u> B </u> | 37) <u> A </u> |
| 8) <u> C </u> | 18) <u> C </u> | 28) <u> A </u> | 38) <u> E </u> |
| 9) <u> E </u> | 19) <u> D </u> | 29) <u> C </u> | *39) <u> 403 </u> |
| 10) <u> B </u> | 20) <u> D </u> | 30) <u> D </u> | *40) <u> 18 </u> |

KEY

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

Note: Correct responses are based on **Java SE Development Kit 14 (JDK 14)** 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 14 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used.

Explanations:

1.	D	$1011_2 * 1100_2 = 10000100_2 = 84_{16}$																																	
2.	A	$8 + 5 - -4 / 2 * 3 =$ $8 + 5 - -2 * 3 =$ $8 + 5 - -6 =$ $13 - -6 =$ $13 + 6 =$ 19																																	
3.	C	When applying a format specifier to a string, the number after the decimal is how many characters from that string are printed. %10.0s causes 10 blank spaces to be printed before the second string is printed right justified within 10 additional spaces.																																	
4.	E	Replaces all occurrences of a lowercase 'a' with a capital 'Z'.																																	
5.	A	$!(T \&\& F) \parallel !(T \wedge F) =$ $!F \parallel !T =$ $T \parallel F =$ T																																	
6.	B	The cube root of -9 is approximately -2.0800838. Take the opposite to get 2.0800838. Math.floor(2.0800838) returns 2.0.																																	
7.	D	$35 + 19.0 / 18 * 9.25 =$ $35 + 1.055 * 9.25 =$ $35 + 9.76 =$ 44.76 $(int) (44.76 + 0.5) =$ $(int) 45.26 =$ 45 r is a double so value is printed as 45.0.																																	
8.	C	<table style="border: none;"> <thead> <tr> <th style="text-align: left;">x</th> <th style="text-align: left;">x%10</th> <th style="text-align: left;">y</th> </tr> </thead> <tbody> <tr><td>2147483647</td><td>7</td><td>0</td></tr> <tr><td>214748364</td><td>4</td><td>3</td></tr> <tr><td>21474836</td><td>6</td><td>6</td></tr> <tr><td>2147483</td><td>3</td><td>7</td></tr> <tr><td>214748</td><td>8</td><td>3</td></tr> <tr><td>21474</td><td>4</td><td>6</td></tr> <tr><td>2147</td><td>7</td><td>12</td></tr> <tr><td>214</td><td>4</td><td>15</td></tr> <tr><td>21</td><td>1</td><td>16</td></tr> <tr><td>2</td><td>2</td><td>17</td></tr> </tbody> </table>	x	x%10	y	2147483647	7	0	214748364	4	3	21474836	6	6	2147483	3	7	214748	8	3	21474	4	6	2147	7	12	214	4	15	21	1	16	2	2	17
x	x%10	y																																	
2147483647	7	0																																	
214748364	4	3																																	
21474836	6	6																																	
2147483	3	7																																	
214748	8	3																																	
21474	4	6																																	
2147	7	12																																	
214	4	15																																	
21	1	16																																	
2	2	17																																	
9.	E	A. Prints 9 ?'s B. Prints 91 ?'s C. Prints 11 ?'s D Prints 9 ?'s																																	
10.	B	Switches characters in 0 and 8, 1 and 9, 2 and 10 ... 7 and 15.																																	
11.	E	When a method uses try ... catch, a throws declaration is not required.																																	
12.	A	<table style="border: none;"> <tr><td>x</td><td>y</td></tr> <tr><td>5.0</td><td>0.0</td></tr> <tr><td>5.2</td><td>0.5</td></tr> <tr><td>5.4</td><td>1.0</td></tr> <tr><td>5.6</td><td>1.5</td></tr> <tr><td>5.8</td><td>2.0</td></tr> <tr><td>6.0</td><td>2.5</td></tr> </table>	x	y	5.0	0.0	5.2	0.5	5.4	1.0	5.6	1.5	5.8	2.0	6.0	2.5																			
x	y																																		
5.0	0.0																																		
5.2	0.5																																		
5.4	1.0																																		
5.6	1.5																																		
5.8	2.0																																		
6.0	2.5																																		
13.	D	$18 / 3 \& 4 + 5 \ll 1 =$ $6 \& 9 \ll 1 =$ $6 \& 18 =$ 2																																	
14.	B	$10 * 15 = 150$. 150 is out of range for a variable that is of type byte. Casting to byte allows the value to be assigned to c but the value will be incorrect.																																	

15.	C	<p>x list</p> <p>1 [blue, green, red, orange, purple, yellow]</p> <p>2 [blue, red, orange, purple, yellow]</p> <p>3 [blue, red, purple, yellow]</p>
16.	A	The ellipsis (...) allows the argument to be passed as a comma separated list and converts it to an array.
17.	E	<p>A radix sort places each element in order based on the least significant digit moving from right to left. A radix sort is stable so elements will remain in the original order relative to one another until a digit differs.</p> <p>Here is the state of the array after each pass of the sort:</p> <p>[7, 2111, 158, 3, 215, 1, 20, 45, 1254] – original list</p> <p>[20, 2111, 1, 3, 1254, 215, 45, 7, 158]</p> <p>[1, 3, 7, 2111, 215, 20, 45, 1254, 158]</p> <p>[1, 3, 7, 20, 45, 2111, 158, 215, 1254]</p> <p>[1, 3, 7, 20, 45, 158, 215, 1254, 2111]</p>
18.	C	<p>A. The outer for loop causes the method to throw an <code>ArrayIndexOutOfBoundsException</code>.</p> <p>B. Returns the array with the sums in the bottom row but does not include the original values.</p> <p>D. Only works correctly if <code>m</code> has the same number of columns and rows. Otherwise, it will throw an <code>ArrayIndexOutOfBoundsException</code>.</p> <p>E. Throws an <code>ArrayIndexOutOfBoundsException</code> because <code>i</code> and <code>j</code> are in the wrong position within the body of the inner for loop.</p>
19.	D	If a protected data element is to be accessed from within different package, the class that accesses the protected element must be a subclass of the class that contains the protected element.
20.	D	One interface cannot implement another. They can, however, extend one another.
21.	C	Interfaces cannot be extended. The reserved word <code>implements</code> is only required once when implementing more than one interface.
22.	A	<code>obj</code> must be down cast to a <code>C</code> type object.
23.	D	<code>new C(1)</code> instantiates a <code>C</code> object and then assigns it to an <code>A</code> reference object. The two parameter <code>cats</code> method is undefined in interface <code>A</code> .
24.	E	$A \wedge !(B \wedge C)$ is equal to $A \wedge (!B \vee C)$. Both expressions will produce the truth table shown.
25.	B	<p>This method implements Heron's formula for estimating a square root. The algorithm begins by making an arbitrary guess at the square root by dividing <code>X</code> by 2. Each subsequent estimate is $Estimate_{next} = (Estimate_{previous} + X / Estimate_{previous}) / 2.0$. The while loop terminates when the next estimate and previous estimates are equal to four decimal places. The method then returns the estimate rounded to two decimal places.</p> <p>Example:</p> <pre>x = 9.0 4.5000 3.2500 3.2500 3.0100 3.0100 3.0000 3.0000 3.0000 square root = 3.00</pre>
26.	B	This code segment adds up all of the odd numbers between 1 and 30 except those that are divisible by 3.
27.	B	$4 * 7 * 10 = 280$
28.	A	<code>keyset()</code> returns the keys for this map. In this case, a set of characters.
29.	C	<p>The keys are characters from the string <code>s</code>. the values associated with each key are the index values for each character. When a duplicate key is placed in the map the existing key and value are replaced. A <code>TreeMap</code> is sorted based on keys, in this case characters. The code prints the last index value of each of the unique characters in the string in alphabetical order.</p> <p>a – 10, c – 2, i – 12, l – 3, m – 5, n – 14, o – 13, t – 11, z – 9</p>
30.	D	<code>i</code> is passed by reference. Any changes to <code>j</code> before <code>j</code> is reassigned are reflected in <code>i</code> . The <code>fill</code> method replaces 1 with 5 in <code>j</code> .

31.	A	"science".getChars(3,6,c,2) replaces the characters in c starting at index 2 with the characters from index 3 to 5 in "science".
32.	B	Math.hypot(6,8) returns the hypotenuse of a right triangle whose legs are 6 and 8.
33.	E	A. Instantiates a new Checkit object which implements the Checker interface. print is looking for an object that implements the check method and Checkit objects do implement the check method. B. This call to the print method passes an anonymous class which implements the necessary code. That must be of type Checker. C. Passes a lambda expression. The formal parameter may be (but do not have to be) enclosed in parenthesis and have a type. D. Passes a lambda expression. The formal parameter does not have to have a type or be enclosed in parenthesis. If a return statement (or any other statement) is passed that statement must be enclosed in curly braces.
34.	D	When the add method is called bottom is pointing at the last element and bottom.nextToy points to null. bottom.nextToy is reassigned so that it points to the new element t which now becomes the last element. bottom is assigned t so that it now points the correct last element.
35.	C	Answer choices A, B and D do not remove the last element. Answer choice E removes the correct element but returns the next to last element instead of the element that has been removed.
36.	A	B. Creates an infinite loop because the while loop never advances to the next element. C. Throws a NullPointerException when the loop reaches the next to last element. D. Never prints the last element. E. Will not compile. Attempts to return a Toy object instead of a String.
37.	A	A ToyBox object is a singly linked list because each node points to the next node or to null if it is the last node and not to the previous one. It can not be a stack or a queue because the remove method may remove an element from anywhere in the list. It can not be a set because duplicate elements are allowed.
38.	E	$A*B+A*(B+C)+B*(B+C) =$ $A*B+A*B+A*C+B*B+B*C =$ (Distributive Law of AND over OR) $A*B+A*B+A*C+B+B*C =$ (Idempotent Law for AND) $A*B+A*C+B+B*C =$ (Idempotent Law for OR) $A*B+A*C+B =$ (Law of Absorption) $B+A*B+A*C=($ Commutative terms) $B+A*C$ (Law of Absorption)
39.	403	method1 returns the decimal value of the string representation of a hexadecimal value. $193_{16} = 403_{10}$
40.	18	method(i, j) returns the quotient when i is divided by j. $416 / 23 = 18$

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

	Score	Initials																							
<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;"># Right:</td> <td style="width: 10%; text-align: center;">×</td> <td style="width: 20%; text-align: center;">6 pts</td> <td style="width: 10%; text-align: center;">=</td> <td style="width: 30%;"></td> </tr> <tr> <td># Wrong:</td> <td style="text-align: center;">×</td> <td style="text-align: center;">-2 pts</td> <td style="text-align: center;">=</td> <td></td> </tr> <tr> <td># Skipped:</td> <td style="text-align: center;">×</td> <td style="text-align: center;">0 pts</td> <td style="text-align: center;">=</td> <td style="text-align: center; font-size: 2em; font-weight: bold;">0</td> </tr> </table>	# Right:	×	6 pts	=		# Wrong:	×	-2 pts	=		# Skipped:	×	0 pts	=	0	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Judge #1:</td> <td style="width: 15%; text-align: center;"><input style="width: 100%;" type="text"/></td> <td style="width: 15%; text-align: center;"><input style="width: 100%;" type="text"/></td> </tr> <tr> <td>Judge #2:</td> <td style="text-align: center;"><input style="width: 100%;" type="text"/></td> <td style="text-align: center;"><input style="width: 100%;" type="text"/></td> </tr> <tr> <td>Judge #3:</td> <td style="text-align: center;"><input style="width: 100%;" type="text"/></td> <td style="text-align: center;"><input style="width: 100%;" type="text"/></td> </tr> </table>	Judge #1:	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	Judge #2:	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	Judge #3:	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>
# Right:	×	6 pts	=																						
# Wrong:	×	-2 pts	=																						
# Skipped:	×	0 pts	=	0																					
Judge #1:	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>																							
Judge #2:	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>																							
Judge #3:	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>																							