



UNIVERSITY INTERSCHOLASTIC LEAGUE

Computer Science Written Study Packet 2020

This Computer Science packet contains tests and keys from **only** 2020 Invitational A, B and District. Region and State are not available.

This item is intended for High School grade levels.

UIL COMPUTER SCIENCE WRITTEN TEST

2020 INVITATIONAL A

JANUARY/FEBRUARY 2020

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 – 2020 INVITATIONAL A

Note: Correct responses are based on **Java SE Development Kit 12 (JDK 12)** 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 12 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.	
<p>Which of the following binary numbers is equivalent to the decimal number 253?</p> <p>A) 11111111 B) 11111101 C) 11111110 D) 11111100 E) 11111001</p>	
Question 2.	
<p>What is the output of the code segment to the right?</p> <p>A) -1 B) 2 C) 9 D) 13 E) 8.5</p>	<pre>out.print(8 + 5 - 9 * 2 / 4);</pre>
Question 3.	
<p>What is the output of the code segment to the right?</p> <p>A) Sep Oct Nov Dec</p> <p>B) Sep\nOct Nov Dec</p> <p>C) Sep OctNovDec</p> <p>D) Sep\nOctNov Dec</p> <p>E) Sep OctNov Dec</p>	<pre>out.print("Sep\nOct"); out.println("Nov"); out.print("Dec");</pre>
Question 4.	
<p>What is the output of the line of code shown on the right?</p> <p>A) 1 B) 2 C) 4</p> <p>D) 9 E) 10</p>	<pre>out.print("mississippi".indexOf('i'));</pre>
Question 5.	
<p>What is the output of the line of code shown on the right?</p> <p>A) true</p> <p>B) false</p>	<pre>out.print(true && !false false);</pre>
Question 6.	
<p>What is the output of the code segment to the right?</p> <p>A) 4 B) 3.8 C) 3.0 D) 4.0 E) 3.7</p>	<pre>int i = 3; double d = 1.25; out.print(Math.round(i * d));</pre>
Question 7.	
<p>What is the output of the code segment to the right?</p> <p>A) 6.0 B) 6.125 C) 9.25 D) 8.75 E) 12.5</p>	<pre>double e = 3.5, f = 2.75; int j = 3, k = 4; out.print(e * j + f - k);</pre>

Question 8.

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

- A) 12 8 5
- B) 16 5 8
- C) 16 5 11
- D) 12 5 11
- E) 12 5 8

```
int x = 12,y = 5,z = 8;
if(x / y > 2.0)
    z = z + 3;
if(z > 8)
    x = y + z;
out.print(x + " " + y + " " + z);
```

Question 9.

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

- A) 6543210
- B) 654321
- C) 543210
- D) 54321
- E) 6

```
int c = 6;
while(c > 0)
{
    out.print(c);
    c--;
}
```

Question 10.

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

- A) [1, 2, 2, 6, 5]
- B) [1, 2, 2, 4, 6]
- C) [1, 1, 6, 4, 5]
- D) [1, 6, 2, 6, 5]
- E) There is no output due to an error.

```
String[] list = {"1","2","3","4","5"};
list[3]="6";
list[2]=list[1];
out.print(Arrays.toString(list));
```

Question 11.

```
public class Q11
{
    public static void main(String[] args) throws IOException
    {
        Scanner f = new Scanner(new File("data.dat"));
        while(f.hasNext())
            out.print(f.next() + " ");
    }
}
```

Which of the following represents the output of the class shown above? You may assume that all necessary import statements are present and correct. The file named `data.dat` contains the following:

abcdef

- A) a
b
c
d
e
f
- B) a b c d e f
- C) a
- D) abcdef
- E) There is no output because the class throws an IOException

<p>Question 12.</p> <p>What is the output of the code segment to the right?</p> <p>A) 56 B) 36 C) 55 D) 45 E) 44</p>	<pre>int t = 0; for(int x = 0;x < 10;x++) t += x; out.print(t);</pre>
<p>Question 13.</p> <p>What is the correct order of operations (from left to right) for the operators listed on the right?</p> <p>A) += && ! + * B) * + ! && += C) ! + * += && D) += && + * ! E) ! * + && +=</p>	<p>+= && ! + *</p>
<p>Question 14.</p> <p>Which of the following values <u>cannot</u> be stored in a variable that is of type byte?</p> <p>A) 32767 B) -32768 C) 32768 D) -32767 E) None of the values shown above can be stored in a variable of type byte.</p>	
<p>Question 15.</p> <p>What is the output of the code segment to the right?</p> <p>A) [5.0, 4.0, 1.3, 3.5, 5.0, 5.25] B) [5.0, 5.8, 3.5, 1.3, 6.1, 5.8, 5.25] C) [5.0, 5.8, 3.5, 1.3, 6.1, 5.8] D) [5.0, 5.8, 3.5, 1.3, 5.8] E) [5.0, 5.8, 4.0, 1.3, 3.5, 5.8, 5.25]</p>	<pre>ArrayList<Double> list = new ArrayList<Double>(); list.add(5.0);list.add(5.8);list.add(4.0); list.add(3.5);list.add(6.1);list.add(5.25); list.remove(2); list.add(3, 1.3); list.set(5, list.get(1)); out.print(list);</pre>
<p>Question 16.</p> <p>What is the output of the code segment to the right?</p> <p>A) -6 B) -1 C) 1 D) 4 E) 6</p>	<pre>String r = "monkey"; String p = "money"; out.print(p.compareTo(r));</pre>
<p>Question 17.</p> <p>What is the output of the code segment to the right?</p> <p>A) 1.56 1.56 1.56 B) 1.56 3.14 2.77 C) No output. Will not compile. D) No output. Throws a NumberFormatException. E) No output. Throws a TypeMismatchException.</p>	<pre>String num = "1.56 3.14 2.77"; double d1 = Double.parseDouble(num); double d2 = Double.parseDouble(num); double d3 = Double.parseDouble(num); out.print(d1 + " " + d2 + " " + d3);</pre>

Question 18.

Which of the following statements correctly calculates the value of x in the formula shown here?

$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

- A) `double x = -b + Math.sqrt(Math.pow(b,2) - 4 * a * c) / 2 * a;`
- B) `double x = (-b + Math.sqrt(Math.pow(b,2) - 4 * a * c)) / (2 * a);`
- C) `double x = (-b + Math.sqrt(Math.pow(2,b) - 4 * a * c)) / 2 * a;`
- D) `double x = -b + Math.sqrt(b * b - 4 * a * c) / 2 * a;`
- E) `double x = -b + Math.sqrt(b * b - 4 * a * c) / (2 * a);`

Question 19.

How many constructors does the class `Uil` contain?

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

**//Use the following class to answer
//questions 19, 20 and 21.**

Question 20.

What is the output of this client code?

```
Uil uil1 = new Uil();
Uil uil2 = new Uil(12,"invB");
out.print(uil1.printUil());
out.print(uil2.printUil());
```

- A) `invA cs 0 null 12 invB`
- B) `cs null 0 invB 12`
- C) `invA cs invB 12`
- D) `invA cs 0 invB 12`
- E) `invA cs null 0 invB 12`

```
public class Uil
{
    private int a;
    private String b;

    public Uil()
    {
        out.print("invA ");
    }

    public Uil(int x,String s)
    {
        a = x;
        b = s;
        out.print("cs ");
    }

    public String printUil()
    {
        return b + " " + a + " ";
    }
}
```

Question 21.

What is printed by the following client code that is in a class other than `Uil`?

```
Uil uil = new Uil(240,"district");
int i = uil.a;
String s = uil.b;
out.print(i + " " + s);
```

- A) `240 district`
- B) `cs 240 district`
- C) `cs`
- D) `cs district`
- E) This code will not compile and prints an error message.

Question 22.

Which of the following methods correctly implements a binary search algorithm? Assume that `list` is sorted in ascending order.

<p>A.</p> <pre>public static int binarySearch(String[] list,String target) { int mid = list.length/2; int front = 0; int back = list.length-1; while(back >= front) { if(list[mid].equals(target)) return mid; if(target.compareTo(list[mid])>0) back = mid - 1; else front = mid + 1; mid = (front + back) / 2; } return -1; }</pre>	<p>B.</p> <pre>public static int binarySearch(String[] list,String target) { int mid = list.length/2; int front = 0; int back = list.length-1; while(back >= front) { if(list[mid].equals(target)) return mid; if(target.compareTo(list[mid])<0) back = mid - 1; else front = mid + 1; mid = (front + back) / 2; } return -1; }</pre>
<p>C.</p> <pre>public static int binarySearch(String[] list,String target) { int mid = list.length/2; int front = 0; int back = list.length-1; while(back >= front) { if(list[mid].equals(target)) return mid; if(target.compareTo(list[mid])<0) front = mid + 1; else back = mid - 1; mid = (front + back) / 2; } return -1; }</pre>	<p>D.</p> <pre>public static int binarySearch(String[] list,String target) { int mid = list.length/2; int front = 0; int back = list.length-1; while(back > front) { if(list[mid].equals(target)) return mid; if(target.compareTo(list[mid])<0) back = mid - 1; else front = mid + 1; mid = (front + back) / 2; } return -1; }</pre>
<p>E. More than one of the above.</p>	

Question 23.

Once a method has correctly implemented a binary search algorithm, which of these would represent the tightest correct runtime on an array of n elements?

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

Question 24.

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

- A) 25
- B) 59
- C) 34
- D) 2030
- E) 1

```
out.print(58^35);
```

Question 25.

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

- A) [5, 6, 7, 8]
[9, 0, 1, 2]
[1, 2, 3, 4]
- B) [9, 0, 1, 2]
[1, 2, 3, 4]
[5, 6, 7, 8]
- C) [1, 2, 3, 4]
[9, 0, 1, 2]
[5, 6, 7, 8]
- D) [5, 6, 7, 8]
[1, 2, 3, 4]
[9, 0, 1, 2]

E) There is no output due to an error.

```
int[][] m = new int[3][];
int[] x = {1,2,3,4};
int[] y = {5,6,7,8};
int[] z = {9,0,1,2};
m[0] = z;
m[2] = y;
m[1] = x;
int[] t = m[0];
m[0] = m[1];
m[1] = t;
for(int[] r:m)
    out.println(Arrays.toString(r));
```

Question 26.

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

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

```
String s = "325-555-1234";
out.print(s.matches(".{10}") + " ");
out.print(s.matches("\\d+-\\d+-\\d+") + " ");
out.print(s.matches("325\\W555\\S1234"));
```

Question 27.

How many abstract methods does every functional interface contain?

- A) 0
- B) 1
- C) Always more than 1.
- D) The number of abstract methods depends on the function of the interface.
- E) The number of arguments passed by the lambda expression determines the number of abstract methods in the interface.

Question 28.

Which of the following represents the output of the code segment listed here?

```
ArrayList<String> list = new ArrayList<String>();
list.add("tomato");list.add("ham");
list.add("turkey");list.add("onion");
Set<String> set = new TreeSet<String>();
set.add("ham");set.add("turkey");
set.add("beef");set.add("cheese");
set.add("mustard");set.add("lettuce");
set.addAll(list);
out.print(set);
```

- A) [ham, turkey, beef, cheese, mustard, lettuce, tomato, onion]
- B) [mustard, ham, turkey, onion, beef, tomato, lettuce, cheese]
- C) [beef, cheese, ham, lettuce, mustard, onion, tomato, turkey]
- D) [beef, cheese, lettuce, mustard, onion, tomato]
- E) [beef, cheese, ham, ham, lettuce, mustard, onion, tomato, turkey, turkey]

Question 29.

Given the statements shown on the right, what is the largest possible value that might be assigned to w?

- A) 101
- B) 9.5
- C) 9
- D) 10.5
- E) 10

```
Random r = new Random();
int w = (int)(r.nextDouble() * 10 + 0.5);
```

Question 30.

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

- A) [homecoming]
- B) [h, m, c, m, ng]
- C) [o, e, o, i]
- D) [oeoi]
- E) [hmcming]

```
String a = "homecoming";
String[] p = a.split("aeiou");
out.print(Arrays.toString(p));
```

Question 31.

Consider the method crunch shown on the right. What is the output of this segment of client code?

```
//client code
int x = 5;
Integer y = 90, z = 15;
for(int i = 1; i <= 5; i++)
    crunch(x, y, z);
out.print(x + " " + y + " " + z);
```

- A) 15 45 480
- B) 5 90 15
- C) 7 85 30
- D) 26 45 480
- E) 7 45 480

```
public static void crunch(int x,
Integer y, Integer z)
{
    y -= x;
    z *= 2;
    x += 2;
}
```

Question 32.

The method shown on the right is a partial implementation of the selection sort algorithm. The array `list` should be sorted in ascending order.

Which of the following must replace **<code1>** to ensure that the method will compile and function as intended.

- A) 0
- B) `y`
- C) `i + 1`
- D) `x - 1`
- E) 1

```
//Use this implementation of an ascending
//selection sort algorithm to answer
//questions 32 - 34.
```

```
public static void selection(int list[])
{
    int x,y;
    for(int i = 0;i < list.length; ++i)
    {
        x = list[i];
        y = i;
        for(int j = <code1>;j < list.length; j++)
            if(<code2>)
            {
                x = list[j];
                y = j;
            }
        list[y] = list[i];
        list[i] = x;
        //comment
    }
}
```

Question 33.

Which of the following must replace **<code2>** to ensure that the method will compile and function as intended?

- A) `list[i] < x`
- B) `list[j] < y`
- C) `list[j] > x`
- D) `list[j] < x`
- E) `list[i] > list[j]`

Question 34.

If the client code shown here is executed, what is the state of the array `list` when `i` equals 3 and the execution of the code has reached the comment?

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

- A) [1, 2, 0, 5, 4, 3, 6, 7, 8, 9]
- B) [0, 1, 2, 8, 7, 9, 6, 5, 4, 3]
- C) [1, 2, 6, 7, 8, 9, 0, 5, 4, 3]
- D) [1, 2, 3, 7, 8, 9, 0, 5, 4, 6]
- E) [0, 1, 2, 3, 7, 9, 6, 5, 4, 8]

Question 35.

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

- A) 4 2 4
- B) 0 6 0
- C) 5 9 4
- D) 5 1 4
- E) There is no output due to an error.

```
int r = 0,p = 6,s = 0;
for(;r < 5;r++)
{
    s = ~s;
    while(p > 1)
    {
        s++;
        p--;
    }
}
out.print(r + " " + p + " " + s);
```

Question 36.

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

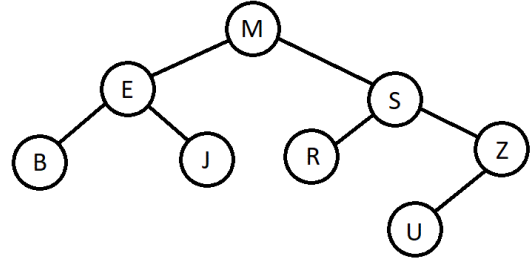
- A) SEptEMbEr
- B) sePTemBeR
- C) PTBR
- D) SEEME
- E) september

```
String s1 = "september";
String s2 = "";
for(int i = 0; i < s1.length(); i++)
    if(s1.charAt(i) % 2 == 0)
        s2 += (char) (s1.charAt(i) - 32);
out.print(s2);
```

Question 37.

Which of the following represents a pre-order traversal of the binary search tree shown on the right?

- A) M E B J S R Z U
- B) B J E R U Z S M
- C) B E J M R S U Z
- D) M E S B J R Z U
- E) U Z R J B E S M

**Question 38.**

Which of the following Boolean expressions is NOT equal to A?

- A) $A * (A + B)$
- B) $\overline{A + B}$
- C) $A + A * B$
- D) $A * A$
- E) $A + A$

Question 39.

What is the output of the client code shown here given the method `rec` shown on the right. Write your answer in the blank provided on the answer document.

```
out.print(rec(3));
```

```
public static int rec(int x)
{
    if(x == 0)
        return 7;
    else
        return 2 * rec(--x);
}
```

Question 40.

Write the signed 8-bit binary two's complement representation of -101 in the blank provided on the answer document.

★ ANSWER KEY – CONFIDENTIAL ★

UIL COMPUTER SCIENCE – 2020 INVITATIONAL A

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

- | | | | |
|------------------|------------------|------------------|--------------------------|
| 1) <u> B </u> | 11) <u> D </u> | 21) <u> E </u> | 31) <u> B </u> |
| 2) <u> C </u> | 12) <u> D </u> | 22) <u> B </u> | 32) <u> C </u> |
| 3) <u> E </u> | 13) <u> E </u> | 23) <u> D </u> | 33) <u> D </u> |
| 4) <u> A </u> | 14) <u> E </u> | 24) <u> A </u> | 34) <u> E </u> |
| 5) <u> A </u> | 15) <u> C </u> | 25) <u> C </u> | 35) <u> D </u> |
| 6) <u> A </u> | 16) <u> A </u> | 26) <u> D </u> | 36) <u> C </u> |
| 7) <u> C </u> | 17) <u> D </u> | 27) <u> B </u> | 37) <u> A </u> |
| 8) <u> E </u> | 18) <u> B </u> | 28) <u> C </u> | 38) <u> B </u> |
| 9) <u> B </u> | 19) <u> C </u> | 29) <u> E </u> | *39) <u> 56 </u> |
| 10) <u> A </u> | 20) <u> E </u> | 30) <u> A </u> | *40) <u> 10011011 </u> |

KEY

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

Note: Correct responses are based on **Java SE Development Kit 12 (JDK 12)** 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 12 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used.

Explanations:

1.	B	<table border="1"> <tr> <td>128</td> <td>64</td> <td>32</td> <td>16</td> <td>8</td> <td>4</td> <td>2</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> </tr> </table> <p>$128 + 64 + 32 + 16 + 8 + 4 + 0 + 1 = 253$</p>	128	64	32	16	8	4	2	1	1	1	1	1	1	1	0	1														
128	64	32	16	8	4	2	1																									
1	1	1	1	1	1	0	1																									
2.	C	$8+5-9*2/4=$ $8+5-18/4=$ $8+5-4=$ $13-4=$ 9																														
3.	E	<p>\n is the escape sequence for a new line println places the cursor on a new line after printing the argument. print does not.</p>																														
4.	A	indexOf returns the <u>first</u> occurrence of the argument 'i'.																														
5.	A	$true \ \&\& \ !false \ \ false =$ $true \ \&\& \ true \ \ false =$ $true \ \ false =$ true																														
6.	A	$3 * 1.25 = 3.75 \rightarrow \text{Math.round}(3.75)$ returns 4. Math.round returns a long type value.																														
7.	C	$3.5*3+2.75-4=$ $10.5+2.75-4=$ $13.25-4=$ 9.25																														
8.	E	$12/5 > 2.0$ $8 > 8$ is false Neither of the if statements are executed. $2 > 2.0$ is false																														
9.	B	<table border="1"> <thead> <tr> <th>C</th> <th>C > 0</th> <th>Print</th> </tr> </thead> <tbody> <tr><td>6</td><td>T</td><td>6</td></tr> <tr><td>5</td><td>T</td><td>5</td></tr> <tr><td>4</td><td>T</td><td>4</td></tr> <tr><td>3</td><td>T</td><td>3</td></tr> <tr><td>2</td><td>T</td><td>2</td></tr> <tr><td>1</td><td>T</td><td>1</td></tr> <tr><td>0</td><td>F</td><td></td></tr> </tbody> </table>	C	C > 0	Print	6	T	6	5	T	5	4	T	4	3	T	3	2	T	2	1	T	1	0	F							
C	C > 0	Print																														
6	T	6																														
5	T	5																														
4	T	4																														
3	T	3																														
2	T	2																														
1	T	1																														
0	F																															
10.	A	<table border="1"> <thead> <tr> <th>index value</th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> </tr> </thead> <tbody> <tr> <td>original list</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> </tr> <tr> <td>list[3]="6"</td> <td>1</td> <td>2</td> <td>3</td> <td>6</td> <td>5</td> </tr> <tr> <td>list[2]=list[1]</td> <td>1</td> <td>2</td> <td>2</td> <td>6</td> <td>5</td> </tr> </tbody> </table>	index value	0	1	2	3	4	original list	1	2	3	4	5	list[3]="6"	1	2	3	6	5	list[2]=list[1]	1	2	2	6	5						
index value	0	1	2	3	4																											
original list	1	2	3	4	5																											
list[3]="6"	1	2	3	6	5																											
list[2]=list[1]	1	2	2	6	5																											
11.	D	The default delimiter for the next method is a space. There are no spaces in the single data item "abcdef" therefore that string is read all at once and the loop terminates.																														
12.	D	<table border="1"> <thead> <tr> <th>x</th> <th>t</th> </tr> </thead> <tbody> <tr><td></td><td>0</td></tr> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> <tr><td>2</td><td>3</td></tr> <tr><td>3</td><td>6</td></tr> <tr><td>4</td><td>10</td></tr> <tr><td>5</td><td>15</td></tr> <tr><td>6</td><td>21</td></tr> <tr><td>7</td><td>28</td></tr> <tr><td>8</td><td>36</td></tr> <tr><td>9</td><td>45</td></tr> </tbody> </table>	x	t		0	0	0	1	1	2	3	3	6	4	10	5	15	6	21	7	28	8	36	9	45						
x	t																															
	0																															
0	0																															
1	1																															
2	3																															
3	6																															
4	10																															
5	15																															
6	21																															
7	28																															
8	36																															
9	45																															
13.	E	The correct order is ! * + && +=																														
14.	E	The range for byte is -128 to 127																														
15.	C	<table border="1"> <tr> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> </tr> <tr> <td>5.0</td> <td>5.8</td> <td>4.0</td> <td>3.5</td> <td>6.1</td> <td>5.25</td> </tr> <tr> <td>5.0</td> <td>5.8</td> <td>3.5</td> <td>6.1</td> <td>5.25</td> <td></td> </tr> <tr> <td>5.0</td> <td>5.8</td> <td>3.5</td> <td>1.3</td> <td>6.1</td> <td>5.25</td> </tr> <tr> <td>5.0</td> <td>5.8</td> <td>3.5</td> <td>1.3</td> <td>6.1</td> <td>5.8</td> </tr> </table>	0	1	2	3	4	5	5.0	5.8	4.0	3.5	6.1	5.25	5.0	5.8	3.5	6.1	5.25		5.0	5.8	3.5	1.3	6.1	5.25	5.0	5.8	3.5	1.3	6.1	5.8
0	1	2	3	4	5																											
5.0	5.8	4.0	3.5	6.1	5.25																											
5.0	5.8	3.5	6.1	5.25																												
5.0	5.8	3.5	1.3	6.1	5.25																											
5.0	5.8	3.5	1.3	6.1	5.8																											
16.	A	ASCII values for 'k' and 'e' are 107 and 101. $101 - 107 = -6$.																														

17.	D	The string being parsed cannot contain a space. Code will compile but then throws a <code>NumberFormatException</code> because of the space.																								
18.	B	Answer choices A and D divide by 2 and multiply by a before adding -b. Answer choice C raises 2 to the power of b rather than b^2 . Answer choice E divides by $2a$ then adds -b.																								
19.	C	<code>public Uil()</code> and <code>public Uil(int x,String s)</code> are the two constructors.																								
20.	E	Each call to a constructor prints. The default constructor does not assign values to a or b so they have default values of 0 and null.																								
21.	E	Private fields cannot be accessed from outside the class they are in.																								
22.	B	Answer choices A and C search in the wrong half of the list with each pass of the while loop. Answer choice D doesn't find the target if it is the first or last element in the list.																								
23.	D	In one step, we do a constant amount of work to cut the number of elements under consideration in half. There are $O(\log n)$ such steps until we're left with a trivial array.																								
24.	A	$58_{10} = 111010_2$ $35_{10} = 100011_2$ <table border="1" style="margin-left: 20px;"> <tr><td></td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td></td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>XOR</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> </table> $011001_2 = 25_{10}$		1	1	1	0	1	0		1	0	0	0	1	1	XOR	0	1	1	0	0	1			
	1	1	1	0	1	0																				
	1	0	0	0	1	1																				
XOR	0	1	1	0	0	1																				
25.	C	First row is array z, second row is array x and third row is array y. <table border="1" style="margin-left: 20px;"> <tr><td>9</td><td>0</td><td>1</td><td>2</td></tr> <tr><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr><td>5</td><td>6</td><td>7</td><td>8</td></tr> </table> Then rows 0 and 1 are swapped. <table border="1" style="margin-left: 20px;"> <tr><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr><td>9</td><td>0</td><td>1</td><td>2</td></tr> <tr><td>5</td><td>6</td><td>7</td><td>8</td></tr> </table>	9	0	1	2	1	2	3	4	5	6	7	8	1	2	3	4	9	0	1	2	5	6	7	8
9	0	1	2																							
1	2	3	4																							
5	6	7	8																							
1	2	3	4																							
9	0	1	2																							
5	6	7	8																							
26.	D	<code>".{10}"</code> any character exactly 10 times is false. 10 digits plus two dashes. <code>"\d+\-\\d+\-\\d+-"</code> is any digit one or more times plus a dash repeated three times is true. <code>"325\\W555\\S1234"</code> is 325 followed by a non-word character followed by 555 followed by a non-whitespace character followed by 1234 is true.																								
27.	B	The definition of a functional interface is that it may contain only one abstract method.																								
28.	C	This is a <code>TreeSet</code> so the elements will be alphabetized. The original set is: beef cheese ham lettuce mustard turkey Sets can not contain duplicates so, when the <code>ArrayList</code> is added, only onion and tomato are added to set. This results in beef cheese ham lettuce mustard onion tomato turkey																								
29.	E	The largest value that might be generated by <code>nextDouble</code> is approximately 0.999. Multiply 0.999 time 10 to get approximately 9.99. Add 0.5 to get approximately 10.49 then cast to an integer to get 10.																								
30.	A	The string "aeiou" is the delimiter for the <code>split</code> method. That exact string does not occur in "homecoming" so the next method returns "homecoming".																								
31.	B	x is passed by value. y and z are passed by reference, however, both <code>Integer</code> and <code>Double</code> are immutable. Since the objects are immutable, when the assignment statements within the method are made, brand new objects are created that then autobox the primitive values, however, no changes are made to the original x and y objects.																								
32.	C	Each iteration of the inner loop begins one spot beyond the elements that have already been placed into the sorted portion of the array.																								
33.	D	The inner loop is searching for the smallest value left in the unsorted portion of the list.																								
34.	E	$i = 0$ [0, 7, 1, 8, 2, 9, 6, 5, 4, 3] $i = 1$ [0, 1, 7, 8, 2, 9, 6, 5, 4, 3] $i = 2$ [0, 1, 2, 8, 7, 9, 6, 5, 4, 3] $i = 3$ [0, 1, 2, 3, 7, 9, 6, 5, 4, 8]																								

35.	D	<pre> r p s for loop 0 0 while loop 0 5 0 while loop 0 4 1 while loop 0 3 2 while loop 0 2 3 while loop 0 1 4 for loop 1 4 for loop 2 -5 for loop 3 4 for loop 4 -5 5 1 4 </pre>
36.	C	The if statement determines if each character's ASCII value is divisible by 2. Those that are divisible by 2 are converted to upper case and printed.
37.	A	Pre-order traversal visits each node in a root – left – right order.
38.	B	DeMorgan's law says that $\overline{A + B}$ is equal to $\overline{A} * \overline{B}$.
39.	56	<pre> rec(3) = 2 * rec(2) = 2 * 28 = 56 rec(2) = 2 * rec(1) = 2 * 14 = 28 rec(1) = 2 * rec(0) = 2 * 7 = 14 rec(0) = 7 </pre>
40.	10011011	<pre> 101₁₀ = 01100101₂ ~01100101 = 10011010 10011010 + 1 = 10011011 </pre>

UIL COMPUTER SCIENCE WRITTEN TEST

2020 INVITATIONAL B

FEBRUARY/MARCH 2020

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 – 2020 INVITATIONAL B

Note: Correct responses are based on **Java SE Development Kit 12 (JDK 12)** 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 12 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used. For all output statements, assume that the **System** class has been statically imported using: `import static java.lang.System.*;`

Question 1. Which of the following is equivalent to 11001011 ₂ ? A) 68 ₁₆ B) CB ₁₆ C) CC ₁₆ D) A5 ₁₆ E) 20 ₁₆	
Question 2. What is the output of the code segment to the right? A) 0 B) 14 C) 2 D) 9 E) 8	<pre>out.print(13 % 5 + 6 % 7);</pre>
Question 3. What is the output of the code segment to the right? A) Invitational\\B B) Invitational B C) InvitationalB D) Invitational B E) Invitational\B	<pre>out.println("Invitational\\B");</pre>
Question 4. What is the output of the code segment to the right? A) TeamGo B) GoTeam C) oTeamG D) oTeamGo E) GoTeamG	<pre>String str = "GoTeamGo"; out.print(str.substring(1, 7));</pre>
Question 5. What is the output of the code segment shown on the right? A) true B) false	<pre>boolean a = true; boolean b = true; boolean c = a ^ b; out.print(!c);</pre>
Question 6. What is the output of the code segment to the right? A) 3.0 B) 3 C) 1.25 D) 1 E) There is no output due to an error.	<pre>int i = 3; double j = 1.25; out.print(Math.max(i, j));</pre>
Question 7. What is the output of the code segment to the right? A) 15.0 B) -1 C) 15 D) 1.0 E) -1.0	<pre>int i = 14, j = 8; double d = 4.0, e = 11.0; out.print(i + d - e - j);</pre>
Question 8. What is the output of the code segment shown on the right? A) 7 B) 1 C) 48 D) 1.1 E) 36	<pre>int r = 19, s = 12, t = 21; if(r % s > t) out.print(r - s); else if(s - t < r) out.print(t / r); else out.print(s * 4);</pre>

Question 9.

Which of the following represents the output of the code shown to the right?

- A) #####
- B) #####
- C) #####
- D) #####
- E) #####

```
int stop = 100;
int go = 1;
do {
    out.print("#");
    go *= 2;
}while(go < stop);
```

Question 10.

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

- A) [3, 8, -4, 5, 0]
- B) [3, 8, -4, 5]
- C) [0, 3, -4, -4, 5]
- D) [0, 3, 8, -4, 5]
- E) [0, 8, -4, 8, 5]

```
int []nums = new int[5];
nums[1] = 3; nums[3] = -4;
nums[2] = 8; nums[4] = 5;
nums[nums[1]] = nums[nums.length-2];
out.print(Arrays.toString(nums));
```

Question 11.

Which of the following is the output of the code segment shown on the right? You may assume that all necessary import statements are present and correct. The file named `data.dat` contains the following:

```
one,two,three,four,five,six
```

- A) one,two,three,four,five,six
- B) one, two, three, four, five, six
- C) onetwothreefourfivesix
- D) one two three four five six
- E) one

```
Scanner f = new Scanner(new
File("data.dat"));
f.useDelimiter(",");
while(f.hasNext())
    out.print(f.next());
f.close();
```

Question 12.

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

- A) 255
- B) 256
- C) 128
- D) 12
- E) 6

```
int m = 2, n = 0, p = 1;
while(n < 7) {
    p = p * m;
    n++;
}
out.print(p);
```

Question 13.

What is the output of the code segment shown here?

```
double x = 1.5, y = 2.25, z = -0.75;
out.print(x + z > y || z - Math.abs(x - y) > z);
```

- A) true
- B) false

Question 14.

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

- A) 2147483647
- B) 32767
- C) 127
- D) 2147483648
- E) 32768

```
out.println(Integer.MAX_VALUE);
```

Question 15.

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

- A) [E, B, D, A]
- B) [E, C, D, A]
- C) [E, C, B, D, A]
- D) This code segment will not compile.
- E) This code segment throws an exception.

```
ArrayList<String> list = new
ArrayList<String>();
list.add("E");list.add("C");list.add("B");
list.add("D");list.add("A");
list.remove(2);list.remove("F");
out.print(list);
```

Question 16.

Which of the following must replace **<code>** in the class shown on the right so that the values passed in parameters *x* and *y* are assigned to the instance variables *x* and *y*?

- A) private
- B) super
- C) public
- D) this
- E) static

```
public class Uil
{
    private int x;
    public int y;
    public static int z;

    public Uil(int x, int y) {
        <code>.x = x;
        <code>.y = y;
        z += 2;
    }

    public int getX() {
        return x;
    }

    public static int getZ() {
        return z;
    }
}
```

Question 17.

If **<code>** has been filled in correctly, which of the following lines of client code will NOT compile and execute correctly?

```
Uil uil = new Uil(5,4);
int a = Uil.y; //line #1
int b = uil.y; //line #2
int c = uil.getZ(); //line #3
int d = Uil.getZ(); //line #4
```

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

Question 18.

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

- A) false false false 0
- B) true true true false
- C) false false true 0
- D) false true true true
- E) false true true 0

```
String r = "apple";
String p = new String("apple");
String o = "apple";
String s = p;
out.print((r == p) + " ");
out.print((r == o) + " ");
out.print((r.equals(p))+" ");
out.print(p.compareTo(s));
```

Question 19.

Consider the class `FunWithNumbers` shown on the right. Which of the following must replace `<code>` to ensure that when executed the main method will produce this output?

[TEN, TWO, TEN, FOUR, TEN, SIX, TEN, EIGHT, TEN, TEN]

- A) `Numbers.values(TEN)`
- B) `TEN`
- C) `Numbers.TEN`
- D) `new Numbers(TEN)`
- E) `10`

```
import static java.lang.System.out;
import java.util.Arrays;
public class FunWithNumbers {

    public enum Numbers{
        ONE, TWO, THREE, FOUR,
        FIVE, SIX, SEVEN, EIGHT,
        NINE, TEN
    }

    public static void main(String[] args) {
        Numbers[] nums = Numbers.values();
        for(int i = 0; i < nums.length; i+=2)
            nums[i] = <code>;
        out.print(Arrays.toString(nums));
    }
}
```

Question 20.

The formula to find the area of a triangle when two sides and the included angle are known is:

$$\frac{1}{2}ab \sin C$$

Which of the following statements will calculate the area of a triangle with sides `a` and `b` and an included angle `c` measured in degrees and assign that value to a variable named `area`.

- A) `double area = 1 / 2 * a * b * Math.sin(Math.toRadians(c));`
- B) `double area = 0.5 * a * b * Math.sin(c);`
- C) `double area = 1.0 / 2.0 * ab(Math.sin(Math.toRadians(c)));`
- D) `double area = 1.0 / 2 * a * b * Math.sin(toRadians(c));`
- E) `double area = 0.5 * a * b * Math.sin(Math.toRadians(c));`

Question 21.

```
public static void main(String[] args) {
    <code> item = r -> r.substring(r.length() / 2);
    out.print(item.getIt("today"));
}

public interface UILInterface {
    public String getIt(String s);
}
```

The code segment shown above should print the second half of the string passed to the `getIt` method. What must replace `<code>` to ensure that the segment will compile and function as intended?

- A) `int`
- B) `public`
- C) `UILInterface`
- D) `String`
- E) No additional code is required.

Question 22.

Which of the following best describes the method shown here?

```
public static int method(String[]list,String t) {
    int i = 0, j = -1;
    while(i < list.length) {
        if(list[i].equals(t))
            j = i;
        i++;
    }
    return j;
}
```

- A) method returns the index value of each occurrence of t in list or 0 if t is not found.
- B) method returns the index value of the first occurrence of t in list or -1 if t is not found.
- C) method returns the String at index value t or -1 if t is out of bounds.
- D) method returns the index value of the last occurrence of t in list or -1 if t is not found.
- E) method always returns -1.

Question 23.

Which of the following shows the output of this client code?

```
InvB list = new InvB();
list.cat("mercury");list.cat("mars");
list.cat("earth");list.cat("jupiter");
list.bird();list.dog();list.bird();
while(!list.snake())
    out.print(list.dog() + " ");
```

- A) earth mars mercury
- B) mars earth jupiter
- C) mercury
- D) jupiter earth mars
- E) jupiter

//Use the class shown here to answer
// questions 23 and 24.

```
import java.util.*;
public class InvB {

    private ArrayList<String> list;

    public InvB() {
        list = new ArrayList<String>();
    }

    public void cat(String s) {
        list.add(s);
    }

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

    public String bird() {
        return list.get(list.size() - 1);
    }

    public boolean snake() {
        return list.size() == 0;
    }
}
```

Question 24.

The class InvB implements a _____.

- A) queue
- B) map
- C) stack
- D) set
- E) priority queue

Question 25.

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

- A) five 4
- B) zero 4
- C) one 5
- D) three 5
- E) two 6

```
Queue<String> list = new
LinkedList<String>();
list.add("two");list.add("one");
list.add("five");list.add("zero");
list.add("three");list.add("four");
list.poll();list.remove();
out.print(list.peek() + " " + list.size());
```

Question 26.

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

- A) ***##1-13
- B) *****2-23
- C) *****-113
- D) ***##2-23
- E) ###** -113

```
int x = 4, y = -4, a = -2;
while(a <= 2) {
    if(a++ > 0 && x-- == ++y)
        out.print("# ");
    else
        out.print("* ");
}
out.print(x + " " + y + " " + a);
```

Question 27.

Which of the following must replace **<code>** to ensure that the method will compile and execute as intended?

- A) k<=0 && list[k]<current
- B) k>=0 || list[k]>current
- C) k>=0 && list[k]>current
- D) list[k]>current
- E) k<=0

Question 28.

Assume that **<code>** has been replaced with the correct code and this client code is executed.

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

What is printed when i equals 4 if this line of code

```
out.println(Arrays.toString(list));
```

replaces the comment?

- A) [1, 2, 0, 5, 4, 3, 6, 7, 8, 9]
- B) [0, 1, 2, 3, 4, 9, 6, 5, 7, 8]
- C) [1, 6, 7, 8, 2, 9, 0, 5, 4, 3]
- D) [1, 2, 6, 7, 8, 9, 0, 5, 4, 3]
- E) [0, 1, 2, 3, 7, 9, 6, 5, 4, 8]

```
//Use the following implementation of
//an insertion sort to answer
//questions 27, 28 and 29.
```

```
public static void insertion(int[] list)
{
    for(int i = 1; i < list.length;i++)
    {
        int current = list[i];
        int k;
        for(k = i - 1; <code>; k--)
            list[k + 1] = list[k];
        list[k + 1] = current;
        //comment
    }
}
```

Question 29.

Once the method `insertion` has been correctly implemented, what is the worst case time complexity for this method?

- A) O(1)
- B) O(n)
- C) O(log n)
- D) O(n log n)
- E) O(n²)

Question 30.

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

- A) [1, 4, 7]
[2, 5, 8]
[3, 6, 9]
- B) [7, 8, 9]
[4, 5, 6]
[1, 2, 3]
- C) [9, 6, 3]
[8, 5, 2]
[7, 4, 1]
- D) [3, 2, 1]
[6, 5, 4]
[9, 8, 7]
- E) [3, 6, 9]
[2, 5, 8]
[1, 4, 7]

```
int[][] mat = {{1,2,3},{4,5,6},{7,8,9}};
int len = mat.length - 1;
for(int r = 0; r < mat.length; r++) {
    for( int c = 0; c < len; c++) {
        int t = mat[r][c];
        mat[r][c]=mat[mat.length - 1 - c][len];
        mat[mat.length - 1 - c][len] = t;
    }
    len--;
}
for(int[] a:mat)
    out.println(Arrays.toString(a));
```

Question 31.

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

- A) 9
- B) 27
- C) 1
- D) 24
- E) 0

```
System.out.print(3<<3);
```

Question 32.

If a particular method whose run time efficiency is $O(n^2)$ requires 1 second to process 12000 elements in a data set, how long will it take to process 48000 elements?

- A) 2 seconds
- B) 4 seconds
- C) 8 seconds
- D) 16 seconds
- E) 32 seconds

Question 33.

Which of the following is NOT a valid identifier?

- A) amount
- B) \$Dollar
- C) %percent
- D) last
- E) three_4

Question 34.

Which of the following is the correct method header for a method that returns a tip amount, given the amount of the check and the desired percent tip?

- A) public static tip(double check, double percent)
- B) public static double tip(double check, double percent)
- C) public static int tip(double check, double percent)
- D) public static double tip(check, percent)
- E) public static double (double check, double percent)

Question 35.Which of the following is the truth table for this expression? $\bar{A} + B \oplus A$

A.

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

B.

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

C.

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

D.

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

E.

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

Question 36.

Which of the following represents output of the main method shown on the right?

- A) *#####r
- B) r#####*
- C) *#####c
- D) *r#####
- E) r*#####

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

public static String rec(String x)
{
    if(x.length()>1)
    {
        return "#" + rec(x.substring(1));
    }
    else
    {
        out.print("*");
        return x;
    }
}
```

Question 37.

What is the value of the postfix expression shown on the right? The operands are 8, 3, 2, 14, and 5.

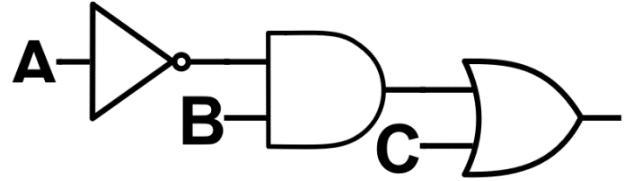
- A) 37
- B) 7
- C) 29
- D) 11
- E) 22

$$8 \ 3 \ * \ 2 \ + \ 14 \ 5 \ + \ -$$

Question 38.

Which of the following Boolean expressions is diagrammed on the right?

- A) $A \ \&\& \ !B \ || \ C$
- B) $!A \ || \ B \ \&\& \ C$
- C) $!(A \ \&\& \ B \ || \ C)$
- D) $A \ \&\& \ B \ || \ C$
- E) $!A \ \&\& \ B \ || \ C$

**Question 39.**

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

```

char[] c = {'a','m','e','x','r','c'};
int h = 200;
for(char ch:c)
    switch(ch) {
        case 'a':h-='a';break;
        case 'm':h+='m';break;
        case 'x':h-='x';
        case 'c':h+='c';break;
        default: h++;
    }
out.print(h);
  
```

Question 40.

If the values shown here are placed into a binary search tree in the order that they are listed, which value will serve as the root node? Write your answer in the blank provided on the answer document.

2 0 8 4 6 7 1 3 5 9

★ ANSWER KEY – CONFIDENTIAL ★

UIL COMPUTER SCIENCE – 2020 INVITATIONAL B

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

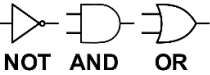
- | | | | |
|------------------|------------------|------------------|-----------------------|
| 1) <u> B </u> | 11) <u> C </u> | 21) <u> C </u> | 31) <u> D </u> |
| 2) <u> D </u> | 12) <u> C </u> | 22) <u> D </u> | 32) <u> D </u> |
| 3) <u> E </u> | 13) <u> B </u> | 23) <u> A </u> | 33) <u> C </u> |
| 4) <u> C </u> | 14) <u> A </u> | 24) <u> C </u> | 34) <u> B </u> |
| 5) <u> A </u> | 15) <u> B </u> | 25) <u> A </u> | 35) <u> E </u> |
| 6) <u> A </u> | 16) <u> D </u> | 26) <u> B </u> | 36) <u> A </u> |
| 7) <u> E </u> | 17) <u> A </u> | 27) <u> C </u> | 37) <u> B </u> |
| 8) <u> B </u> | 18) <u> E </u> | 28) <u> D </u> | 38) <u> E </u> |
| 9) <u> E </u> | 19) <u> C </u> | 29) <u> E </u> | *39) <u> 292 </u> |
| 10) <u> D </u> | 20) <u> E </u> | 30) <u> C </u> | *40) <u> 2 </u> |

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

Note: Correct responses are based on **Java SE Development Kit 12 (JDK 12)** 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 12 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used.

Explanations:

1.	B	$1100_2 \mid 1011_2 = 12_{10}$ and $11_{10} = CB_{16}$																											
2.	D	$13\%5 + 6\%7 = 3 + 6 = 9$																											
3.	E	\\ is the escape sequence used to insert a backslash.																											
4.	C	<table border="1" style="display: inline-table; vertical-align: top;"> <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> </tr> <tr> <td>G</td><td>o</td><td>T</td><td>e</td><td>a</td><td>m</td><td>G</td><td>o</td> </tr> </table> <p>Two argument substring method returns a substring beginning at the first index and ending at second argument minus one.</p>	0	1	2	3	4	5	6	7	G	o	T	e	a	m	G	o											
0	1	2	3	4	5	6	7																						
G	o	T	e	a	m	G	o																						
5.	A	$a \wedge b = T \wedge T = F$ and $!F = T$																											
6.	A	i is promoted to a double when the max method is called. The max method with two double parameters returns a double.																											
7.	E	$i + d - e - j =$ $14 + 4.0 - 11.0 - 8 =$ $18.0 - 11.0 - 8 =$ $7.0 - 8 =$ -1.0																											
8.	B	$r \% s > t = 19 \% 12 > 21 = 7 > 21 = \text{false}$ $s - t < r = 12 - 21 < 19 = -9 < 19 = \text{true}$ $t / r = 21 / 19 = 1$																											
9.	E	<table border="1" style="display: inline-table; vertical-align: top;"> <thead> <tr> <th>stop</th> <th>go</th> <th>output</th> </tr> </thead> <tbody> <tr><td>100</td><td>1</td><td>#</td></tr> <tr><td>100</td><td>2</td><td>##</td></tr> <tr><td>100</td><td>4</td><td>###</td></tr> <tr><td>100</td><td>8</td><td>####</td></tr> <tr><td>100</td><td>16</td><td>#####</td></tr> <tr><td>100</td><td>32</td><td>#####</td></tr> <tr><td>100</td><td>64</td><td>#####</td></tr> <tr><td>100</td><td>128</td><td></td></tr> </tbody> </table>	stop	go	output	100	1	#	100	2	##	100	4	###	100	8	####	100	16	#####	100	32	#####	100	64	#####	100	128	
stop	go	output																											
100	1	#																											
100	2	##																											
100	4	###																											
100	8	####																											
100	16	#####																											
100	32	#####																											
100	64	#####																											
100	128																												
10.	D	<p>After the values are assigned, the array looks like this:</p> <table border="1" style="display: inline-table; vertical-align: top;"> <tr> <td>index</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td> </tr> <tr> <td>value</td><td>0</td><td>3</td><td>8</td><td>-4</td><td>5</td> </tr> </table> <p>nums[nums[1]] is nums[3] nums[nums.length-2] is nums[3] So -4 is assigned back into the same place.</p>	index	0	1	2	3	4	value	0	3	8	-4	5															
index	0	1	2	3	4																								
value	0	3	8	-4	5																								
11.	C	As the strings are read the delimiter is removed.																											
12.	C	<table border="1" style="display: inline-table; vertical-align: top;"> <tr> <td>m</td><td>p</td><td>n</td> </tr> <tr><td>2</td><td>1</td><td>0</td></tr> <tr><td>2</td><td>2</td><td>1</td></tr> <tr><td>2</td><td>4</td><td>2</td></tr> <tr><td>2</td><td>8</td><td>3</td></tr> <tr><td>2</td><td>16</td><td>4</td></tr> <tr><td>2</td><td>32</td><td>5</td></tr> <tr><td>2</td><td>64</td><td>6</td></tr> <tr><td>2</td><td>128</td><td>7</td></tr> </table>	m	p	n	2	1	0	2	2	1	2	4	2	2	8	3	2	16	4	2	32	5	2	64	6	2	128	7
m	p	n																											
2	1	0																											
2	2	1																											
2	4	2																											
2	8	3																											
2	16	4																											
2	32	5																											
2	64	6																											
2	128	7																											
13.	B	$x + z > y \mid\mid z - \text{Math.abs}(x - y) > z$ $1.5 + -0.75 > 2.25 \mid\mid -0.75 - \text{Math.abs}(1.5 - 2.25) > -0.75$ $1.5 + -0.75 > 2.25 \mid\mid -0.75 - \text{Math.abs}(-0.75) > -0.75$ $1.5 + -0.75 > 2.25 \mid\mid -0.75 - 0.75 > -0.75$ $0.75 > 2.25 \mid\mid -1.5 > -0.75$ false $\mid\mid$ false false																											
14.	A	The range of values for the int data type is -2147483648 to 2147483647																											
15.	B	Removing a non-existent element does not change the array or cause an error.																											
16.	D	The this keyword is a reference to the current object and distinguishes the instance variables from the formal parameters.																											
17.	A	Cannot make a static reference to a non-static instance variable using the class name instead of an object.																											

18.	E	r == p is false because they are different objects. r == o is true because they both point at the same string constant. r.equals(p) is true because the equals method returns true if both store the same string. p.compareTo(s) returns 0 (zero) when p and s are equivalent.
19.	C	Constants in an enumerated data type are accessed by using the type name, dot operator and the constant name.
20.	E	A. $1/2 = 0$ B. Does not convert c to radians. C. Missing the multiplication operator between a and b. D. Missing Math class for call to toRadians method.
21.	C	A lambda expression must be assigned to a variable that is the same type as the functional interface.
22.	D	Loop does not stop when the first occurrence is found. Actually checks every element in the array and then returns the value stored in j.
23.	A	The class InvB implements a stack. The cat method is push. The dog method is pop. The bird method is peek and snake is the isEmpty method. Elements are added and removed in a first in, last out fashion.
24.	C	See #23.
25.	A	poll() retrieves and removes the element at the head of the queue. A queue adds and removes elements in a first in, first out fashion.
26.	B	The Boolean expression for the if statement never evaluates as true so only * are printed. x and y are not decremented and incremented until after a becomes equal to 2 because of short circuit evaluation.
27.	C	The inner for loop is searching for the proper place to insert the current element by working back through the elements that are already in place searching for the first element that is less than the current element.
28.	D	i = 1 [6, 7, 1, 8, 2, 9, 0, 5, 4, 3] i = 2 [1, 6, 7, 8, 2, 9, 0, 5, 4, 3] i = 3 [1, 6, 7, 8, 2, 9, 0, 5, 4, 3] i = 4 [1, 2, 6, 7, 8, 9, 0, 5, 4, 3]
29.	E	best $O(n)$ average $O(n^2)$ worst $O(n^2)$
30.	C	This code segment transposes the values in the matrix across the diagonal that runs from bottom left to top right.
31.	D	$3 * 2^3 = 3 * 8 = 24$
32.	D	$48000 / 12000 = 4$ and $4^2 = 16$
33.	C	An identifier may not begin with any non-letter character other than an underscore (_) or a dollar sign (\$).
34.	B	A. Does not have a return type. C. Returns a int instead of a double. Money will need to be represented as a decimal number. D. Formal parameters do not have a type. E. Does not have a method name.
35.	E	Where A and B are both true the expression evaluates to !T T ^ T = False. This eliminates answer choices A and D. Where A and B are both false the expression evaluates as !F F ^ F = True. This eliminates answer choices B and C.
36.	A	With each call to the method a # is concatenated to the string. When the base case is reached an * is printed FIRST then the single letter r is concatenated to the END of the string.
37.	B	$8 * 3 * 2 + 14 * 5 + - = 24 * 2 + 14 * 5 + - = 26 * 14 * 5 + - = 26 * 19 - = 7$
38.	E	 NOT AND OR
39.	292	ch = a, h = 103 ch = m, h = 212 ch = e, h = 213 ch = x, h = 192 case 'x' is missing a break statement and adds 'c' to the total as well. ch = r, h = 193 ch = c, h = 292
40.	2	The first value (element) inserted into any BST is always be the root node.

UIL COMPUTER SCIENCE WRITTEN TEST

2020 DISTRICT

MARCH 2020

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 – 2020 DISTRICT

Note: Correct responses are based on **Java SE Development Kit 12 (JDK 12)** 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 12 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>Which of the following is NOT equal to $1A_{16} + 23_{16}$?</p> <p>A) $3D_{16}$ B) 61_{10} C) 111101_2 D) 75_8 E) All are equal.</p>	
<p>Question 2.</p> <p>What is the output of the code segment to the right?</p> <p>A) 5 B) 51.28 C) 52 D) 5.52 E) 42</p>	<pre>out.print(19 + 33 - 6 * 3 / 25)</pre>
<p>Question 3.</p> <p>What is the output of the code segment to the right?</p> <p>A) InvAInvB District</p> <p>B) InvA InvBDistrict</p> <p>C) InvA InvB District</p> <p>D) InvA InvB District</p> <p>E) InvAInvBDistrict</p>	<pre>out.print("InvA"); out.println("InvB"); out.print("District");</pre>
<p>Question 4.</p> <p>What is the output of the code segment to the right?</p> <p>A) 2 B) 3 C) 4</p> <p>D) 5 E) 9</p>	<pre>out.print("greenEraser".lastIndexOf('e', 6));</pre>
<p>Question 5.</p> <p>What is the output of the code segment shown on the right?</p> <p>A) true</p> <p>B) false</p>	<pre>boolean yes = true; boolean no = false; out.print(!yes && !no);</pre>
<p>Question 6.</p> <p>What is the output of the code segment to the right?</p> <p>A) 11 B) 11.0 C) 12 D) 12.0 E) 90.0</p>	<pre>out.print(Math.pow(3,2) + Math.sqrt(9));</pre>
<p>Question 7.</p> <p>What is the output of the code segment to the right?</p> <p>A) 53.625 B) 5.625 C) 5.925 D) 5</p> <p>E) There is no output due to an error.</p>	<pre>int i = 10; double d = 6.25; char c = '5'; out.print(c + d / i);</pre>

Question 8.

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

I. 30 II. 14 III. 33 IV. 53 V. 54

- A) I, II, III and V
- B) II and III
- C) III and IV
- D) I, II and V
- E) II and V

```
int x = <value>;
if(x < 30 || x % 2 == 0)
    out.print("Option 1");
else
    out.print("Option 2");
```

Question 9.

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

- A) 38
- B) 10
- C) 11
- D) 12
- E) 33

```
for(int x = 5; x < 38; x += 3)
    out.print("#");
```

Question 10.

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

- A) [4, 5, 2, 3, 3]
- B) [3, 1, 2, 3, 0]
- C) [5, 4, 2, 3, 3]
- D) [4, 5, 2, 3, 5]
- E) There is no output due to an error.

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

```
public class Q11
{
    public static void main(String[] args) throws IOException
    {
        Scanner f = new Scanner(<missing code>);
        while(f.hasNext())
            out.print(f.next());
        f.close();
    }
}
```

Question 11.

Which of the following must replace **<missing code>** in the class shown above to ensure that the class will compile and execute correctly? You may assume that all necessary classes have been imported.

- A) File("data.dat")
- B) new File("data.dat")
- C) new File(data.dat)
- D) "data.dat"
- E) new File()

Question 12.

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

- A) 0 -137
- B) -3 -26
- C) 0 97
- D) -3 -29
- E) 0 -29

```
int x, y = 34;
for(x = 18; x > 0; x -= 3)
    y = y - x;
out.print(x + " " + y);
```

<p>Question 13.</p> <p>What is the correct order of operations for the operators listed on the right?</p> <p>A) I II III B) II III I C) III II I D) II I III E) III I II</p>	<p>I. % II. && III. </p>
<p>Question 14.</p> <p>Which of the following represents the output of the code segment shown on the right?</p> <p>A) -83 B) 83 C) -84 D) 84 E) -82</p>	<pre>out.print(~83);</pre>
<p>Question 15.</p> <p>What is the output of the code segment to the right?</p> <p>A) [1, 5, 3, 3, 4, 9, 7, 2, 8] B) [1, 5, 3, 9, 7, 2, 8] C) [1, 5, 3, 3, 9, 2, 8] D) [1, 5, 3, 3, 9, 7, 2, 8] E) [5, 1, 5, 3, 3, 9, 7, 2, 8]</p>	<pre>int[] i = {5,1,3,4,9,7,2,8}; ArrayList<Integer> nums = new ArrayList<Integer>(); for(int x:i) nums.add(x); nums.set(3, 3); nums.add(2, 5); nums.remove(0); nums.get(5); out.print(nums);</pre>
<p>Question 16.</p> <p>How many ordered pairs make this Boolean expression false?</p> <p>A) 0 B) 1 C) 2 D) 3 E) 4</p>	$\overline{A * B}$
<p>Question 17.</p> <p>What is the output of the code segment shown here?</p> <pre>out.print("static".compareTo("public")>4?"void":"main");</pre> <p>A) void B) main C) true D) false E) 3</p>	
<p>Question 18.</p> <p>Given the declaration of r shown below, which of the following statements will always assign a random whole number to the variable rando that is between 50 (inclusive) and 60 (exclusive)?</p> <pre>Random r = new Random();</pre> <p>A) <code>int rando = r.nextInt(60);</code> B) <code>int rando = (int)r.nextDouble() * 10 + 50;</code> C) <code>int rando = (int)(r.nextDouble() + 50);</code> D) <code>int rando = (int)(r.nextDouble() * 10) + 60;</code> E) <code>int rando = (int)(r.nextDouble() * 10) + 50;</code></p>	

Question 19.

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

- A) 0
- B) 2
- C) 3
- D) 6
- E) 7

Question 20.

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

```
District d1 = new District();
out.println(d1);
```

- A) null 0 0.0
- B) District@4517d9a3
- C) 0 0.0
- D) There is no output and there is no error.
- E) There is no output due to an error.

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

```
public class District {
    private String var1;
    private int var2;
    public double var3;

    public District() {}

    public District(String s, int i, double d)
    {
        var1 = s;
        var2 = i;
        var3 = d;
    }

    public String getVar1() {return var1;}

    public String toString() {
        return var1 + " " + var2 + " " + var3;
    }
}
```

Question 21.

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

```
District d1 = new District("hello",14,3.14);
out.println(d1.var1+" "+d1.var2+" "+d1.var3);
```

- A) hello 14 3.14
- B) 3.14
- C) hello 14
- D) There is no output because this code will not compile.
- E) There is no output because this code throws an exception. }

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);
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.

<p>Question 23.</p> <p>What is printed by the code segment shown on the right?</p> <p>A) 4 B) 9 C) 6 D) 12 E) This segment throws an <code>ArrayIndexOutOfBoundsException</code></p>	<pre>int[][] mat = new int[4][5]; for(int r = 1; r < mat.length; r++) for(int c = 1; c < mat[r].length; c++) mat[r][c] = r * c; out.println(mat[3][3]);</pre>
<p>Question 24.</p> <p>Which of the following must replace <code><missing code></code> to instantiate a <code>Stack</code> object that will store <code>String</code> objects?</p> <p>A) <code>Stack<String> s = new Stack<String>();</code> B) <code>Stack<String> s = new Stack(String);</code> C) <code>Stack<String> s = new Stack<>();</code> D) <code>Stack<String> s = Stack<String>();</code> E) More than one of the above.</p>	<p>//Use the following code segment //to answer questions 24, 25 and 26.</p> <p><missing code></p>
<p>Question 25.</p> <p>What is the output of line #1 shown on the right?</p> <p>A) monday B) tuesday C) wednesday D) thursday E) friday</p>	<pre>s.push("monday"); s.add("tuesday"); s.push("wednesday"); s.push("thursday"); s.add(1, "friday"); Queue<String> q = new LinkedList<String>(); while(!s.empty()) q.add(s.pop()); while(!q.isEmpty()) s.push(q.poll()); out.println(s.peek()); //line #1 out.println(q.peek()); //line #2</pre>
<p>Question 26.</p> <p>What is the output of line #2 shown on the right?</p> <p>A) monday B) tuesday C) friday D) null E) There is no output due to an error.</p>	<p>(This cell contains the continuation of the code from the previous row, which is already shown in the right column of the previous row.)</p>
<p>Question 27.</p> <p>Which of the following is the 8-bit binary two's complement equivalent of -78?</p> <p>A) 10110001 B) 01001110 C) 10110010 D) 01001101 E) 00010110</p>	<p>(This cell is empty.)</p>

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

```
public static void sort(ArrayList<String> list)
{
    for(int i = 1; i < list.size();i++)
    {
        String current = list.get(i);
        int k = i - 1;
        while(<code 1>)
        {
            <code 2>;
            k--;
        }
        list.set(k + 1, current);
        out.println(list);
    }
}
```

Question 28.

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

- A) `k <= 0 && list.get(k).compareTo(current) <= 0`
- B) `k >= 0 && list.get(k).compareTo(current) > 0`
- C) `k >= 0 && list.remove(k).compareTo(list.get(k + 1)) > 0`
- D) `list.get(k).compareTo(current) < 0`
- E) `k >= 0 && list.get(k).compareTo(i) > 0`

Question 29.

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

- A) `list.set(k, list.get(k + 1))`
- B) `list.set(i, list.get(k))`
- C) `list.set(k + 1, current)`
- D) `list.set(k + 1, list.get(k))`
- E) `list.add(k + 1, list.get(k))`

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 printed when *i* equals 3?

- A) [aardvark, cat, dog, panda, zebra, mouse]
- B) [aardvark, dog, cat, mouse, panda, zebra]
- C) [aardvark, mouse, panda, dog, zebra, cat]
- D) [panda, zebra, cat, mouse, dog, aardvark]
- E) [aardvark, dog, mouse, panda, zebra, cat]

```
ArrayList<String> list = new
ArrayList<String>();
list.add("panda");list.add("mouse");
list.add("aardvark");list.add("dog");
list.add("zebra");list.add("cat");
sort(list);
```

Question 31.

If it takes `sort` 4 seconds to sort 3 million elements, what is the best estimate of how long will it take to sort 6 million elements?

- A) 16 seconds
- B) 8 seconds
- C) 5 seconds
- D) 12 seconds
- E) 9 seconds

Question 32.

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

- A) de#yk
- B) adcxdz
- C) ac9e#3py7xz6yk
- D) 3
- E) adc9de# py7xdz6yk

```
Scanner s = new
Scanner("adc9de#3py7xdz6yk");
s.useDelimiter("\\d");
while(s.hasNext())
    if(s.next().matches("\\w{3}")
        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 -9?

```
int num = 8;
<missing code>
System.out.print(x.apply(num));
```

- A) Function<Integer, Integer> num = y -> {y++; y = -y; return y};
- B) Function<Integer, Integer> x = y -> {y++; y = -y; return y};
- C) Function<Integer> x = y -> {y++; y = -y; return y};
- D) Function<Integer, Integer> x = {y++; y = -y; return y};
- E) Function<Integer, Integer> x = y -> {y++; y = -y};

Question 34.

Which of the following must replace /*code*/ in the class shown on the right?

- A) implements Comparable<>
- B) extends Comparable<WeirdString>
- C) implements Comparable
- D) implements Comparable<WeirdString>
- E) implements Comparable<String>

```
//Use class WeirdString to answer questions
// 34 and 35
```

```
public class WeirdString /*code*/{
    private String str;

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

    public int compareTo(WeirdString ws) {
        String s = ws.toString();
        if(str.length() > s.length())
            return 1;
        else if(str.length() < s.length())
            return -1;
        else
            return str.compareTo(s);
    }

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

Question 35.

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

```
ArrayList<WeirdString> ws = new
ArrayList<WeirdString>();
ws.add(new WeirdString("apple"));
ws.add(new WeirdString("zebra"));
ws.add(new WeirdString("aardvark"));
ws.add(new WeirdString("monday"));
Collections.sort(ws);
for(WeirdString weird:ws)
    out.print(weird + " ");
```

- A) aardvark monday apple zebra
- B) aardvark apple monday zebra
- C) zebra monday apple aardvark
- D) apple aardvark monday zebra
- E) apple zebra monday aardvark

Question 36.

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

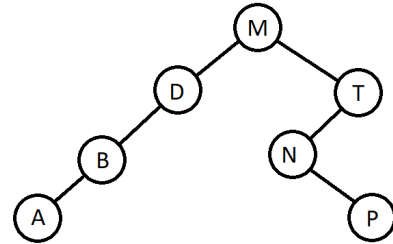
- A) 8
- B) 64
- C) 10
- D) 00000008
- E) 1000

```
out.print(Integer.toString(8, 8));
```

Question 37.

Which of the following represents a pre-order traversal of the binary search tree shown on the right?

- A) M D B A T N P
- B) A B D P N T M
- C) A B D M N P T
- D) T P N M D B A
- E) A P B N D T M

**Question 38.**

Which of the following would be returned if x is assigned 4 when the method is called?

- A) 80
- B) 72
- C) 84
- D) 96
- E) 78

```
public static int rec(int x)
{
    if(x == 0)
        return 4;
    else if(x == 1)
        return 12;
    else
        return rec(x - 1) + 2 * rec(x - 2);
}
```

Question 39.

Write the output of the code segment shown on the right in the blank provided on the answer document.

```
int a = 5, b = 3, c = 2;
while(c>0) {
    int d = b << c;
    a &= d;
    b--;
    c--;
}
out.println(a);
```

Question 40.

How many edges does a complete graph with 8 vertices have? Write your answer in the blank provided on the answer document.

★ ANSWER KEY – CONFIDENTIAL ★

UIL COMPUTER SCIENCE – 2020 DISTRICT

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

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

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

Note: Correct responses are based on **Java SE Development Kit 12 (JDK 12)** 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 12 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used.

Explanations:

1.	E	$1A_{16} = 26_{10}$ $23_{16} = 35_{10}$ $26 + 35 = 61$ Answer choices A, B, C and D are all equal to 61_{10} .																														
2.	C	$19 + 33 - 6 * 3 / 25 = 19 + 33 - 18 / 25 = 19 + 33 - 0 = 19 + 33 = 52$																														
3.	A	print leaves the cursor on the line, println moves the cursor to the next line.																														
4.	B	lastIndexOf('e',6) finds the last occurrence of 'e' before index value 6.																														
5.	B	!T && !F = F && T = F																														
6.	D	$3^2 + 3 = 9 + 3 = 12.0$ Both of the methods return a double value.																														
7.	A	$53 + 6.25 / 10 = 53 + 0.625 = 53.625$																														
8.	D	$x < 30 x \% 2 == 0$ evaluates as true whenever x is less than 30 or an even number. 30 and 54 are both even and 14 is less than 30 and even.																														
9.	C	x takes the values 5 8 11 14 17 20 23 26 29 32 and 35 and prints each time.																														
10.	E	The third line in the segment throws an IndexOutOfBoundsException because num[1] is 5. The last index in the array is 4.																														
11.	B	To associate a file with a Scanner object a new File object must be constructed and passed to the Scanner class constructor.																														
12.	E	$x = 0$ $y = 34$ $x = 18$ $y = 16$ $x = 15$ $y = 1$ $x = 12$ $y = -11$ $x = 9$ $y = -20$ $x = 6$ $y = -26$ $x = 3$ $y = -29$ $x = 0$ $y = -29$																														
13.	A	Order of precedence is % then && then .																														
14.	C	~ is the complement operator. Add one to the operand and take the opposite sign.																														
15.	D	Original list - [5, 1, 3, 4, 9, 7, 2, 8] nums.set(3, 3); - [5, 1, 3, 3, 9, 7, 2, 8] nums.add(2, 5); - [5, 1, 5, 3, 3, 9, 7, 2, 8] nums.remove(0); - [1, 5, 3, 3, 9, 7, 2, 8] nums.get(5); - [1, 5, 3, 3, 9, 7, 2, 8]																														
16.	B	!(A&&!B) can be simplified to !A B <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>A</th> <th>B</th> <th>!A B</th> </tr> </thead> <tbody> <tr> <td>T</td> <td>T</td> <td>T</td> </tr> <tr> <td>T</td> <td>F</td> <td>F</td> </tr> <tr> <td>F</td> <td>T</td> <td>T</td> </tr> <tr> <td>F</td> <td>F</td> <td>T</td> </tr> </tbody> </table>	A	B	!A B	T	T	T	T	F	F	F	T	T	F	F	T															
A	B	!A B																														
T	T	T																														
T	F	F																														
F	T	T																														
F	F	T																														
17.	B	"static".compareTo("public") returns 3. 3 is less than 4 so "main" is printed.																														
18.	E	nextDouble() returns a double value between 0 (inclusive) and 1 (exclusive). Multiplying by 10 gets a double value between 0 (inclusive) and 10 (exclusive). This number is cast to a whole number using (int). Then add 50 to get a range of 50 (inclusive) to 60 (exclusive).																														
19.	C	s, i and d are all local to the District constructor and are therefore not instance variables.																														
20.	A	The default constructor does not assign values to the instance variables so the default values for each are printed by the toString method.																														
21.	D	var1 and var2 are private.																														
22.	A	All classes automatically extend the Object class.																														
23.	B	No values are added to the first row or column. The remaining cells are assigned the product of the row and column. <table border="1" style="margin-left: 20px;"> <tbody> <tr> <td></td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> </tr> <tr> <td>2</td> <td>0</td> <td>2</td> <td>4</td> <td>6</td> <td>8</td> </tr> <tr> <td>3</td> <td>0</td> <td>3</td> <td>6</td> <td>9</td> <td>12</td> </tr> </tbody> </table>		0	1	2	3	4	0	0	0	0	0	0	1	0	1	2	3	4	2	0	2	4	6	8	3	0	3	6	9	12
	0	1	2	3	4																											
0	0	0	0	0	0																											
1	0	1	2	3	4																											
2	0	2	4	6	8																											
3	0	3	6	9	12																											

24.	E	In answer choice A the type String is explicitly stated. In answer choice C the type is inferred from the variable s. Both are legal declarations.
25.	A	push adds an element to the top of the stack. add adds an element to the top of the stack. add(1, "Friday") places the element at index 1 within the stack. All of the elements in s are popped into the queue q. All of the elements in q are pushed back into the stack s until q is empty. s.peek() returns the element at the top of the stack s which is "Monday"
26.	D	q is empty. q.peek() returns null.
27.	C	78 is 01001110_2 Take the complement to get 10110001 Add one to get 10110010
28.	B	The while loop is working backwards searching for the first element that comes before current alphabetically.
29.	D	Shift each element to the right until the correct location for current is found.
30.	E	i = 1 [mouse, panda, aardvark, dog, zebra, cat] i = 2 [aardvark, mouse, panda, dog, zebra, cat] i = 3 [aardvark, dog, mouse, panda, zebra, cat] i = 4 [aardvark, dog, mouse, panda, zebra, cat] i = 5 [aardvark, cat, dog, mouse, panda, zebra]
31.	A	The run time efficiency for an insertion sort is $O(n^2)$. 6 million is 2 times greater than 3 million. $2^2 = 4$. $4 \times 4 = 16$.
32.	A	\d is the regex character for any digit. \w{3} means any three word characters. Each call to next() (including in the if statement) move the pointer to the next token.
33.	B	A re-declares num as a functional type. C The interface Function requires two types. D Missing the lambda operator. E Does not have a return statement.
34.	D	The formal parameter of the compareTo method must match the type to which the class has been parameterized.
35.	E	WeirdString objects are first compared by length where shorter strings come before longer ones and then alphabetically.
36.	C	Integer.toString(8,8) returns the string representation of the decimal value 8 in base 8.
37.	A	A pre-order traversal visits the root, then left node, then right node.
38.	C	rec(4) = rec(3) + 2 * rec(2) = 44 + 2 * 20 = 84 rec(3) = rec(2) + 2 * rec(1) = 20 + 2 * 12 = 44 rec(2) = rec(1) + 2 * rec(0) = 12 + 2 * 4 = 20 rec(1) = 12 rec(0) = 4
39.	4	<< is the same as multiplying by that power of 2. &= is the compound assignment operator for bitwise AND. a b c d 5 3 2 4 2 1 12 4 1 0 4
40.	28	The formula is $\frac{1}{2} n(n-1) = \frac{1}{2} 8(8-1) = 28$