UIL COMPUTER SCIENCE WRITTEN TEST

2026 DISTRICT

MARCH 2026

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 – 2026 DISTRICT

Note: Correct responses are based on **Java SE Development Kit 22 (JDK 22)** 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 22 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 the expression $456_8 + 711_8$?

A) 745_{10} **B)** 744_8 **C)** $2E7_{16}$ **D)** 1011100111_2 **E)** 1367_8

Question 2

What is output by the code to the right?

A) `10.5`

B) `24`

C) `23.25`

D) `23`

E) `25`

```
int a = 15;
int b = 4;
int c = 3;

int result = a + b * c - a / b;

out.println(result);
```

Question 3

What is output by the code to the right?

A)
```
Val: 5 and 3.50
Path: C:\\temp
```

B)
```
Val:
5 and 3.5
Path: C:\temp
```

C)
```
Val: 5 and 3.5
Path: C:\temp
```

D)
```
Val: 5 and 3.5
Path: C:\\temp
```

```
int x = 5;
double y = 3.5;
out.print("Val: ");
out.printf("%d and %.1f\n", x, y);
out.println("Path: C:\\temp");
```

Question 4

What is output by the code to the right?

A) `hexlo 5`

B) `Hello World 11`

C) `Hexxo 5`

D) `hexxo 5`

E) `hello 5`

```
String text = "  Hello World  ";
String result = text.trim().toLowerCase()
    .substring(0, 5).replace('l', 'x');

out.println(result + " " +
result.length());
```

Question 5

What is output by the code to the right?

A) `truetruefalsefalse`

B) `falsetruetruefalse`

C) `truetruetruefalse`

D) `falsetruefalsefalse`

E) `falsefalsefalsefalse`

```
boolean a = true;
boolean b = false;
boolean c = true;

boolean r1 = a && b || c;
boolean r2 = a || b && !c;
boolean r3 = a ^ c;
boolean r4 = !(a && c) || b;

out.println(r1 + "" + r2 + r3 + r4);
```

Question 6

What is output by the code to the right?

A) 9267

B) 8367

C) 8377

D) 826-7

```
double a = Math.floor(8.9);
double b = Math.ceil(2.1);
long c = Math.round(6.4);
int d = Math.abs(-7);

out.println((int)a + "" + (int)b + c + d);
```

Question 7

What is output by the code to the right?

A) 89 **B)** 88 **C)** 90 **D)** 91 **E)** 88

```
int a = 26 & 21 + 3;
int b = 67 | 420 / 69;
a ^= b - 13 / 2;
out.println(a);
```

Question 8

What is output by the code to the right?

A) B

B) B1

C) B12

D) C2

E) A12

```
int x = 7;
int y = 3;
String result = "";

if (x > 5)
    if (y > 5)
        result = "A";
    else
        result = "B";
else if (x > 3)
    result = "C";
else
    result = "D";

switch (result) {
    case "A":
    case "B":
        result += "1";
    case "C":
        result += "2";
        break;
    default:
        result += "3";
}
out.println(result);
```

Question 9

On which line of code to the right marked by comments should we place the following loop to center the output pyramid?

```
for(int j = 4; j > i; j--) {
 out.print(" ");
}
```

A) line 1 **B)** line 2

C) line 3 **D)** line 4

```
for (int i=1; i<=4;i++) {
    //line 1
    for (int k=1; k<=2*i-1;k++) {
        //line 2
        out.print("*");
        // line 3
    }
    // line 4
    out.println();
}
```

Question 10

What is output by the code to the right?

A) `10 6 21`

B) `6 4 15`

C) `10 4 14`

D) `10 4 19`

```
int[] arr = {2, 4, 6, 8, 10};

arr[1] = arr[0] + arr[3];
arr[4] = arr[1] - arr[2];

int result = arr[1] + arr[4] + arr.length;

out.println(arr[1] + " " + arr[4] + " " +
            result);
```

Question 11

Given a file "`data.txt`" containing the below data. What is output by the code to the right?

```
Hello World
25 3.75
```

A) `Hello 25 28`

B) `Hello World 25 28`

C) `Hello 25 29`

D) `World 25 28`

```
Scanner file = new Scanner(
        new File("data.txt"));

String word = file.next();
file.nextLine();
int num = file.nextInt();
double dec = file.nextDouble();

file.close();

out.println(word + " " + num + " " +
            (num + (int)dec));
```

Question 12

What is output by the code to the right?

A) `15 12`

B) `12 15`

C) `12 8`

D) `20 15`

E) `6 15`

```
int[] nums = {3, 5, 2, 4, 6};
int sum = 0;
int product = 1;
for (int n : nums) {
    if (n % 2 == 0) {
        sum += n;
    } else {
        product *= n;
    }
}
out.println(sum + " " + product);
```

Question 13

What is output by the code to the right?

A) `12 3 5 1`

B) `10 3 5 2`

C) `13 3 5 1`

D) `14 3 5 1`

```
int a = 3;
int b = 4;
int c = 2;
int result = a + ++b * c-- - b % c;
out.println(result + " " + a + " " +
            b + " " + c);
```

Question 14

Which of the following statements about Java primitive data types is TRUE?

A) A `byte` can store values from `-128` to `128`.

B) When a `short` with value `32767` is incremented, it becomes `32768`.

C) A `long` and a `double` both use `32` bits of memory.

D) When a `byte` with value `-128` is decremented, it becomes `127`.

E) An `int` can store any value that a `long` can store.

Question 15

What is output by the code to the right?

A) `15 3`

B) `20 3`

C) `15 4`

D) `30 3`

E) There is no output due to a runtime error.

```
ArrayList<Integer> list =
        new ArrayList<>();

list.add(10);
list.add(20);
list.add(30);
list.add(1, 15);
list.remove(2);
list.set(0, 5);

out.println(list.get(1) + " " +
            list.size());
```

Question 16

What is output by the code to the right given that the file `a.txt` does not exist?

A) `23`

B) `13`

C) There is no output due to a compile error.

D) There is output, followed by a runtime error.

E) There is no output due to a runtime error.

```
try {
   String i = "a.txt";
   File f = new File(i);
   Scanner s = new Scanner(i);
   s.nextLine();
   out.print(1);
} catch (IOException e) {
   out.print(2);
} finally {
   out.print(3);
}
```

Question 17

What could replace **<1*>** in the code to the right so that it will compile and run without error (removing items).

A) `pop` **B)** `remove` **C)** `poll`

D) A and B. **E)** All of the above.

Question 18

Assuming **<1*>** is filled in correctly, what is output by the line marked `//q18` code to the right?

A) `[1, 189, 3]` **B)** `[189, 3]`

C) Output cannot be determined until runtime

D) There is no output due to a compile error.

E) There is no output due to a runtime error.

Question 19

Assuming **<1*>** is filled in correctly, what is output by the line marked `//q19` code to the right?

A) `189` **B)** `true` **C)** `false`

D) There is no output due to a compile error

E) There is no output due to a runtime error.

```
LinkedHashSet<Integer> struct;
struct = new LinkedHashSet<>();

struct.add(1);
struct.add(2);
struct.add(1);
struct.add(189);
struct.add(3);
struct.<1*>(2);
struct.<1*>(1);

out.println(struct); //q18
out.println(struct.remove(0)); //q19
```

Question 20

Which of the lines in the code to the right first causes an error?

A) `//1`

B) `//2`

C) `//3`

D) `//4`

E) None of the above lines cause any errors.

```
double g = Math.ceil(6); //1
int i = Math.min(g, 2);  //2
i = Math.round(g);       //3
g = Math.log(i);         //4
```

Question 21

What can replace **<1*>** in the code to the right so that the `wings` instance variable is assigned value `2` and compiles without error?

A) `static` **B)** `static final`

C) `final` **D)** Nothing is required.

E) B or C. **F)** Any of the above.

Question 22

Which of the following describes the scope of the `city` instance variable in the `Pigeon` class?

A) More restrictive than `private`.

B) Less restrictive than `private`; more than `protected`.

C) Less restrictive than `protected`; more than `public`.

D) Less restrictive than `public`.

Question 23

What can replace **<2*>** in the code to the right so that the `fly()` method returns value `2` and the `Bird` interface compiles without warning?

A) `{return wings;}` **B)** `;`

C) `{return 2;}` **D)** A or C.

E) Any of the above.

Question 24

What can replace **<3*>** in the code to the right so that the `fly()` method returns the value `2`?

A) `super.wings` **B)** `wings`

C) `2` **D)** B or C.

E) Any of the above.

Question 25

Assuming **<1*>**, **<2*>**, and **<3*>** are filled in correctly, what is output by the line marked `//q27` in the code to the right, assuming any errors below line `//q27` are commented out?

A) `Caw!Caw!Caw!`

B) `Caw! NYC Caw! LA Caw!`

C) `Caw! NYC Caw! LA Caw! Miami`

D) There is no output due to a compile error.

E) There is no output due to a runtime error.

Question 26

Assuming **<1*>**, **<2*>**, and **<3*>** are filled in correctly, and any error from the previous question has been commented out, what is output by the line marked `//q28` in the code to the right?

A) `0 0 0` **B)** `2 2 2`

C) `2 0 0` **D)** `2 2 0`

E) There is no output due to a compile error.

```
interface Bird {
    <1*> int wings = 2;
    public String call();
    private int fly() <2*>
}

class Pigeon implements Bird {
    String city;

    public Pigeon(String c) {
        city = c;
    }

    public String call() {
        return "Caw! " + city;
    }

    public int fly() {
        return <3*>;
    }
}

////////////client code///////////
Pigeon p1 = new Pigeon("NYC");
Pigeon p2 = new Pigeon("LA");
Bird b1 = new Pigeon("Miami");
String o = p1.call();
o += " " + p2.call();
o += " " + b1.call();
out.println(o); //q27
o = "" + p1.fly();
o += " " + p2.fly();
o += " " + b1.fly();
out.println(o); //q28
```

Question 27

What is output if the following call is made to the function `mystery` to the right? `mystery("Ahs2");`

A) `-102` **B)** `108`

C) `99` **D)** `117`

E) There is no output due to a runtime error.

Question 28

What is output if the following call is made to the function `mystery` to the right? `mystery("K38knd38kOP1");`

A) `210` **B)** `-183`

C) `-237` **D)** `-234`

E) There is no output due to a runtime error.

```
int mystery(String s) {
  if (s.isEmpty())
    return 1;
  char c = s.charAt(0);
  s = s.substring(1);
  if (Character.isUpperCase(c))
    return 3 * mystery(s);
  if (Character.isDigit(c))
    return 5 + mystery(s);
  if (c < '1')
    return -1 * mystery(s);
  return 1 + 2 * mystery(s);
}
```

Question 29

What is the intended purpose of method the method `goodtime(int i, int j)` to the right?

A) Return the greatest common divisor of two numbers.

B) Return the least common multiple of two numbers.

C) Return the least prime number between the two.

D) Return the greatest prime number between the two.

E) Return the product of greatest factors of two numbers.

Question 30

What is output by the client code to the right?

A) `1 1` **B)** `216 216`

C) `36 126` **D)** `6 1`

E) There is no output due to a runtime error.

```
int goodtime(int i, int j) {
    if(i == 0 || j == 0)
        return 0;
    int s = Math.abs(i * j);
    while(j != 0) {
        int t = i % j;
        i = j;
        j = t;
    }
    return s / i;
}
////////////client code///////////
out.print(goodtime(12, 18));
out.print(" ");
out.print(goodtime(14, 9));
```

Question 31

What is output by the client code to the right?

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

C) `true false` **D)** `true true`

E) There is no output due to a compile error.

```
String str = "BEEPbeepONtheSTREET";
String reg =
    "([A-Z]+[a-z]+)*|((\\w\\w)+)+";
out.print(str.matches(reg) + " ");
reg = "\\w\\D{10,17}....";
out.println(str.matches(reg));
```

Question 32

What is output by the code to the right?

A) `TT` **B)** `TF`

C) `FT` **D)** `FF`

E) There is no output due to a runtime error.

```
int a = 5;
double b = 5.2;
out.print((a >= b) ? 'T' : 'F');
out.println((b <= a) ? 'T' : 'F');
```

Question 33

Which of the following ranges accurately describes the list of possible integers that `rand` can be?

A) [5,18] **B)** [5,18)

C) [5,13] **D)** [5,13)

E) None of the above.

```
int rand = (Math.random() * 13) + 5;
```

Question 34

What is output by the line marked `//Q34` in the code to the right?

A) `1` **B)** `4` **C)** `5`

D) There is no output due to a runtime error as the `byte[]` entity does not implement `hashCode()`, which is required for all types that are stored in a `HashSet`.

E) There is no output due to a compile time error as the `byte` arrays are being filled with `char`'s, not `byte`'s.

Question 35

Disregarding the output produced by the line marked `//Q34` in the code to the right, what is output by the code to the right?

A)
```
[97, 98, 99, 100]
[100, 97, 98, 99]
[99, 100, 97, 98]
[97, 98, 99, 100]
[98, 99, 100, 97]
```

B)
```
abcd
bcda
cdab
dabc
abcd
```

C)
```
[a, b, c, d]
[b, c, d, a]
[c, d, a, b]
[d, a, b, c]
```

D) It is impossible to know the output until runtime as the output will be dependent on the memory addresses of the arrays, since arrays use the identity hash code.

E) It is impossible to know the output until runtime due to the nature of it being a `HashSet` and storing arrays; however, the output would have been a permutation of the lines from one of either option A, B, or C.

F) There is still no output due to an error.

Question 36

Consider if `uniques` was instead storing `char[]`'s, each newly created array added to `uniques` was instead of type `char[]`, and `unique` was of type `char[]` in the for loop. Then, what is output by the line marked `//Q34` in the code to the right?

A) `1` **B)** `4` **C)** `5`

D) There is no output due to a runtime error as the `char[]` entity does not implement `hashCode()`, which is required for all types that are stored in a `HashSet`.

E) There is no output due to a compile time error.

Question 37

Using the same situation described by Question 36 above, and disregarding the output produced by the line marked `//Q34`, what is output by the code to the right?

A) The same answer as option A from question 35.

B) The same answer as option B from question 35.

C) The same answer as option C from question 35.

D) The same answer as option D from question 35.

E) The same answer as option E from question 35.

F) The same answer as option F from question 35.

```
HashSet<byte[]> uniques =
        new HashSet<byte[]>();
uniques.add(new byte[] {
        'a', 'b', 'c', 'd'
});
uniques.add(new byte[] {
        'b', 'c', 'd', 'a'
});
uniques.add(new byte[] {
        'c', 'd', 'a', 'b'
});
uniques.add(new byte[] {
        'd', 'a', 'b', 'c'
});
uniques.add(new byte[] {
        'a', 'b', 'c', 'd'
});

out.println(uniques.size()); //Q34
for (byte[] unique : uniques) {
    out.println(unique);
}
```

Question 38

What is output by the code to the right?

A) T1T1T1T1T1T1T1T1T1T1T2T2T2T2T2T2T2T2T2T2

B) T2T2T2T2T2T2T2T2T2T2T1T1T1T1T1T1T1T1T1T1

C) T1T2T1T2T1T2T1T2T1T2T1T2T1T2T1T2T1T2T1T2T1T2

D) T2T1T2T1T2T1T2T1T2T1T2T1T2T1T2T1T2T1T2T1T2T1

E) It is impossible to determine the output until runtime.

F) There is no output due to an error.

```
final Lock lock = new ReentrantLock();
StringBuilder sb = new StringBuilder();

Thread t1 = new Thread(() -> {
    for (int i = 0; i < 10; i++) {
        lock.lock();
        sb.append("T1");
        lock.unlock();
    }
});
Thread t2 = new Thread(() -> {
    for (int i = 0; i < 10; i++) {
        lock.lock();
        sb.append("T2");
        lock.unlock();
    }
});

t1.start();
t2.start();
try {
    t1.join();
    t2.join();
} catch (InterruptedException e) {
    e.printStackTrace();
}
out.println(sb);
```

Question 39

What is the ***height*** of the Binary Search Tree (BST) of integers resulting from inserting the elements to the right in order? Write your answer in the blank provided on your answer sheet for this question.

Question 40

What is the ***maximum width*** of the Binary Search Tree (BST) of integers resulting from inserting the elements to the right in order? Write your answer in the blank provided on your answer sheet for this question.

```
First: 59
  |    56
  |    42
  |    16
  |    72
  |    83
  |    77
  |    58
  v    32
 Last: 73
```