UIL COMPUTER SCIENCE WRITTEN TEST

2022 INVITATIONAL A

JANUARY/FEBRUARY 2022

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 (best) answer. There is a 2-point penalty for each incorrect answer.
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 – 2022 INVITATIONAL A

Note: Correct responses are based on **Java SE Development Kit 14 (JDK 17)** 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 17 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 decimal numbers is equivalent to the binary number 100101001_2?

A) 247 **B)** 297 **C)** 177 **D)** 231 **E)** 144

Question 2

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

A) 0 **B)** 0.2 **C)** 73 **D)** 72 **E)** 72.2

```
out.print(9 * 8 + 7 % 6 / 5);
```

Question 3

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

A) 002.34
B) 02.34
C) 02.35
D) 002.35
E) None of the above

```
out.printf("%05.2f", 2.345);
```

Question 4

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

A) 4
B) 5
C) 6
D) 7
E) 8

```
String str = "hello world";
out.print(str.lastIndexOf('o'));
```

Question 5

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

A) true
B) false

```
boolean t = true;
boolean f = false;
out.print(f^t^t^f);
```

Question 6

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

A) 2.5 **B)** 3.0 **C)** 2.0 **D)** 2 **E)** 3

```
out.print(Math.round(2.45));
```

Question 7

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

A) 16.0 **B)** 17.0 **C)** 18.0 **D)** 19.0 **E)** 20.0

```
int x = 6, y = 10;
double a = 1.5, b = 1.25;
out.print(x * a + y / b);
```

Question 8

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

A) fizz
B) fizzbuzz
C) fizzbuzz!
D) There is no output due to a runtime error.
E) There is no output due to a compile error.

```
switch("test"){
   case "test":
      out.print("fizz");
   case "Test":
      out.print("buzz");
      break;
   case "TEST":
      out.print("!");
}
```

Question 9

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

A) ********

B) ******

C) *****

D) *******

E) ***

```
String str = "";
for(int i = 0;i<3;i++)
{
   str = str + "*";
   out.print(str);
}
```

Question 10

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

A) [6, 5, 1, 3, 1, 1]

B) [6, 5, 2, 3, 2, 1]

C) [6, 5, 2, 3, 2, 1]

D) [6, 5, 2, 3, 4, 1]

E) There is no output due to an error.

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

Question 11

The contents of input.txt are as follows:

```
9
10
11
12
```

What is output by the code segment to the right?

A) 9

B) 18

C) 19

D) 42

E) There is no output due to an error

```
File f = new File("input.txt");
Scanner scan = new Scanner(f);
int sum = 0;
for(int i = 0;i<2;i++)
{
   sum += scan.nextInt();
   scan.nextLine();
}
out.print(sum);
```

Question 12

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

A) 18

B) 15

C) 24

D) 16

E) 10

```
int a = 1;
int h = 0;
for(int g = 1; g <= 5; g++) {
      h += a;
     a++;
}
out.print(h);
```

Question 13

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

A) 5

B) 3

C) 2

D) 14

E) 16

```
int a = 4, b = 2, c = 1;
out.print(a++ + ++b >> c);
```

Question 14

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

A) 8

B) 16

C) 32

D) 64

E) 4

```
out.println(Float.SIZE);
```

Question 15

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

A) [7, 5, 3, 1]

B) [1]

C) [7, 5, 3]

D) [7, 5, 1]

E) There is no output due to a runtime error

```
ArrayList<Integer> list;
list = new ArrayList<Integer>();
list.add(7);
list.add(5);
list.add(3);
list.add(1);
list.remove(3);
out.println(list);
```

Question 16

How many combinations of assigned values make the following boolean expression true?

A) 0 **B)** 2 **C)** 4 **D)** 6 **E)** 7

$$(A + B) \oplus (C + D)$$

Question 17

What is output by a call to `M1(12,12)`?

A) 6 6 1

B) 12 12 0

C) 0 0 1

D) 12 12 1

E) There is no output due to an infinite loop

Question 18

What is output by a call to `M1(108,24)`?

A) 12 12 5

B) 24 24 4

C) 12 12 6

D) 6 6 5

E) There is no output due to an infinite loop

```
public static void M1(int x, int y){
   int z = 0;
   do {
      x -= y;
      if(y > x)
      {
         int save = x;
         x = y;
         y = save;
      }
      z++;
   }while(x != y);
   out.print(x+" "+y+" "+z);
}
```

Question 19

How many asterisks are printed by `M2(10)`?

A) 40
B) 44
C) 50
D) 21
E) none of the above

Question 20

How many asterisks are printed by `M2(16)`?

A) 60
B) 64
C) 80
D) 41
E) none of the above

```
public static void M2(int n){
   for(int i = 0;i<n;i++)
   {
      for(int j = 1;j<n;j*=2)
      {
         out.print("*");
      }
   }
}
```

Question 21

What is the tightest upper bound for the Big O complexity of `M2`?

A) O (n)
B) O (nlogn)
C) O (n^2)
D) O (2^n)
E) none of the above

Question 22

Which of the following could be generated by the following line of code?

```
out.print((int)(Math.random() * 2 + 10));
```

I. 9 II. 10 III. 11 IV. 12

A) I and II
B) II and III
C) III and IV
D) II, III, and IV

Question 23

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

A) 8
B) 16
C) 9
D) 17
E) None of the above

```
out.print(1 << 3 + 1);
```

Question 24

What is printed by the line labeled **<Code Location 1>** shown on the right?

A) 0

B) 4

C) 5

D) 6

E) There is no output due to an error.

```
TreeSet<Integer> set;
set = new TreeSet<Integer>();
set.add(6);
set.add(9);
set.add(5);
set.add(0);
set.add(2);
set.add(4);
//<Code Location 1>
out.print(set.floor(5));
//<Code Location 2>
out.print(set.higher(5));
```

Question 25

What is printed by the line labeled **<Code Location 2>** shown on the right?

A) 4

B) 5

C) 6

D) 9

E) There is no output due to an error.

Question 26

Which of the following is closest to the expected output of the code to the right?

A) 500000.0

B) 0.5

C) 0.0

D) 1.0

E) -1.0

```
int it = 1000000;
double sum = 0;
for(int i = 0;i<it;i++)
{
   sum += Math.random();
}
out.println(sum/it);
```

Question 27

What is output by the following client code?

```
DataStructure<Integer> ds;
ds = new DataStructure<>();
for(int i: new int[] {1,5,2,7,4,8,9})
   ds.add(i);
while(!ds.isEmpty())
   out.print(ds.remove()+" ");
```

A) 1 5 2 7 4 8 9

B) 9 8 4 7 2 5 1

C) 1 2 4 5 7 8 9

D) 9 8 7 5 4 2 1

E) There is no output due to a runtime error

Question 28

What is output by the following client code?

```
DataStructure<Integer> ds;
ds = new DataStructure<>();
for(int i: new int[] {1,5,2,7,4,8,9})
   ds.add(i);
while(!ds.isEmpty())
   ds.remove();
out.print(ds.next());
```

A) 0

B) o

C) null

D) 1

E) There is no output due to a runtime error

Question 29

What data structure does DataStructure represent?

A) Queue

B) PriorityQueue

C) Stack

D) Set

E) none of the above

Question 30

The O class is an example of what?

A) recursion

B) encapsulation

C) enumeration

D) inheritance

E) none of the above

```
private class DataStructure<E>{
   private class O{
      E elem;
      O next;
      public O(E e, O n)
      {
         this.elem = e;
         this.next = n;
      }
   }

   O h;
   int sz;

   public void add(E e)
   {
      h = new O(e, h);
      sz++;
   }

   public E next()
   {
      return h.elem;
   }

   public E remove()
   {
      E e = next();
      h = h.next;
      sz--;
      return e;
   }

   public boolean isEmpty() {
      return sz == 0;
   }
}
```

Question 31

Which of the following is the output of the line shown below?

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

A) -1

B) 0

C) 1

D) 2

E) There is no ouput due to a runtime error

Question 32

What is output by the code to the right?

A) [3, 2, 3]

B) [2, 1, 2]

C) [2, 3, 2]

D) [1, 1, 1]

E) There is no output due to a runtime error

```
int N = 3;
int[][] ints = new int[N][N];
int[][] d;
d = new int[][] {{1,0,-1,0},{0,1,0,-1}};
Queue<Integer> que = new LinkedList<>();
que.add(1);
que.add(1);
que.add(1);
while(!que.isEmpty())
{
   int a = que.poll();
   int b = que.poll();
   int c = que.poll();
   if(Math.min(a, b) >= 0
      && Math.max(a, b)  < N
      && ints[a][b] == 0){
      ints[a][b] = c;
      for(int i = 0;i<4;i++)
      {
         que.add(a+d[0][i]);
         que.add(b+d[1][i]);
         que.add(c+1);
      }
   }
}
out.println(Arrays.toString(ints[N-1]));
```

Question 33

What sort does method `sort` implement?

A) quick sort

B) radix sort

C) heap sort

D) insertion sort

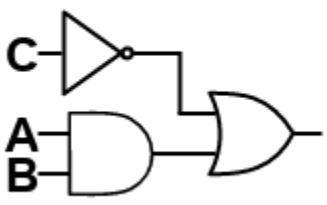
E) selection sort

```
public void sort(int[] ints)
{
    PriorityQueue<Integer> que;
    que = new PriorityQueue<>();
    for(int i:ints)
       que.add(i);
    int x = 0;
    while(!que.isEmpty())
       ints[x++] = que.poll();
}
```

Question 34

How many combinations of assigned values will make the following circuit evaluate to false?

A) 1
B) 2
C) 3
D) 7
E) None of the above



Question 35

Convert the postfix expression to the right to prefix

A) (A + B – C) * (D / E)

B) + A B – C / * D E

C) * - A B C / D E

D) * - + A B C / D E

E) None of the above

```
A B + C - D E / *
```

Question 36

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

A) []

B) [10, 11, 12, 13, 14, 15, 16, 17]

C) [2, 5, 10, 12, 14, 15]

D) [10, 12, 14, 16]

E) There is no output due to a runtime error

```
ArrayList<Integer> list;
list = new ArrayList<Integer>();
for(int i = 0;i<18;i += 2)
list.add(i);
for(int i =0;i<9;i+=2)
list.remove(i);
out.println(list);
```

Question 37

What is the output of the code to the right?

A) 000010010010

B) 000010101111

C) 110010101110

D) 000010101110

E) None of the above

```
boolean[] sv = new boolean[12];
for(int i = 2;i<sv.length;i++)
{
   if(!sv[i])
      for(int j = i*2;j<sv.length;j+=i)
         sv[j] = true;
}
for(int i = 0;i<sv.length;i++)
   if(sv[i])
out.print(1);
   else
out.print(0);
```

Question 38

What is the output of the code to the right?

A) 17

B) 30

C) 31

D) 0

E) None of the above

```
int p = 17;
boolean flag = (p&1) == 1;
out.print(flag ? (p | 14) : (p & 14));
```

Question 39

A bridge is a graph edge who's removal would cause a graph to become disconnected. To the right is the adjacency matrix of graph G where adj[I][J] = adj[J][I] = 1 if a bidirectional edge exists between vertices I and J. How many bridges exist in G?

Adj:

0	1	1	0	0
1	0	1	0	0
1	1	0	0	1
0	0	0	0	1
0	0	1	1	0

Question 40

Write the signed 8-bit binary two's complement representation of the lowest number that is possible to be represented in this form

★ANSWER KEY – CONFIDENTIAL★

UIL COMPUTER SCIENCE – 2022 INVITATIONAL A

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

1) B	11) C	21) B	31) A
2) D	12) B	22) B	32) A
3) C	13) B	23) B	33) C
4) D	14) C	24) C	34) C
5) B	15) C	25) C	35) D
6) D	16) D	26) B	36) E
7) B	17) E	27) B	37) D
8) B	18) A	28) E	38) C
9) B	19) A	29) C	*39) 2
10) A	20) B	30) B	*40) 10000000

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

Note: Correct responses are based on **Java SE Development Kit 17 (JDK 17)** 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 17 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used.

Explanations:

1.	1 + 8 + 32 + 256 = 297
2.	Order of operations and integer division, (9*8) + ((7 % 6) / 5) = (72) + (0) = 72
3.	Round to 2 decimals and print at least 5 characters, zero not space -> 02.35
4.	The last o to appear in str is at index 7
5.	Chained xor is true if there are an odd number of trues, false otherwise
6.	2.45 rounds down to 2 (Math.round returns a long)
7.	(6 * 1.5) + (10 / 1.25) = (9.0) + (8.0) = 17.0
8.	There is no break after the first case, so the second case will execute as well
9.	3 loop iterations, each adding a star to str and then printing. 1 star + 2 star + 3 star = 6 stars total
10.	Initial list: [6,5,4,3,2,1] After first move: [6,5,4,3,1,1] After second move: [6,5,1,3,1,1] (be careful to reference the altered array instead of the original)
11.	nextLine() will pick up only the carriage return after calling nextInt(), so no numbers are skipped
12.	A H 1 0 2 1 3 3 4 6 5 10 6 15 After all loop iterations, H = 15
13.	Order of operations, additions and ++b will come before shifts a will be incremented after the operation is completed 4 + 3 >> 1 = 7 >>1 = 3
14.	Floats are 32 bits
15.	ArrayList.remove(x) removes index X, not value X when given a primitive int
16.	Only one way to make A+B false, and 3 ways to make C+D true. The same math applies to the opposite combination. 1*3 + 3*1 = 6
17.	x -= y makes x zero then they swap so x-=y leaves x unchanged with y zero so condition is false and nothing changes. This leaves us in an infinite loop
18.	The code is finding GCD using Eulers method. GCD(108, 24) is 12. Tracing will find that this takes 5 iterations to converge
19.	Inner loop will execute 4 times per outer loop, outer loop iterates 10 times. 10*4 = 40. Take note of the loop pointer doubling at each iteration instead of increasing by one
20.	Inner loop will still execute 4 times per outer loop, 4*16 = 64
21.	Due to doubling each time, inner loop is logarithmic with respect to n. outer loop is linear, so overall is NlogN
22.	The random*2 will generate a real number in the range [0, 2), plus 10 will be in [10, 12). Truncating will result in only 10 or 11
23.	Addition comes before shifting, 4 << 1 = 8
24.	Floor is the largest element less than or equal to the input
25.	Higher is the smallest element strictly greater than the input
26.	Assuming Math.random() is relatively fair and unbiased, on average each iteration will contribute .5 to the average. In the long run after 1000000 iterations, this will converge to a .5 average.
27.	Adding always sets the head to the new value, and removing and calling next returns the head element. Hence we are using a LIFO data structure. Take this into account and trace the code
28.	While loop removes all items, stack is empty with head null. If head is null, h.elem will throw a null pointer exception
29.	LIFO data structure is Stack
30.	The O class encapsulates the data of a node
31.	Min Value is –2^32, and max value is 2^32 – 1. Hence, their sum is -1

32.	This simulates BFS on a 3x3 grid starting at (1,1) with an initial cost of 1, allowing up, down, left, right as directions. The entire grid will be: [3,2,3] [2,1,2] [3,2,3]
33.	PriorityQueue uses a min heap internally, so sorting with PQ is a heapsort
34.	for the result to be false, C must be true. (A*B) must also evaluate to false, which there are 3 ways to do. hence the answer is 3
35.	To convert to prefix from postfix, we look at the groups each operator is associated with, and move the operator to the front instead of the back. Order of letters must be preserved. The * applies to all 5 symbols The / applies to only D and E The – applies to A B and C The + applies to only A and B
36.	first loop inserts 9 items, second loop attempts to remove 9 items but is increasing location by +2 while array is shrinking. the location exceeds list size and causes an exception.
37.	This marks an array index i true If i is a multiple of some smaller index encountered other than 1. Hence composite number will all be marked to true, and primes will be false
38.	17 & 1 = 1, so we will evaluate the first half of the ternary operator. 17 | 14 = 10001 | 01110 = 11111 = 31
39.	Removing the edge 2-4 will disconnect (3,4) and (0,1,2) Removing the edge 3-4 will disconnect (0,1,2,3) and (4)
40.	The last bit of a 2's complement number can be viewed as a negative 2^n. Hence the smallest value that can be made is only the last bit and no positive bits to increase it. Hence, 10000000.