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.

package java.lang

class Object
```
boolean equals(Object anotherObject)
String toString()
int hashCode()
```

interface Comparable<T>
```
int compareTo(T anotherObject)
   Returns a value < 0 if this is less than anotherObject.
   Returns a value = 0 if this is equal to anotherObject.
   Returns a value > 0 if this is greater than anotherObject.
```

class Integer implements Comparable<Integer>
```
Integer(int value)
int intValue()
boolean equals(Object anotherObject)
String toString()
String toString(int i, int radix)
int compareTo(Integer anotherInteger)
static int parseInt(String s)
```

class Double implements Comparable<Double>
```
Double(double value)
double doubleValue()
boolean equals(Object anotherObject)
String toString()
int compareTo(Double anotherDouble)
static double parseDouble(String s)
```

class String implements Comparable<String>
```
int compareTo(String anotherString)
boolean equals(Object anotherObject)
int length()
String substring(int begin)
  Returns substring(begin, length()).
String substring(int begin, int end)
  Returns the substring from index begin through index (end - 1).
int indexOf(String str)
  Returns the index within this string of the first occurrence of str.
  Returns –1 if str is not found.
int indexOf(String str, int fromIndex)
  Returns the index within this string of the first occurrence of str,
  starting the search at fromIndex. Returns –1 if str is not found.
int indexOf(int ch)
int indexOf(int ch, int fromIndex)
char charAt(int index)
String toLowerCase()
String toUpperCase()
String[] split(String regex)
boolean matches(String regex)
String replaceAll(String regex, String str)
```

class Character
```
static boolean isDigit(char ch)
static boolean isLetter(char ch)
static boolean isLetterOrDigit(char ch)
static boolean isLowerCase(char ch)
static boolean isUpperCase(char ch)
static char toUpperCase(char ch)
static char toLowerCase(char ch)
```

class Math
```
static int abs(int a)
static double abs(double a)
static double pow(double base, double exponent)
static double sqrt(double a)
static double ceil(double a)
static double floor(double a)
static double min(double a, double b)
static double max(double a, double b)
static int min(int a, int b)
static int max(int a, int b)
static long round(double a)
static double random()
  Returns a double greater than or equal to 0.0 and less than 1.0.
```

package java.util

interface List<E>
class ArrayList<E> implements List<E>
```
boolean add(E item)
int size()
Iterator<E> iterator()
ListIterator<E> listIterator()
E get(int index)
E set(int index, E item)
void add(int index, E item)
E remove(int index)
```

class LinkedList<E> implements List<E>, Queue<E>
```
void addFirst(E item)
void addLast(E item)
E getFirst()
E getLast()
E removeFirst()
E removeLast()
```

class Stack<E>
```
boolean isEmpty()
E peek()
E pop()
E push(E item)
```

interface Queue<E>
class PriorityQueue<E>
```
boolean add(E item)
boolean isEmpty()
E peek()
E remove()
```

interface Set<E>
class HashSet<E> implements Set<E>
class TreeSet<E> implements Set<E>
```
boolean add(E item)
boolean contains(Object item)
boolean remove(Object item)
int size()
Iterator<E> iterator()
boolean addAll(Collection<? extends E> c)
boolean removeAll(Collection<?> c)
boolean retainAll(Collection<?> c)
```

interface Map<K,V>
class HashMap<K,V> implements Map<K,V>
class TreeMap<K,V> implements Map<K,V>
```
Object put(K key, V value)
V get(Object key)
boolean containsKey(Object key)
int size()
Set<K> keySet()
Set<Map.Entry<K, V>> entrySet()
```

interface Iterator<E>
```
boolean hasNext()
E next()
void remove()
```

interface ListIterator<E> extends Iterator<E>
```
void add(E item)
void set(E item)
```

class Scanner
```
Scanner(InputStream source)
Scanner(String str)
boolean hasNext()
boolean hasNextInt()
boolean hasNextDouble()
String next()
int nextInt()
double nextDouble()
String nextLine()
Scanner useDelimiter(String regex)
```

Package `java.util.function`

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

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

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

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

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

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

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

Note: Correct responses are based on **Java SE Development Kit 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 NOT equal to $1A_{16} + 23_{16}$?

A) $3D_{16}$ **B)** 61_{10} **C)** 111101_2 **D)** 75_8 **E)** All are equal.

Question 2.

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

A) 5 **B)** 51.28 **C)** 52 **D)** 5.52 **E)** 42

```
out.print(19 + 33 - 6 * 3 / 25)
```

Question 3.

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

A) `InvAInvB`
    `District`

B) `InvA`
    `InvBDistrict`

C) `InvA`
    `InvB`
    `District`

D) `InvA InvB District`

E) `InvAInvBDistrict`

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

Question 4.

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

A) 2 **B)** 3 **C)** 4

D) 5 **E)** 9

```
out.print("greenEraser".lastIndexOf('e', 6));
```

Question 5.

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

A) `true`

B) `false`

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

Question 6.

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

A) 11 **B)** 11.0 **C)** 12 **D)** 12.0 **E)** 90.0

```
out.print(Math.pow(3,2) + Math.sqrt(9));
```

Question 7.

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

A) 53.625 **B)** 5.625 **C)** 5.925 **D)** 5

E) There is no output due to an error.

```
int i = 10;
double d = 6.25;
char c = '5';
out.print(c + d / i);
```

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);
```

Question 13.

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

- **A)** I II III
- **B)** II III I
- **C)** III II I
- **D)** II I III
- **E)** III I II

```
I. %

II. &&

III. ||
```

Question 14.

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

- **A)** -83
- **B)** 83
- **C)** -84
- **D)** 84
- **E)** -82

```
out.print(~83);
```

Question 15.

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

- **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]

```
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);
```

Question 16.

How many ordered pairs make this Boolean expression false?

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

$$\overline{A * \bar{B}}$$

Question 17.

What is the output of the code segment shown here?

```
out.print("static".compareTo("public")>4?"void":"main");
```

- **A)** void
- **B)** main
- **C)** true
- **D)** false
- **E)** 3

Question 18.

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)?

```
Random r = new Random();
```

- **A)** int rando = r.nextInt(60);
- **B)** int rando = (int)r.nextDouble() * 10 + 50;
- **C)** int rando = (int)(r.nextDouble() + 50);
- **D)** int rando = (int)(r.nextDouble() * 10) + 60;
- **E)** int rando = (int)(r.nextDouble() * 10) + 50;

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

A) 0

B) 2

C) 3

D) 6

E) 7

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.

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.

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.

```
//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 23.

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

A) 4

B) 9

C) 6

D) 12

E) This segment throws an ArrayIndexOutOfBoundsException

```
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]);
```

Question 24.

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

A) `Stack<String> s = new Stack<String>();`

B) `Stack<String> s = new Stack(String);`

C) `Stack<String> s = new Stack<>();`

D) `Stack<String> s = Stack<String>();`

E) More than one of the above.

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

<missing code>
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
```

Question 25.

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

A) `monday`

B) `tuesday`

C) `wednesday`

D) `thursday`

E) `friday`

Question 26.

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

A) `monday`

B) `tuesday`

C) `friday`

D) `null`

E) There is no output due to an error.

Question 27.

Which of the following is the 8-bit binary two's complement equivalent of -78?

A) 10110001

B) 01001110

C) 10110010

D) 01001101

E) 00010110

```java
//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]`

```java
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

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());
```

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;};`

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>`

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`

```
//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;
  }
}
```
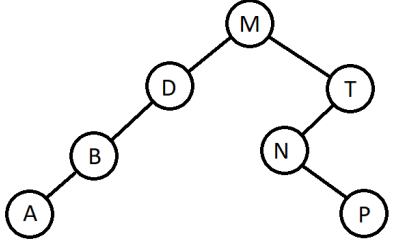
What is the output of the line of code shown on the right?

A) 8

B) 64

C) 10

D) 00000008

E) 1000

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

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

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);
}
```

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

A) 4

B) 3

C) 2

D) 1

e) "Hello World"

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

How many edges does a complete graph with 8 vertices have?

a) 56   b) 28   c) 32   d) 64   e) It varies