UIL COMPUTER SCIENCE WRITTEN TEST

2019 INVITATIONAL B

FEBRUARY/MARCH 2019

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)
```

UIL COMPUTER SCIENCE WRITTEN TEST – 2019 INVITATIONAL B

Note: Correct responses are based on **Java SE Development Kit 8 (JDK 8)** 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 8 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 hexadecimal values is equivalent to 10111001_2?

A) $1F_{16}$ **B)** 98_{16} **C)** $A8_{16}$ **D)** $B9_{16}$ **E)** CA_{16}

Question 2.

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

A) 5.67 **B)** 5.0 **C)** 3 **D)** 2 **E)** 5

```
out.print(29%15/3+1);
```

Question 3.

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

A)
```
Random
Object
Math\nString
```

B)
```
Random\nObject
Math\\nString
```

C)
```
Random
Object
Math\\nString
```

D)
```
Random
Object
MathnString
```

E)
```
Random
ObjectMath\nString
```

```
out.println("Random\nObject");
out.print("Math\\nString");
```

Question 4.

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

A) 1 **B)** 2 **C)** 5 **D)** 6 **E)** 7

```
String str="superduper";
out.print(str.indexOf(str.charAt(6)));
```

Question 5.

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

A) true **B)** false

```
boolean a=true,b=true,c;
c=a&&!b^(a||b);
out.print(c);
```

Question 6.

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

A) 8 **B)** 8.0 **C)** 4 **D)** 4.0

E) Error. Will not compile due to a duplicate identifier.

```
double cbrt=Math.cbrt(64);
out.print(cbrt);
```

Question 7.

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

A) 6.0 **B)** −5.75 **C)** 5.75 **D)** 5 **E)** 6

```
double a;int x;
a=8.75;x=-3;
out.print(a+x);
```

Question 8.

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

A) one

B) two

C) three

D) four

E) five

```
if(3*8>=11+12)
 if(5-18<-14)
  out.print("one");
 else
  out.print("two");
else
 if(43/10==4)
  out.print("three");
 else
  if(63-45>0)
   out.print("four");
  else
   out.print("five");
```

Question 9.

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

A) -4 -1 2 5 8 11

B) -4 -1 2 5 8

C) -2 0 2 4 6 8 10

D) -1 2 5 8

E) -1 2 5 8 11

```
int z=-4;
while(z<10) {
     z+=3;
     out.print(z+" ");
}
```

Question 10.

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

A) [9, 8, -4, 2, 0, -4]

B) [9, 8, -4, 2, -4, -4]

C) [9, 4, 0, 2, 8, -4]

D) [9, 4, -4, 2, 8, -4]

E) Error. Throws an ArrayIndexOutOfBoundsException.

```
int[] nums= {9,4,-4,2,0,5};
nums[nums[1]]=8;
nums[5]=nums[nums[3]];
out.print(Arrays.toString(nums));
```

```
import static java.lang.System.out;
import java.util.*;
import java.io.*;
public class Q11 {

     public static void main(String[] args) throws IOException{
          Scanner file=new Scanner(<code>);
          while(file.hasNextInt()) {
               out.print(file.nextInt()+" ");
          }
     }
}
```

Question 11.

Which of the following must replace **<code>** in the class implemented above? Assume *datafile.dat* exists and is in the correct directory.

A) `new File("datafile.dat")`

B) `"datafile.dat"`

C) `new File()`

D) `File("datafile.dat")`

E) No additional code is required.

Question 12.

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

A) `54`

B) `55`

C) `65`

D) `52`

E) `11`

```
int n=0;
for(int m=100;m>0;m-=10)
     n=n+m/10;
out.print(n);
```

Question 13.

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

A) `106`

B) `80`

C) `true`

D) `false`

E) Error. Will not compile.

```
int g=10,h=7,i=12;
out.print(++h*g>i*h+g);
```

Question 14.

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

A) `1 8`

B) `8 64`

C) `4 32`

D) `2 16`

E) `16 128`

```
out.print(Integer.BYTES+" ");
out.print(Integer.SIZE);
```

Question 15.

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

A) main 4

B) static 3

C) void 3

D) void 4

E) static 4

```
ArrayList<String> list=new
ArrayList<String>();
list.add("public");list.add("static");
list.add("void");list.add("main");
out.print(list.get(2)+" "+list.size());
```

Question 16.

How many times does the code segment shown on the right print `true`?

A) 0

B) 15

C) 3

D) 4

E) 18

```
String str="abc^245#s&890jhy%165x";
for(int i=0;i<str.length()-3;i++) {
  String s=str.substring(i, i+3);
  out.println(s.matches("[a-z]\\W\\d"));
  }
```

Question 17.

What is the output of this line of code?

```
out.print("saturation".compareTo("saturday"));
```

A) 3

B) 6

C) true

D) false

E) -3

```
String str="aB2C7*9dE4$";
char[] chrs=str.toCharArray();
int i=0,j=0;
while(i<chrs.length) {
     if(Character.isDigit(chrs[i])||Character.isLowerCase(chrs[i]))
           continue;
     else
           if(Character.isUpperCase(chrs[i]))
                 j++;
           else
                 break;
     i++;
}
out.print(j);
```

Question 18.

What is printed by the code segment shown above?

A) 0 **B)** 2 **C)** 3 **D)** 5 **E)** There is no output.

Question 19.

Which of the following must replace **<code>** to ensure that the code segment on the right will print all the values stored in array `list`?

A) `i`

B) `list`

C) `list[i]`

D) `list[]`

E) `i[list]`

```
int []list= {5,8,2,4,6,3,7,1,9};
for(int i:list)
     out.print(<code>+" ");
```

Question 20.

Which of the following methods will correctly return the least common multiple of parameters `a` and `b`? Assume that the product of `a` and `b` does not exceed `Integer.MAX_VALUE`.

A.

```
public static int findlcm(int a,int b) {
  int lcm=a*b;
  int lcv=lcm;
  while(lcv>=Math.max(a, b)) {
    if(lcv%a==0&&lcv%b==0)
      return lcv;
    lcv--;
  }
  return lcv;
}
```

B.

```
public static int findlcm(int a,int b) {
  int lcm=a*b;
  int lcv=lcm;
  while(lcv>=Math.max(a, b)) {
    if(a%lcv==0||b%lcv==0)
      lcm=lcv;
    lcv--;
  }
  return lcm;
}
```

C.

```
public static int findlcm(int a,int b) {
  int lcm=a*b;
  int lcv=lcm;
  while(lcv>=a*b) {
    if(lcv%a==0&&lcv%b==0)
      lcm=lcv;
    lcv++;
  }
  return lcm;
}
```

D.

```
public static int findlcm(int a,int b) {
  int lcm=a*b;
  int lcv=lcm;
  while(lcv>=Math.max(a, b)) {
    if(lcv%a==0&&lcv%b==0)
      lcm=lcv;
    lcv--;
  }
  return lcm;
}
```

E. More than one of the above.

Question 21.

Which of the following can replace **<code>** in the segment shown on the right?

A) ArrayList

B) LinkedList

C) List

D) Queue

E) Set

Question 22.

Assuming **<code>** has been replaced correctly, what is the output of the code segment before the comment?

A) one two three four five

B) four two three one five

C) five one three two four

D) five four one three two

E) three two one four five

Question 23.

Assuming **<code>** has been replaced correctly, what is printed by just the code listed after the comment?

A) 1

B) 2

C) 3

D) 4

E) 5

```
//Use this code segment to answer
//questions 21, 22 and 23.
Queue<String> list=new <code><String>();
list.add("five");list.add("one");
list.add("three");list.add("two");
list.add("four");
for(String s:list)
     out.print(s+" ");
//comment
list.peek();list.poll();list.remove();
out.print("\n"+list.size());
```

Question 24.

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

A) 81

B) 208

C) 34

D) 127

E) 46

```
out.print(121&87);
```

Question 25.

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

A) 21

B) 20

C) 14

D) 10

E) 15

```
int i=0;
for(int j=1;j<5;j++) {
  int k=j;
  while(k>0) {
    i++;
    k--;
  }
}
out.print(i);
```

Question 26.

Which of the following must replace **<code>** in the method shown on the right to ensure that the array list is sorted in ascending order?

A) `list[j]>x`

B) `list[i]<x`

C) `list[j]<y`

D) `x<y`

E) `list[j]<x`

Question 27.

Assume **<code>** *has been filled in correctly to answer questions 27, 28 and 29.*

What is printed by the line of code following the comment when `i` is equal to 2 if `list` is initialized as follows?

```
int[] list= {2,9,3,7,0,1,4};
```

A) `[0, 1, 2, 7, 3, 9, 4]`

B) `[9, 7, 4, 2, 0, 1, 3]`

C) `[2, 3, 9, 7, 0, 1, 4]`

D) `[0, 1, 2, 3, 7, 9, 4]`

E) `[2, 3, 7, 9, 0, 1, 4]`

Question 28.

Which of the following algorithms is implemented by the `sort` method shown on the right?

A) Index Sort

B) Selection Sort

C) Radix Sort

D) Insertion Sort

E) Merge Sort

Question 29.

What is the worst-case run time efficiency for the method `sort` where n is the size of array `list`?

A) O(1)

B) O(n log n)

C) O(n)

D) O(log n)

E) $O(n^2)$

```
//Use this method to answer questions
//26 - 29.
public static void sort(int[] list) {
  int x,y;
  for(int i=0;i<list.length;i++) {
    x=list[i];
    y=i;
    for(int j=i+1;j<list.length;j++) {
      if(<code>) {
        x=list[j];
        y=j;
      }
    }
    list[y]=list[i];
    list[i]=x;
    //comment
    out.println(Arrays.toString(list));
  }
}
```

Question 30.

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

A) `a&&b^c`

B) `a||b^!c`

C) `a&&b^!c`

D) `a&&b&&!c`

E) `a||b&&c`

a	b	c	
T	T	T	T
T	T	F	F
T	F	F	T
F	F	F	F
F	F	T	F
F	T	T	F

Question 31.

Which of the following can replace **<code1>** in the class `Item` and will correctly assign the value stored in the parameter `id` to the field named `id`?

A) `this(id)`

B) `id=id`

C) `super(id)`

D) `id=this.id`

E) More than one of the above.

Question 32.

Which of the following must replace **<code2>** to ensure that the method `toString` will compile and execute correctly?

A) `Item`

B) `void`

C) `static`

D) `String`

E) No additional code is required.

Question 33.

Which of the following is the correct implementation of a method that will return the value stored in the field `cost`?

A.
```
public getCost() {
  return cost;
}
```
B.
```
public int getCost() {
  return cost;
}
```
C.
```
public void getCost() {
  System.out.println(cost);
}
```
D.
```
public double getCost() {
  return cost;
}
```
E.
```
public double getCost() {
  return Item.cost;
}
```

```
//Use the code listed here for class Item to
//answer questions 31 - 34
public class Item {
   public static int count=0;
   private int id;
   private double cost;
   public Item(int id, double cost) {
     <code1>;
     this.cost = cost;
     count++;
   }
   public Item(int id) {
     this.id=id;
     count++;
   }
   public Item() { }
   public void setId(int id) {
     this.id = id;
     count++;
   }
   public void setCost(double cost) {
     this.cost = cost;
   }
   public <code2> toString() {
   return "id= " + id + ", cost= " + cost;
   }
}
```

Question 34.

Assuming **<code1>** and **<code2>** are correct, what is the output of the client code shown on the right?

A) 0 **B)** 2 **C)** 3 **D)** 4

E) There is no output due to an error.

```
Item r=new Item(123,55.25);
Item s=new Item(321);
Item t=new Item();
out.println(Item.count);
```

Question 35.

All binary operators ***except*** _____ operators are left associative.

A) arithmetic

B) relational

C) Boolean

D) bitwise

E) assignment

Question 36.

Which of the following Boolean expressions is <u>not</u> equal to A?

A) $A + A$ **B)** $A + 1$ **C)** $A * 1$ **D)** $A * A$ **E)** $A + 0$

Question 37.

Given the class `DataStructure` shown on the right, which of the following data structures must the class be implementing?

A) A set

B) A stack

C) A queue

D) A linked list

E) A tree

```
import java.util.ArrayList;

public class DataStructure {

	private ArrayList<String> list;

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

	public boolean add(String s) {
		if(!list.contains(s)) {
			list.add(s);
			return true;
		}
		else
			return false;
	}

	public boolean remove(String s) {
		return list.remove(s);
	}
}
```

Question 38.

The following expressions are written using prefix, postfix and infix notation. Which one is <u>not</u> equivalent to all the others?

A) A B C / + D *

B) A / B + C * D

C) A B / C D * +

D) + / A B * C D

E) None of the above. They are all equivalent.

Question 39.

How many edges are in a complete undirected graph that contains 7 vertices? Write your answer in the blank provided on the answer document.

Question 40.

How many leaves are contained within a binary search tree that is constructed by inserting the following values in the order shown?

58 19 18 3 4 1 67 60 82

Write your answer in the blank provided on the answer document.