UIL COMPUTER SCIENCE WRITTEN TEST

2017 REGION

APRIL 2017

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 – 2017 REGION

Note: Correct responses are based on **Java SE Development Kit 8 (JDK 8)** 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 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 is less than 100_8?

A) 40_{16} **B)** 32_{10} **C)** 01001000_2 **D)** 98_{16} **E)** 01000000_2

Question 2.

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

A) 6 **B)** 0 **C)** 0.0 **D)** 8 **E)** 6.0

```
out.println(8+5%4-3.0);
```

Question 3.

What is the output of the code segment to the right? *Dashes - indicate blank spaces.*

A) -----Total=$1234.57

B) Total=$-----1234.56

C) Total=$1234.57-----

D) Total=-----$1234.57

E) Total=$-----1234.57

```
out.printf("Total=$%12.2f",1234.5678);
```

Question 4.

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

A) ttgadget **B)** gogogadgeg **C)** tototadtet

D) togogadget **E)** Error. Throws a StringIndexOutOfBoundsException.

```
String s="gogogadget";
out.print(s.replace(s.substring(0, 1),
      s.substring(s.length()-1)));
```

Question 5.

Which of the following combinations will make this Boolean expression evaluate as true?

```
a||b&&c^d
```

A) a is false, b is false, c is true, d is false

B) a is true, b is false, c is false, d is false

C) a is false, b is true, c is true, d is true

D) All of the above.

E) None of the above.

Question 6.

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

A) 790.0 **B)** 790 **C)** 789 **D)** 789.0

E) Error. Will not compile.

```
double d=789.8653;
int i=Math.round(d);
out.print(i);
```

Question 7.

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

A) 37.39 5.56

B) 32.83 4.56

C) 32.00 5.56

D) 32.00 4.56

E) 32 5.56

```
double m=8.2;
double n=4.56;
m=(int)(--m*n++);
out.printf("%1.2f %1.2f",m,n);
```

Question 8.

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

A) Error. First case statement cannot be empty.

B) No output. Produces an infinite loop.

C) 5

D) 6

E) 1

```
int x=1;
while(x<=5){
      switch(x){
      case 1:
      case 2: x++;break;
      case 3: x+=2;break;
      case 4: x=1;break;
      case 5: x++;break;
      default:x=1;
      }
}
out.print(x);
```

Question 9.

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

A) sehcodgocan

B) ecdoa

C) shogcn

D) sehcodgoca

E) shogc

```
String s="nacogdoches";
for(int i=s.length()-1;i>0;i-=2)
      out.print(s.charAt(i));
```

Question 10.

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

A) 5 4 3 2 1

B) 5 4 3 2 0

C) 0 2 3 4 5

D) 1 2 3 4 5

E) Error. Will not compile.

```
int a[]={1,2,3,4,5};
int[] b=new int[a.length];
for(int i=a.length-1;i>0;i--)
      b[i]=a[i];
for(int i:b)
      out.print(i+" ");
```

Question 11.

The class Abc contains an error. Which of the following best describes that error? The file *datafile.dat* is present in the correct directory and contains these values all listed on the same line.

2.5 1.75 3.25 4 5.85

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

     public static void main(String[] args) throws IOException{
          File f=new File("datafile.dat");
          Scanner s=new Scanner(f);
          double sum=0;
          while(s.hasNext())
               sum=sum+s.next();
          out.print(sum);
     }
}
```

A) Type mismatch: cannot convert from String to double

B) Type mismatch: cannot convert from int to double

C) Throws a ClassNotFoundException.

D) Scanner cannot be resolved to a type.

E) Unhandled exception: type FileNotFoundException.

Question 12.

The file datafile.dat is present in the correct directory and contains these values all listed on the same line.

2.5 1.75 3.25 4 5.85

Assume that this line of code is contained within the main method:

```
Scanner f=new Scanner(new File("datafile.dat);
```

Which of the following segments of code will read all of the values in Scanner object f, calculate the average of those values, and then print the average?

A.
```
double a=0;
int b=1;
do{
      a+=f.nextDouble();
      b++;
}while(f.hasNext());
out.print(a/b);
```

B.
```
double a=0;
int b=0;
double[] c=new double[100];
while(f.hasNext())
      c[b]=f.nextDouble();
for(double d:c)
      a+=d;
out.print(a/b);
```

C.
```
double a=0;
int b=0;
while(f.hasNext()){
   a+=f.nextDouble();
   b=b+a;
}
out.print(a/b);
```

D.
```
double a=0;
int b=0;
while(f.hasNext()){
      b++;
      a+=f.nextDouble();
}
out.print(a/b);
```

E. More than one of these will correctly calculate and print the average.

Question 13.

What is the output of the code segement shown to the right?

A) −1

B) 0

C) 4

D) 16

E) 32

```
int e=4,f=16;
out.print(f>>e-2);
```

Question 14.

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

A) −4 **B)** 3 **C)** −3 **D)** 0 **E)** 4

```
int d=3;
out.print(~d);
```

Question 15.

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

A) [grouse, dove, chukar, quail, turkey]

B) [dove, chukar, quail, pheasant, turkey, grouse]

C) [dove, quail, chukar, pheasant, turkey, grouse]

D) [dove, quail, chukar, turkey, grouse]

E) [dove, chukar, pheasant, turkey, grouse]

```
ArrayList<String> a=new ArrayList<String>();
String[] list=
      {"dove","quail","pheasant","turkey"};
for(String s:list)
      a.add(s);
a.add("grouse");
a.set(2, "chukar");
out.print(a);
```

```
// Use the following code to answer questions 16, 17, 18, 19 and 20.

public <code 1> class Vehicle {
      private String vin,make,model;
      private int year;
      private double odometer;
      public Vehicle(String vin, String make, String model, int i,double odometer) {
            System.out.print("Vehicle1 ");
            this.vin = vin;
            this.make = make;
            this.model = model;
            this.year = i;
            this.odometer=odometer;
      }
      public Vehicle(){System.out.print("Vehicle2 ");}
      public String getVin() {return vin;}
      public void setVin(String vin) {this.vin = vin;}
      public String getMake() {return make;}
      public void setMake(String make) {this.make = make;}
      public String getModel() {return model;}
      public void setModel(String model) {this.model = model;}
      public int getYear() {return year;}
      public void setYear(int year) {this.year = year;}
      public double getOdometer() {return odometer;}
      public void setOdometer(double miles){odometer=miles;}
      public abstract void drive(double miles);
}
public class Gas extends Vehicle {

      private double mpg,tank=0;

      public Gas(String vin, String make, String model, int i,double odometer,double mpg)
            {
            super(vin, make, model, i,odometer);
            this.mpg=mpg;
            System.out.print("Gas1 ");}
      public Gas() {System.out.print("Gas2 ");}
      public double getMpg() {return mpg;}
      public void setMpg(double mpg) {this.mpg = mpg;}
      public void addGas(double gallons){tank+=gallons;}
      public void drive(double miles){
            tank-=miles/mpg;
            <code 2>;}
}
public class Electric extends Vehicle {

      private double mpkWh,battery=0;
      private final double BATCAP=30;
      public Electric(String vin, String make, String model, int year, double odometer,
                        double mpkwh)
            {
            super(vin, make, model, year, odometer);
            this.mpkWh=mpkwh;}
      public Electric() {}

      public void drive(double miles) {
            battery-=miles/mpkWh;
            <code 2>;
      }
      public void charge(){battery=BATCAP;}
}
```

Question 16.

Which of the following should replace **<code 1>** to ensure that the Vehicle class will compile correctly?

A) `inherits` **B)** `extends` **C)** `final` **D)** `implements` **E)** `abstract`

Question 17.

Which of the following could replace **<code 2>** so that the `drive` method changes the `odometer` field to reflect the number of miles driven?

A) `odometer+=miles`

B) `setOdometer(getOdometer()+miles)`

C) `setOdometer(odometer+miles)`

D) `odometer=getOdometer()+miles`

E) `setOdometer(miles)`

Question 18.

Assuming that **<code 1>** and **<code 2>** have been filled in correctly, what is the output of the client code shown here?

```
Vehicle v1=new Gas("123456789","Toyota","Tundra",2012,42000,17.0);
out.println();
Electric v2=new Electric("987654321","Nissan","Leaf",2015,125.75,3);
out.println();
Vehicle v3=new Gas();
```

A)
```
Vehicle1 Gas1
  Gas2
```
B)
```
Vehicle1 Gas1
  Vehicle1
  Vehicle2 Gas2
```
C)
```
Gas1 Vehicle1
  Vehicle1 Vehicle2
  Gas2
```
D)
```
Vehicle1 Gas1
  Vehicle1
  Gas2
```
E) Error. Will not compile.

Question 19.

Assume that **<code 1>** and **<code 2>** have been filled in correctly. One of the following lines of client code contains an error, which one is it?

A) `Gas v1=new Gas("123456789","Toyota","Tundra",2012,42000,17.0);`

B) `Vehicle v2=new Gas("123456789","Toyota","Tundra",2012,42000,17.0);`

C) `Electric v3=new Electric("987654321","Nissan","Leaf",2015,125.75,3);`

D) `Electric v4=new Vehicle("987654321","Nissan","Leaf",2015,125.75,3);`

E) `Vehicle v4=new Electric("987654321","Nissan","Leaf",2015,125.75,3);`

Question 20.

Assuming that **<code 1>** and **<code 2>** has been filled in correctly, what is the output of the client code shown here?

```
Vehicle v1=new Electric("7G54B100J321","Tesla","Model S",2016,100,4);
v1.charge();
v1.drive(100);
System.out.print(v1.getOdometer());
```

A) 30

B) 0

C) 100

D) 200

E) Error. Will not compile.

Question 21.

Which of the following values for `s` will make this line of code print true?

```
out.print(s.matches("[A-Z]{1}[a-z]+, [A-Z]{1}[a-z]+"));
```

A) John Smith

B) John,Smith

C) Smith, John

D) smith, john

E) All of the above will make the code print true.

// Use the code shown here to answer questions 22 and 23.

```
Map<Integer,String> m=new TreeMap<Integer,String>();
int[] locker={125,211,91,173,18};
String[] student={"Susan","Larry","Alex","Susan","Brianne"};
for(int i=0;i<locker.length;i++)
      m.put(locker[i], student[i]);
out.println(m); //line #1
m.put(91, "Lori");
out.print(m.get(91)+" "+m.remove(125)+" "+m.size()); //line #2
```

Question 22.

What is printed by **line #1** in the code shown above?

A) {18=Brianne, 91=Alex, 125=Susan, 173=Susan, 211=Larry}

B) {125=Susan, 211=Larry, 91=Alex, 173=Susan, 18=Brianne}

C) {Brianne, Alex, Susan, Susan, Larry}

D) {Brianne, Alex, Susan, Larry}

E) {91=Alex, 18=Brianne, 211=Larry, 125=Susan, 173=Susan}

Question 23.

What is printed by **line #2** in the code shown above?

A) Lori Susan 5.

B) Lori Susan 3

C) Alex Susan 3

D) Alex Susan 4

E) Lori Susan 4

Question 24.

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

A)
```
[2, -6, 0]
[2, 0, 1]
[8, 0, 4]
[-4,6,-1]
```
B)
```
[2, 2, 8, -4]
[-6, 0, 0, 6]
[4, 0, -1, 1]
```
C)
```
[2, -6, 0]
[2, 0, 1]
[8, 0, 4]
[-4,6,-1]
```
D)
```
[2, 2, 8, -4]
[-6, 0, 0, 6]
[0, 1, 4, -1]
```
E) There is no output due to an error.

```
int[][] a={{2,-1},{0,3},{1,0}};
int[][] b={{0,1,4,-1},{-2,0,0,2}};
int[][] p=new int[a.length][b[1].length];
for(int r=0;r<a.length;r++)
      for(int c=0;c<b[0].length;c++){
            int s=0;
            for(int x=0;x<b.length;x++){
                  s+=a[r][x]*b[x][c];}
            p[r][c]=s;}
for(int[]r:p)
      out.println(Arrays.toString(r));
```

Question 25.

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

A) 1 **B)** 2 **C)** 3 **D)** 4 **E)** 5

```
String s="fileeditproject";
String[] s2=s.split("e");
out.println(s2.length);
```

Question 26.

Every Java class is implicitly a subclass of _____.

A) Java **B)** Object **C)** all Java standard classes **D)** an Interface **E)** a package

Question 27.

If each of the data structures listed contains elements that implement the Comparable interface, which one can not be sorted by the Collections.sort(x) method?

A) ArrayList

B) LinkedList

C) Stack

D) All of the above can be sorted by Collections.sort(x).

E) None of the above can be sorted by Collections.sort(x).

```
// Use the code listed here to answer questions 28, 29 and 30.

public static void sort(int[] list){
      if(list.length<=1) return;
      int[] list1 = Arrays.copyOfRange(list, 0, list.length/2);
      int[] list2 = Arrays.copyOfRange(list, list.length/2, list.length);
      <code 1>
      arraycopy(tm, 0, list, 0, tm.length);
      }

public static int[] merge(int[] list1,int[] list2){
      int list1Index=0;
      int list2Index=0;
      int tempIndex=0;
      int[] temp = new int[list1.length+list2.length];
      // while loop referred to in question 29.
      while(list1Index<list1.length&&list2Index<list2.length){
            if(list1[list1Index]<list2[list2Index]){
                  temp[tempIndex]=list1[list1Index];
                  list1Index++;
                  tempIndex++;                  }
            else
                  {
                  temp[tempIndex]=list2[list2Index];
                  list2Index++;
                  tempIndex++;
                  }
      }
      // End of while loop referred to in question 29.
      while(list1Index<list1.length)
            {
            temp[tempIndex]=list1[list1Index];
            list1Index++;
            tempIndex++;
            }
      while(list2Index<list2.length)
                  {
            temp[tempIndex]=list2[list2Index];
            list2Index++;
            tempIndex++;
            }
      return temp;
      }
```

Question 28.

The code shown above is intended to implement the Merge Sort algorithm. What should replace **<code 1>** in the `sort` method so that it will compile and execute correctly?

A)
```
sort(list1);
int[] tm=merge(list1,list2);
sort(list2);
int[] tm=merge(list1,list2);
```

B)
```
int[] tm=merge(list1,list2);
sort(list1);
sort(list2);
```

C)
```
sort(list1);
sort(list2);
int[] tm=merge(list1,list2);
```

D)
```
int[] tm=merge(list1,list2);
```

E)
```
sort(list1);
sort(list2);
```

Question 29.

Which of the following best describes the function of the while loop marked by comments in the `merge` method?

A) Switch the elements in `list1` with the elements in `list2` and then place the resulting list into the array `temp`.

B) Place all of the elements in `list1` into the array `temp` and then place all of the elements from `list2` into the array `temp`.

C) Find the smallest element in both `list1` and `list2` and place each into the array `temp`.

D) As long as either `list1` or `list2` still contains elements, compare the current element in each and place the smaller of the two into the array `temp`.

E) As long as both `list1` and `list2` still contain elements, compare the current element in each and place the smaller of the two into the array `temp`.

Question 30.

What is the worst case relative time complexity (Big-O value) for a Merge Sort?

A) O(1)

B) O(n)

C) $O(n^2)$

D) O(log n)

E) O(n log n)

Question 31.

What is the output of method `abc` shown to the right if the value passed to x is 6?

A) `#$#$#*^&^&^`

B) `^&^&^*#$#$#`

C) `$#$#$*&^&^&`

D) `&^&^&*$#$#$`

E) `#^#^#$&$&$*`

```
public static void abc(int x){
      if(x==1)
            out.print("*");
      else if(x%2==0)
            {
            out.print("#");
            abc(x-1);
            out.print("^");
            }
      else
            {
            out.print("$");
            abc(x-1);
            out.print("&");
            }
}
```

Question 32.

Given the method `xyz` shown on the right what is the output of the client code shown here?

```
int m=20,n=2;
out.print(xyz(m,n)+" "+m+" "+n);
```

A) `90 20 2`

B) `84 20 2`

C) `84 34 6`

D) `84 29 5`

E) `90 34 6`

```
public static int xyz(int m,int n){
int c=m+n;
int d=m*n;
while(d>c){
      m+=n;
      n++;
      d-=n;}
return m+n+c+d;
}
```

Question 33.

What is the result of a preorder traversal of a binary search tree created by inserting the values 10, 5, 15, 20, 12, 6, and 1 in that order?

A) 1 5 6 10 12 15 20

B) 1 6 5 12 20 15 10

C) 10 5 1 6 15 12 20

D) 10 5 15 1 6 12 20

E) 20 15 12 10 6 5 1

Question 34.

What is the output of the code segment shown here?

```
String s1="Dallas",s2="Amarillo";
int x=s1.length()>s2.length()?s1.length():s2.length();
out.print(x);
```

A) 0 **B)** 14 **C)** 6 **D)** 8 **E)** There is no output due to an error.

Question 35.

What is the output of this line of code?

```
out.println(5<<3>>2&17);
```

A) 0 **B)** 40 **C)** 10 **D)** 1 **E)** 24

Question 36.

Which of the following truth tables shows all of the possible values for the expression $A * \bar{B} \oplus C$?

A)

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

B)

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

C)

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

D)

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

E)

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

Question 37.

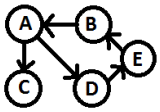
All of the following values are shown using signed 8-bit 2s complement binary representation. Which one would be closest to 0 (zero) on a number line?

A) 00010011 **B)** 10101101 **C)** 11111111 **D)** 00000101 **E)** 11011001

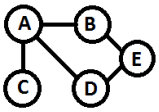
Question 38.

Which of the following graphs does the adjacency matrix to the right represent?

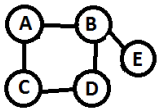
A)



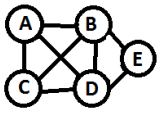
B)



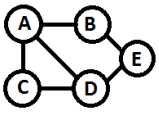
C)



D)



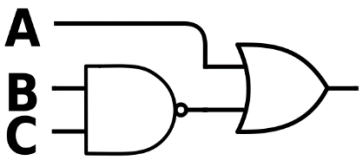
E)




```
int[][] am={
   {0,1,1,1,0},//A
   {1,0,0,0,1},//B
   {1,0,0,0,0},//C
   {1,0,0,0,1},//D
   {0,1,0,1,0}};//E
```

Question 39.

Write the Boolean expression that is represented by the digital electronics diagram shown on the right?



Question 40.

What is the relative time complexity (Big O value) for access within a doubly linked list that contains n elements?

★ANSWER KEY – CONFIDENTIAL★

UIL COMPUTER SCIENCE – 2017 REGION

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

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

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

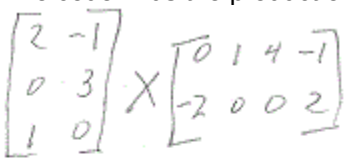
Note: Correct responses are based on **Java SE Development Kit 8 (JDK 8)** from Sun Microsystems, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g., "error" is an answer choice) and any necessary Java SE 8 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used.

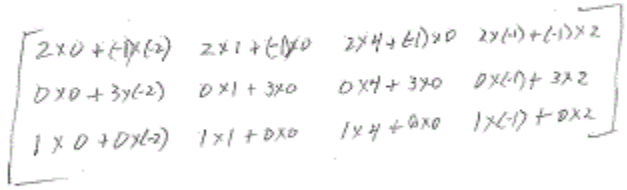
Explanations:

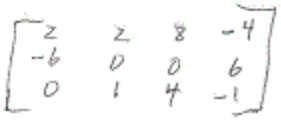
1. B $100_8=64_{10}$, $40_{16}=64_{10}$, $01001000_2=72_{10}$, $98_{16}=152_{10}$, $01000000_2=64_{10}$
2. E 8+5%4-3.0 = 8+1-3.0 = 9-3.0 = 6.0
3. E `%12.2f` means a decimal number, right justified in 12 spaces and rounded to two decimal places.
4. C `s.substring(0, 1)` returns a **g**. `s.substring(s.length()-1)` returns a **t**. Therefore, we are going to replace all of the g's with t's.
5. B

a	b	c	d	a||b&&c^d
F	F	T	F	F
T	F	F	F	T
F	T	T	T	F

6. E Math.round returns a long which can not be assigned to an int type variable.
7. C m is decremented **before** it is multiplied times n. n is incremented **after** the multiplication. The result of the multiplication is 32.832. Casting to an int then truncates the result to 32 which is then promoted back to a double with assignment leaving 32.00.
8. D When a switch statement encounters an empty case statement it will fall through to the next case statement. The values for x are 1, 2, 3, 5 then 6.
9. E Begins with the last letter in the string (`i=s.length()-1`), moves backwards printing every other letter (`i-=2`). Stops before the first letter (`i>0`).
10. C The for loop takes each value in the array `a,` starting at the back, and places it into array `b,` starting at the back, but stops before the zero (0) index is reached therefore `b[0]` is unassigned. The default value is 0 (zero).
11. A The Scanner method next() returns a String which cannot be added to or assigned to a double type variable.
12. D Answer A creates an inaccurate count of the values because b is initialized to 1. Answer B never increments variable b. In answer C, variable b finds the sum of the values instead of the count.
13. C Subtraction comes before bit shifting in the order of operations. >> is division by a power of 2. 16>>4-2 = 16>>2 = $16/2^2$ = 16/4 = 4.
14. A Add one (1) and change the sign.
15. D The set method **replaces** the element at the specified position.
16. E Any class that contains an abstract method must also be abstract.
17. B The odometer field is a private field within the Vehicle class and can not be directly access from within the Electric class.
18. B The default Gas class constructor does not make an explicit call to a super constructor. Therefore, there is an implicit call to the default Vehicle constructor before the output statement is executed.
19. D Objects may not be instantiated from abstract classes.
20. E The method charge() is undefined for the type Vehicle.
21. C Match any capital letter once followed by any lower case letter one or more times followed by a comma then a space followed by any capital letter once followed by any lower case letter one or more times.
22. A Elements in a Map are stored in ascending order of the key values, in this case, locker numbers.
23. E The put method replaces the value for that key. The get method returns the value associated with the designated key but does not remove it.
24. D The code finds the product of matrices a and b.



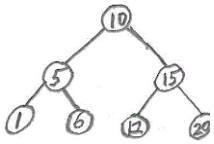




25. D s2 contains [fil, , ditproj, ct]

0	1	2	3
"fil"	" "	"ditproj"	"ct"

26. B All classes inherit from the Object class. Object is the cosmic super class.
27. D While it is counter intuitive to sort a Stack, any class that implements the List interface may be sorted using Collections.sort.
28. C Merging is the last step in the algorithm.
29. E
30. E The merge sort has O(n log n) complexity in all situations, best, worst, and average.
31. A Determines if x is even or odd. Answer choice A is an illustration of the call stack where # is the bottom and ^ is on top.
32. B The values of m and n never change in the client code because they are passed by value to the method.
    m=20 n=2 c=22 d=40
    m=22 n=3 c=22 d=37
    m=25 n=4 c=22 d=33
    m=29 n=5 c=22 d=28
    m=34 n=6 c=22 d=22
    84 20 2
33. C Preorder traversal visits node, left then right.



34. D Same as:

```
int x;
if(s1.length()>s2.length())
      x=s1.length();
else
      x=s1.length();
```

35. A Bit shift before bitwise AND. 5*8/4=10. 10&17=0.

```
    0  1  0  1  0
&   1  0  0  0  1
    0  0  0  0  0
```

36. E * is AND, ⊕ is XOR, and the over bar means NOT. Same as A & !B ^ C.
37. C

10101101	11011001	11111111	00000101	00010011
-83	-39	-1	5	19

38. B C contains the correct number of edges but they connect to different nodes. D contains to many edges. A is a directed graph and would have half as many connections. Therefore, C is correct.
39. Also accept A || !(B && C); A or not (B and C); A|!(B&C); !(B&&C)||A; !(C&&B) || A; A|!(BC).
40. Linked lists do not allow for direct access via an index, so O(n).