

UIL COMPUTER SCIENCE WRITTEN TEST

2019 INVITATIONAL A

JANUARY/FEBRUARY 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 A

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.

What is the decimal equivalent of $2AB_{16}$?

- A) 700 B) 667 C) 342 D) 751 E) 683

Question 2.

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

- A) 34 B) 18 C) -24 D) 50 E) 46

```
out.print(45-8*2/3+-6);
```

Question 3.

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

- A) Socks
 Bubba
 Binky
- B) SocksBubbaBinky
- C) Socks
 BubbaBinky
- D) SocksBubba
 Binky
- E) Socks Bubba Binky

```
out.println("Socks");
out.print("Bubba");
out.println("Binky");
```

Question 4.

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

- A) defghab B) efghbc C) defghabc
- D) efghbcd E) fghbcd

```
String str="abcdefgh";
out.print(str.substring(4)+
          str.substring(1, 3));
```

Question 5.

Which of the lines shown on the right will print false?

- A) line #1
- B) line #2
- C) line #3
- D) line #4
- E) More than one of the above.

```
boolean a=true,b=true,c=true;
out.println(a&&b&&c); //line #1
out.println(a&&b||c); //line #2
out.println(a&&b^c); //line #3
out.println(a||b^c); //line #4
```

Question 6.

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

- A) 3 B) 3.0 C) 4 D) 4.0 E) 3.2

```
double x=Math.PI;
double y=Math.ceil(x);
out.print(y);
```

Question 7.

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

- A) 10.875 B) 3.375 C) 10 D) 11 E) 4.0

```
int i=10;
double d=3.5;
long g=4;
out.print(i+d/g);
```

Question 8. What is the output of the code segment to the right? A) 20 12 B) 20 -8 C) 5 20 D) 5 -3 E) 7 -3	<pre>int m=15,n=-8; if (m>=n) n=20; if (n>=0) m=5; if (n*m<=100) { m=7; n=-3;} out.print(m+" "+n);</pre>
Question 9. How many asterisks are printed by the code shown to the right? A) 4 B) 5 C) 0 D) 8 E) 9	<pre>int x; for (x=4;x<=8;x++) out.print("*");</pre>
Question 10. What is the output or the error of the code segment to the right? A) 3 B) 5 C) Error. ArrayIndexOutOfBoundsException exception. D) Error in line #1. E) Error in line #2.	<pre>String []list=new String[5];//line #1 list[0]="one"; list[1]="two"; list[4]="three"; out.print(list.length);//line #2</pre>
Question 11. Which of the following lines in the code segment shown on the right contains an error? All file input setup is correct. A) line #1 B) line #2 C) line #3 D) line #4 E) None of the above. There are no errors.	<pre>File f=new File("datafile.dat");//line #1 Scanner scr=new Scanner(f);//line #2 while(scr.hasNextInt()) //line #3 out.print(scr.nextInt()+" ");//line #4</pre>
Question 12. What is the output of the code segment to the right? A) 36 71 B) 49 154 C) 42 105 D) 35 70 E) 48 153	<pre>int x=0,y=0; while (x<40) { y+=x; x+=7; } out.print(x+" "+y);</pre>
Question 13. What is the correct order of operations (from left to right) for the operators listed on the right? A) ++ * < && += B) ++ && * < += C) += ++ && * < D) ++ < * && += E) * < ++ && +=	&& += ++ < *

Question 14.

Which of the following lines of code will NOT compile?

- A) `byte b=32768;`
- B) `short s=32768;`
- C) `int i=32768;`
- D) A and B
- E) B and C

Question 15.

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

- A) 0 4
- B) -9 4
- C) 14 5
- D) -9 3
- E) 0 5

```
ArrayList<Integer> list=new  
ArrayList<Integer>();  
list.add(3);list.add(14);list.add(-9);  
list.set(1, 0);list.add(0, 5);  
out.print(list.get(3)+" "+list.size());
```

Question 16.

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

- A) true true true
- B) true false true
- C) true false false
- D) false true false
- E) false true true

```
String s1=new String("InvitationalA");  
String s2=new String("InvitationalA");  
out.print((s1==s2)+" ");  
out.print(s1.equals(s2)+" ");  
out.print(s2.equals("InvitationalA"));
```

Question 17.

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

- A) 9.0
- B) 10
- C) 9
- D) 9.781
- E) There is no output due to an error.

```
double d1=5.031,d2=4.75;  
int i=(int)d1+(int)d2;  
out.print(i);
```

Question 18.

Which of the following methods will return the sum of the digits in the parameter n? n will always be positive.

A)

```
public static long sumOfDigs(long n){
    long sum=0;
    for(long x=n;x>0;x/=10)
        sum+=x/10;
    return sum;
}
```

C)

```
public static long sumOfDigs(long n) {
    long sum=0;
    do {
        sum+=n%10;
        n=n/10;
    }while(n>0);
    return sum;
}
```

E) More than one of the above.**B)**

```
public static long sumOfDigs(long n) {
    long sum=0;
    String str=Long.toString(n);
    for(int i=0;i<str.length();i++)
        sum+=Long.parseLong(
            str.substring(i, i+1));
    return sum;
}
```

D)

```
public static long sumOfDigs(long n) {
    long sum=0;
    while(n>0) {
        n%=10;
        sum+=n;
        n/=10;
    }
    return sum;
}
```

Question 19.

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

- A)** [b, tm, nsuperm, n, qu, m, n]
- B)** [b, tmansupermanaquaman]
- C)** [batmansuperm, naquaman]
- D)** [ba, tma, nsuperma, na, qua, ma, n]
- E)** [a, a, a, a, a, a]

```
String s="batmansupermanaquaman";
String[] spl=s.split("a");
out.print(Arrays.toString(spl));
```

Question 20.

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

- A)** [8, 5, 2] **B)** [9, 6, 1] **C)** [8, 2, 5]
 [3, 4, 7] [7, 4, 3] [3, 7, 4]
 [1, 6, 9] [2, 5, 8] [1, 9, 6]
- D)** [2, 5, 8] **E)** [9, 6, 1]
 [7, 4, 3] [2, 5, 8]
 [9, 6, 1] [7, 4, 3]

```
int[][] mat= {{2,5,8},
               {7,4,3},
               {9,6,1}};
int[] x=mat[0];
mat[0]=mat[2];
mat[2]=x;
for(int[] m:mat)
    out.println(Arrays.toString(m));
```

Question 21.

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

- A)** 11.25
- B)** Total=11.25
- C)** Total=11
- D)** Total=74.25
- E)** Total=74

```
String s="Total=";
int i=7;
double d=4.25;
out.print(s+i+d);
```

Question 22. The method shown on the right is intended to implement a binary search algorithm. Which of the following must replace <code 1> both places it occurs to ensure the method will compile and execute as intended? A) <code>list.length/2</code> B) <code>(front+back)/2</code> C) <code>list.length-1/2</code> D) <code>(front+list.length)/2</code> E) <code>front+back/2</code>	//Use the following to answer questions 22, 23 and 24. <pre> public static int bs(int[] list,int e) { int i=-1; int front=0,back=list.length-1; int mid=<code 1>; while(back>=front) { if(list[mid]==e) { i=mid; break; } else if(e<list[mid]) back=mid-1; else front=mid+1; mid=<code 1>; } return <code 2>; } </pre>
Question 23. Which of the following must replace <code 2> to ensure the method will compile and execute as intended? A) <code>i</code> B) <code>mid</code> C) <code>front</code> D) <code>back</code> E) No additional code is required.	
Question 24. Once implemented correctly, and if <code>n</code> is the length of the array, which of the following is the strictest correct runtime? A) <code>O(1)</code> B) <code>O(n)</code> C) <code>O(n²)</code> D) <code>O(log n)</code> E) <code>O(n log n)</code>	
Question 25. What is the output of the code segment to the right? A) <code>a@ \$2#d*5</code> B) <code>a w d</code> C) <code>@ \$ # *</code> D) <code>a w 2 d 5</code> E) <code>@ \$ 2 # * 5</code>	<pre> String s="a@w\$2#d*5"; Scanner scr=new Scanner(s); scr.useDelimiter("\\w"); while(scr.hasNext()) out.print(scr.next()+" "); </pre>
Question 26. Given the declaration of <code>r</code> shown on the right, which of the following will print only values between 20 (inclusive) and 40 (exclusive)? A) <code>out.print(r.nextInt(40));</code> B) <code>out.print(r.nextInt()+20);</code> C) <code>out.print(r.nextInt(20)+19);</code> D) <code>out.print(r.nextInt(39));</code> E) <code>out.print(r.nextInt(20)+20);</code>	<pre> Random r=new Random(); </pre>

Question 27. Which of the following could serve as a correct alternative for the condition of the while loop marked with the comment and still produce the same output? A) <code>q.next() != null</code> B) <code>q.peek() != null</code> C) <code>q.hasNext()</code> D) <code>q.size() >= 0</code> E) <code>Queue.hasNext(q)</code>	
Question 28. Which of the following represents the output of the code segment listed on the right before line #1 has executed? A) gamma delta chi B) delta gamma chi C) beta chi delta gamma D) alpha beta chi delta gamma E) chi delta gamma	<pre>//Use the following code segment to //answer questions 27, 28 and 29. Queue<String> q=new PriorityQueue<String>(); q.add("delta");q.add("beta"); q.add("gamma");q.add("alpha"); q.poll();q.add("chi");q.remove(); while(!q.isEmpty())//comment out.print(q.remove()+" "); out.print("\n"+q.size());//line #1</pre>
Question 29. Which of the following represents the output of just line #1 in the code segment? A) 0 B) 3 C) 4 D) 5 E) There is no output because line #1 throws an exception	
Question 30. Which of the following values would be returned by this call to method <code>mtd</code> shown on the right? <code>mtd(9)</code> A) 55 B) 34 C) 21 D) 15 E) 54	<pre>public static int mtd(int n) { if(n==0) return 0; else if(n==1) return 1; else return mtd(n-1)+mtd(n-2); }</pre>
Question 31. Which of the following represents the output of the <code>main</code> method shown on the right? A) 125 5 50 10 B) 125 55 50 10 C) 275 5 50 10 D) 275 55 50 5 E) 0 50 5 10	<pre>public static void main(String[] args) { int x=5,y=50,z=10; out.print(go(x,y,z)+" "); out.print(x+" "+y+" "+z); } public static int go(int x,int y,int z) { int p=0; while(x<=y) { p=p+x; x+=z; } return p; }</pre>

Question 32.

The constructor for the Contestant class shown on the right is _____.

- A) overridden
- B) overwritten
- C) overloaded
- D) extended
- E) inherited

Question 33.

Which of the following will not correctly instantiate a Contestant object?

- A) Contestant c1=new Contestant("Bob", "123", 232);
- B) Contestant c2=new Contestant("Sue", "321");
- C) Contestant c3=new Contestant();
- D) Contestant c4=new Contestant("Al", 231, 85);
- E) None of the above. All are correct.

Question 34.

What is the output of the client code shown here?

```
Contestant c5=new Contestant();
out.print(c5.getName()+" "+c5.getIdNum()
        +" "+c5.getScore());
```

- A) 0
- B) null null 0
- C) null 0
- D) There is no output and there are no errors.
- E) There is no output due to an error.

Question 35.

Which of the following methods, when added to the Contestant class, will compile and correctly assign a value to the instance variable score?

- A) public int setScore(int s) {
 score = s;}
- B) public void setScore(int s) {
 return score;}
- C) public void setScore(int s) {
 s = score;}
- D) public void setScore() {
 score = s;}
- E) public void setScore(int s) {
 score = s;}

```
//Use the class implemented here to answer
//questions 32 - 35.
public class Contestant {

    private String name, idNum;
    private int score;

    public Contestant(String n, String id, int
s){
        name=n;
        idNum=id;
        score=s;}

    public Contestant(String n, String id) {
        name=n;
        idNum=id;}

    public Contestant() {}

    public int getScore() {
        return score;}

    public String getName() {
        return name;}

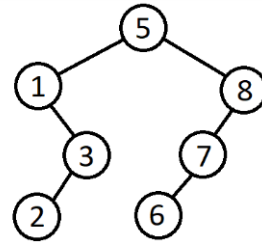
    public String getIdNum() {
        return idNum;}

}
```

Question 36.

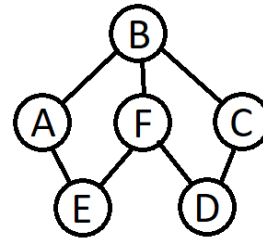
Which of the following represents a post-order traversal of the binary search tree illustrated on the right?

- A) 5 1 3 2 8 7 6
- B) 2 3 1 6 7 8 5
- C) 1 2 3 5 6 7 8
- D) 5 1 8 3 7 2 6
- E) 8 7 6 2 3 1 5

**Question 37.**

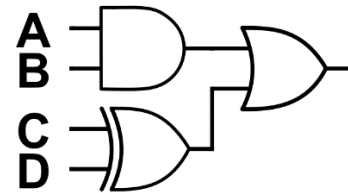
Which of the following represents the longest simple cycle within the graph shown on the right?

- A) A B C D E A
- B) A E F B A
- C) C D F E A B C
- D) A B C D F B C
- E) A B C D F E

**Question 38.**

Which of the following Boolean expressions is diagrammed on the right?

- A) $A * B + C \oplus D$
- B) $A + B * C \oplus D$
- C) $A * B + \overline{C + D}$
- D) $A + B \oplus C * D$
- E) $A * B + \overline{C \oplus D}$

**Question 39.**

Evaluate the postfix expression shown on the right and write your answer in the blank provided on the answer document.

8 3 4 * 2 / + 5 -

Question 40.

Write the 8-bit binary two's complement representation of -90 in the blank provided on the answer document.

UIL COMPUTER SCIENCE WRITTEN TEST

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

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

FOR ADMINISTRATIVE USE ONLY

# Right:	×	6 pts	=	
# Wrong:	×	-2 pts	=	
# Skipped:	×	0 pts	=	0

	Score	Initials
Judge #1:		
Judge #2:		
Judge #3:		

★ ANSWER KEY – CONFIDENTIAL ★

UIL COMPUTER SCIENCE – 2019 INVITATIONAL A

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

- | | | | |
|------------------|------------------|------------------|------------------------|
| 1) <u> E </u> | 11) <u> E </u> | 21) <u> D </u> | 31) <u> A </u> |
| 2) <u> A </u> | 12) <u> C </u> | 22) <u> B </u> | 32) <u> C </u> |
| 3) <u> C </u> | 13) <u> A </u> | 23) <u> A </u> | 33) <u> D </u> |
| 4) <u> B </u> | 14) <u> D </u> | 24) <u> D </u> | 34) <u> B </u> |
| 5) <u> C </u> | 15) <u> B </u> | 25) <u> C </u> | 35) <u> E </u> |
| 6) <u> D </u> | 16) <u> E </u> | 26) <u> E </u> | 36) <u> B </u> |
| 7) <u> A </u> | 17) <u> C </u> | 27) <u> B </u> | 37) <u> C </u> |
| 8) <u> E </u> | 18) <u> E </u> | 28) <u> E </u> | 38) <u> A </u> |
| 9) <u> B </u> | 19) <u> A </u> | 29) <u> A </u> | *39) <u> 9 </u> |
| 10) <u> B </u> | 20) <u> B </u> | 30) <u> B </u> | *40) <u> 10100110 </u> |

* 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.	E	A ₁₆ =10 ₁₀ B ₁₆ =11 ₁₀ 11*1=11 10*16=160 2*256=512 11+160+512=683																	
2.	A	45*8*2/3+-6 = 45-16/3-6 = 45-5-6 = 40-6 = 34																	
3.	C	println moves the cursor to the next line. print does not.																	
4.	B	String index values start at 0. <table border="1"><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td>a</td><td>b</td><td>c</td><td>d</td><td>e</td><td>f</td><td>g</td><td>h</td></tr></table> str.substring(4) returns efgh str.substring(1, 3) returns bc		0	1	2	3	4	5	6	7	a	b	c	d	e	f	g	h
0	1	2	3	4	5	6	7												
a	b	c	d	e	f	g	h												
5.	C	^ (XOR) evaluates as true if one <u>but not both</u> operands are true. && (AND) evaluates as true if <u>both</u> operands are true. true&&true^true = true&&false = false																	
6.	D	Math.PI = 3.141592653589793 Math.ceil(x) returns returns the smallest <u>double</u> value that is greater than or equal to the argument and is equal to a mathematical integer.																	
7.	A	10+3.5/4 = 10+0.875 = 10.875																	
8.	E	<table border="1"><tr><td></td><td>m</td><td>n</td></tr><tr><td></td><td>15</td><td>-8</td></tr><tr><td>15>-8 is true</td><td>15</td><td>20</td></tr><tr><td>20>=0 is true</td><td>5</td><td>20</td></tr><tr><td>20*5<=100 is true</td><td>7</td><td>-3</td></tr></table>			m	n		15	-8	15>-8 is true	15	20	20>=0 is true	5	20	20*5<=100 is true	7	-3	
	m	n																	
	15	-8																	
15>-8 is true	15	20																	
20>=0 is true	5	20																	
20*5<=100 is true	7	-3																	
9.	B	x takes on the values 4, 5, 6, 7, and 8. Loop iterates 5 times and prints 5 stars.																	
10.	B	Arrays have a fixed length set at the time of initialization (line #1) even if values are not assigned to every index value.																	
11.	E	The segment of code will compile and execute as intended.																	
12.	C	<table border="1"><tr><td>x</td><td>y</td></tr><tr><td>0</td><td>0</td></tr><tr><td>7</td><td>0</td></tr><tr><td>14</td><td>7</td></tr><tr><td>21</td><td>21</td></tr><tr><td>28</td><td>42</td></tr><tr><td>35</td><td>70</td></tr><tr><td>42</td><td>105</td></tr></table>		x	y	0	0	7	0	14	7	21	21	28	42	35	70	42	105
x	y																		
0	0																		
7	0																		
14	7																		
21	21																		
28	42																		
35	70																		
42	105																		
13.	A	Standard Java operator precedence rules.																	
14.	D	<table border="1"><tr><td>Data Type</td><td>Range of Values</td></tr><tr><td>byte</td><td>-128 to 127</td></tr><tr><td>short</td><td>-32768 to 32767</td></tr><tr><td>int</td><td>-2147483648 to 2147483647</td></tr></table>		Data Type	Range of Values	byte	-128 to 127	short	-32768 to 32767	int	-2147483648 to 2147483647								
Data Type	Range of Values																		
byte	-128 to 127																		
short	-32768 to 32767																		
int	-2147483648 to 2147483647																		
15.	B	<table border="1"><tr><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><td>3</td><td>14</td><td>-9</td><td></td></tr><tr><td>3</td><td>0</td><td>-9</td><td></td></tr><tr><td>5</td><td>3</td><td>0</td><td>-9</td></tr></table>		0	1	2	3	3	14	-9		3	0	-9		5	3	0	-9
0	1	2	3																
3	14	-9																	
3	0	-9																	
5	3	0	-9																
16.	E	Both s1 and s2 are String objects. When comparing two objects the equality operator (==) returns true if the variables are pointing at the same object. In this case s1 and s2 are not the same object. The equals method determines if two String objects store the exact same string of characters.																	
17.	C	Casting truncates the values stored in the variables d1 and d2. So, 5 + 4 = 9.																	
18.	E	Both B and C are correct.																	
19.	A	The split method divides the string on "a" and removes the "a".																	
20.	B	The code segment switches the top and bottom rows in the 2D array mat.																	
21.	D	Since s is a String, the + operator performs concatenation throughout the expression.																	

22.	B	A binary search works by dividing the list in two on each iteration of the loop and discarding the half of the list that does not contain the target value and continuing the search in the half that does.
23.	A	The method returns the location within the list (index value) of the target value or -1 if the target is not found. In this case the index is stored in <code>i</code> .
24.	D	Reducing the size of the list by $\frac{1}{2}$ on each iteration results in logarithmic time efficiency.
25.	C	The regular expression <code>\\w</code> matches any word character (letters and numbers). The delimiter (in this case letters and numbers) is not printed.
26.	E	<code>nextInt(n)</code> returns the next random integer between 0 (inclusive) and <code>n</code> (exclusive).
27.	B	A, C and E are not valid code. D makes the code throw a <code>NoSuchElementException</code> .
28.	E	Because <code>q</code> is a priority queue, elements are removed in alphabetical order. After adding the original four elements to the queue, the call to <code>poll</code> removes "alpha". After the addition of "chi" the call to <code>remove</code> removes "beta". The elements are then printed in alphabetical order.
29.	A	The loop removes all of the elements in the queue.
30.	B	Returns the 9 th term in the Fibonacci sequence. Here is the call stack. Count the ones. 9 8 7 6 5 4 3 2 1 0 1 2 1 0 3 2 1 0 1 4 3 2 1 0 1 2 1 0 5 4 3 2 1 0 1 2 1 0 3 2 1 0 1 6 5 4 3 2 1 0 1 2 1 0 3 2 1 0 1 4 3 2 1 0 1 2 1 0 7 6 5 4 3 2 1 0 1 2 1 0 3 2 1 0 1 4 3 2 1 0 1 2 1 0 5 4 3 2 1 0 1 2 1 0 3 2 1 0 1
31.	A	<code>x</code> , <code>y</code> and <code>z</code> are passed by value, so their values are never changed in the main method. Here is a trace of the values within the <code>go</code> method: x y z p 5 50 10 0 15 50 10 5 25 50 10 20 35 50 10 45 45 50 10 80 55 50 10 125
32.	C	A class can have more than one method (in this case the constructors) with the same name as long as their parameter lists differ in either number of parameters or type.
33.	D	The second argument, the integer 231, does not match the type of the 2 nd parameter in the three-argument constructor. It would have to be a <code>String</code> .
34.	B	The no argument constructor for the <code>Contestant</code> class does not assign any values to the instance variables. The default values are null for <code>Strings</code> and zero for <code>int</code> .
35.	E	Setter methods should be <code>void</code> and not return anything. The method should assign the parameter <code>s</code> to the field <code>score</code> .
36.	B	Post order visits the nodes left, right, and then root.
37.	C	A cycle is a connected path that begins and ends at the same vertex.
38.	A	$\neg D$ is * (AND). $\neg D$ is + (OR). $\neg D$ is $\oplus$ (XOR).

39.	9	$834 \times 2 / + 5 - =$ $8122 / + 5 - =$ $86 + 5 - =$ $145 - =$ 9
40.	10100110	Write down 90 in binary ignoring the minus sign. 01011010 Find the complement by flipping all of the bits. 10100101 Add one. 10100110