

# UIL COMPUTER SCIENCE WRITTEN TEST

# 2017 DISTRICT

March 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 DISTRICT

**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 the sum of  $A4_{16}$  and  $3E_{16}$ ?

- A)  $228_{10}$                       B)  $341_8$                       C)  $E2_{16}$                       D)  $11100011_2$                       E) None of the above.

## Question 2.

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

- A) 3.0      B) 3      C) 4      D) 4.0      E) 1

```
out.println(12/4.0*5/5);
```

## Question 3.

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

- A) Computer  
Science  
B) \"Computer\"  
\"Science\"  
C) \"Computer\"  
\"Science\"  
D) Computer Science  
E) \"Computer\" n \"Science\"

```
out.print(\"\"Computer\"\"\n\"Science\"");
```

## Question 4.

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

- A) sketball      B) asketball      C) all  
D) ll                      E) basketball

```
String s="basketball";  
out.print(s.substring(s.indexOf("a")));
```

## Question 5.

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

- A) true                      B) false

```
boolean b=false,c=true,d=false,e;  
e=b^(c||d);  
out.print(e);
```

## Question 6.

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

- A) 9.0      B) 8.0      C) 9      D) 6.0      E) 8

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

## Question 7.

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

- A) e      B) E32      C) 133      D) 101  
E) Error. Will not compile because of a type mismatch.

```
char c='E';  
int d=32;  
out.print(c+d);
```

## Question 8.

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

- A) Warning  
B) Warning 38.5  
C) Warning Danger 38.5  
D) Danger 28.0  
E) Warning 28.0

```
double p=38.5;  
if(p<32)  
    out.print("OK");  
else if(p<45)  
    out.print("Warning");  
else  
    out.print("Danger");  
    p-=10.5;  
out.print(" "+p);
```

<p><b>Question 9.</b></p> <p>Which of the following represents the output of this code segment?</p> <p>A) 0 2 4 6 8  B) 2 4 6 8 10  C) 2 4 6 8  D) 0 4 8 12  E) 0 2 4 6 8 10</p>	<pre>int a=0; do{     out.print(a+" ");     a+=2; }while(a&lt;10); out.print(a);</pre>
<p><b>Question 10.</b></p> <p>What is the output of the code segment to the right?</p> <p>A) [0, 3, 2, 2, 4]  B) [0, 0, 2, 6, 2]  C) [0, 0, 2, 6, 6]  D) [0, 0, 2, 5, 5]  E) Error. Throws an ArrayIndexOutOfBoundsException</p>	<pre>int a[]={1,3,2,5,4}; a[a[0]]=0; a[0]=a[a[0]]; a[4]=a[a[3]]; out.print(Arrays.toString(a));</pre>
<p><b>Question 11.</b></p> <p>Which of the following can replace <b>&lt;code&gt;</b> in this class so that the class will compile and run correctly?</p> <pre>import static java.lang.System.out; import java.io.*; import java.util.*; public class Abc {      public static void main(String[] args) <b>&lt;code&gt;</b>{         Scanner s=new Scanner(new File("datafile.dat"));         while(s.hasNext())             out.println(s.next());     } }</pre> <p>A) throws IOException  B) throws FileNotFoundException  C) IOException  D) throws ScannerException  E) No additional code is required.</p>	
<p><b>Question 12.</b></p> <p>What is the output of the code segment to the right?</p> <p>A) dcoagt  B) dogcat  C) cdaotg  D) gt  E) No output. Throws StringIndexOutOfBoundsException.</p>	<pre>String s1="dog",s2="cat",s3=""; int i=0; while(i&lt;s1.length()){     s3=s3+s1.charAt(i)+s2.charAt(i);     i++; } out.print(s3);</pre>
<p><b>Question 13.</b></p> <p>What is the value of the expression shown to the right?</p> <p>A) true  B) false  C) 225  D) 15  E) 9</p>	<p>14 11&amp;9</p>

**Question 14.**

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

- A) 32767
- B) 32768
- C) 2147483647
- D) 2147483648
- E) 9223372036854775807

```
out.print(Integer.MAX_VALUE);
```

**Question 15.**

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

- A) 3.14
- B) 2.72
- C) 1.62
- D) 6.28
- E) 0.0

```
ArrayList<Double> a=new ArrayList<Double>();  
a.add(3.14);  
a.add(2.72);  
a.set(0, 1.62);  
a.set(0, 6.28);  
out.print(a.get(1));
```

**Question 16.**

To ensure that the values stored in parameters *f* and *l* are correctly assigned to the fields *first* and *last*, which of the following lines of code can replace **<code1>** in the class listed to the right?

- A) `super(f, l);`
- B) `this.Student(f, l);`
- C) `Student.super(f, l);`
- D) `Student.this(f, l);`
- E) `this(f, l);`

**Question 17.**

Which of the following should replace **<code2>** to ensure that the method `calculate()` will compile and run correctly?

- A) `float`
- B) `int`
- C) `void`
- D) `double`
- E) No additional code is required.

**Question 18.**

Assume that **<code1>** and **<code2>** have been replaced correctly in the class `Student` shown to the right. What is the output of the client code shown here?

```
Student s1=new
Student("Bob", "Smith", "1234");
s1.add(90);
s1.add(65);
s1.add(93);
out.println(s1);
```

- A) Bob Smith 1234 82.0
- B) Bob Smith 1234 83.0
- C) Bob Smith 1234 83
- D) Bob Smith 1234 82
- E) Bob Smith 1234 82.666666666666667

**Question 19.**

Assume that **<code1>** and **<code2>** have been replaced correctly in the class `Student` shown to the right. What is the output of the client code shown here?

```
Student s2=new Student("Jane", "Jones");
s2.add(95);
out.println(s2);
```

- A) Jane Jones
- B) Jane Jones 95.0
- C) Jane Jones null 95.0
- D) Error. Throws a `NullPointerException`.
- E) Error. Will not compile.

```
public class Student {
    private String first;
    private String last;
    private String id;
    private ArrayList<Integer> grades = new
    ArrayList<Integer>();

    public Student(){}

    public Student(String f,String l){
        first=f;
        last=l;
    }

    public Student(String f,String l,String i){
        <b><code1></b>
        id=i;
    }

    public void add(int g){
        grades.add(g);
    }

    public <b><code2></b> calculate(){
        double total=0.0;
        for(int g:grades)
            total+=g;
        return Math.round(total/grades.size());
    }

    public String toString(){
        return first+" "+last+" "+id+"
        "+calculate();
    }
}
```

<p><b>Question 20.</b></p> <p>What is the output of the code segment to the right?</p> <p>A) Telx hsem  B) Telewax h  C) TNOeekxwl  D) Tx sem  E) There is no output due to an error.</p>	<pre>char[][] c=new char[3][]; c[0]="Texas".toCharArray(); c[1]="New Mexico".toCharArray(); c[2]="Oklahoma".toCharArray(); for(int x=0;x&lt;c[0].length;x+=2){     out.print(c[0][x]);     out.print(c[1][x+1]);     out.print(c[2][x+2]); }</pre>
<p><b>Question 21.</b></p> <p>What is the output of <b>line #1</b> in the code segment to the right?</p> <p>A) [lizard, frog, snake, frog] [dog, cat, snake]  B) [lizard, frog, snake] [dog, cat, snake]  C) [frog, lizard, snake] [cat, dog, snake]  D) [frog, frog, lizard, snake] [cat, dog, snake]  E) [frog, lizard, snake] [cat, dog]</p>	<pre>Set&lt;String&gt; s1=new TreeSet&lt;String&gt;(); Set&lt;String&gt; s2=new TreeSet&lt;String&gt;(); s1.add("lizard"); s1.add("frog"); s1.add("snake"); s2.add("dog"); s2.add("cat"); s1.add("frog"); s2.add("snake"); out.println(s1+" "+s2); //line #1 s1.retainAll(s2); out.println(s1); //line #2</pre>
<p><b>Question 22.</b></p> <p>What is the output of <b>line #2</b> in the code segment to the right?</p> <p>A) [frog, cat, snake, lizard, dog]  B) [snake]  C) [lizard, frog]  D) [frog, lizard]  E) [cat, dog, frog, lizard, snake]</p>	<pre>out.println(Integer.toString(8, 2));</pre>
<p><b>Question 23.</b></p> <p>What is the output of the code segment to the right?</p> <p>A) 2  B) 64  C) 8.00  D) 8 8  E) 1000</p>	<pre>out.println(Integer.toString(8, 2));</pre>
<p><b>Question 24.</b></p> <p>Which of the following Java expressions is equivalent to <code>a&amp;&amp;(b  c)</code>?</p> <p>A) <code>(a  b)&amp;&amp;(a  c)</code>    B) <code>a&amp;&amp;b  a&amp;&amp;c</code>    C) <code>(a  b)  c</code>    D) <code>a  b&amp;&amp;c</code>    E) <code>!(a&amp;&amp;b)  !c</code></p>	
<p><b>Question 25.</b></p> <p>Which of the following cannot be the output of the code segment to the right?</p> <p>A) 21  B) 20  C) 29  D) 30  E) All of the above are possible output.</p>	<pre>double r=Math.random(); int s=(int)(r*10+20); out.print(s);</pre>

**Question 26.**

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

- A) 14
- B) 19
- C) 6
- D) 10
- E) 15

```
int t=0;
for(int x=1;x<=5;x++){
    switch(x){
        case 1:t+=x;break;
        case 2:t+=x+1;
        case 3:t*=2;break;
        case 4:t+=3;
        default: t-=x;
    }
}
out.print(t);
```

**Question 27.**

The least restrictive run time efficiency (Big O value) of a sequential search is  $O(n)$ . If a method that implements the sequential search algorithm can search a list of 30,000 items in 0.75 seconds, how long will it take for that same method to search a list of 90,000 items?

- A) 2.25 seconds
- B) 6.75 seconds
- C) 1.5 seconds
- D) 0.48 seconds
- E) 3.0 seconds

**Question 28.**

What is the output of this client code given the implementation of method abc shown on the right?

```
out.println(abc(0));
```

- A) 51
- B) 96
- C) 24
- D) 53
- E) 30

```
public static int abc(int x){
    if(x>10)
        return x-3;
    else
    {
        x*=3;
        return x+abc(x+2);
    }
}
```

**Question 29.**

If the four methods shown to the right are all contained within the same class, which pair will cause the class to have a compile error?

- A) I and II
- B) I and III
- C) II and III
- D) II and IV
- E) III and IV

```
//Method I
public static double abc(int x, int y){
    return x+y;
}
//Method II
public static double abc(int x, double y){
    return x+y;
}
//Method III
public static int abc(String s){
    return s.length();
}
//Method IV
public static String abc(String s){
    return s.substring(0);
}
```



**Question 30.**

What should replace `<code>` in the method `sort` shown to the right so that `list` is sorted in ascending order?

- A) `list[j]<x`
- B) `x<list[j]`
- C) `x<j`
- D) `x>list[j]`
- E) `list[j]<list[j+1]`

**Question 31.**

Assuming that `<code>` has been replaced correctly, which sorting algorithm does method `sort` implement?

- A) bubble sort
- B) insertion sort
- C) selection sort
- D) merge sort
- E) quicksort

**Question 32.**

What is the least restrictive worst case running time (Big O value) for the method `sort`?

- A)  $O(n \log n)$
- B)  $O(\log n)$
- C)  $O(n^2)$
- D)  $O(n)$
- E)  $O(1)$

```
public static void sort(int[] list){
for(int i=1;i<list.length;i++){
    int x=list[i];
    int j=i-1;
    while(j>=0&&<code>){
        list[j+1]=list[j];
        j--;
    }
    list[j+1]=x;
}
```

**Question 33.**

Which of the following methods will correctly determine if any value of x is prime?

**A)**

```
public static boolean isPrime(int x){
    boolean p=false;
    int i=2;
    while(p&&i<=Math.sqrt(x)){
        if(x%i==0){
            p=false;
            break;
        }
        else
            i++;
    }
    return p;
}
```

**B)**

```
public static boolean isPrime(int x){
    boolean p=true;
    int i=2;
    while(p&&i<Math.sqrt(x)){
        if(x%i==0){
            p=false;
            break;
        }
        else
            i++;
    }
    return p;
}
```

**C)**

```
public static boolean isPrime(int x){
    boolean p=true;
    int i=4;
    while(p&&i<x){
        if(x%i==0){
            p=false;
            break;
        }
        else
            i++;
    }
    return p;
}
```

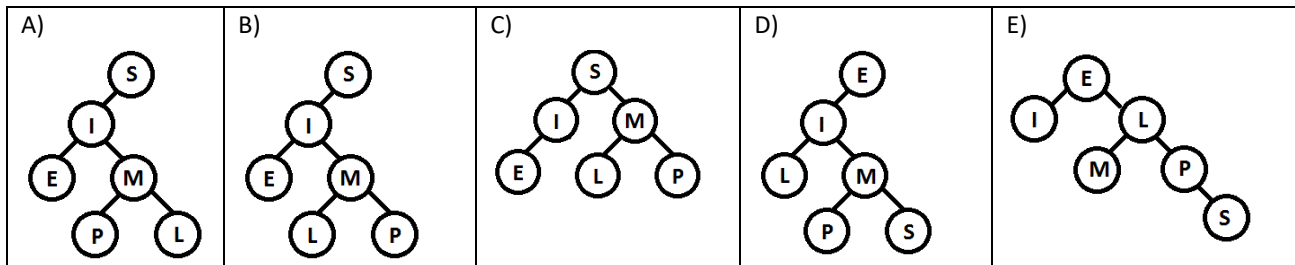
**D)**

```
public static boolean isPrime(int x){
    boolean p=true;
    int i=2;
    while(p&&i<=Math.sqrt(x)){
        if(x%i==0){
            p=false;
            break;
        }
        else
            i++;
    }
    return p;
}
```

**E)** All of the above will correctly determine if any value of x is prime.

**Question 34.**

If the letters in the word SIMPLE are placed into a binary search tree starting with S and ending with E, which of the following is a correct representation of that tree?



**Question 35.**

The code segment shown to the right will not compile. Which of the lines marked are the cause of the compilation error?

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

```
int x=0,y=0;
for(;x<10;){ //line #I
    double z=5.0;
    y=y+(int) (x*z); //line #II
    x++;
}
out.print(x+y+z); //line #III
```

**Question 36.**

How many ordered pairs will make the Boolean expression shown to the right false?

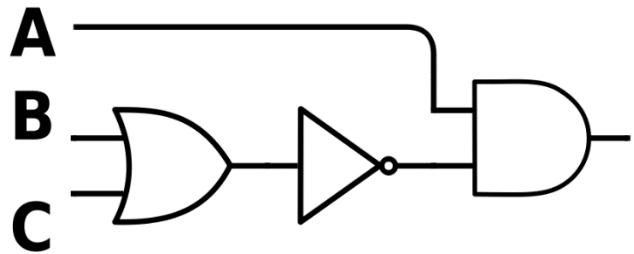
- A) 0
- B) 1
- C) 2
- D) 3
- E) 4

$$\bar{A} + B$$

**Question 37.**

Which of the following Java expressions is equivalent to the Boolean expression depicted in the illustration on the right?

- A) A || ! (B&&C)
- B) A&& (!B || !C)
- C) ! (A&&B | !C)
- D) A&&! (B^C)
- E) A&&! (B | !C)

**Question 38.**

Which of the following is the signed 8-bit two's complement representation of -54?

- A) 00110110
- B) 11001001
- C) 11001000
- D) 01001001
- E) 11001010

**Question 39.**

After the code listed below executes, which element is at the head of the queue?

```
String[] list={"milk", "eggs", "bread", "sugar", "flour", "chips", "apples", "coffee"};
PriorityQueue<String> q=new PriorityQueue<String>();
for(String s:list) q.add(s);
q.remove();
q.remove();
q.remove();
```

**Question 40.**

Rewrite this expression for calculating the area of a trapezoid using reverse polish notation (postfix).

$$\frac{1}{2}(b_1+b_2)h$$

# ★ ANSWER KEY – CONFIDENTIAL ★

## UIL COMPUTER SCIENCE – 2017 DISTRICT

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

- |                  |                  |                  |                                     |
|------------------|------------------|------------------|-------------------------------------|
| 1) <u>  C  </u>  | 11) <u>  A  </u> | 21) <u>  C  </u> | 31) <u>  B  </u>                    |
| 2) <u>  A  </u>  | 12) <u>  A  </u> | 22) <u>  B  </u> | 32) <u>  C  </u>                    |
| 3) <u>  C  </u>  | 13) <u>  D  </u> | 23) <u>  E  </u> | 33) <u>  D  </u>                    |
| 4) <u>  B  </u>  | 14) <u>  C  </u> | 24) <u>  B  </u> | 34) <u>  B  </u>                    |
| 5) <u>  B  </u>  | 15) <u>  B  </u> | 25) <u>  D  </u> | 35) <u>  C  </u>                    |
| 6) <u>  A  </u>  | 16) <u>  E  </u> | 26) <u>  D  </u> | 36) <u>  B  </u>                    |
| 7) <u>  D  </u>  | 17) <u>  D  </u> | 27) <u>  A  </u> | 37) <u>  E  </u>                    |
| 8) <u>  E  </u>  | 18) <u>  B  </u> | 28) <u>  D  </u> | 38) <u>  E  </u>                    |
| 9) <u>  E  </u>  | 19) <u>  C  </u> | 29) <u>  E  </u> | *39) <u>  coffee  </u>              |
| 10) <u>  E  </u> | 20) <u>  A  </u> | 30) <u>  B  </u> | *40) <u>  1 2 / b1 b2 + * h *  </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. C  $A4_{16} = 164_{10}$  and  $3E_{16} = 62_{10}$ .  $164 + 62 = 226$ .  $226_{10} = E2_{16}$ .  $341_8 = 225_{10}$ .  $11100011_2 = 227_{10}$ .
2. A  $12/4.0*5/5 = 3.0*5/5 = 15.0/5 = 3.0$ .
3. C `\` is the escape sequence to insert a quotation mark. `\n` inserts a new line.
4. B `indexOf("a")` returns the first occurrence of the letter a. The substring method with just one argument starts at the index value of a and goes to the end of the string.
5. B `false^(true||false) = false^true = false^false = false`
6. A `Math.pow` raises the first argument to the power of the second argument and returns the result as a double.  $3^2 = 9.0$
7. D ASCII value of E is 69.  $69 + 32 = 101$ .
8. E `p=10.5` is executed regardless of the results of the if/else statement. Indentation does not affect execution of the code.
9. E After the 8 is printed within the loop, a gets 10. That terminates the loop and the 10 is printed by the last statement.
10. E `a[3]` stores 5. `a[5]` throws the out of bounds exception.
11. A The File class might throw an exception, for example, if the file referenced is not present. That exception must be handled either by catching it or by rethrowing it.
12. A `charAt` returns the character at the index number of the argument. So, concatenate d and c then o and a and finally g and t.
13. D Bitwise AND is done before bitwise OR.

	11	1011
AND	9	1001
		1001
OR	14	1110
	15	1111




14. C The largest value that can be stored in a variable of type `int` is 2147483647.
15. B `set` replaces the value at the value at the index value specified by the argument. It does not add any additional elements to the list.
16. E `this(f,1)` will call the constructor within this class with parameters that match the arguments in the call. `super(f,1)` would call a matching constructor in a parent class.
17. D The method `calculate` returns the results of a call to `Math.round` which returns a double value.
18. B `calculate` finds the average of all of the values in the ArrayList `grades`. The `toString` method returns the first and last name, the id number and a call to `calculate`.
19. C The two argument constructor leaves the id field unassigned. The default value for a String type variable is null.
20. A
 

T	e	x	a	s					
N	e	w		M	e	x	i	c	o
O	k		a		o		a		
21. C A TreeSet does not allow duplicates and stores elements in sorted order.
22. B `retainAll` returns the intersection of the two sets. Snake is the only element contained in both sets.
23. E `Integer.toString(x,y)` returns the value x in the radix (numbering system) y. In this case 8 in binary is 1000.
24. B Distributive property of AND over OR.
25. D `Math.random()` returns a random value x such that  $0.0 \leq x < 1.0$ . Therefore,  $r * 10$  will never be 10. Casting to an integer truncates so the largest value this expression will ever return is 29.
26. D When `x = 2` both case 2 and case 3 are executed because case 2 does not include a `break` statement. When `x = 4` both the case 4 statement and the default statement are executed for the same reason.
27. A A time complexity of  $O(n)$  means that search time is directly correlated to list size. Since 90,000 is three times larger than 30,000 then search time will be three times greater.  $0.75 * 3 = 2.25$ .
28. D  $23 + 24 + 6 + 0 = 53$
29. E Overloaded methods must have either a different number of parameters or different type parameters. The return type does not make them different.
30. B You are working your way back through the list from position i looking for the first place that x is less than `list[j]`. When you find that spot you insert x at j+1.
31. B
32. Nested loops usually (but not always) have a BigO value of  $O(n^2)$ .

33. D Answer choice A incorrectly initializes p to false. Answer choice B terminates the while loop too soon. Answer choice C incorrectly shows that 2, 4, 6 and 9 are prime.
34. B Elements are arranged in a binary search tree so that every element in the left branch of a particular node is less than that node and every element in the right branch are greater than that node. The first element to be entered into the tree is the root node and in this case it is the largest so there is no right branch for the root.
35. C In line #3 the variable z is out of scope. Since z is declared within the for loop the scope of z is restricted to the body of the loop.

36. B

A	B	!A OR B
T	T	T
T	F	F
F	T	T
F	F	T

37. E  is OR,  is NOT, and  is AND.

38. E 54 in binary is 00110110. Take the complement (flip it) to get 11001001. Add one to get 11001010.

39. coffee Elements are removed from a PriorityQueue in order. In this case in alphabetical order. Apples, bread, and chips are removed from the queue which leaves coffee at the head of the queue.

40.  $\frac{1}{2}(b_1+b_2)h = \frac{1}{2} * (b_1 + b_2) * h = \frac{1}{2} / b_1 b_2 + * h *$