

UIL COMPUTER SCIENCE WRITTEN TEST

2019 STATE

MAY 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 – STATE

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 values is NOT equal to 133₇?

- A) 73₁₀ B) 49₁₆ C) 112₈ D) 01001001₂ E) 243₅

Question 2.

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

- A) 12 B) -12 C) -22 D) 22 E) 7

```
out.print(-(5*-7/2+9-4));
```

Question 3.

What is the output of the code segment to the right? #'s indicate blank spaces.

- A) ####39.000
B) #####39
C) ###+39.000
D) 39.0000000
E) +39.000###

```
int x=12;
double y=3.25;
out.printf("%+ ,10.3f",x*y);
```

Question 4.

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

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

```
String s1="marchmadness";
out.print(s1.indexOf('a', 2));
```

Question 5.

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

- A) true B) false

```
out.println(true&&!false^true||false);
```

Question 6.

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

- A) 9.0 B) 6.0 C) 12.0 D) 12 E) 9

```
out.print(Math.floor(Math.PI)*Math.ceil(Math.E));
```

Question 7.

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

- A) 23 B) 74 C) -20 D) -28 E) 51

```
int x=14;
byte y=14;
short z=12;
out.print(x+y-z*y/3);
```

Question 8.

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

- A) 146
B) 23
C) 236
D) 56
E) 36

```
int a=5,b=-2,c=0,d=-10;
if(a-d>=b)
    if(b*a>c)
        out.print("1");
    else
    {
        out.print("2");
        if(b==d+8)
            out.print("3");
    }
else
    if(c+d>c)
        out.print("4");
    else
        out.print("5");
out.print("6");
```

<p>Question 9.</p> <p>What is the output of the code segment shown on the right?</p> <p>A. B. C. D. E.</p> <pre> A. B. C. D. E. ^^^^ ^^^^^ ^ ^ ^^^^^ ^^^ ^^^^^ ^^ ^^ ^^^^^ ^^ ^^^ ^^ ^^ ^^^ ^ ^^ ^^^^^ ^^^^^ ^^ ^ </pre>	<pre> for(int x=5;x>0;x--) { for(int y=x;y>=1;y--) out.print("^"); out.println(); } </pre>
<p>Question 10.</p> <p>What is the output of the code segment to the right?</p> <p>A) [6, 5, 1, 4, 11, 9, 8, 2, 3] B) [8, 12, 7, 4, 3, 5, 1, 2, 6] C) [8, 7, 4, 3, 15, 5, 1, 2, 6] D) [1, 6, 3, 4, 13, 5, 1, 8, 9] E) There is not output. Throws an ArrayIndexOutOfBoundsException.</p>	<pre> int i[]= {2,4,8,6,9,7,1,3,5}; i[i.length-i[1]]=i[3]+i[i.length-1]; int j[]= {8,9,7,4,3,5,1,2,6}; j[i[6]]=j[i[j[4]]]+i[j[5]]; out.print(Arrays.toString(j)); </pre>
<p>Question 11.</p> <p>Consider the class shown on the right and make no assumptions regarding import statements. Which of the following must replace <code> to allow the class to compile and execute correctly and allow the user to enter two values from the keyboard?</p> <p>A) System.out B) in C) Scanner.in D) Scanner.nextInt() E) System.in</p>	<pre> import java.util.Scanner; public class Q11 { public static void main(String[] args) { Scanner keyboard=new Scanner(<code>); System.out.print("Enter first number: "); int num1=keyboard.nextInt(); System.out.print("Enter second number: "); int num2=keyboard.nextInt(); System.out.print("Sum = "+(num1+num2)); } } </pre>
<p>Question 12.</p> <p>What is the output of the code segment shown on the right?</p> <p>A) 25.5 B) 25.0 C) 22.75 D) 26.0 E) 25.75</p>	<pre> double v=0.0; do { v=v+3.75; v=v-0.5; }while(v<25.5); out.print(v); </pre>
<p>Question 13.</p> <p>What is the output of the code segment to the right?</p> <p>A) 0 B) 1 C) 2 D) 13 E) 14</p>	<pre> int i=158; i%=5^4<<3%2; out.print(i); </pre>

Question 14.

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

- A) 9
- B) 18
- C) 17
- D) 10
- E) 36

```
out.println(Byte.BYTES+Byte.SIZE);
```

Question 15.

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

- A) [f, n, d, m, r, p]
- B) [m, r, f, n, d, p]
- C) [m, r, f, n, d]
- D) [f, m, r, p, n, d]
- E) [f, n, m, r, p, d]

```
ArrayList<String> az=new
ArrayList<String>();
az.add("m");az.add("r");az.add("p");
ArrayList<String> by=new
ArrayList<String>();
by.add("f");by.add("n");by.add("d");
by.addAll(2, az);
out.print(by);
```

Question 16.

Which of the following classes will compile correctly?

A.

```
public abstract class Abs {
    abstract int mtd() {};
}
```

B.

```
public abstract class Abs {
    abstract int mtd();
}
```

C.

```
public class Abs {
    abstract void mtd();
}
```

D.

```
abstract class Abs {
    protected void mtd();
}
```

E. More than one of the above.

Question 17.

What is the output of the code segment here?

```
String s="computer";
String t=s.length()<s.charAt(1)?s.substring(1, 4):s.substring(3, 7);
out.print(t);
```

- A) o
- B) 7
- C) 8
- D) omp
- E) pute

Question 18.

What is the output of the code segment shown here?

```
List<String> list=new LinkedList<String>();
list.add("A");list.add("R");
list.add("O");list.add("B");
list.add("W");list.add("N");
ListIterator<String> li=list.listIterator(list.indexOf("B"));
out.print(li.next());
li.next();
while(li.hasPrevious()) {
    out.print(li.previous());
}
```

- A) OBORA
- B) WBORA
- C) BBORA
- D) BWBORA
- E) WWBORA

Question 19.

Which line in the code segment shown here will print true?

```
final int flags=Pattern.CASE_INSENSITIVE|Pattern.LITERAL;
Pattern p=Pattern.compile("[abc]+", flags);
out.println(p.matcher("[aBc]+").matches());//line #1
out.println(p.matcher("abc").matches());//line #2
out.println(p.matcher("aBcAbC").matches());//line #3
```

- A) line #1
- B) line #2
- C) line #3
- D) All of the above.
- E) None of the above.

Question 20.

Given the code segment shown on the right, which of the following lines will NOT compile?

- A) Collections.sort(arraylist);
- B) Collections.sort(stack);
- C) Collections.sort(priorityqueue);
- D) Collections.sort(array);
- E) More than one of the above.

```
ArrayList<Integer> arraylist=new
ArrayList<Integer>();
Stack<Integer> stack=new Stack<Integer>();
Queue<Integer> priorityqueue=new
PriorityQueue<Integer>();
int[] array=new int[10];
```

//Use the class Data to answer questions 21 - 25.

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

    private final <code 1> int SIZE=15;
    private int[] list = new int[SIZE];

    public Data() {
        for(int i=SIZE-1;i>=0;i--)
            list[i]=SIZE-i;
    }

    interface DataIterator <code 2> Iterator<Integer>{}

    private class SpecialIterator <code 3> DataIterator{

        private int ni=0;

        public boolean hasNext() {
            return ni<=SIZE-1;
        }

        public Integer next() {
            Integer i=<code 4>;
            ni+=2;
            return i;
        }
    }

    public void print() {
        SpecialIterator si=new SpecialIterator();
        while(si.hasNext())
            out.print(si.next()+" ");
    }

    public static void main(String[] args) {
        Data d=new Data();
        d.print();    //CLIENT CODE
        out.println();
        out.print(Data.SIZE);
    }
}
```

Question 21.

Which of the following must replace **<code 1>** for the class `Data` to compile and execute correctly?

- A) `class`
- B) `abstract`
- C) `static`
- D) `void`
- E) no additional code is required

Question 22.

Which of the following must replace **<code 2>** and **<code 3>** to ensure that class `Data` will compile and execute? For each answer choice the first word is meant to replace **<code 2>** and the second is meant to replace **<code 3>**.

- A) `implements extends`
- B) `implements implements`
- C) `extends extends`
- D) `class extends`
- E) `extends implements`

Question 23.

Which of the following must replace **<code 4>** to ensure that class `Data` will compile and execute?

- A) `(Integer)list[ni]`
- B) `list[ni]`
- C) `ni`
- D) `Integer.valueOf(list[ni])`
- E) More than one of the above.

Question 24.

Which of the following represents the output of the line of code marked as **CLIENT CODE**?

- A) 15 13 11 9 7 5 3 1
- B) 14 12 10 8 6 4 2 0
- C) 1 3 5 7 9 11 13 15
- D) 2 4 6 8 10 12 14 16
- E) 14 12 10 8 6 4 2

Question 25.

`SpecialIterator` is _____.

- A) an inner class
- B) an anonymous class
- C) a local class
- D) an interface
- E) a nested static class

Question 26.

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

- A) U52V49
- B) U10V13
- C) U10V49
- D) UV1013
- E) There is no output due to an error.

```
out.print("U"+5*2+"V"+4+9);
```

Question 27.

Which of the following should replace **<code 1>** in the add method shown on the right?

- A) f
- B) y
- C) x
- D) e
- E) g

Question 28.

Which of the following should replace **<code 2>** in the add method shown on the right?

- A) e
- B) x.item
- C) y
- D) new Node<E>(e)
- E) e.item

Question 29.

Once all missing code has been correctly inserted into the class, what is the output of the client code shown here?

```
UIL<String> uil=new UIL<>("m");
uil.add("f");uil.add("r");
uil.add("b");uil.add("w");
uil.add("c");uil.print();
```

- A) b c f m r w
- B) m f b c r w
- C) c b f w r m
- D) w r m f c b
- E) m f r b w c

Question 30.

Once all missing code has been correctly inserted into the class, what data structure does class UIL implement?

- A) max heap
- B) min heap
- C) doubly linked list
- D) priority queue
- E) binary search tree

**//Use class UIL to answer questions
//27 - 30.**

```
public class UIL <E extends Comparable<E>>{

    private Node<E> m;
    private int size=0;

    public UIL(E e) {m=new Node<E>(e);}

    public void add(E e) {
        Node<E> x=null;
        Node<E> y=m;
        while(<b>code 1</b>!=null)
            if(e.compareTo(y.item)<0) {
                x=y;
                y=y.f;}
            else {
                x=y;
                y=y.g;}
        if(e.compareTo(x.item)<0)
            x.f=<b>code 2</b>;
        else
            x.g=<b>code 2</b>;
    }

    public void print() {
        print(m);}

    public void print(Node<E> m) {
        if(m==null) return;
        print(m.f);
        System.out.print(m.item+" ");
        print(m.g);}

    public class Node<E>{
        public E item;
        public Node<E> f;
        public Node<E> g;
        public Node(E e) {
            item=e;}
    }
}
```

//Use the class Merge_sort to answer questions 31, 32 and 33.

```
import java.util.Arrays;
public class Merge_sort {

    private static char[] aux;

    public static void sort(char[] a) {
        int n=a.length;
        aux=new char[n];
        for(int len=1;len<n;len*=2) {
            out.println(Arrays.toString(a));//line #1
            for(int lo=0;lo<n-len;lo+=len+len)
                merge(a,lo,lo+len-1,Math.min(lo+len+len-1, n-1));
        }
    }

    public static void merge(char[] a, int lo,int mid,int hi) {
        int i=lo,j=mid+1;
        for(int x=lo;x<=hi;x++)
            aux[x]=a[x];
        for(int k=lo;k<=hi;k++)
            if(i>mid)
                a[k]=aux[j++];
            else if(j>hi)
                a[k]=aux[i++];//line #2
            else if(aux[j]<aux[i])
                a[k]=aux[j++];
            else
                a[k]=aux[i++];
    }

    public static void main(String[] args) {
        char[] list= "zerblaoin".toCharArray();
        sort(list);
        out.print(Arrays.toString(list));
    }
}
```

Question 31.

What is the output of **line #1** when len = 2?

- A) [i, e, a, b, l, r, o, z, n]
- B) [a, e, r, b, l, z, o, i, n]
- C) [e, z, b, r, a, l, i, o, n]
- D) [b, e, r, z, a, i, l, o, n]
- E) [a, b, e, i, l, n, o, r, z]

Question 32.

Which of the following is true if **line #2** executes?

- A) All of the elements in aux have been merged into a.
- B) All of the elements in the left half of aux have been merged into a.
- C) All of the elements in a are in sorted order.
- D) All of the elements in aux are in sorted order.
- E) All of the elements in the right half of aux have been merged into a.

Question 33.

What is the worst case relative run time efficiency (complexity) of the sort method in the `Merge_sort` class shown on the previous page?

- A) $O(n)$ B) $O(n^2)$ C) $O(\log n)$ D) $O(n \log n)$ E) $O(n^3)$

Question 34.

Which of the following will always print a random lowercase letter?

- A) `out.print((char)(r.nextInt(26)+97));`
 B) `out.print((char)(r.nextInt(97)+26));`
 C) `out.print((char)(r.nextInt(26)));`
 D) `out.print((char)(r.nextInt(26)+65));`
 E) `out.print((char)(r.nextInt(27)+97));`

Question 35.

If x is a power of 2 then the method shown on the right returns the equivalent of ____.

- A) 2^x
 B) $\log_2 x$
 C) $\log_{10} x$
 D) x^2
 E) 0

```
public static int mtd(int x) {
    if(x>1) return mtd(x/2)+1;
    return 0;
}
```

Question 36.

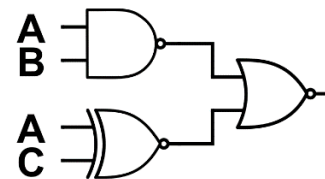
Which of the following lies closest to zero on a number line? All of the values are shown using signed 8-bit binary two's complement notation.

- A) 10100011
 B) 10001010
 C) 11011101
 D) 11001101
 E) 11010110

Question 37.

Which of the following combinations of values for A, B and C will make the Boolean expression diagrammed on the right evaluate to true?

- A) A is true, B is true and C is true
 B) A is false, B is true and C is true
 C) A is true, B is false and C is true
 D) A is true, B is true and C is false
 E) A is true, B is false and C is false



Question 38.

Which of the following is equivalent to the Boolean expression shown on the right?

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

$$\overline{A * (A + B) + (B + B * A)}$$

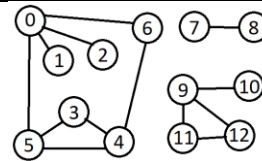
Question 39.

Write the value of the expression shown on the right in the blank provided on the answer document. The expression is written using reverse polish notation and integer division. The operands are 15, 8, 26, 5, 43, and 7.

$$15 \ 8 \ - \ 26 \ 5 \ + \ + \ 43 \ 7 \ / \ *$$

Question 40.

How many unique simple cycles does the graph shown on the right contain? (Two cycles are different if there is an edge used in one cycle but not the other.) Write your answer in the blank provided on the answer document.



★ ANSWER KEY – CONFIDENTIAL ★

UIL COMPUTER SCIENCE – 2019 STATE

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

- | | | | |
|------------------|------------------|------------------|-----------------------|
| 1) <u> C </u> | 11) <u> E </u> | 21) <u> C </u> | 31) <u> C </u> |
| 2) <u> A </u> | 12) <u> D </u> | 22) <u> E </u> | 32) <u> E </u> |
| 3) <u> E </u> | 13) <u> C </u> | 23) <u> E </u> | 33) <u> D </u> |
| 4) <u> D </u> | 14) <u> A </u> | 24) <u> A </u> | 34) <u> A </u> |
| 5) <u> B </u> | 15) <u> E </u> | 25) <u> A </u> | 35) <u> B </u> |
| 6) <u> A </u> | 16) <u> B </u> | 26) <u> C </u> | 36) <u> C </u> |
| 7) <u> D </u> | 17) <u> D </u> | 27) <u> B </u> | 37) <u> D </u> |
| 8) <u> C </u> | 18) <u> D </u> | 28) <u> D </u> | 38) <u> E </u> |
| 9) <u> B </u> | 19) <u> A </u> | 29) <u> A </u> | *39) <u> 228 </u> |
| 10) <u> B </u> | 20) <u> E </u> | 30) <u> E </u> | *40) <u> 4 </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	$133_7 = 73_{10}$ $122_8 = 74_{10}$ All other answer choices equal 73_{10} .
2.	A	$-(5 \cdot 7/2 + 9 - 4) =$ $-(-35/2 + 9 - 4) =$ $-(-17 + 9 - 4) =$ $-(-8 - 4) =$ $-(-12) =$ 12
3.	E	$3.25 \times 12 = 39.00$. The format specifier has 3 flags, - which left aligns, + which forces a plus sign for positive values and , which requires a comma separator (not used in this case). 10.3 specifies the use of 10 spaces and three decimal places.
4.	D	<code>indexOf('a', 2)</code> begins to look for the character 'a' at the index value 2. The first 'a' in the string is found at index number 1 so that value is not returned. The next occurrence of 'a' is at index value 6.
5.	B	<code>T && !F ^ T F =</code> <code>T && T ^ T F =</code> <code>T && F F =</code> <code>F F =</code> <code>F</code>
6.	A	<code>Math.PI = 3.141592653589793</code> <code>Math.E = 2.718281828459045</code> <code>Math.floor(3.141592653589793) = 3.0</code> <code>Math.ceil(2.718281828459045) = 3.0</code> $3.0 \times 3.0 = 9.0$
7.	D	$14 + 14 - 12 \cdot 14 / 3 =$ $14 + 14 - 168 / 3 =$ $14 + 14 - 56 =$ $28 - 56 =$ -28
8.	C	$5 - -10 >= -2$ is true. $-2 \cdot 5 > 0$ is false. Print the 2. $-2 == -10 + 8$ is true. Print the 3. Last line is not part of the if else statement so print the 6.
9.	B	Each iteration of the inner loop prints x number of ^. x is decremented by the outer loop from 5 down to 1.
10.	B	After second line i becomes: 2 4 8 6 9 11 1 3 5 After fourth line j becomes: 8 12 7 4 3 5 1 2 6
11.	E	The <code>in</code> object is a reference to the default input stream. <code>System</code> has not been imported so it must be included.
12.	D	Here is the value of v for each iteration of the loop: 3.25 6.5 9.75 13.0 16.25 19.5 22.75 26.0
13.	C	$5^4 \ll 3 \cdot 2 =$ $5^4 \ll 1 =$ $5^8 =$ 13 $158 \% 13 = 2$
14.	A	<code>Byte.BYTES = 1</code> and <code>Byte.SIZE = 8</code> . $8 + 1 = 9$.
15.	E	<code>az</code> → m r p <code>by</code> → f n d <code>by.addAll(2, az)</code> means insert all of <code>az</code> into <code>by</code> starting at index 2. <code>by.addAll(2, az)</code> → f n m r p d

16.	B	Answer choice A implements the method. Abstract methods are not implemented. Answer choice C is incorrect because any class containing an abstract method must also be declared as abstract. The method in answer choice D is not abstract therefore it must be implemented (have a body).
17.	D	<code>condition?true:false</code> In this example the length of the string is 8 which is less than the ASCII value of 'o', which is 111, so <code>print s.substring(1,4)</code> .
18.	D	The iterator is initially placed between the O and the B. <code>out.print(li.next())</code> prints the B and moves the iterator between the B and the W. <code>li.next()</code> moves the iterator between the W and the N. The loop then prints the letters in reverse order starting at W.
19.	A	The flag <code>CASE_INSENSITIVE</code> ignores case. The flag <code>LITERAL</code> ignores all metacharacters and escape sequences. Therefore, <code>[abc]</code> is not viewed as a character class in a regular expression. The actual string <code>"[abc]"</code> must be present for a match with the exception that the characters can be both upper and lower case.
20.	E	Answer choices A and B will compile. Answer choice C will not because <code>Queue</code> does not implement <code>Comparable</code> . Answer choice D will not because primitive data types are not objects which in turn do not implement <code>Comparable</code> .
21.	C	The last line in the class makes a static reference to <code>SIZE</code> .
22.	E	<code>DataIterator</code> is an interface and can not implement any methods so it must extend <code>Iterator</code> . Since <code>DataIterator</code> is an interface it must be implemented by <code>SpecialIterator</code> .
23.	E	Choices A, B and D are all valid.
24.	A	The call to the <code>Data</code> constructor fills the list array as follows: <code>[15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]</code> <code>SpecialIterator</code> skips every other element in the array.
25.	A	An inner class is any class that is declared within another class. It is an instance member of the enclosing class just like instance variables and instance methods. A local class is defined within a block. Typically local classes are declared within a method. They are not an instance member of the enclosing class. <code>SpecialIterator</code> is not declared within a method. Anonymous classes are declared and instantiate at the same time and do not have a name. <code>SpecialIterator</code> clearly has a name. <code>SpecialIterator</code> is a class not an interface. <code>SpecialIterator</code> is a nested class, however, it has not been declared as static so it is not a nested static class .
26.	C	<code>"U"+5*2+"V"+4+9=</code> <code>"U"+10+"V"+4+9=</code> <code>"U10V49"</code>
27.	B	Class <code>UIL</code> is a partial implementation of a binary search tree. <code>f</code> is the left child and <code>g</code> is the right child for each parent node. <code>m</code> is the root. Within the <code>add</code> method <code>x</code> is a parent node and <code>y</code> is the root of this subtree. The while loop moves down the tree searching for the proper location to add the element <code>e</code> . When the root becomes <code>null</code> the loop stops and the element is added as the appropriate child node.
28.	D	Add the new element as either the left or right child of the current parent.
29.	A	<code>print</code> is an in order traversal of the this tree: <pre> m / \ f r / \ b w \ c </pre>
30.	E	See #27

31.	C	<p>Merge_sort is a bottom up implementation of the merge sort algorithm. Here is a print out of each iteration of the outer loop in sort.</p> <pre> 1 [z, e, r, b, l, a, o, i, n] 2 [e, z, b, r, a, l, i, o, n] 4 [b, e, r, z, a, i, l, o, n] 8 [a, b, e, i, l, o, r, z, n] </pre>
32.	E	If j is greater than hi, then there are no more elements to compare in the right half of aux.
33.	D	Merge sort is $O(n \log n)$ in all cases.
34.	A	nextInt(n) returns a random integer between 0 inclusive and n exclusive. 97 is the ASCII value of 'a'.
35.	B	Example: mtd(1024) returns 10. $2^{10} = 1024$. $\log_2(1024)=10$.
36.	C	<pre> 10100011 = -93 10001010 = -118 11011101 = -35 11001101 = -51 11010110 = -42 </pre>
37.	D	<p>The expression diagrammed is $\neg(\neg(A \& B) \vee \neg(A \wedge C))$.</p> <p>$\neg(\neg(T \& T) \vee \neg(T \wedge F)) =$</p> <p>$\neg(\neg T \vee \neg T) =$</p> <p>$\neg(F \vee F) =$</p> <p>$\neg F =$</p> <p>T</p>
38.	E	$\overline{A * (A + B) + (B + B * A)}$ $\overline{A + B}$ Law of absorption $\overline{A * B}$ DeMorgan's law
39.	228	<pre> 15 8 - 26 5 + + 43 7 / * = 7 31 + 6 * = 38 6 * = 228 </pre> <p>The problem clearly states the integer division should be used. Do not accept an answer that is not a whole number.</p>
40.	4	<p>A simple cycle is a path whose first and last vertices are the same and that contains no repeated edges or vertices. In this case:</p> <pre> 0 5 3 4 6 0 0 5 4 6 0 3 4 5 3 9 11 12 9 </pre>