

UIL COMPUTER SCIENCE WRITTEN TEST

2023 STATE

MAY 2023

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)
```

STANDARD CLASSES AND INTERFACES – SUPPLEMENTAL REFERENCE

Package `java.util.function`

Interface `BiConsumer<T,U>`
void `accept(T t, U u)`

Interface `BiFunction<T,U,R>`
R `apply(T t, U u)`

Interface `BiPredicate<T,U>`
boolean `test(T t, U u)`

Interface `Consumer<T>`
void `accept(T t)`

Interface `Function<T,R>`
R `apply(T t)`

Interface `Predicate<T>`
boolean `test(T t)`

Interface `Supplier<T>`
T `get()`

UIL COMPUTER SCIENCE WRITTEN TEST – 2023 STATE

Note: Correct responses are based on **Java SE Development Kit 17 (JDK 17)** 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 17 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

If the values of the five numbers were converted to a common base and sorted, which one would be in the middle - larger than two numbers and smaller than the other two?

- A) 11011101_2 B) 265_8 C) 180_{10} D) $B9_{16}$ E) 3333_4

Question 2

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

- A) -3 B) 20 C) 21 D) 53 E) 9

```
out.print(23 + 37 / 7 % 2 - 24 % 5);
```

Question 3

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

- A) 68-13.60-c
B) 68-13.00-c
C) 68-13.60-D
D) 68-13.00-D
E) 54.40-D

```
int May = 68;  
double State = May / 5;  
String St = "%d-%.2f-%c";  
out.printf(St, May, State, May);
```

Question 4

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

- A) INAS B) STXA C) USAS D) TIXA E) INXA

```
String One = "TEXAS";  
String Two = "AUSTIN";  
int A = Two.indexOf("T");  
int B = One.indexOf("S");  
String Three = One.substring(A);  
String Four = Two.substring(B);  
out.print(Four+Three);
```

Question 5

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

- A) true
B) false

```
int A = 4;  
int B = 7;  
int C = 2;  
boolean D = (A>C) && (B<A);  
boolean E = (A + B == 10) || (A >= C*2);  
boolean F = D ^ E;  
out.print(F);
```

Question 6

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

- A) 11 B) 12 C) 13 D) 14 E) 15

```
int H = 5;  
int J = (int) (Math.pow(H,3));  
int L = (int) (Math.sqrt(J));  
out.print(L);
```

Question 7

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

- A) 50 B) 52 C) 54 D) 56 E) 58

```
int A = 10;  
int B = 25;  
int Z = A + B % A * A - B / A;  
out.print(Z);
```

<p>Question 8</p> <p>What is the output of the code segment to the right?</p> <p>A) 35 B) 45 C) 55 D) 65 E) 75</p>	<pre>int AA = 20; int BB = 15; int CC = 10; if (AA - BB > CC / 2) AA += CC; if (BB % CC == 0) AA += CC; else AA += BB; if (AA % CC == 5) AA += CC; else if (AA > 100) AA -= 100; out.print(AA);</pre>
<p>Question 9</p> <p>What is the output of the code segment to the right?</p> <p>A) 11121314151 B) 2345 C) 14 D) 1021324354 E) 111213141</p>	<pre>for(int x = 1; x <= 50; x += 10) out.print(x / 10 * 10 + x % 10);</pre>
<p>Question 10</p> <p>What is the output of the code segment to the right?</p> <p>A) 14 B) 18 C) 21 D) 23 E) 30</p>	<pre>int[] Miller = {5,1,2,9,2,6,7,4,1,7}; int F = Miller[Miller.length-1]; for(int x=0; x<Miller.length; x++) if (Miller[x]%2==0) F += Miller[x]; out.print(F);</pre>
<p>Question 11</p> <p>What is output by the code segment to the right?</p> <p>A) EAG B) ECG C) FBF D) FDJ E) BDF</p>	<pre>String St = "AB CD EF GH IJ"; St += St; Scanner Sc = new Scanner(St); for(int x = 1; x <= 3; x++) { Sc.next(); Sc.next(); out.print(Sc.next().substring(1)); }</pre>
<p>Question 12</p> <p>What is the output of the code segment to the right?</p> <p>A) 201 B) 209 C) 219 D) 221 E) 229</p>	<pre>int G = 0; for (int x=10; x<=50; x++) G += (int)(Math.sqrt(x)+0.5); out.print(G);</pre>

<p>Question 13</p> <p>What is the output of the code segment to the right?</p> <p>A) 7 B) 12 C) 15 D) 21 E) 23</p>	<pre>int R = 20; int Y = R >> 2 ^ 2 + R & 11; out.print(Y);</pre>
<p>Question 14</p> <p>What is the output of the code segment shown on the right?</p> <p>A) 2 B) 4 C) 6 D) 8 E) 12</p>	<pre>int K = Integer.SIZE; int L = Double.SIZE; int M = Byte.SIZE; out.println(K / M + L / K);</pre>
<p>Question 15</p> <p>What is output by the code segment to the right?</p> <p>A) [10, 20, 30, 40, 50] B) [50, 40, 30, 20, 10] C) [10, 10, 20, 20, 30] D) [50, 50, 40, 40, 30] E) [1, 0, 2, 0, 3]</p>	<pre>ArrayList<Integer> soup; soup = new ArrayList<Integer>(); for(int x = 1; x<=5; x++) { soup.add(x); soup.add(0,x*10); soup.remove(soup.size()-1); } out.print(soup);</pre>
<p>Question 16</p> <p>What is the output of the code segment shown on the right?</p> <p>A) 100 B) 121 C) 144 D) 169 E) 625</p>	<pre>int N = 25; int z=0; for(int x=1; x<=N; x+=2) z +=x; out.print(z);</pre>
<p>Question 17</p> <p>What is the output of the code segment shown on the right?</p> <p>A) 75 B) 80 C) 84 D) 100 E) 105</p>	<pre>int N=0; String St = ""; for(int x=0; x<=20; x++) { St = Integer.toBinaryString(x); N += St.length(); } out.print(N);</pre>
<p>Question 18</p> <p>What is the output of the code segment shown on the right?</p> <p>A) 498 B) 992 C) 1000 D) 1024 E) 2000</p>	<pre>int N = 1000; for (int x=1; x<=5; x++) N = N>>1; for (int x=1; x<=5; x++) N = N<<1; out.print(N);</pre>

Question 19

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

- A) AB BC CD DE EF FG GH
- B) AB BC GH FG EF DE CD
- C) GH FG AB BC EF DE CD
- D) GH AB BC CD DE EF FG
- E) GH FG AB BC CD DE EF

```
String[]VV = {"AB","BC","CD","DE","EF","FG","GH"};
for(int x=VV.length-1;x>=2;x--)
{
    String Closet = VV[x];
    VV[x] = VV[x-2];
    VV[x-2] = Closet;
}
for(int x=0; x<VV.length; x++)
    out.print(VV[x]+ " ");
```

Question 20

In the code segment to the right, what is the output of line 1?

- A) 50 B) 60 C) 70 D) 80 E) 90

Question 21

In the code segment to the right, what is the output of line 2?

- A) 50 B) 60 C) 61 D) 62 E) 70

Question 22

In the code segment to the right, what is the output of line 3?

- A) 50 B) 61 C) 69 D) 70 E) 71

```
public class Rockford
{
    private int A;
    private int B;
    private int C;

    public Rockford(int D)
    {
        A = D;
        B = D++;
        C = --D;
    }

    public int getOne()
    {
        return A * 5;
    }

    public int getTwo()
    {
        return getOne() + B;
    }

    public int getThree()
    {
        return getTwo() + C;
    }
}
////////////////////////////////////
// Client code
Rockford Jim = new Rockford(10);
System.out.println(Jim.getOne()); //Line 1
System.out.println(Jim.getTwo()); //Line 2
System.out.println(Jim.getThree()); //Line 3
```

Question 23

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

- A) 4 B) 6 C) 8 D) 10 E) 12

```
String St = "BEAR OWL DOG CAT LION ";
St += "ZEBRA RAT PIG TIGER GORILLA";
int N = 0;
Scanner Sue = new Scanner(St);
while(Sue.hasNext())
{
    Sue.next();
    String A = Sue.next();
    if(A.matches("..O. ")) N++;
    if(A.matches("[A-C].*")) N++;
    if(A.matches("..G")) N++;
    if(A.matches(".*R.*")) N++;
}
out.print(N);
```

Question 24

In the code segment to the right, what is the output of line 1?

- A) 2 B) 3 C) 6 D) 10 E) 14

Question 25

In the code segment to the right, what is the output of line 2?

- A) 2 B) 3 C) 6 D) 10 E) 14

```
HashSet<Integer>HS;
HS =new HashSet<Integer>();
int []Nums = {8,0,6,3,5,2,4,3,6,3};
int []More = {5,1,2,6,5,3,1,4,6,4};
for(int x=0; x<Nums.length; x++)
    HS.add(Nums[x]);
for(int x=0; x<More.length; x++)
    HS.add(More[x]);
TreeMap<String,Integer>Tree;
Tree = new TreeMap<String,Integer>();
Tree.put("BIG",0);
Tree.put("SMALL",0);
Tree.put("STRANGE",0);
for (int Bob: HS)
{
    int A = Tree.get("BIG");
    int B = Tree.get("SMALL");
    int C = Tree.get("STRANGE");
    if (Bob>5)
        Tree.put("BIG",A+1);
    else
        Tree.put("SMALL",B+1);
    if (Bob%2==1)
        Tree.put("STRANGE",C+1);
}
out.println(Tree.get("BIG")); //Line 1
out.println(Tree.get("SMALL")); //Line 2
out.println(Tree.get("STRANGE")); //Line 3
```

Question 26

In the code segment to the right, what is the output of line 3?

- A) 2 B) 3 C) 6 D) 10 E) 14

Question 27

Find the value of Go(33).

- A) 3 B) 6 C) 33 D) 36 E) 39

Question 28

Find the value of Go(20).

- A) 14 B) 18 C) 20 D) 22 E) 24

Question 29

Find the value of Go(1).

- A) 11 B) 55 C) 76 D) 114 E) 132

```
public static int Go(int N)
{
    int A = N / 10;
    int B = N % 10;
    if (A==B)
        return N;
    if (A > B)
        return A + Go(N+2);
    return B + Go(N+3);
}
```

Question 30

List the operators to the right in order from highest precedence to lowest precedence ?

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

- I %
 II ^
 III >>

Question 31

In the code to the right, what is output by line #1?

- A) 22
 B) 24
 C) 25
 D) 27
 E) 28

```
public static int Find(int nums[], int L, int R,
int T)
{
    int middle = (L + R)/2;
    int C = 0;
    while(L <= R)
    {
        C++;
        if (nums[middle] < T )
            L = middle + 1;
        else if(nums[middle] > T)
            R = middle -1;
        else
            return middle + C;

        middle = (L + R)/2;
    }
    return -1;
}
```

Question 32

In the code to the right, what is output by line #2?

- A) 20
 B) 21
 C) 22
 D) 23
 E) 24

```
////////////////////////////////////
/// Client Code

int []nums =
{1,2,2,2,2,4,6,7,7,7,9,10,11,11,11,23,34,45,45,45,56};

out.print(Find(nums,0,20,56)); // Line 1
out.print(Find(nums,0,20,45)); // Line 2
out.print(Find(nums,0,20,2)); // Line 3
```

Question 33

In the code to the right, what is output by line #3?

- A) 2
 B) 3
 C) 4
 D) 5
 E) 6

Question 34

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

- A) 0
- B) 7
- C) 14
- D) 21
- E) 28

```
public class One
{
    private int A;
    public int B;
    public One(int Z)
    {
        A = Z;
        B = Z & (Z - 1);
    }
    public int getA()
    {
        return A ^ (A - 1);
    }

    public int getB()
    {
        return B & (B - 1);
    }
}
////////////////////////////////////
// Client Code
One Uno = new One(7);
One Dos = new One(12);
out.print(Uno.getB()*Dos.getA());
```

Question 35

Assume that class One from problem #34 exists. Now assume that class Two to the right extends class one . Which five lines of the client code to the right will cause a compiler error?

- A) 4 11 14 15 16
- B) 4 11 12 14 16
- C) 4 12 14 15 16
- D) 6 7 11 12 13
- E) 6 7 10 13 16

```
public class Two extends One
{
    private int A;
    private int E;
    public Two(int H, int G)
    {
        super(H*G);
        A = H;
        E = G;
    }
    public int getB()
    {
        return B;
    }
    public int getC()
    {
        return E;
    }
}
////////////////////////////////////
Possible Client Code
One Alpha = new One(12); // Line 1
One Beta = new Two(3,4); // Line 2
Two Gamma = new Two(12,16); // Line 3
Two Delta = new One(20); // Line 4
out.println(Alpha.getA()); // Line 5
out.println(Beta.getA()); // Line 6
out.println(Gamma.getA()); // Line 7
out.println(Alpha.getB()); // Line 8
out.println(Beta.getB()); // Line 9
out.println(Gamma.getB()); // Line 10
out.println(Alpha.getC()); // Line 11
out.println(Beta.getC()); // Line 12
out.println(Gamma.getC()); // Line 13
out.println(Gamma.A); // Line 14
out.println(Gamma.B); // Line 15
out.println(Gamma.E); // Line 16
```

Question 36

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

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

```
String A = new String("Apple");
String B = new String("Banana");
String C = new String("Banana");
String D = new String("Apple");
String E = A;
A = B;
int N = 0;
if(A==B) N++;
if(B==C) N++;
if(C==D) N++;
if(E==A) N++;
if (A.equals(C)) N++;
if (B.equals(D)) N++;
if (C.equals(E)) N++;
if (D.equals(B)) N++;
out.print(N);
```

Question 37

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

- A) 5 B) 10 C) 15 D) 20 E) 25

```
int[][] Box = new int[5][5];
for(int x=1; x<Box.length; x++)
    for(int y=1; y<Box[0].length; y++)
        Box[x][y]=Box[x-1][y-1]+x*y;
out.print(Box[4][3]);
```

Question 38

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

- A) 24 B) 25 C) 26 D) 27 E) 28

```
int N = 16;
for(int x=28; x>=21; x--)
    N ^= x;
out.print(N);
```

Question 39

Evaluate the prefix expression to the right. Write the value in the blank reserved for #39.

+ / * 3 4 - 17 * 5 3 / + 6 6 3

Question 40

In the code to the right, we are conducting a Boolean Algebra test. What number will be output by the code? Write the value in the blank reserved for #40.

```
int N=0;
for(int A = 0; A<=1; A++)
for(int B = 0; B<=1; B++)
for(int C = 0; C<=1; C++)
for(int D = 0; D<=1; D++)
{
    boolean AA = (A==1);
    boolean BB = (B==1);
    boolean CC = (C==1);
    boolean DD = (D==1);
    boolean One = (AA&&BB) || (CC&&DD);
    boolean Two = (AA || BB) && (CC || DD);
    if(One==Two)
        N++;
}
out.println(N);
```

★ ANSWER KEY – CONFIDENTIAL ★

UIL COMPUTER SCIENCE – 2023 STATE

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

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

KEY

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

Note: Correct responses are based on **Java SE Development Kit 17 (JDK 17)** 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 17 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used.

Explanations:

1.	D	<p>Convert all 5 to a common base... I suggest base 8.</p> <p>A) $11\ 011\ 101_2 = 335_8$ B) $265_8 = 265_8$ C) $180_{10} = 264_8$ D) $B9_{16} = 271_8$ E) $3333_4 = 377_8$</p>
2.	B	<p>$23 + 37 / 7 \% 2 - 24 \% 5$ $23 + 5 \% 2 - 24 \% 5$ $23 + 1 - 24 \% 5$ $23 + 1 - 4$ $24 - 4$ 20</p>
3.	D	<pre>int May = 68; May = 68 double State = May / 5; State = 13.00000 String St = "%d-%.2f-%c"; Creates the format integer-2-decimal double-character out.printf(St, May, State, May); 68-13.00-D</pre>
4.	A	<pre>String One = "TEXAS"; String Two = "AUSTIN"; int A = Two.indexOf("T"); A=3 int B = One.indexOf("S"); B=4 String Three = One.substring(A); Three = "AS" String Four = Two.substring(B); Four = "IN" out.print(Four+Three); "INAS"</pre>
5.	A	<pre>int A = 4; int B = 7; int C = 2; boolean D = (A>C) && (B<A); D = false boolean E = (A + B == 10) (A >= C*2); E = true boolean F = D ^ E; XOR F = true out.print(F);</pre>
6.	A	<pre>int H = 5; int J = (int)(Math.pow(H,3)); J = 125 int L = (int)(Math.sqrt(J)); L = (int)(11.18) = 11 out.print(L);</pre>
7.	E	<pre>int A = 10; int B = 25; int Z = A + B % A * A - B / A; 10 + 25 % 10 * 10 - 25 / 10 10 + 5 * 10 - 25 / 10 10 + 50 - 25 / 10 10 + 50 - 2 60 - 2 = 58</pre>

8.	B	<pre>int AA = 20; int BB = 15; int CC = 10; if (AA - BB > CC / 2) false!!!! AA += CC; if (BB % CC == 0) false!!! do the else AA += CC; else AA += BB; AA is now 35 if (AA % CC == 5) true!!! AA += CC; AA is now 45 else if (AA > 100) AA -= 100; out.print(AA);</pre>
9.	E	<pre>for(int x = 1; x <= 50; x += 10) out.print(x / 10 * 10 + x % 10); Iteration 1: x is 1 1/10*10 + 1%10 = 0+1=1 Iteration 2: x is 11 11/10*10 + 11%10 = 10+1=11 Iteration 3: x is 21 21/10*10 + 11%10 = 20+1=21 Iteration 4: x is 31 31/10*10 + 11%10 = 30+1=31 Iteration 5: x is 41 41/10*10 + 11%10 = 40+1=41</pre>
10.	C	<pre>int[] Miller = {5,1,2,9,2,6,7,4,1,7}; int F = Miller[Miller.length-1]; for(int x=0; x<Miller.length; x++) if (Miller[x]%2==0) F += Miller[x]; out.print(F); F = 7 before the loop begins Then, we add all the even values to F F = 21</pre>
11.	D	<pre>String St = "AB CD EF GH IJ"; St += St; Scanner Sc = new Scanner(St); for(int x = 1; x <= 3; x++) { Sc.next(); Sc.next(); out.print(Sc.next().substring(1)); }</pre> <p>St = "AB CD EF GH IJAB CD EF GH IJ"; Notice no space between IJ and AB</p> <p>The loop skips 2 strings then processes the 3rd, three times.</p> <p>EF CD IJ Printing letter #1 in each gives us FDJ</p>

12.	C	<pre>int G = 0; for (int x=10; x<=50; x++) G += (int)(Math.sqrt(x)+0.5); out.print(G);</pre> <p>41 iterations 10-12 Add 3 3*3 = 9 13-20 Add 4 8*4=32 21-30 Add 5 10*5=50 31-42 Add 6 12*6=72 43-50 Add 7 8*7=56 G = 219</p>
13.	A	<pre>int R = 20; int Y = R >> 2 ^ 2 + R & 11; out.print(Y);</pre> <p>Order of precedence: + >> & ^ 20 >> 2 ^ 2 + 20 & 11 20 >> 2 ^ 22 & 11 5 ^ 22 & 11 5 ^ 2 101 ^ 010 = 111 = 7</p>
14.	C	<pre>int K = Integer.SIZE; int L = Double.SIZE; int M = Byte.SIZE; out.println(K / M + L / K);</pre> <p>32 64 8 32/8 + 64/32 = 4+2 = 6</p>
15.	B	<pre>ArrayList<Integer> soup; soup = new ArrayList<Integer>(); for(int x = 1; x<=5; x++) { soup.add(x); soup.add(0,x*10); soup.remove(soup.size()-1); } out.print(soup);</pre> <p>Iteration 1: [] then [1] then [10, 1] then [10] Iteration 2: [10] then [10,2] then [20,10, 2] then [20,10] Iteration 3: [20,10] then [20,10,3] then [30,20,10, 3] then [30,20,10] Iteration 4: [30,20,10] then [30,20,10,4] then [40,30,20,10, 4] then [40,30,20,10] Iteration 5: [40,30,20,10] then [40,30,20,10,5] then [50,40,30,20,10, 5] then [50,40,30,20,10]</p>
16.	D	<pre>int N = 25; int z=0; for(int x=1; x<=N; x+=2) z +=x; out.print(z);</pre> <p>Add the first N odds and you get N² This adds the first 13 odds and gets 169</p>
17	A	<p>This adds the lengths of the binary strings from 0 to 20.</p> <p>0,1 = 2 10,11 = 4 100,101,110,111 = 12 1000,1001,1010,1011,1100,1101, 1110, 1111 = 32 10000,10001,10010,10011,10100 = 25 The sum of the lengths is 75</p>

18.	B	This does $1000 \gg 5$ followed by a $\ll 5$ 1000,500,250,125,62, 31 31, 62, 124, 248, 496, 992
19.	E	"AB","BC","CD","DE","EF","FG","GH" Starting at the last element, this routine swaps the element x with element x-2. This continues until =2. x=6 "AB","BC","CD","DE","GH","FG","EF" x=5 "AB","BC","CD","FG","GH","DE","EF" x=4 "AB","BC","GH","FG","CD","DE","EF" x=3 "AB","FG","GH","BC","CD","DE","EF" x=2 "GH","FG","AB","BC","CD","DE","EF"
20.	A	Rockford Jim = new Rockford(10); When the constructor is called, A, B, and C are all initialized to 10. getOne() returns a 50
21.	B	getTwo() returns getOne() + 10 = 60
22.	D	getThree() returns getTwo() + 10 = 70
23.	A	"BEAR OWL DOG CAT LION ZEBRA RAT PIG TIGER GORILLA" The loop tests these values for A: "OWL CAT ZEBRA PIG GORILLA" if(A.matches("..O. ")) N++; no matches if(A.matches("[A-C].*")) N++; CAT if(A.matches("..G")) N++; PIG if(A.matches(".*R.*")) N++; ZEBRA GORILLA 4 matches
24.	A	The HashSet HS takes on these values {0,1,2,3,4,5,6,8} in some order. Sets cannot have duplicate values. A TreeMap is set up using the Strings BIG, SMALL, and STRANGE. BIG numbers: Those in HS that are greater than 5 2 is the answer here.
25.	C	SMALL numbers: Those in HS that are not greater than 5 6 is the answer here.
26.	B	STRANGE numbers: Those in HS that are odd 3 is the answer here.
27.	C	go(33) = 33
28.	E	go(20) = 2 + go(22) = 24 go(22) = 22
29.	D	go(1) = 1 + go(4) = 114 go(4) = 4 + go(7) = 113 go(7) = 7 + go(10) = 109 go(10) = 1 + go(12) = 102 go(12) = 2 + go(15) = 101 go(15) = 5 + go(18) = 99 go(18) = 8 + go(21) = 94 go(21) = 2 + go(23) = 86 go(23) = 3 + go(26) = 84 go(26) = 6 + go(29) = 81 go(29) = 9 + go(32) = 75 go(32) = 3 + go(34) = 66 go(34) = 4 + go(37) = 63 go(37) = 7 + go(40) = 59 go(40) = 4 + go(42) = 52 go(42) = 4 + go(44) = 48 go(44) = 44

30.	D	% >> ^
31.	C	<p>{1,2,2,2,2,4,6,7,7,7,9,10,11,11,11,23,34,45,45,45,56}</p> <p>This is the code for a binary search, which will be performed on the sorted list.</p> <p>When a target is found, it returns the index of the "found" target + how many "visits" it took to find it.</p> <p>To find the 56, we visit the 10th item, the 9. (Visit 1)</p> <p>Then we visit the 15th item, the 23 (Visit 2)</p> <p>Then we visit the 18th item, the 45. (Visit 3)</p> <p>Then we visit the 19th item, another 45. (Visit 4)</p> <p>Then we find the 56 at position 20. (Visit 5)</p> <p>20 + 5 = 25</p>
32.	B	<p>To find the 45, we visit the 10th item, the 9. (Visit 1)</p> <p>Then we visit the 15th item, the 23 (Visit 2)</p> <p>Then we visit the 18th item, the 45. (Visit 3)</p> <p>18 + 3 = 21</p>
33.	E	<p>To find the 45, we visit the 10th item, the 9. (Visit 1)</p> <p>Then we visit the 4th item, the 2.</p> <p>Then we visit the 1st item, another 2.</p> <p>Then we find the target at position 0.</p>
34.	E	<pre> One Uno = new One(7); A = 7 B = 7 ^ 6 (111^110) = 1 Uno.getB() = 6 & 5 (110 & 101) = 4 One Dos = new One(12); A = 12 B = 12^11 (1100 ^ 1011) = 7 Dos.getA() = 12 ^ 11 (1100^1011) = 7 out.print(Uno.getB()*Dos.getA()); 4 * 7 = 28 </pre>
35.	B	<pre> Two Delta = new One(20); // Line 4 Two is-a One, not vice versa. Thus, this declaration is invalid. out.println(Alpha.getC()); // Line 11 class One does not have a getC() method out.println(Beta.getC()); // Line 12 This error occurs at compilation time, class One does not contain a getC() method out.println(Gamma.A); // Line 14 A is a private instance variable which cannot be accessed in this manor. out.println(Gamma.E); // Line 16 E is a private instance variable which cannot be accessed in this manor. </pre>
36.	B	<pre> if(A==B) N++; true because A and B now represent the same object. if(B==C) N++; false if(C==D) N++; false if(E==A) N++; false if (A.equals(C)) N++; true because A now contains the String "Banana" if (B.equals(D)) N++; false if (C.equals(E)) N++; false if (D.equals(B)) N++; false </pre>

37.	D	<p>Starting with position 1,1 and moving Left to Right down the matrix, each cell will be the cell up and to the left of the place plus the product of the row and column.</p> <pre> 0 0 0 0 0 0 1 2 3 4 0 2 5 8 11 0 3 8 14 20 0 4 11 20 30 </pre> <p>Box[4][3] = 20 (and so does Box[3][4]) See the pattern emerging?</p>																																																																																					
38.	A	<pre> 16 ^ 28 = 10000 ^ 11100 = 01100 12 ^ 27 = 01100 ^ 11011 = 10111 23 ^ 26 = 10111 ^ 11010 = 01101 13 ^ 25 = 01101 ^ 11001 = 10100 20 ^ 24 = 10100 ^ 11000 = 01100 12 ^ 23 = 01100 ^ 10111 = 11011 27 ^ 22 = 11011 ^ 10110 = 01101 16 ^ 21 = 01101 ^ 10101 = 11000 = 24 </pre>																																																																																					
39.	10	<pre> + / * 3 4 - 17 * 5 3 / + 6 6 3 + / * 3 4 - 17 * 5 3 / 12 3 + / * 3 4 - 17 * 5 3 4 + / * 3 4 - 17 15 4 + / * 3 4 2 4 + / 12 2 4 + 6 4 = 10 </pre>																																																																																					
40.	10	<p>A truth table would be apropos</p> <table border="1"> <thead> <tr> <th>ABCD</th> <th>AB + CD</th> <th>(A + B) * (C + D)</th> <th></th> <th></th> </tr> </thead> <tbody> <tr><td>0000</td><td>0</td><td>0</td><td>same</td><td>1</td></tr> <tr><td>0001</td><td>0</td><td>0</td><td>same</td><td>2</td></tr> <tr><td>0010</td><td>0</td><td>0</td><td>same</td><td>3</td></tr> <tr><td>0011</td><td>1</td><td>0</td><td></td><td></td></tr> <tr><td>0100</td><td>0</td><td>0</td><td>same</td><td>4</td></tr> <tr><td>0101</td><td>0</td><td>1</td><td></td><td></td></tr> <tr><td>0110</td><td>0</td><td>1</td><td></td><td></td></tr> <tr><td>0111</td><td>1</td><td>1</td><td>same</td><td>5</td></tr> <tr><td>1000</td><td>0</td><td>0</td><td>same</td><td>6</td></tr> <tr><td>1001</td><td>0</td><td>1</td><td></td><td></td></tr> <tr><td>1010</td><td>0</td><td>1</td><td></td><td></td></tr> <tr><td>1011</td><td>1</td><td>1</td><td>same</td><td>7</td></tr> <tr><td>1100</td><td>1</td><td>0</td><td></td><td></td></tr> <tr><td>1101</td><td>1</td><td>1</td><td>same</td><td>8</td></tr> <tr><td>1110</td><td>1</td><td>1</td><td>same</td><td>9</td></tr> <tr><td>1111</td><td>1</td><td>1</td><td>same</td><td>10</td></tr> </tbody> </table>	ABCD	AB + CD	(A + B) * (C + D)			0000	0	0	same	1	0001	0	0	same	2	0010	0	0	same	3	0011	1	0			0100	0	0	same	4	0101	0	1			0110	0	1			0111	1	1	same	5	1000	0	0	same	6	1001	0	1			1010	0	1			1011	1	1	same	7	1100	1	0			1101	1	1	same	8	1110	1	1	same	9	1111	1	1	same	10
ABCD	AB + CD	(A + B) * (C + D)																																																																																					
0000	0	0	same	1																																																																																			
0001	0	0	same	2																																																																																			
0010	0	0	same	3																																																																																			
0011	1	0																																																																																					
0100	0	0	same	4																																																																																			
0101	0	1																																																																																					
0110	0	1																																																																																					
0111	1	1	same	5																																																																																			
1000	0	0	same	6																																																																																			
1001	0	1																																																																																					
1010	0	1																																																																																					
1011	1	1	same	7																																																																																			
1100	1	0																																																																																					
1101	1	1	same	8																																																																																			
1110	1	1	same	9																																																																																			
1111	1	1	same	10																																																																																			