UIL COMPUTER SCIENCE WRITTEN TEST

2018 INVITATIONAL A

JANUARY/FEBRUARY 2018

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 – 2018 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.

Which of the following decimal numbers is equivalent to 11111111_2?

A) 64 **B)** 128 **C)** 127 **D)** 255 **E)** 256

Question 2.

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

A) 238 **B)** 76.2 **C)** 77 **D)** 76 **E)** -19

```
out.println(6*4*3+7-14/5);
```

Question 3.

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

A)
```
Fourth Of July
```
B)
```
Fourth
Of\nJuly
```
C)
```
Fourth
OfJuly
```
D)
```
FourthOf
July
```
E)
```
Fourth
Of
July
```

```
out.println("Fourth");
out.print("Of\nJuly");
```

Question 4.

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

A) `##Hall  oween**`

B) `Hall  oween`

C) `##Halloween**`

D) `Halloween`

E) Error. Cannot call a method using a String literal.

```
out.print("##Hall  oween**".trim());
```

Question 5.

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

A) `true` **B)** `false`

```
out.print(!true^!false);
```

Question 6.

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

A) 5.64 **B)** 5.0 **C)** 5 **D)** 4.99 **E)** 6.0

```
double a=3.65,b=1.99;
double c=Math.round(a)+Math.floor(b);
out.print(c);
```

Question 7.

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

A) 5 **B)** 5.0 **C)** 9.75 **D)** -4.75 **E)** 5.25

```
int i=5;
double d=4.75;
double a=i-d+i;
out.print(a);
```

Question 8.

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

A) `X Y`

B) `X`

C) `Y Z`

D) `X Y Z`

E) `Y`

```
double d1=1.8,d2=1.81,d3=1.8;
if(d1>=d2)
  if(d1==d3)
     out.print("X ");
if(d1==d3)
  if(d2>d3)
     out.print("Y ");
out.print("Z");
```

Question 9.

How many hash tags are printed by the code shown to the right?

A) 0 **B)** 1 **C)** 4 **D)** 5 **E)** 6

```
int x=0;
do {
     out.print("# ");
}while(x++<5);
```

Question 10.

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

A) `[One, Two, Three, Four, null]`

B) `[null, One, Two, Three, Four, null]`

C) `[One, Two, Three, Four]`

D) `[null, One, Two, Three, Four]`

E) `[ , One, Two, Three, Four]`

```
String s[]=new String[5];
s[1]="One";
s[2]="Two";
s[3]="Three";
s[4]="Four";
out.print(Arrays.toString(s));
```

Question 11.

The file `datafile.dat` contains five words each listed on a separate line. Which of the following can correctly replace **<code>** in the code segment shown on the right so that each word in the file will be printed on the same line separated by a space? Assume that all required classes have been imported correctly.

A) `s.next()`

B) `s.hasNext()`

C) `f.hasNext()`

D) `s.hasNextInt()`

E) `f.next()`

```
File f=new File("datafile.dat");
Scanner s=new Scanner(f);
while(<code>)
     out.print(s.next()+" ");
```

Question 12.

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

A) `5 0`

B) `4 0`

C) `5 24`

D) `4 24`

E) `6 24`

```
int t=1,x;
for(x=0;x<5;x++)
     t*=x;
out.print(x+" "+t);
```

Question 13.

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

A) -9

B) -19

C) -18

D) -22

E) There is no output due to a type mismatch error.

```
short s=5;
int i=-8;
double d=3.5;
int b=++s+i*(int)d;
out.print(b);
```

Question 14.

Which of the following values can be stored in a variable that is of type byte?

A) 64 **B)** 128 **C)** 256 **D)** 32767 **E)** All of these values can be stored in a byte type variable.

Question 15.

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

A) [pig, kid, calf]

B) [lamb, kid, pig, calf]

C) [calf, pig, lamb, kid]

D) [pig, lamb, kid, calf]

E) Error. Throws an IndexOutOfBoundsException.

```
ArrayList<String> a=new
ArrayList<String>();
a.add(0, "lamb");
a.add(1, "kid");
a.add(0, "pig");
a.add("calf");
out.print(a);
```

Question 16.

What is the output of the code segment shown here?

```
String s="abcde";
for(int i=0;i<s.length();i++)
    out.print(s.substring(i, i+1).matches("[^aeiou]")+" ");
}
```

A) true false false false true

B) false true true true false

C) false false false false false

D) true false false false false

E) false true true true true

Question 17.

What is printed by the client code shown here given the implementation of the method abc shown to the right?

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

A) 16

B) 30

C) 580

D) 21

E) 31

```
public static int abc(int i) {
if(i>5)
    return 1;
else
    return 2*i+abc(++i);
}
```

Question 18.

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

A) `[earth, water, wind]`

B) `false [wind, earth, water]`

C) `false [earth, water, wind]`

D) `true [earth, water, wind]`

E) `false [earth, water, wind, wind]`

```
Set<String> s=new TreeSet<String>();
s.add("wind");
s.add("earth");
s.add("fire");
s.add("water");
s.add("wind");
s.remove("fire");
out.print(s.remove("fire")+" "+s);
```

Question 19.

Which of the following is the largest possible value that the code segment on the right might print?

A) 20

B) 40

C) 38

D) 39

E) 19

```
Random r=new Random();
System.out.print((r.nextInt(10)+10)*2);
```

Question 20.

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

A) `[5, 7, null, 9, 8]`

B) `[5, 7, 0, 9, 8]`

C) `[5, 7, 9, 8]`

D) Prints the hexadecimal value of the memory location for each of the objects stored in the array.

E) There is no output due to an error.

```
Object[] o=new Object[5];
o[0]=5;
o[1]=new Integer(7);
o[3]=9;
o[4]="8";
out.print(Arrays.toString(o));
```

Question 21.

Variables a, b and c are called _____?

A) instance variables

B) fields

C) class variables

D) both A and B.

E) A, B, and C

Question 22.

Which of the following reserved words must replace **<code>** in the `setC` method?

A) `super`

B) `this`

C) `null`

D) `instanceof`

E) `static`

Question 23.

What is the output of the client code shown here if **<code>** has been filled in correctly?

```
Uil a=new Uil(3,4,5);
a.a=6;
a.b=7;
a.c=8;
out.println(a.a+" "+a.b+" "+a.getC());
```

A) `6 7 8`

B) `6 7 5`

C) `3 4 5`

D) `3 4 8`

E) No output. Will not compile.

Question 24.

What is the output of the client code shown here if **<code>** has been filled in correctly?

```
Uil b=new Uil();
b.setC(6);
b.a=1;
out.print(b.a+" "+b.b+" "+b.getC());
```

A) `1 6`

B) `1 null 6`

C) `1 4 6`

D) `1 0 6`

E) No output. Will not compile.

```
//Use the code shown here to answer
//questions 21 – 24.

public class Uil {

      int a,b;
      private int c;

      public Uil(int x, int y, int z) {
            a = x;
            b = y;
            c = z;
      }

      public Uil() {}

      public int getC() {
            return c;
      }

      public void setC(int c) {
            <code>.c = c;
      }
}
```

Question 25.

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

A) -32

B) 32

C) 8

D) -8

E) -1

```
String s="String";
String t="strange";
out.print(s.compareTo(t));
```

Question 26.

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

A) 9 36

B) 3 15

C) 9 16

D) 3 16

E) 3 6

```
int[][] i= {{5,7,4},{3,0,2},{1,8,6}};
out.print(i.length+" ");
int s=0;
for(int x=1;x<3;x++)
     for(int y=1;y<3;y++)
          s+=i[x][y];
out.print(s);
```

Question 27.

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

A) 17 12 -4

B) 18 12 -6

C) 18 1 -4

D) 17 1 -6

E) 18 1 -6

```
byte r=17,s=18,t=-5;
s/=++r+t--;
out.print(r+" "+s+" "+t);
```

Question 28.

Which of the following methods will return the index number of the last occurrence of the character passed as parameter `c` or -1 if the character is not present in the string?

A.

```
public static int lastIndexOf(String s,char c)
{
    int k=-1;
    for(int i=s.length()-1;i>=0;i++) {
      if(s.charAt(i)==c){
        k=i;
        break;
        }
    }
    return k;
}
```

B.

```
public static int lastIndexOf(String s,char c)
{
    int k=-1;
    for(int i=s.length()-1;i>=0;i--) {
      if(s.charAt(i)==c){
        k=i;
        break;
        }
    }
    return k;
}
```

C.

```
public static int lastIndexOf(String s,char c)
{
    int k=0;
    for(int i=s.length()-1;i>=0;i--) {
      if(s.charAt(i)==c){
        k=i;
        break;
        }
    }
    return k;
}
```

D.

```
public static int lastIndexOf(String s,char c)
{
    int k=-1;
    for(int i=0;i< s.length()-1;i++) {
      if(s.charAt(i)==c){
        k=i;
        break;
        }
    }
    return k;
}
```

E.

```
public static int lastIndexOf(String s,char c)
{
    int k=-1;
    for(int i=s.length()-1;i>=0;i--) {
      if(s.charAt(i)==c){
        k=i;
        break;
        }
    }
}
```

Question 29.

Which of the following Java statements will compile and correctly calculate the Celsius temperature when given a Fahrenheit temperature? The formula is shown on the right where f is the Fahrenheit temperature and c is the Celsius temperature.

$$C=5/9(F-32)$$

A) `double c=5.0/9.0(f-32);`

B) `double c=5/9*(f-32);`

C) `double c=5.0/9.0*(f-32);`

D) All of the above.

E) None of the above.

Question 30.

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

A) 49.0 33.0 17.0

B) 61.0 41.0 21.0

C) 58.0 40.0 22.0

D) 50.0 37.0 24.0

E) 47.0 35.0 23.0

```
double x=10.0,y=5.0,z=8.0;
while(x<50) {
     x=y--+z;
     y+=5;
     z=y*2-x;
}
out.println(x+" "+y+" "+z);
```

Question 31.

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

A) 420

B) 212

C) 312

D) 303

E) 0

```
String s="amigdexaul";
int sum=0;
for(int i=0;i<s.length();i++) {
 if(s.charAt(i)<110)
  char c=s.charAt(i);
  boolean vowel=c=='a'||c=='e'||c=='i'
                    ||c=='o'||c=='u';
  if(vowel)
   continue;
  else
   sum+=s.charAt(i);
 else
  break;
}
out.print(sum);
```

Question 32.

Which of the following shows the order of the elements in array a when code execution reaches the comment statement and i equals 3 given the following client code?

```
int[] a= {3,2,4,1,0};
sort(a);
```

A) [1, 2, 3, 4, 0]

B) [2, 3, 4, 1, 0]

C) [0, 1, 2, 3, 4]

D) [1, 0, 2, 3, 4]

E) [0, 1, 2, 4, 3]

Question 33.

Which of the following sorting algorithms is implemented by the sort method shown on the right?

A) selection sort

B) insertion sort

C) bubble sort

D) quick sort

E) merge sort

```
//Use the following to answer
//questions 32, 33 and 34.
public static void sort(int[] a) {
  for(int i=1;i<a.length;i++) {
    int ce=a[i];
    int k;
    for(k=i-1;k>=0&&a[k]>ce;k--) {
      a[k+1]=a[k];
    }
    a[k+1]=ce;
    //comment
  }
}
```

Question 34.

What is the worst case run time efficiency (Big O value) for the sort method shown on the right?

A) O(n)

B) $O(n^2)$

C) $O(n^3)$

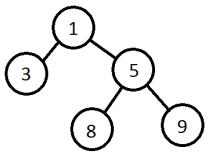
D) O(log n)

E) O(n log n)

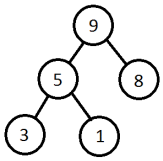
Question 35.

If 1, 8, 3, 5 and 9 are placed into a binary search tree, in that order, which of the following is the correct representation of that tree?

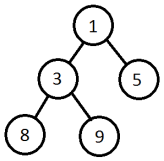
A)



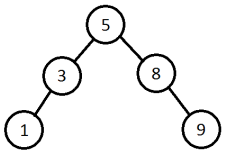
B)



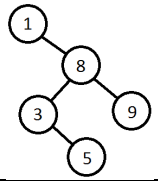
C)



D)



E)



Question 36.

Which of the following best describes the graph shown on the right?

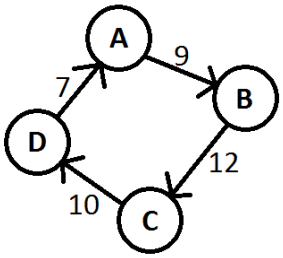
A) weighted, directed and complete

B) weighted and directed

C) unweighted and directed

D) weighted and undirected

E) unweighted and undirected

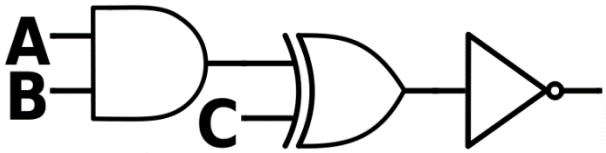


Question 37.

If A is true, B is false and C is false, what is the result of the expression shown in the diagram shown on the right?

A) true

B) false



Question 38.

What is the worst case time complexity for accessing an element in a linked list?

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

Question 39.

Evaluate the postfix expression shown on the right and write your answer in the blank provided. The operands are 19, 4, 5, -5, and 3.

```
19 4 5 + -5 - 3 * -
```

Question 40.

Write the 8-bit two's complement binary equivalent of -50.

★ANSWER KEY – CONFIDENTIAL★

UIL COMPUTER SCIENCE – 2018 INVITATIONAL A

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

1) D	11) B	21) D	31) C
2) C	12) A	22) B	32) A
3) E	13) C	23) E	33) B
4) A	14) A	24) D	34) B
5) A	15) D	25) A	35) E
6) B	16) B	26) D	36) B
7) E	17) E	27) E	37) A
8) C	18) C	28) B	38) C
9) E	19) C	29) C	*39) -23
10) D	20) A	30) D	*40) 11001110

** 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:

<table>
<tr><td>1.</td><td>D</td><td>128+64+32+16+8+4+2+1=255</td></tr>
<tr><td>2.</td><td>C</td><td>6*4*3+7-14/5 = 72+7-2 = 79-2 = 77</td></tr>
<tr><td>3.</td><td>E</td><td><code>println</code> and <code>\n</code> both produce a new line.</td></tr>
<tr><td>4.</td><td>A</td><td>The <code>trim</code> method removes whitespace from either end of a string.</td></tr>
<tr><td>5.</td><td>A</td><td>!true ^ !false = false ^ true = true. Exclusive or (^) is true if one but not both of the operands are true.</td></tr>
<tr><td>6.</td><td>B</td><td>4 + 1.0 = 5.0</td></tr>
<tr><td>7.</td><td>E</td><td>5 - 4.75 + 5 = 0.25 + 5 = 5.25</td></tr>
<tr><td>8.</td><td>C</td><td>1.8 = 1.8 and 1.81 > 1.8</td></tr>
<tr><td>9.</td><td>E</td><td>Value of x Number of hash tags
0 #
1 ##
2 ###
3 ####
4 #####
5 ######</td></tr>
<tr><td>10.</td><td>D</td><td>Default value of a String variable is null.<table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>Null</td><td>One</td><td>Two</td><td>Three</td><td>Four</td></tr></table></td></tr>
<tr><td>11.</td><td>B</td><td>The File class does not contain a <code>hasNext</code> method. The <code>next</code> method within the Scanner class returns a String, not a boolean.</td></tr>
<tr><td>12.</td><td>A</td><td>Since the initial value of x is zero, the value of t becomes and stays zero throughout the loop because of multiplication by zero. The value of x reaches 5, which ends the loop."</td></tr>
<tr><td>13.</td><td>C</td><td>++5 + (-8) * (int) 3.5 = 6 + (-8) * 3 = 6 + (-24) = -18. Casting a double to an int truncates 3.5 to be just 3.</td></tr>
<tr><td>14.</td><td>A</td><td>Range of values for the byte data type is -128 to 127.</td></tr>
<tr><td>15.</td><td>D</td><td><table><tr><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><td>lamb</td><td></td><td></td><td></td></tr><tr><td>lamb</td><td>kid</td><td></td><td></td></tr><tr><td>pig</td><td>lamb</td><td>kid</td><td></td></tr><tr><td>pig</td><td>lamb</td><td>kid</td><td>calf</td></tr></table></td></tr>
<tr><td>16.</td><td>B</td><td>[^aeiou] is a set that includes any character that is <u>not</u> a vowel. Therefore, the code prints true if a letter is a consonant and false if it is a vowel.</td></tr>
<tr><td>17.</td><td>E</td><td>The call stack is popped as follows 1+10+8+6+4+2+0, which equals 31.</td></tr>
<tr><td>18.</td><td>C</td><td><code>TreeSet</code> stores elements in sorted order with no duplicates. The <code>remove</code> method returns false if the element to be removed is not in the set.</td></tr>
<tr><td>19.</td><td>C</td><td>nextInt(x) returns a random whole number from 0 inclusive to x exclusive. Therefore, (9 + 10) * 2 = 38 is the largest possible value.</td></tr>
<tr><td>20.</td><td>A</td><td>Any value or object may be stored as an <code>Object</code> data type. <code>null</code> is the default value for an unassigned object within an array.</td></tr>
<tr><td>21.</td><td>D</td><td>Field and instance variable have the same meaning. They are variables that contained within and belong to each instance of the class (every object that is instantiate from that class). A class variable is a static variable. There is only one class variable regardless of how many objects are instantiated from the class.</td></tr>
<tr><td>22.</td><td>B</td><td>The reserved word <code>this</code> designates the variable as an instance variable and not a local variable.</td></tr>
<tr><td>23.</td><td>E</td><td>Cannot access the field <code>c</code> from outside the class because it has been designated as private.</td></tr>
<tr><td>24.</td><td>D</td><td>The instance variable <code>b</code> was never assigned. Default value is 0.</td></tr>
<tr><td>25.</td><td>A</td><td>The ASCII value of S is 83 and s is 115. 83 – 115 = -32.</td></tr>
<tr><td>26.</td><td>D</td><td>A two-dimensional array is an array of arrays. So, the length is how many arrays are in the array. In this case 3. 0 + 2 + 8 + 6 = 16</td></tr>
<tr><td>27.</td><td>E</td><td>s = 18/(++17+(-5)--) = 18/(18-6) = 18/12 = 1. <code>r</code> is incremented by one and <code>t</code> is decremented by 1. So, r=18, s=1 and t=-6.</td></tr>
<tr><td>28.</td><td>B</td><td>A increments i, thus moving the wrong direction. C returns 0 if the character is not present in the string. D finds the first occurrence. E never returns anything.</td></tr>
</table>

<table>
<tr><td>29.</td><td>C</td><td>A is missing the multiplication operator. B uses integer division. 5/9=0.</td></tr>
<tr><td>30.</td><td>D</td><td>These are the values for each variable at the beginning of each iteration of the loop and the final line is the values after the loop has stopped execution.
10.0 5.0 8.0
13.0 9.0 5.0
14.0 13.0 12.0
25.0 17.0 9.0
26.0 21.0 16.0
37.0 25.0 13.0
38.0 29.0 20.0
49.0 33.0 17.0
50.0 37.0 24.0</td></tr>
<tr><td>31.</td><td>C</td><td><code>continue</code> skips the remainder of the loop body. <code>break</code> stops loop execution. The loop executes until it encounters a character greater than or equal to n. Prior to that if the character is a vowel it is skipped. If it is a consonant, that characters ASCII value is added to sum. 109 + 103 + 100 = 312.</td></tr>
<tr><td>32.</td><td>A</td><td><table><tr><td>i</td><td></td></tr><tr><td>1</td><td>2 3 4 1 0</td></tr><tr><td>2</td><td>2 3 4 1 0</td></tr><tr><td>3</td><td>1 2 3 4 0</td></tr></table></td></tr>
<tr><td>33.</td><td>B</td><td>An insertion sort works by choosing the next element in the array and then placing it into its proper location within the already sorted portion of the array. A common analogy is picking up playing cards from a table and placing them in order in your hand.</td></tr>
<tr><td>34.</td><td>B</td><td>Best Case O(n), Average Case O(n^2), Worst Case O(n^2)</td></tr>
<tr><td>35.</td><td>E</td><td>The first value is the root. After inserting the root, in this case 1, each value is inserted to the right if it is greater than the root or to the left if it is less than the root. After moving to the left or right of the root, the next node is considered the root and the process is repeated until there is no longer another node to compare to.</td></tr>
<tr><td>36.</td><td>B</td><td>Weighted means each edge has a value. Directed means you can only travel one direction along an edge. This graph is not complete because not every pair of vertices are connected.</td></tr>
<tr><td>37.</td><td>A</td><td>!((true&&false)^false) = !(false ^ false) = !false = true</td></tr>
<tr><td>38.</td><td>C</td><td>Elements within a linked list must be traversed from the first node and progressing one node at a time. The element to be accessed might be the last element in the list.</td></tr>
<tr><td>39.</td><td>-23</td><td>19 4 5 + -5 – 3 * - = 19 9 -5 – 3 * - = 19 14 3 * - = 19 42 - = -23</td></tr>
<tr><td>40.</td><td>11001110</td><td>Binary value of 50 is 00110010. Find the complement (flip the bits) to get 11001101, which represents the value -51 (complement is opposite, minus 1). Add 1 to get 11001110.</td></tr>
</table>