UIL COMPUTER SCIENCE WRITTEN TEST

2021 INVITATIONAL A

JANUARY/FEBRUARY 2021

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 – 2021 INVITATIONAL A

Note: Correct responses are based on **Java SE Development Kit 14 (JDK 14)** 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 14 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 the binary number 11100111_2?

A) 247 **B)** 239 **C)** 115 **D)** 231 **E)** 103

Question 2.

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

A) 16 **B)** 18.0 **C)** 11 **D)** 11.0 **E)** 11.75

```
out.print(9 / 4 * 3 + 7 - 2);
```

Question 3.

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

A) Earth
Mars
Venus

B) Earth Mars Venus

C) EarthMarsVenus

D) “Earth” “Mars” “Venus”

E) “Earth”
“Mars”
“Venus”

```
out.print("Earth");
out.print("Mars");
out.print("Venus");
```

Question 4.

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

A) 3 **B)** 2 **C)** 5
D) 6 **E)** 1

```
String str = "java.lang.System.out";
out.print(str.indexOf('a'));
```

Question 5.

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

A) true

B) false

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

Question 6.

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

A) -1.01 **B)** 0.0 **C)** 0.01 **D)** 2.02 **E)** – 0.01

```
out.print(Math.ceil(1.01) +
          Math.floor(-1.01));
```

Question 7.

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

A) 16.0 **B)** 12.67 **C)** 19.2 **D)** 17.2 **E)** 18.0

```
int x = 8, y = 9;
double a = 1.5, b = 1.25;
out.print(x * b + y / a);
```

Question 8.

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

A) 13 5614

B) 13 25

C) 25

D) There is no output.

E) There is no output due to an error.

```
final int M = 13;
int x = 5, y = 6, z = 14;
if(x * y >= M)
     if(z - 1 == M)
          if(x + y - M > z)
               out.print(M + " ");
               out.print(x + y + z);
```

Question 9.

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

A) ********

B) ******

C) *****

D) *******

E) *

```
int x = 0;
do {
     x++;
     out.print("*");
}while(x < 7);
```

Question 10.

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

A) [0, 5, 2, 6, 3, 5, 6]

B) [0, 5, 2, 6, 3, 5, 7]

C) [0, 1, 2, 6, 3, 5, 7]

D) [5, 4, 2, 6, 3, 5, 7]

E) There is no output due to an error.

```
int[] list = {0,4,2,6,3,5,1};
list[1] = list[list.length - 2];
list[list[3]] = list.length;
out.print(Arrays.toString(list));
```

Question 11.

```
import static java.lang.System.out;
import java.util.Scanner;
import java.io.File; //LINE #1
import java.io.IOException;
public class Q11 {
     public static void main(String[] args) throws IOException{ //LINE #2
           File file = new File("values.dat");
           Scanner scanner = new Scanner(file); //LINE #3
           while(scanner.hasNext()) //LINE #4
                 out.println(scanner.next());
           file.close(); //LINE #5
     }
}
```

One of the lines in the class shown above contains an error. Which line is it?

A) LINE #1 **B)** LINE #2 **C)** LINE #3 **D)** LINE #4 **E)** LINE #5

Question 12.

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

A) 36

B) 55

C) 44

D) 45

E) 46

```
int h = 1;
for(int g = 2; g < 10; g++) {
     h += g;
}
out.print(h);
```

Question 13.

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

A) 14

B) 22

C) 17

D) 15

E) 24

```
int w = 5, x = 6, y = 4, z = 1;
out.print(++w + x * y >> z);
```

Question 14.

Which of the following data types may be used to store 32768 and requires the smallest amount of memory to do so?

A) `byte`

B) `short`

C) `double`

D) `long`

E) `int`

Question 15.

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

A) 524

B) 634

C) 534

D) 625

E) 521

```
ArrayList<Integer> list = new
ArrayList<Integer>();
list.add(6);list.add(5);
list.add(4);list.add(3);
list.add(2);list.add(1);
out.print(list.get(1));
out.print(list.remove(4));
out.print(list.indexOf(1));
```

Question 16.

How many constructors are present in the class `Class` shown on the right?

A) None **B)** 1 **C)** 2 **D)** 3 **E)** 4

Question 17.

What is the output of **LINE #1** in the client code shown here?

```
Class obj1 = new Class();
Class obj2 = new
  Class(obj1.d().length(),0.75,"computer");
out.println(obj2.d()); //LINE #1
```

A) scmputer

B) mputer

C) puter

D) coience

E) com

Question 18.

What is the output of **LINE #1** in the client code shown here?

```
Class obj1 = new Class();
Class obj2 = new
  Class(obj1.d().length(),0.75,"computer");
out.println(obj2.d());
Class obj3 = new
  Class(obj1.a(),obj2.c(),"competition");
out.println(obj3.d); //LINE #1
```

A) petition

B) mpetition

C) etition

D) scipetition

E) There is no output due to an error.

```
//Use the class shown below to answer
//questions 16, 17 and 18.

 public class Class {

 private int i;
 private double d;
 private String s;

 public Class(int m, double n, String o) {
      i = m;
      d = n;
      s = o;
 }

 public Class() {
      i = 8;
      d = 0.5;
      s = "science";
 }

 public int a() {
      return (int)(d * b());
 }

 public int b() {
      return i;
 }

 public double c() {
      return d;
 }

 public String d() {
      return s.substring(a());
 }
 }
```

Question 19.

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

```
ArrayList<Integer> x = new
ArrayList<Integer>();
ArrayList<Integer> y = new
ArrayList<Integer>();
Integer[] i = {5,6,4,3,7,1,8,9,2};
for(int m:i) {
  x.add(m);
  y.add(m);
}
System.out.print((x == y) + " ");
System.out.print(x.equals(y));
```

A) `true true`

B) `true false`

C) `false true`

D) `false false`

E) Will not compile. Type mismatch error.

Question 20.

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

```
double a = 4.99, b = 3.5;
double c = (int)a + b;
out.print(c);
```

A) `8.5`

B) `8`

C) `7.0`

D) `7.5`

E) `7`

Question 21.

What is the output of the class `PetsTest`?

```
public class PetsTest {
enum Pets {
  DOG, CAT, LIZARD,
  RABBIT, HAMSTER, PIG
 }

public static void main(String[] args) {
  Pets[] myPets = Pets.values();
  System.out.print(myPets[3].name());
  System.out.print(Pets.CAT);
 }
}
```

A) `LIZARD1`

B) `RABBITCAT`

C) `RABBIT1`

D) `RABBIT2`

E) `1CAT`

Question 22.

Which of the following represents the output of the code segment shown here?

```
char[][] letters = new char[5][5];
char letter = 'A';
for(int i = 0; i < letters.length; i++)
     for(int j = 0; j < letters[i].length; j++)
           letters[i][j] = letter++;
for(char[] arr:letters)
     System.out.println(Arrays.toString(arr));
```

A
```
[A, B, C, D, E]
[F, G, H, I, J]
[K, L, M, N, O]
[P, Q, R, S, T]
[U, V, W, X, Y]
```

B
```
[A, B, C, D, E]
[F, G, H, I, J]
[K, L, M, N, O]
[P, Q, R, S, T]
[U, V, W, X, Y, Z]
```

C
```
[A, F, K, P, U]
[B, G, L, Q, V]
[C, H, M, R, W]
[D, I, N, S, X]
[E, J, O, T, Y]
```

D
```
[B, C, D, E, F]
[G, H, I, J, K]
[L, M, N, O, P]
[Q, R, S, T, U]
[V, W, X, Y, Z]
```

E
```
[B, G, L, Q, V]
[C, H, M, R, W]
[D, I, N, S, X]
[E, J, O, T, Y]
[F, K, P, U, Z]
```

Question 23.

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

A) 77

B) 77.25

C) 5-3A4.25

D) 2A4.25

E) 71.25

```
int x = -3;
out.print(5 + x + 'A' + 4.25);
```

Question 24.

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

A) 0.84

B) 7.15

C) 7

D) 0.0

E) There is no output due to an error.

```
double j = 12.85;
double k = 6.3;
double p = k += j %= 3;
out.print(((int)(p * 100)) / 100.0);
```

Question 25.

What is printed by the line of code shown on the right?

A) 708

B) 00101100

C) 0

D) 44

E) There is no output due to an error.

```
out.print(0b10110001 >> 0b00000010);
```

Question 26.

Which of the following represents the output of the code segment shown on the right?

A) 3528

B) 4536

C) 2016

D) 504

E) 1527

```
int y = 5;
for(int x = 0; x < 10; x++)
     switch(x) {
     case 0:
     case 1:
     case 3: y += x; break;
     case 4:
     case 5:break;
     case 6: y -= 2; break;
     case 7:
     case 8:
     case 9: y *= x; break;
     default: y /= x;
     }
out.print(y);
```

Question 27.

Which of the following statements about functional interfaces is **false**?

A) A functional interface may contain numerous default methods.

B) A functional interface may not contain any fields.

C) A functional interface may contain exactly one abstract method.

D) A functional interface may override the `public boolean equals(Object obj)` method from the `Object` class.

E) Every method within a functional interface must be abstract.

Question 28.

Consider the code segment shown here. Which of the following can replace the missing code to ensure that all the elements in `list` are placed into the same row and column of `table` as they are in `list`?

```
ArrayList<String[]> table = new ArrayList<String[]>();
String[][] list = {{"a","b","t","e"},
                   {"r","q","u","c"},
                   {"b","z","s","o"}};
//missing code
```

A)
```
for(int i = 0; i < list.length; i++)
   for(String[] letters : list)
      table.add(i,letters);
```

B)
```
for(int i = 0; i < list.length; i++)
   for(String[] letters : list)
      table.get(i).add(letters[i]);
```

C)
```
for(String[] letters : list)
   table.add(letters);
```

D)
```
for(int i = 0; i < list.length; i++)
   table.get(i).add(list[i][i]);
```

E) More than one of the above.

Question 29.

Consider the code segment shown on the right. What is the output of that code segment?

A) `121`

B) `110`

C) `512`

D) `1048576`

E) `262144`

```
final int N = 1024;
long total = 0;
for(int x = N; x > 0; x/=2)
   for(int y = 1; y <= N; y*=2)
      total++;
System.out.print(total);
```

Question 30.

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

A) `M`

B) `ssissippi`

C) `ss`

D) `pp`

E) There is no output due to an error.

```
String[] s = "Mississippi".split("i", 2);
out.print(s[2]);
```

Question 31.

Which of the following is the output of the main method shown here?

```
  public static void main(String[] args) {
          method("abcde");
          method("abcdef");
     }

     public static void method(String s) {
          if(s.length() % 2 == 0) {
               method(s.substring(s.length() / 2));
               method(s.substring(0, s.length() / 2));
          }
          else if(s.length() > 1) {
               out.print(s.substring(s.length() / 2, s.length() / 2 + 1));
               method(s.substring(0, s.length() / 2));
               method(s.substring(s.length() / 2 + 1));
          }
          else
               out.print(s);
     }
```

A) `abcdeabcdef` **B)** `fedcbaedcba` **C)** `cabdebacedf` **D)** `cedbaefdbca` **E)** `cbaededfbac`

Question 32.

Which of the following code segments must be added to this sort method to complete the implementation of an ascending insertion sort?

```
public static void sort(int[] list)
{
//missing implementation
}
```

A)

```
for(int i = 0; i < list.length - 1; i++)
{
int k = i;
for(int j = i + 1; j < list.length; j++)
  if(list[j] < list[k])
    k = j;
int t = list[i];
list[i] = list[k];
list[k] = t;
}
```

B)

```
for(int x = 1; x < list.length; x++)
{
  int y = list[x];
  int z = x;
  while(z > 0 && y < list[z - 1] )
  {
    list[z] = list[z - 1];
    z--;
  }
  list[z] = y;
 }
```

C)

```
for(int m = 0; m < list.length - 1; m++)
  for(int n = m + 1; n < list.length; n++)
    if(list[n] < list[m])
    {
      int t = list[n];
      list[n] = list[m];
      list[m] = t;
    }
```

D)

```
if(i == list.length) return;
int k = i;
for(int j = i + 1; j < list.length; j++)
  if(list[j] < list[k])
    k = j;
int t = list[i];
list[i] = list[k];
list[k] = t;
sort(list, i + 1);
```

E) More than one of the above.

Question 33.

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

A) `A D B C E`

B) `E C B D A`

C) `C B D A E`

D) `E A D B C`

E) `A B C D E`

```
Stack<String> s = new Stack<String>();
s.push("A");s.push("B");s.add("C");
s.add(1, "D");s.push("E");
s.add(0, s.pop());
while(!s.empty())
  out.print(s.pop() + " ");
```

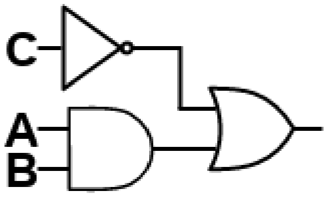
Question 34.

What is the value of the expression shown in the diagram on the right if C is `false`?

A) `true`

B) `false`

C) Cannot be determined.



Question 35.

Which of the following shows the output of the code segment shown on the right?

A) `[7, 1, 4, 6, 8, 9, 2, 5]`

B) `[7, 1, 2, 4, 6]`

C) `[7, 1, 2, 4, 6, 8]`

D) `[7, 1, 4, 6, 8, 9, 2]`

E) `[2, 3, 4]`

```
int[] a = {3,7,1,9,8,6,4,2,5,0};
Arrays.sort(a, 3, 7);
int[] b = Arrays.copyOfRange(a, a[2], a[5]);
out.print(Arrays.toString(b));
```

Question 36.

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

A) `12`

B) `5`

C) `6`

D) `10`

E) `7`

```
String str = "UIL_Comp^Sci-2021";
int r = 0;
for(int i = 0; i < str.length(); i++)
  if(str.substring(i, i + 1).matches("[^a-z]"))
    r = r + 1;
out.print(r);
```

Question 37.

Which of the following is the output of the line of code shown on the right?

A) `26`

B) `10`

C) `7`

D) `30`

E) `19`

```
out.print(18 | 3 & 5);
```

Question 38.

Which of the following statements is false?

A) An interface may not have any constructors.

B) An abstract class may not have any constructors.

C) All variables within an interface must be declared as public static final.

D) All methods within an interface must be public abstract instance methods.

E) In a nonabstract subclass extended from an abstract class, all the abstract methods must be implemented.

Question 39.

What is the most restrictive run time efficiency for the code segment shown on the right for exceptionally large values of `n`. Express your answer <u>using Big O notation</u> and write it in the blank provided on the answer document.

```
for(int m = 1; m <= n; m++)
  for(int o = n; o > 1; o/=2)
    x++;
out.println(x);
```

Question 40.

Write the signed 8-bit binary two's complement representation of -104 in the blank provided on the answer document.

★ANSWER KEY – CONFIDENTIAL★

UIL COMPUTER SCIENCE – 2021 INVITATIONAL A

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

1) D

2) C

3) C

4) E

5) A

6) B

7) A

8) C

9) D

10) B

11) E

12) D

13) D

14) E

15) A

16) C

17) B

18) E

19) C

20) D

21) B

22) A

23) E

24) B

25) D

26) C

27) E

28) C

29) A

30) E

31) E

32) B

33) C

34) A

35) D

36) A

37) E

38) B

*39) O(n log n)

*40) 10011000

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

Note: Correct responses are based on **Java SE Development Kit 14 (JDK 14)** 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 14 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><table><tr><td>128</td><td>64</td><td>32</td><td>16</td><td>8</td><td>4</td><td>2</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr></table>128 + 64 + 32 + 4 + 2 +1 = 231</td></tr>
<tr><td>2.</td><td>C</td><td>9 / 4 * 3 + 7 – 2 = 2 * 3 + 7 – 2 = 6 + 7 – 2 = 13 – 2 = 11</td></tr>
<tr><td>3.</td><td>C</td><td>The print methods leave the cursor on the same line where the text is printed. There are no spaces in the string literals so none are printed.</td></tr>
<tr><td>4.</td><td>E</td><td>indexOf(‘a’) returns the index value of the first occurrence ‘a’ in the string.</td></tr>
<tr><td>5.</td><td>A</td><td>^ (XOR) will evaluate as true if one or the other but not both operands are true.
true ^ true ^ true =
false ^ true =
true</td></tr>
<tr><td>6.</td><td>B</td><td>ceil(1.01) → 2.0,   floor(-1.01) → -2.0,   2.0 +-2.0 = 0.0</td></tr>
<tr><td>7.</td><td>A</td><td>8 *1.25 + 9 / 1.5 = 10.0 + 9 / 1.5 = 10.0 + 6.0 = 16.0</td></tr>
<tr><td>8.</td><td>C</td><td>5 * 6 &gt;= 13 is true
14 – 1 == 13 is true
5 + 6 – 13 > 14 is false
First output statement is not executed.
x + y + z = 5 + 6 + 14 = 25
Indentation does not create a block statement.</td></tr>
<tr><td>9.</td><td>D</td><td><table><tr><td>x</td><td>Output</td></tr><tr><td>0</td><td></td></tr><tr><td>1</td><td>*</td></tr><tr><td>2</td><td>**</td></tr><tr><td>3</td><td>***</td></tr><tr><td>4</td><td>****</td></tr><tr><td>5</td><td>*****</td></tr><tr><td>6</td><td>******</td></tr><tr><td>7</td><td>*******</td></tr></table></td></tr>
<tr><td>10.</td><td>B</td><td><table><tr><th>index</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th></tr><tr><td>int[] list = …</td><td>0</td><td>4</td><td>2</td><td>6</td><td>3</td><td>5</td><td>1</td></tr><tr><td>list[1] = …</td><td>0</td><td>5</td><td>2</td><td>6</td><td>3</td><td>5</td><td>1</td></tr><tr><td>list[list[3]] = …</td><td>0</td><td>5</td><td>2</td><td>6</td><td>3</td><td>5</td><td>7</td></tr></table></td></tr>
<tr><td>11.</td><td>E</td><td>The scanner object should be closed instead of the file object.</td></tr>
<tr><td>12.</td><td>D</td><td>g is added to h with each iteration of the loop.<table><tr><th>h</th><th>g</th></tr><tr><td>1</td><td>2</td></tr><tr><td>3</td><td>3</td></tr><tr><td>6</td><td>4</td></tr><tr><td>10</td><td>5</td></tr><tr><td>15</td><td>6</td></tr><tr><td>21</td><td>7</td></tr><tr><td>28</td><td>8</td></tr><tr><td>36</td><td>9</td></tr><tr><td>45</td><td>10</td></tr></table></td></tr>
<tr><td>13.</td><td>D</td><td>++5 + 6 * 4 &gt;&gt; 1 =
6 + 6 * 4 &gt;&gt; 1 =
6 + 24 >> 1 =
30 &gt;&gt; 1 = <i>(this is the same as 30 / 2)</i>
15</td></tr>
<tr><td>14.</td><td>E</td><td><i>byte</i> -128 to 127
<i>short</i> -32768 to 32767
<i>int</i> -2147483648 to 2147483647</td></tr>
<tr><td>15.</td><td>A</td><td>get returns the element at that index number.
remove removes the argument or returns false if it is not present.
indexOf returns the index number of the argument.</td></tr>
<tr><td>16.</td><td>C</td><td>Constructors have the same name as the class and no return type.</td></tr>
</table>

<table>
<tr><td>17.</td><td>B</td><td>The default constructor sets i, d and s to 8, 0.5 and “science”. The first argument in the call to the constructor to instantiate obj2 is a call to obj1.d() which returns the string “nce” whose length is 3. So, the fields i, d and s in obj2 are assigned 3, 0.75 and “computer”. The call to obj2.d() returns “mputer”.</td></tr>
<tr><td>18.</td><td>E</td><td>obj3.d will not compile. d is a private field.</td></tr>
<tr><td>19.</td><td>C</td><td>x and y are two different list that contain the same elements. Therefore, x == y is false. The equals method is inherited from the abstract class AbstractList<E> which implements equals as follows: Compares the specified object with this list for equality. Returns true if and only if the specified object is also a list, both lists have the same size, and all corresponding pairs of elements in the two lists are equal.</td></tr>
<tr><td>20.</td><td>D</td><td>Cast first then add. Result is a double.</td></tr>
<tr><td>21.</td><td>B</td><td>values() returns an array containing each of the constants in the enumerated type.
name() returns the name of the constant.
Pets.CAT references the specific constant CAT.</td></tr>
<tr><td>22.</td><td>A</td><td>Letters are placed in the 2D array in row major order.</td></tr>
<tr><td>23.</td><td>E</td><td>5 + -3 + 65 + 4.25 =
2 + 65 + 4.25 =
67 + 4.25 =
71.25</td></tr>
<tr><td>24.</td><td>B</td><td>p = 6.3 += 12.85 %= 3 → p = 6.3 += 0.85 → p = 7.15 Compound assignment is done from right to left.</td></tr>
<tr><td>25.</td><td>D</td><td>Shift the bits two places to the right in the left operand to get 00101100_2 which is 44_{10}. The print method converts binary to decimal.</td></tr>
<tr><td>26.</td><td>C</td><td><table>
<tr><td>x</td><td>y</td></tr>
<tr><td></td><td>5</td></tr>
<tr><td>0</td><td>5</td></tr>
<tr><td>1</td><td>6</td></tr>
<tr><td>2</td><td>3</td></tr>
<tr><td>3</td><td>6</td></tr>
<tr><td>4</td><td>6</td></tr>
<tr><td>5</td><td>6</td></tr>
<tr><td>6</td><td>4</td></tr>
<tr><td>7</td><td>28</td></tr>
<tr><td>8</td><td>224</td></tr>
<tr><td>9</td><td>2016</td></tr>
</table></td></tr>
<tr><td>27.</td><td>E</td><td>A functional interface may only contain one abstract method but may also include default methods.</td></tr>
<tr><td>28.</td><td>C</td><td>table is an ArrayList of String arrays. The enhanced for loop gets each array in list and adds it to the end of the table as a row.</td></tr>
<tr><td>29.</td><td>A</td><td>Both the outer and inner loops execute 11 times. 11 * 11 = 121</td></tr>
<tr><td>30.</td><td>E</td><td>The second argument of the split method restricts the size of s to two elements. print(s[2]) throws an indexOutOfBounds exception.</td></tr>
<tr><td>31.</td><td>E</td><td>If the string passed to method has an even length each half of the string is processed in a right to left fashion. If the string passed to the method has odd number of characters the middle character is printed then the remaining halves are processed from left to right. If the string passed to method has one character, that character is printed.</td></tr>
<tr><td>32.</td><td>B</td><td>A and D are selection sorts. C is a bubble sort.</td></tr>
<tr><td>33.</td><td>C</td><td><table>
<tr><td>Original Stack</td><td>add(1,”D”)</td><td>push “E”</td><td>add(0,pop())</td></tr>
<tr><td>C</td><td>C</td><td>E</td><td>C</td></tr>
<tr><td>B</td><td>B</td><td>C</td><td>B</td></tr>
<tr><td>A</td><td>D</td><td>B</td><td>D</td></tr>
<tr><td></td><td>A</td><td>D</td><td>A</td></tr>
<tr><td></td><td></td><td>A</td><td>E</td></tr>
</table></td></tr>
<tr><td>34.</td><td>A</td><td>Equivalent to NOT false OR A AND B. NOT false is true. true OR anything is always true.</td></tr>
<tr><td>35.</td><td>D</td><td>Arrays.sort(a, 3, 7) sorts the portion of the array a from index 3 to 6 in ascending order. Arrays.copyOfRange(a, a[2], a[5]) is equivalent to copyOfRange(a,1, 8) which returns {7,1,4,6,8,9,2}</td></tr>
<tr><td>36.</td><td>A</td><td>Counts any character that is not a lower case letter.</td></tr>
</table>

37.	E	Apply & then |. 18 | 3 & 5 = 10010 | 011 & 101 = 10010 | 001 = 10011 = 19
38.	B	A class declared as an abstract class cannot be instantiated but an abstract class can have a constructor. Each abstract class must have a concrete subclass which will implement the abstract methods of that abstract class. Creating an object of any subclass invokes all the constructors in the corresponding inheritance tree. The same case applies to abstract classes. Even though an object of an abstract class cannot be instantiated, when an object of a class which is concrete and is a subclass of an abstract class is instantiated, the constructor of the abstract class is automatically invoked. Therefore, there can be a constructor in abstract class.
39.	O(n log n)	The outer loop executes n times and the inner loop executes $\log_2 n$ times. Multiply and ignore the base to get n * log n. Must be written using Big-O notation.
40.	10011000	Convert 104 to binary → 01101000 Flip all the bits → 10010111 Add one → 10011000