

UIL COMPUTER SCIENCE WRITTEN TEST

2021 INVITATIONAL B

FEBRUARY/MARCH 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 B

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 12 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used. For all output statements, assume that the `System` class has been statically imported using: `import static java.lang.System.*;`

Question 1.

Which of the following is equivalent to 80_{16} ?

- A) 1000000_2 B) 0100000_2 C) 1111111_2 D) 0100111_2 E) 1111110_2

Question 2.

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

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

```
out.print(12 % 5 - 7 % 6);
```

Question 3.

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

- A) Invitational\tB
B) Invitational
 B
C) Invitational B
D) InvitationalB
E) InvitationaltB

```
out.println("Invitational\tB");
```

Question 4.

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

- A) 0 B) a C) b
D) m E) e

```
String str = "Ramble On";  
out.print(str.charAt(str.length() - 7));
```

Question 5.

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

- A) true
B) false

```
boolean a = false;  
boolean b = true;  
boolean c = !(a && b);  
out.print(c);
```

Question 6.

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

- A) -5.0 B) 5.0 C) -5.5 D) 6.0
E) -6.0

```
double d = - Math.abs(-5.5);  
int i = (int) Math.round(-5.5);  
out.print(Math.max(d, i));
```

Question 7.

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

- A) 0.0
B) 35.0625
C) 35.0
D) 18.125
E) 24.125

```
double e = 3.5, f = 8.25;  
double g = 5, h = 2;  
out.print(e + g * f / h);
```

Question 8.

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

- A) m
- B) c
- C) z
- D) h
- E) UIL

```
char c = 'm', d = 'c';
char e = 'z', f = 'h';
if(c < d)
    if(d < e)
        out.print(f);
    else
        out.print(c);
else if(e > f)
    if(f < c)
        out.print(e);
    else
        out.print(d);
else
    out.print("UIL");
```

Question 9.

Which of the following represents the output of the code shown to the right?

- A) 28 21 14 7 0
- B) 35 5 0
- C) 35 28 21 14 7 0
- D) 35 28 21 14 7
- E) 35 5

```
for(int x = 35; x > 0; x /= 7)
    out.print(x + " ");
```

Question 10.

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

- A) [0, 7, 5, -1, 5]
- B) [7, 7, 5, -1, 0]
- C) [3, 2, 1, 0, 0]
- D) [-1, 7, 5, 0, 7]
- E) [5, 7, 5, -1, 0]

```
int[] values = new int[5];
values[3] = -1; values[2] = 5;
values[1] = 7; values[0] = 0;
values[values[4]] = values[values.length-3];
out.print(Arrays.toString(values));
```

Question 11.

Which of the following is the output of the code segment shown on the right? You may assume that all necessary import statements are present and correct. The file named `data.dat` contains the following:

one,two,three,four,five,six

- A) one,two,three,four,five,six
- B) t,t,i,i
- C) ttii
- D) one,wo,hree,four,fve,sx
- E) There is no output due to an error.

```
Scanner f = new Scanner(new
File("data.dat"));
f.useDelimiter("[ti]");
while(f.hasNext())
    out.print(f.next());
f.close();
```

Question 12.

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

- A) 256 75
- B) 64 65
- C) 128 70
- D) 76 75
- E) 1028 0

```
int d = 2, e = 1, f = 105;
while(e < f) {
    e *= d;
    f -= 5;
}
out.print(e + " " + f);
```

Question 13.

What is the output of the code segment shown here?

```
double x = 3.0, y = 2.75, z = 0.5;
out.print(x - z < y && z - Math.abs(y - x) <= z);
```

- A) true
- B) false

Question 14.

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

- A) -128
- B) -32767
- C) -32768
- D) 0
- E) -127

```
out.println(Short.MIN_VALUE);
```

Question 15.

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

- A) [E, F, C, B, D, A]
- B) [E, C, F, B, D, A]
- C) [E, A, C, F, B, D]
- D) [E, C, F, B, D]
- E) There is no output due to an error.

```
ArrayList<String> list = new
ArrayList<String>();
list.add("E");list.add("C");list.add("B");
list.add("D");list.add("A");
list.add(2, "F");list.add(1, "A");
list.remove("A");
out.print(list);
```

Question 16.

Consider the class `Uil` and the client code segment shown on the right. Which of the following would be the output of only line#1 once the client code has been executed?

- A) 0 0
- B) 2.0 0.5
- C) 2 0
- D) 22 6.5
- E) 1 1

Question 17.

Which of the following would be the output of only line#2 once the client code has been executed?

- A) 0
- B) 1
- C) 2
- D) 3
- E) **line#2** produces an error and there will be no output.

//Use the class shown here to answer

//questions 16 and 17.

```
public class Uil
{
    private int x;
    public int y;
    public static int z;

    public Uil(int x, int y) {
        this.x = x * 2;
        this.y = y / 2;
        System.out.println(x + " " + y);
        z++;
    }

    public int getX() {
        return x;
    }

    public int getZ() {
        return z;
    }
}
```

//client code

```
Uil one = new Uil(5,4);
Uil two = new Uil(10,9);
Uil three = new Uil(1,1); //line #1
System.out.println(one.getZ()); //line #2
```

Question 18.

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

- A) lemon 0
- B) lemon -1
- C) pear 1
- D) pearlimelemon 8
- E) There is no output due to an error.

```
String r = "pear";
String p = new String("lime");
String o = "lemon";
String s = r = p = o;
out.print(s + " ");
out.print(s.compareTo(o) + " ");
```

Question 19.

Consider the enumerated data type shown here:

```
public enum Day {
    SUNDAY, MONDAY, TUESDAY, WEDNESDAY,
    THURSDAY, FRIDAY, SATURDAY
}
```

Which of the code segments shown on the right will compile, execute and print a reaction to each day of the week?

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

I.

```
Day days[] = Day.values();
for(Day day: days) {
    switch(day) {
        case MONDAY:out.print("Boo!");break;
        case FRIDAY:out.print("TGIF!");break;
        case SATURDAY:
        case SUNDAY:out.print("Yeah!");break;
        default:out.print("Meh");
    }
}
```

II.

```
Day days[] = Day.values();
for(int i=0; i<days.length; i++) {
    switch(days[i]) {
        case MONDAY:out.print("Boo!");break;
        case FRIDAY:out.print("TGIF!");break;
        case SATURDAY:
        case SUNDAY:out.print("Yeah!");break;
        default:out.print("Meh");
    }
}
```

III.

```
for(Day d=Day.SUNDAY; d<Day.SATURDAY; d++)
{
    switch(d) {
        case MONDAY:out.print("Boo!");break;
        case FRIDAY:out.print("TGIF!");break;
        case SATURDAY:
        case SUNDAY:out.print("Yeah!");break;
        default:out.print("Meh");
    }
}
```

Question 20.

The formula to find the distance between two points is:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Which of the following statements will calculate the distance between points (x1,y1) and (x2,y2) and then assign that value to a variable named d.

- A) `double d = Math.sqrt(Math.pow(x2 - x1,2) + Math.pow(y2 - y1,2));`
- B) `double d = Math.sqrt(Math.sqr(x2 - x1) + Math.sqr(y2 - y1));`
- C) `double d = Math.sqrt(x2 - x1 * x2 - x1 + y2 - y1 * y2 - y1);`
- D) `double d = Math.sqrt(Math.pow(2,x2 - x1) + Math.pow(2,y2 - y1));`
- E) More than one of the above.

Question 21.

```

ArrayList<Integer> numbers = new ArrayList<Integer>();
numbers.add(5);
numbers.add(9);
numbers.add(8);
numbers.add(1);
<code>

```

Consider the code segment shown above. Which of the following can replace **<code>** to ensure that the code segment will print all the elements stored in `list`?

- A) `numbers.forEach(n -> System.out.println(n));`
- B) `numbers.forEach((n) -> System.out.println(n));`
- C) `numbers.forEach((n) -> {System.out.println(n);});`
- D) `numbers.forEach(n -> {System.out.println(n);});`
- E) All of the above.

Question 22.

Which of the following methods will return the index value of the last occurrence of `str` in `list` or `-1` if `str` is not in `list`?

A)

```

public static int lastOccurence(ArrayList
list, String str)
{
    int i = 0, j = -1;
    while(i < list.size()) {
        if(list.get(i).equals(str)) {
            j = i;
            return j;
        }
        i--;
    }
    return j;
}

```

C)

```

public static int lastOccurence(ArrayList
list, String str)
{
    int i = list.size() - 1, j = -1;
    while(i >= 0) {
        if(list.get(i).equals(str)) {
            j = i;
            return j;
        }
        i--;
    }
    return j;
}

```

B)

```

public static int lastOccurence(ArrayList
list, String str)
{
    int i = list.size(), j = -1;
    while(i >= 0) {
        if(list.get(i).equals(str)) {
            j = i;
            return j;
        }
        i--;
    }
    return j;
}

```

D)

```

public static int lastOccurence(ArrayList
list, String str)
{
    int i = list.size() - 1, j = -1;
    while(i >= 0) {
        if(list.get(i).equals(str))
            j = i;
        i--;
    }
    return j;
}

```

- E) More than one of the above.

Question 23.

Which of the following shows the output of the class InvB shown on the right?

- A) dog cat bird turtle pig rat dog dog
- B) cat bird turtle pig rat dog cat
- C) turtle pig rat dog bird
- D) cat dog rat pig turtle bird cat
- E) bird dog pig rat dog turtle

Question 24.

The class InvB implements a _____.

- A) queue
- B) map
- C) stack
- D) set
- E) priority queue

//Use the class shown here to answer // questions 23 and 24.

```
import java.util.*;
public class InvB {
    private ArrayList<String> list;

    public InvB() {
        list = new ArrayList<String>();
    }

    public void abc(String s) {
        list.add(s);
    }

    public String xyz() {
        return list.remove(0);
    }

    public String mno() {
        return list.get(0);
    }

    public boolean efg() {
        return list.size() == 0;
    }

    public static void main(String[] args) {
        InvB obj = new InvB();
        obj.abc("dog");obj.abc("cat");
        obj.abc("bird");obj.abc("turtle");
        obj.abc("pig");obj.abc("rat");
        obj.mno();obj.xyz();obj.abc("dog");
        obj.abc(obj.mno());
        while(!obj.efg())
            System.out.print(obj.xyz() + " ");
    }
}
```

Question 25.

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

- A) Geometry History Calculus PE Art
- B) Geometry History Calculus Art
- C) Art PE Geometry English
- D) Geometry English Calculus PE Art
- E) Tom Bill Carol Latisha

```
Map<String,String> m = new
TreeMap<String,String>();
m.put("Tom", "Art");
m.put("Bill", "Geometry");
m.put("Carol", "History");
m.put("Susan", "Geography");
m.put("Latisha", "Calculus");
m.put("Simon", "PE");
m.remove("Susan");
m.remove("Simon", "Art");
m.put("Carol", "English");
Set<String> s = m.keySet();
for(String key:s)
    System.out.print(m.get(key) + " ");
```

Question 26.

What is the output of this line of code?

```
out.print(15 * 2 / 5 >= 19 - 4 * 3 && 8 / 0 == 0 * 40);
```

- A) true
- B) false
- C) There is no output due to an error.

Question 27.

Which of the following must replace `<code>` to ensure that the method will compile and execute as intended?

- A) `j = k`
- B) `k = i`
- C) `k = 0`
- D) `j = i`
- E) `k = j`

Question 28.

Assume that `<code>` has been replaced with the correct code and this client code is executed.

```
int[] list = {6,7,1,8,2,9,0,5,4,3};
sort(list);
```

What is printed when `i` equals 4 if this line of code

```
out.println(Arrays.toString(list));
```

replaces the comment?

- A) [1, 2, 0, 5, 4, 3, 6, 7, 8, 9]
- B) [0, 1, 2, 3, 4, 9, 6, 5, 7, 8]
- C) [1, 6, 7, 8, 2, 9, 0, 5, 4, 3]
- D) [1, 2, 6, 7, 8, 9, 0, 5, 4, 3]
- E) [0, 1, 2, 3, 7, 9, 6, 5, 4, 8]

Question 29.

Once the method `sort` has been correctly implemented, what is the worst case time complexity for this method?

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

Question 30.

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

- A) eleven
twelve
thirteen
- B) thirteen
eleven
twelve
- C) eleven
eleven
eleven
- D) twelve
eleven
thirteen
- E) twelve
thirteen
eleven

```
//Use the following implementation of
//a selection sort to answer
//questions 27, 28 and 29.
```

```
public static void sort(int[] list)
{
    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])
                <code>;
        int t = list[i];
        list[i] = list[k];
        list[k] = t;
        //comment
    }
}
```

```
char[][] mat = new char[3][5];
String s1 = "eleven";
String s2 = "twelve";
String s3 = "thirteen";
mat[1] = s1.toCharArray();
mat[2] = s2.toCharArray();
mat[0] = s3.toCharArray();
mat[2] = mat[1];
mat[0] = mat[2];
for(char[] arr:mat) {
    for(int i = 0; i < arr.length; i++)
        out.print(arr[i]);
    out.println();
}
```

Question 31.

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

- A) 4
- B) 8
- C) 2
- D) 1
- E) 0

```
System.out.print(32>>3);
```

Question 32.

What is the average time complexity for insertion into a somewhat balanced binary search tree?

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

Question 33.

Which of the following is NOT a Java keyword?

- A) bit
- B) class
- C) package
- D) throw
- E) final

Question 34.

Which of the following is the correct method header for a method that returns a tip amount, given the amount of the check and the desired percent tip?

- A) AAAAAAAAAAAAAAAAAA
- B) AABBBABABAABABA
- C) BBBBBBBBBBBBBBBB
- D) BBBAABABABBABAB
- E) There is no output due to an error.

```
String s = "ComPUterSCIencE";
char[] list=s.toCharArray();
for(char a:list)
    out.print(a < 77 || a > 109?"A":"B");
```

Question 35.

Which of the following Boolean expressions is the correct representation of the truth table shown on the right?

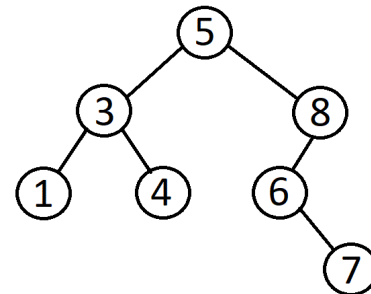
- A) $A + B * A$
- B) $A + B \oplus A$
- C) $A + \overline{B} \oplus \overline{A}$
- D) $\overline{A} + B \oplus A$
- E) $\overline{A} * \overline{B} \oplus A$

A	B	
T	T	T
T	F	F
F	T	T
F	F	T

Question 36.

Which of the following represents a post order traversal of the binary search tree shown on the right?

- A) 7 1 4 6 3 8 5
- B) 1 3 4 5 6 7 8
- C) 5 3 1 4 8 6 7
- D) 1 4 3 7 6 8 5
- E) 5 3 8 1 4 6 7



Question 37.

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

- A) 37
- B) 7
- C) 15
- D) 11
- E) 22

```
System.out.print(~-15 | 5);
```

Question 38.

Given the code segment shown on the right, what is the greatest possible number of elements that could be within Set s?

- A) 7
- B) 1000
- C) 1001
- D) 7000
- E) 6

```
Random r = new Random();  
Set<Integer> s = new TreeSet<Integer>();  
for(int n = 0; n <= 1000; n++)  
    s.add(r.nextInt(7));
```

Question 39.

Determine the output of the code segment shown on the right and write your answer in the blank provided on the answer document.

```
public static void main(String[] args) {  
    out.print(method(5));  
}  
  
public static int method(int i) {  
    if(i < 1)  
        return 10;  
    else if(i % 2 == 0)  
        return 2 * i + method(--i);  
    else  
        return 3 * i + method(--i);  
}
```

Question 40.

The expression shown here has been written using prefix notation. Evaluate the expression and write the value you get in the blank provided on the answer document. The operands are 4, 5, 12 and 2.

+ * 4 5 / 12 2

★ ANSWER KEY – CONFIDENTIAL ★

UIL COMPUTER SCIENCE – 2021 INVITATIONAL B

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

- | | | | |
|------------------|------------------|------------------|--------------------|
| 1) <u> A </u> | 11) <u> D </u> | 21) <u> E </u> | 31) <u> A </u> |
| 2) <u> B </u> | 12) <u> C </u> | 22) <u> C </u> | 32) <u> D </u> |
| 3) <u> C </u> | 13) <u> A </u> | 23) <u> B </u> | 33) <u> A </u> |
| 4) <u> D </u> | 14) <u> C </u> | 24) <u> A </u> | 34) <u> B </u> |
| 5) <u> A </u> | 15) <u> B </u> | 25) <u> D </u> | 35) <u> E </u> |
| 6) <u> A </u> | 16) <u> E </u> | 26) <u> B </u> | 36) <u> D </u> |
| 7) <u> E </u> | 17) <u> D </u> | 27) <u> E </u> | 37) <u> C </u> |
| 8) <u> C </u> | 18) <u> A </u> | 28) <u> B </u> | 38) <u> A </u> |
| 9) <u> E </u> | 19) <u> C </u> | 29) <u> E </u> | *39) <u> 49 </u> |
| 10) <u> E </u> | 20) <u> A </u> | 30) <u> C </u> | *40) <u> 26 </u> |

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

Note: Correct responses are based on **Java SE Development Kit 12 (JDK 12)** 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 12 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used.

Explanations:

1.	A	Each hex digit represents 4 binary digits. $8_{16} = 1000_2$ $0_{16} = 0000_2$ 10000000_2																																								
2.	B	$12 \% 5 - 7 \% 6 =$ $2 - 7 \% 6 =$ $2 - 1 =$ 1																																								
3.	C	\t is the escape sequence for a tab character.																																								
4.	D	The length of str is 9. $9 - 7 = 2$. charAt(2) returns 'm'.																																								
5.	A	!(false && true) = !(false) = true																																								
6.	A	Math.abs(-5.5) returns 5.5 then take the opposite to get -5.5 Math.round(-5.5) returns -5.0 and casting gets -5 Math.max(-5.5,-5) returns -5.0																																								
7.	E	$3.5 + 5 * 8.25 / 2 =$ $3.5 + 41.25 / 2 =$ $3.5 + 20.625 =$ 24.125																																								
8.	C	'm' is not less than 'c' so move on to the else. 'z' is greater than 'h' and 'h' is less than 'm' so print 'z'.																																								
9.	E	Print 35 with first iteration. $35 / 7 = 5$. Print 5. $5 / 7 = 0$. 0 is not greater than 0 stop the loop stops.																																								
10.	E	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">0</td> <td style="width: 20%;">1</td> <td style="width: 20%;">2</td> <td style="width: 20%;">3</td> <td style="width: 20%;">4</td> </tr> <tr> <td>0</td> <td>7</td> <td>5</td> <td>-1</td> <td>0</td> </tr> </table> <p>After first 3 lines are executed values[4] is 0 by default. values.length returns 5. $5 - 3 = 2$. Values[2] is 5. So values[0] = 5. Leaving:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">0</td> <td style="width: 20%;">1</td> <td style="width: 20%;">2</td> <td style="width: 20%;">3</td> <td style="width: 20%;">4</td> </tr> <tr> <td>5</td> <td>7</td> <td>5</td> <td>-1</td> <td>0</td> </tr> </table>	0	1	2	3	4	0	7	5	-1	0	0	1	2	3	4	5	7	5	-1	0																				
0	1	2	3	4																																						
0	7	5	-1	0																																						
0	1	2	3	4																																						
5	7	5	-1	0																																						
11.	D	The regular expression [ti] makes "t" or "i" delimiters. Delimiters are not printed																																								
12.	C	<table style="border-collapse: collapse;"> <tr> <td style="padding-right: 10px;">d</td> <td style="padding-right: 10px;">e</td> <td>f</td> </tr> <tr> <td>2</td> <td>1</td> <td>105</td> </tr> <tr> <td>2</td> <td>2</td> <td>100</td> </tr> <tr> <td>2</td> <td>4</td> <td>95</td> </tr> <tr> <td>2</td> <td>8</td> <td>90</td> </tr> <tr> <td>2</td> <td>16</td> <td>85</td> </tr> <tr> <td>2</td> <td>32</td> <td>80</td> </tr> <tr> <td>2</td> <td>64</td> <td>75</td> </tr> <tr> <td>2</td> <td>128</td> <td>70</td> </tr> </table>	d	e	f	2	1	105	2	2	100	2	4	95	2	8	90	2	16	85	2	32	80	2	64	75	2	128	70													
d	e	f																																								
2	1	105																																								
2	2	100																																								
2	4	95																																								
2	8	90																																								
2	16	85																																								
2	32	80																																								
2	64	75																																								
2	128	70																																								
13.	A	$x - z < y \ \&\& \ z - \text{Math.abs}(y - x) \leq z$ $3.0 - 0.5 < 2.75 \ \&\& \ 0.5 - \text{Math.abs}(2.75 - 3.0) \leq 0.5$ $3.0 - 0.5 < 2.75 \ \&\& \ 0.5 - \text{Math.abs}(-0.25) \leq 0.5$ $3.0 - 0.5 < 2.75 \ \&\& \ 0.5 - 0.25 \leq 0.5$ $2.5 < 2.75 \ \&\& \ 0.25 \leq 0.5$ true && true true																																								
14.	C	The minimum value that can be represented using the short data type (16 bits) is -32768.																																								
15.	B	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td></td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> </tr> <tr> <td></td> <td>"E"</td> <td>"C"</td> <td>"B"</td> <td>"D"</td> <td>"A"</td> <td></td> <td></td> </tr> <tr> <td>add(2,"F")</td> <td>"E"</td> <td>"C"</td> <td>"F"</td> <td>"B"</td> <td>"D"</td> <td>"A"</td> <td></td> </tr> <tr> <td>add(1,"A")</td> <td>"E"</td> <td>"A"</td> <td>"C"</td> <td>"F"</td> <td>"B"</td> <td>"D"</td> <td>"A"</td> </tr> <tr> <td>remove("A")</td> <td>"E"</td> <td>"C"</td> <td>"F"</td> <td>"B"</td> <td>"D"</td> <td>"A"</td> <td></td> </tr> </table>		0	1	2	3	4	5	6		"E"	"C"	"B"	"D"	"A"			add(2,"F")	"E"	"C"	"F"	"B"	"D"	"A"		add(1,"A")	"E"	"A"	"C"	"F"	"B"	"D"	"A"	remove("A")	"E"	"C"	"F"	"B"	"D"	"A"	
	0	1	2	3	4	5	6																																			
	"E"	"C"	"B"	"D"	"A"																																					
add(2,"F")	"E"	"C"	"F"	"B"	"D"	"A"																																				
add(1,"A")	"E"	"A"	"C"	"F"	"B"	"D"	"A"																																			
remove("A")	"E"	"C"	"F"	"B"	"D"	"A"																																				
16.	E	The variables x and y within the print statement in the Uil constructor are the formal parameters, not the fields with the same names.																																								
17.	D	z is a static field shared by all Uil objects and it is incremented each time that a Uil object is instantiated.																																								

18.	A	Assignment is applied from right to left. p gets o then r gets p then s gets r which means that s gets "lemon". s.compareTo(o) returns 0 because both store "lemon".
19.	C	The operator < is undefined for the enumerated type Day. So, code segment III will not compile.
20.	A	B. There is not a sqrt method. C. Math.sqrt(x2 - x1 * x2 - x1 + y2 - y1 * y2 - y1) uses the wrong order of operations. D. The arguments are in the wrong order in the call to the pow method.
21.	E	The syntax is correct in each of the statements. Parenthesis around the parameter are optional as are the curly braces around the output statement.
22.	C	A. Begins the search at the beginning of the list and decrements i to -1 on the first iteration of the loop. B. Begins the search at list.size() which throws an exception. D. Will return the first occurrence.
23.	B	Class InvB implements a queue. abc adds an element to the tail and xyz removes the element at the head. mno returns but does not remove the element at the head. efg returns true if the queue is empty.
24.	A	See #23
25.	D	Maps parameters are a key and a value. In this case the names are the keys and the subjects are the values. remove("Susan") removes Susan and Geography. remove("Simon", "Art) Removes the entry for the specified key only if it is currently mapped to the specified value. Simon is not removed because he is mapped to PE. TreeMaps are sorted on the keys.
26.	B	15 * 2 / 5 >= 19 - 4 * 3 && 8 / 0 == 0 * 40 6 >= 7 && 8 / 0 == 0 * 40 false && 8 / 0 == 0 * 40 Once the left operand is false the right hand operator is never evaluated and the division by zero error never occurs.
27.	E	k represents the smallest value left in the unsorted portion of the list.
28.	B	i = 0 [0, 7, 1, 8, 2, 9, 6, 5, 4, 3] i = 1 [0, 1, 7, 8, 2, 9, 6, 5, 4, 3] i = 2 [0, 1, 2, 8, 7, 9, 6, 5, 4, 3] i = 3 [0, 1, 2, 3, 7, 9, 6, 5, 4, 8] i = 4 [0, 1, 2, 3, 4, 9, 6, 5, 7, 8] i = 5 [0, 1, 2, 3, 4, 5, 6, 9, 7, 8] i = 6 [0, 1, 2, 3, 4, 5, 6, 9, 7, 8] i = 7 [0, 1, 2, 3, 4, 5, 6, 7, 9, 8] i = 8 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
29.	E	Nested loops that increment by one with each iteration typically execute in quadratic time.
30.	C	mat[2] = mat[1]; makes the last row "eleven" mat[0] = mat[2]; makes the first row "eleven" Which means that all three rows in mat are the same.
31.	A	32 >> 3 = 32 / 8 = 4
32.	D	Best and average case time complexity are both O(log n) in a well balanced tree. In a completely imbalanced tree (all right or all left) the complexity would be O(n).
33.	A	class, package, throw, and final are all keywords.
34.	B	The expression a < 77 a > 109 evaluates to true if a character is an uppercase letter that comes before M or is a lowercase letter that comes after m. Therefore, A is printed if the expression is true or prints B if not.
35.	E	NOT(A AND B) XOR A The expression only evaluates as false if XORs left operand is the same as the right. The only time that occurs is when A is true and B is false.
36.	D	Nodes are visited in a left, right, root fashion.
37.	C	The complement operator adds one and takes the opposite. ~-15 = 14 14 5 = 1110 ₂ 101 ₂ = 1111 ₂ = 15
38.	A	nextInt(7) returns a random whole number between and including 0 and 6. A set data structure will not allow the insertion of duplicate elements.

39.	49	$\text{method}(5) = 3 * 5 + \text{method}(4)$ $\text{method}(4) = 2 * 4 + \text{method}(3)$ $\text{method}(3) = 3 * 3 + \text{method}(2)$ $\text{method}(2) = 2 * 2 + \text{method}(1)$ $\text{method}(1) = 3 * 1 + \text{method}(0)$ $\text{method}(0) = 10$	$\text{method}(5) = 3 * 5 + 34 = 49$ $\text{method}(4) = 2 * 4 + 26 = 26$ $\text{method}(3) = 3 * 3 + 17 = 26$ $\text{method}(2) = 2 * 2 + 13 = 17$ $\text{method}(1) = 3 * 1 + 10 = 13$ $\text{method}(0) = 10$	
40.	26	$+ * 4 5 / 12 2 =$ $+ 20 / 12 2 =$ $+ 20 6 =$ 26		