

A+ Computer Science

M/C Written Test

General Directions:

- 1) DO NOT OPEN EXAM UNTIL TOLD TO DO SO.
- 2) **NO CALCULATORS of any kind may be used.**
- 3) You have 45 minutes to complete this contest. If you are in the process of actually writing an answer when the signal to stop is given, you may finish writing that answer.
- 4) Papers may not be turned in until forty-five minutes have elapsed. If you finish the test before the end of the allotted time, remain at your seat and retain your paper until told to do otherwise. You may use this time to check your answers.
- 5) All answers must be written on the answer sheet/Scantron card provided. Indicate your answers in the appropriate blanks provided on the answer sheet or on the Scantron card. Clean erasures are necessary for accurate Scantron grading.
- 6) You may place as many notations as you desire anywhere on the test paper except on the answer sheet or Scantron card which is reserved for answers only.
- 7) You may use additional scratch paper provided by the contest director.
- 8) All questions have ONE and only ONE correct (BEST) answer. There is a penalty for all incorrect answers. **All provided code segments are intended to be syntactically correct, unless otherwise stated (i.e. `error` is an answer choice). Ignore any typographical errors and assume any undefined variables are defined as used.**
- 9) A reference to commonly used Java classes is provided with the test and you may use this reference during the contest. You may detach the reference sheets from the test booklet but DO NOT DO SO UNTIL THE CONTEST BEGINS.
- 10) Assume that any necessary import statements for Standard Java 23 Packages and classes (e.g. `.lang`, `.util`, `System`, `Math`, `Double`, etc.) are included in any programs or code segments that refer to methods from these classes and/or packages.

Scoring:

- 1) All questions will receive 6 points if answered correctly; no points will be given or subtracted if unanswered; 2 points will be deducted for each incorrect answer.

For more Computer Science practice tests and materials,

go to www.apluscompsci.com

Standard Classes and Interfaces — Supplemental Reference**class java.lang.Object**

- o boolean equals(Object other)
- o String toString()
- o int hashCode()

interface java.lang.Comparable<T>

- o int compareTo(T other)
Return value < 0 if this is less than other.
Return value = 0 if this is equal to other.
Return value > 0 if this is greater than other.

class java.lang.Integer implements**Comparable<Integer>**

- o Integer(int value)
- o int intValue()
- o boolean equals(Object obj)
- o String toString()
- o int compareTo(Integer anotherInteger)
- o static int parseInt(String s)

class java.lang.Double implements**Comparable<Double>**

- o Double(double value)
- o double doubleValue()
- o boolean equals(Object obj)
- o String toString()
- o int compareTo(Double anotherDouble)
- o static double parseDouble(String s)

class java.lang.String implements**Comparable<String>**

- o int compareTo(String anotherString)
- o boolean equals(Object obj)
- o int length()
- o String substring(int begin, int end)
Returns the substring starting at index begin and ending at index (end - 1).
- o String substring(int begin)
Returns substring(from, length()).
- o int indexOf(String str)
Returns the index within this string of the first occurrence of str. Returns -1 if str is not found.
- o int indexOf(String str, int fromIndex)
Returns the index within this string of the first occurrence of str, starting the search at the specified index.. Returns -1 if str is not found.
- o charAt(int index)
- o int indexOf(int ch)
- o int indexOf(int ch, int fromIndex)
- o String toLowerCase()
- o String toUpperCase()
- o String[] split(String regex)
- o boolean matches(String regex)

class java.lang.Character

- o static boolean isDigit(char ch)
- o static boolean isLetter(char ch)
- o static boolean isLetterOrDigit(char ch)
- o static boolean isLowerCase(char ch)
- o static boolean isUpperCase(char ch)
- o static char toUpperCase(char ch)
- o static char toLowerCase(char ch)

class java.lang.Math

- o static int abs(int a)
- o static double abs(double a)
- o static double pow(double base, double exponent)
- o static double sqrt(double a)
- o static double ceil(double a)
- o static double floor(double a)
- o static double min(double a, double b)
- o static double max(double a, double b)
- o static int min(int a, int b)
- o static int max(int a, int b)
- o static long round(double a)
- o static double random()
Returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.

interface java.util.List<E>

- o boolean add(E e)
- o int size()
- o Iterator<E> iterator()
- o ListIterator<E> listIterator()
- o E get(int index)
- o E set(int index, E e)
Replaces the element at index with the object e.
- o void add(int index, E e)
Inserts the object e at position index, sliding elements at position index and higher to the right (adds 1 to their indices) and adjusts size.
- o E remove(int index)
Removes element from position index, sliding elements at position (index + 1) and higher to the left (subtracts 1 from their indices) and adjusts size.

class java.util.ArrayList<E> implements List<E>**class java.util.LinkedList<E> implements****List<E>, Queue<E>**

Methods in addition to the List methods:

- o void addFirst(E e)
- o void addLast(E e)
- o E getFirst()
- o E getLast()
- o E removeFirst()
- o E removeLast()

class java.util.Stack<E>

- o boolean isEmpty()
- o E peek()
- o E pop()
- o E push(E item)

interface java.util.Queue<E>

- o boolean add(E e)
- o boolean isEmpty()
- o E peek()
- o E remove()

class java.util.PriorityQueue<E>

- o boolean add(E e)
- o boolean isEmpty()
- o E peek()
- o E remove()

interface java.util.Set<E>

- o boolean add(E e)
- o boolean contains(Object obj)
- o boolean remove(Object obj)
- o int size()
- o Iterator<E> iterator()
- o boolean addAll(Collection<? extends E> c)
- o boolean removeAll(Collection<?> c)
- o boolean retainAll(Collection<?> c)

class java.util.HashSet<E> implements Set<E>

class java.util.TreeSet<E> implements Set<E>

interface java.util.Map<K,V>

- o Object put(K key, V value)
- o V get(Object key)
- o boolean containsKey(Object key)
- o int size()
- o Set<K> keySet()
- o Set<Map.Entry<K, V>> entrySet()

class java.util.HashMap<K,V> implements Map<K,V>

class java.util.TreeMap<K,V> implements Map<K,V>

interface java.util.Map.Entry<K,V>

- o K getKey()
- o V getValue()
- o V setValue(V value)

interface java.util.Iterator<E>

- o boolean hasNext()
- o E next()
- o void remove()

**interface java.util.ListIterator<E> extends
java.util.Iterator<E>**

Methods in addition to the Iterator methods:

- o void add(E e)
- o void set(E e)

class java.lang.Exception

- o Exception()
- o Exception(String message)

class java.util.Scanner

- o Scanner(InputStream source)
- o boolean hasNext()
- o boolean hasNextInt()
- o boolean hasNextDouble()
- o String next()
- o int nextInt()
- o double nextDouble()
- o String nextLine()
- o Scanner useDelimiter(String pattern)

Note: Correct responses are based on **Java SE Development Kit 23** 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 23 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 the expression $5_{12} + 8_{13}$?	
A. 13_{10} B. F_{16} C. 15_9 D. 1_{18} E. 21_{11}	
QUESTION 2	
What is output by the code to the right?	<code>out.println(17 / 2 * 3.0);</code>
A. 23.5 B. 25.5 C. 24.0 D. 25.0 E. There is no output due to an error.	
QUESTION 3	
What is output by the code to the right?	<code>double b=(int)2.17; int i=8; out.println(i + b);</code>
A. 10.17 B. 10 C. 11 D. 11.0 E. 10.0	
QUESTION 4	
What is output by the code to the right?	<code>String s = "apuluscompSci.com"; s = s.substring(13)+s.substring(0,3); out.println(s);</code>
A. omap1 B. comap1 C. comap D. omap E. There is no output due to a runtime error.	
QUESTION 5	
What is output by the code to the right?	<code>boolean a = true; boolean b = false; boolean c = a ^ b; out.print(a b & c);</code>
A. True B. False C. false D. true E. There is no output due to a syntax error.	
QUESTION 6	
Which of the following could NOT be output by the code to the right?	<code>int i=2; i+=(int) (Math.random()*9); i+=(int) (Math.random()*7); out.println(i);</code>
A. 17 B. 16 C. 2 D. 3 E. There is no output due to a compile error.	
QUESTION 7	
What is the output by the code to the right?	<code>int y = 8; if(y < 10) out.print(1); else if(y > 5) out.print(2); else out.print(3); out.print(4);</code>
A. 134 B. 14 C. 1 D. 234 E. 1234	

QUESTION 8 What is the output by the code to the right? A. 91 B. 95 C. 76 D. 88 E. There is no output due to an infinite loop.	<pre>int sum = 0; for(int i = 0; i < 14; i++) { sum += i; } out.println(sum);</pre>
QUESTION 9 What is output by the code to the right? A. 6 B. 14 C. 10 D. 5 E. There is no output due to an error.	<pre>out.println(3 * 4 ^ 7 + 2);</pre>
QUESTION 10 What is the output by the code to the right ? A. [13, 1, 12, 1] B. [12, 1, 14, 1] C. [13, 1, 14, 1] D. [13, 1, 13, 1] E. There is no output due to an error.	<pre>int[] i = new int[4]; i[0] = 13; i[1] = ++i[3]; i[2] = i[0] + i[1]; String outs = ""; outs = Arrays.toString(i); out.println(outs);</pre>
QUESTION 11 What is the output by the code to the right? A. 89 13 21 B. 54 -9 6 21 33 C. There is no output, and the code executes correctly. D. There is some output, followed by an error. E. There is no output due to an error.	<pre>String s="54 89 2 -9 13 6 21 32 33"; Scanner sc=new Scanner(s); while(sc.hasNext()) if(sc.nextInt()%3==0) out.print(sc.nextInt()+" ");</pre>
QUESTION 12 What is the output by the code to the right? A. 389 B. 350 C. 399 D. 375 E. There is no output due to a compile error.	<pre>String t = "APLUS"; int s = 0; for(int i = 0; i < t.length(); i++) { s = s + t.charAt(i); } out.println(s);</pre>
QUESTION 13 What is the order of precedence for the operators to the right ? A. I, IV, III, II B. IV, II, I, III C. I, IV, II, III D. IV, I, II, III E. IV, III, II, I	<pre>I. * II. < III. && (logical) IV. ++</pre>
QUESTION 14 What is the output by the code to the right ? A. 64 B. 16 C. 8 D. 4 E. 32	<pre>out.println(Short.SIZE);</pre>

<p>QUESTION 15</p> <p>What is the output by the code to the right ?</p> <p>A. [0, 1, 9, -17] B. [17, 0, 1, 212, 9, -17] C. [17, 0, 1, -17] D. There is no output due to a compile error. E. There is no output due to a runtime error.</p>	<pre>ArrayList<Integer> a; a=new ArrayList<Integer>(); a.add(17); a.add(212); a.add(9); a.set(1, 0); a.set(2, 1); a.add(-17); out.println(a);</pre>
<p>QUESTION 16</p> <p>What is output by the code to the right ?</p> <p>A. 20 B. 17 C. 21 D. 13 E. There is no output due to an error.</p>	<pre>int n = 13; String s = Integer.toString(n, 7); n = Integer.parseInt(s, 9); s = Integer.toOctalString(n); out.println(s);</pre>
<p>QUESTION 17</p> <p>What is output by the code to the right ?</p> <p>A. 12.0 B. 12.9 C. 12.899999999999999 D. There is no output due to a compile error. E. There is no output due to a runtime error.</p>	<pre>Double d=3; d*=4.3; out.println(d);</pre>
<p>QUESTION 18</p> <p>Which of the following can replace <1*> in the code to the right so that the code executes as intended?</p> <p>A. (int) (k) B. k C. k / 1 D. B and C. E. All of the above.</p>	<pre>double k=5; int y=8; y+=<1*>;</pre>
<p>QUESTION 19</p> <p>Assuming <?> is replaced with some integer value, what is the value of cnt at the end of the execution of the loop, in terms of x?</p> <p>A. $N \log_2(N)$ B. $N \log_3(N)$ C. $N^{(\log_3(N))}$ D. N^N E. N^3</p>	<pre>int x=<?>; int cnt = 1; for(int y = 1; y <= x; y *= 3) cnt *= x;</pre>
<p>QUESTION 20</p> <p>What is the worst-case runtime of an insertion sort?</p> <p>A. $O(N)$ B. $O(N \log N)$ C. $O(\log N)$ D. $O(N^2)$ E. None of the above.</p>	
<p>QUESTION 21</p> <p>What is the output by the code to the right?</p> <p>A. false true B. true true C. true false D. false false E. There is no output due to a runtime error.</p>	<pre>String s1 = "One of the Above"; String s2 = "A, B, or C"; String m = "[A-Z].{3,7} [A-Z]\\w*"; String res = "" + s1.matches(m); res += " " + s2.matches(m); out.println(res);</pre>

QUESTION 22

Which of the following could replace **<1*>** in the code to the right so that the Cat constructor works as intended?

- A. age = age; B. age = this.age;
 C. this.age = age; D. self.age = ag
 E. More than one of the above.

QUESTION 23

Which of the following could replace **<2*>** in the code to the right so that the Tiger constructor works as intended, and the noise instance variable of the parent class is set to "Roar"?

- A. super(a, "Roar")
 B. super(this.a, "Roar")
 C. super(a)
 D. noise = "Roar"
 E. noise = "Roar";
 age = a;

QUESTION 24

Assuming that **<1*>** and **<2*>** are filled correctly, what is the output by the line marked //q24 ?

- A. MeowMeow
 B. MeowRoar
 C. RoarMeow
 D. RoarRoar
 E. This is no output due to a runtime error.

QUESTION 25

Assuming that **<1*>** and **<2*>** are filled correctly, what is the output by the line marked //q25 ?

- A. 710
 B. 711
 C. 77
 D. 1111
 E. This is no output due to a runtime error.

QUESTION 26

What is output by the code to the right?

- A. 213
 B. 212
 C. 214
 D. This is no output due to a compile error.
 E. This is no output due to a runtime error.

```
class Cat{
    int age;
    String noise = "Meow";
    public Cat(int age, String n) {
        <1*>;
        noise = n;
    }
    public String yell() {
        return noise;
    }
    public int older() {
        return ++age;
    }
}

class Tiger extends Cat{
    public Tiger(int a) {
        <2*>;
    }
    public int older() {
        return super.older() +
               super.older();
    }
}

//////////client code//////////
Cat c = new Cat(6, "Meow");
Cat t = new Tiger(9);
out.println(c.yell()+t.yell()); //q24
t.older();
c.older();
out.println(""+c.age+t.age); //q25
```

```
int x = 212;
x++;
out.println(++x);
```

QUESTION 27 What is output by the code to the right? A. 20 B. 21 C. 15 D. 17 E. There is no output due to an error.	<pre>out.println (100 >> 5 + 2 17);</pre>
QUESTION 28 What is output by the line marked //q28? A. true B. false C. 8 D. 9 E. There is no output due to a runtime error.	<pre>Stack<Integer> s; s = new Stack<>(); s.push(76); s.push(129); s.push(3); s.push(1); s.push(-2); s.push(8); out.println(s.push(9)); //q28 s.peek(); s.peek(); out.println(s.pop()); //q29</pre>
QUESTION 29 What is output by the line marked //q29? A. -2 B. 1 C. 8 D. 9 E. There is no output due to a runtime error.	
QUESTION 30 What is the output by the code to the right? A. p B. r C. E D. B E. 0	<pre>String s = "practice.apluscompsci.com"; s = "BEST auto-graded UIL Prep"; out.println(s.charAt(0));</pre>
QUESTION 31 What could replace <1*> in the code to the right so that it compiles without error? A. long a = 54; B. byte a = 54; C. double a = 54.0; D. float a = 54.0; E. More than one of the above.	<pre><1*>; switch(a) { //implementation }</pre>

QUESTION 32

What is the output by //1 in the code to the right?

- A. true B. false
C. null D. -1
E. There is no output due to a syntax error.

```
PriorityQueue<Integer> r;
r = new PriorityQueue<>();
r.add(3);
r.add(-6);
r.add(11);
r.add(-9);
r.add(11);
r.add(2);
r.add(26);

out.println(r.add(4));    //1

r.remove();

out.println(r);    //2
```

QUESTION 33

What is the output by //2 in the code to the right?

- A. [-6, 2, 4, 3, 11, 11, 26]
B. [-6, 3, 2, 4, 11, 26, 11]
C. [-6, 3, 2, 11, 4, 11, 26]
D. [-6, 2, 3, 4, 11, 11, 26]
E. [-6, 3, 2, 4, 11, 11, 26]

QUESTION 34

What is the post-order traversal of the binary search tree created by inserting the list of values to the right in the order they appear?

- A. 13 2 -5 1 9 8 17 212 43 34 32 23 33 25
B. -5 1 2 8 9 13 17 22 23 25 32 33 34 212
C. 212 34 33 32 25 23 22 17 13 9 8 2 1 -5
D. 1 -5 8 9 2 22 25 23 32 34 43 212 17 13
E. 1 -5 22 25 9 8 33 34 2 43 32 25 23 212

13 2 17 -5 212 43 34 32 23 9 8 22 25 1

QUESTION 35

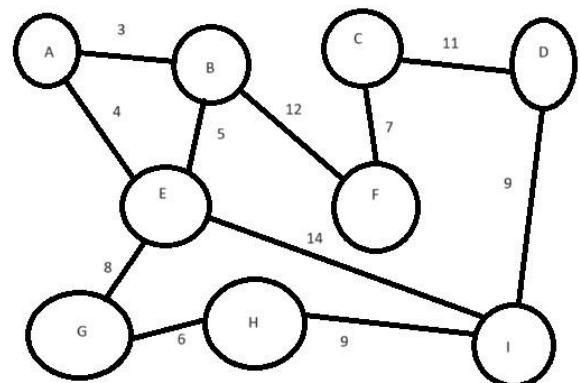
What is the diameter of the binary search tree created by inserting the list of values to the right in the order they appear?

- A. 10 B. 7 C. 3 D. 6 E. 11

QUESTION 36

What kind of graph is shown in the image to the right

- A. Connected Weighted Directed
B. Connected Weighted Undirected
C. Weighted Undirected
D. Unweighted Undirected
E. Weighted Directed


QUESTION 37

What is the cost of the shortest path between nodes A and C in the graph to the right ?

- A. 41 B. 28 C. 22 D. 38 E. 47

QUESTION 38

What is output by the code to the right?

- A. 12 21
- B. 14 21
- C. 14 23
- D. 14 19
- E. There is no output due to an infinite loop.

```
int i = 20, j = 0;
while (i >= j) {
    j++;
    i--;
    if(j % 7 < 4)j += 6;
    else j += 4;
    if(i % 10 < 5)i += 9;
    else i--;
}
out.println(i+" "+j);
```

QUESTION 39

What is the 8-bit two's compliment representation of the following value?

-71_{10}

QUESTION 40

What is the sum of all the values remaining in the stack after executing the following stack psuedocode?

```
push 9
push 8
push 17
pop X
push 212
push 2
pop X
pop X
push -7
pop X
push 4
push 43
pop X
```