

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 12 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 20 (JDK 20)** 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 20 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 number 37_{10} ?

- A. 32_4 B. 46_8 C. 212_4 D. 123_5 E. 31_{12}

QUESTION 2

What is output by the code to the right?

- A. 8 B. 2
C. 4 D. 0
E. There is no output due to an error.

```
out.println(56 % 3 * 4 / 3);
```

QUESTION 3

What is output by the code to the right?

- A. zzzElep B. Elephants
C. ELEPHANTS D. ELEP
E. zzzElephants

```
out.printf("%-3.4S", "Elephants");
```

QUESTION 4

What is output by the code to the right?

- A. lusaplu
B. ulpasul
C. compsci
D. apluscs
E. There is no output due to an error.

```
String s = "aplus";
String t = "compsci";
String r = s.substring(0,3);
r += s.substring(
    t.lastIndexOf(t.charAt(r.length())));
r =
    r.substring(2) + r.substring(0, 4);
out.println(r);
```

QUESTION 5

What is output by the code to the right?

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

```
boolean a = true;
boolean b = false;
a = a ^ a & b;
out.println(a);
```

QUESTION 6

What is output by the code to the right?

- A. 4.0 B. 5.0 C. 5.4 D. 4.0
E. There is no output due to an error.

```
double g = 5.4;
int n = 4;
out.println(Math.max(g, n));
```

QUESTION 7

What is the output by the code to the right?

- A. 14 B. 24
C. 23 D. 13
E. There is no output due to an error.

```
int g = 3;
if(g >= 3 && ++g < 7)
    out.print(1);
out.println(g);
```

<p>QUESTION 8</p> <p>What is the output by the code to the right?</p> <p>A. [1, 1, 1, 0] B. [2, 1, 1, 0] C. [2, 0, 1, 1] D. [2, 0, 1, 1] E. There is no output due to an error.</p>	<pre>int[] i = new int[4]; i[1] = i[2]+1; i[0] = i[1] + i[2]; i[2] = i[0]++; out.println(Arrays.toString(i));</pre>
<p>QUESTION 9</p> <p>What is output by the code to the right?</p> <p>A. 14 B. 12 C. 18 D. 13 E. There is no output due to an error.</p>	<pre>int sum = 0; for(int i = 1; i <= 8; i += 3) for(int j = i; j < 13; j +=2) sum++; out.println(sum);</pre>
<p>QUESTION 10</p> <p>What is output by the code to the right?</p> <p>A. 20 B. 23 C. 25 D. 28 E. There is no output due to an error.</p>	<pre>String str = ""; for(int s = 0; s < 4; s++) for(int t = s; t < 4; t++) str += "***"; out.println(str.length());</pre>
<p>QUESTION 11</p> <p>What is the output by the code to the right?</p> <p>A. 211 B. 213 C. 210 D. 212 E. There is no output due to an error.</p>	<pre>String t; t="23 51 46 78 9 64 37 84 29 14 dun"; Scanner s = new Scanner(t); int sum=0; while(s.hasNextInt()){ if(s.nextInt()%2==1) sum+=s.nextInt(); } out.print(sum);</pre>
<p>QUESTION 12</p> <p>What is the output by the code to the right?</p> <p>A. 22 B. 23 C. 17 D. 25 E. There is no output due to a compile error.</p>	<pre>out.println(2 ^ 17 5 ^ 19);</pre>
<p>QUESTION 13</p> <p>What is the correct order of precedence for the operators to the right ?</p> <p>A. II, I, III, IV B. I, II, III, IV C. III, IV, I, II D. I, IV, III, II E. II, I, IV, III</p>	<p>I. ++ (pre) II. ++ (post) III. * IV. + (additive)</p>
<p>QUESTION 14</p> <p>What is the output by the code to the right ?</p> <p>A. 6 B. 12 C. 10 D. 8 E. 16</p>	<pre>int[] s = new int[6]; s[0] = Double.BYTES; s[1] = Integer.BYTES; s[2] = Float.BYTES; s[3] = Long.BYTES; s[4] = Short.BYTES; Arrays.sort(s); out.println(s[2] + s[4]);</pre>

QUESTION 15

What is the worst case runtime of adding an item to the end of a single linked list?

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

QUESTION 16

Which of the following can replace **<*1>** in the code to the right so that the code segment compiles without error?

- I. 15.0
II. 15
III. 15.0f

- A. I only B. I and III only
C. II and III only D. I and II only
E. I, II, and III

```
ArrayList<Double> a;
a = new ArrayList<Double>();
a.add(<*1>);
```

QUESTION 17

What is output by the `a.foreach` command on the line marked //q17?

- A. 2 3 6 6 3 9 4 4 B. 2 6 6
C. 0 1 2 2 10 12 D. 1 7 5
E. There is no output due to an error.

```
ArrayList<Integer> a;
a = new ArrayList<Integer>();
for(int i = 1; i < 9; i ++)
```

```
a.add(i);

a.forEach((i) -> {
    int j = i % 2;
    i *= j == 0 ? 3 : 4;
    i = i / 4 > 4 ? i / 6 : i / 2;
    if(i % 5 < 3)
        out.print(i+" ");
}); //q17
```

QUESTION 18

What is output by the line marked //q18?

- A. [2, 3, 31, 6, -3, 8]
B. [2, 3, 31, 5, -3, 6, 8]
C. [2, 3, 31, 6, -3]
D. [2, 3, 31, 5, -3, 8]
E. There is no output due to an error.

```
a.removeIf(n -> (n % 3 == 1));
a.add(2, 31);
a.set(4, -3);
out.println(a); //q18
```

QUESTION 19

What is output by the code to the right?

- A. 0 B. -2
C. 2 D. Integer.MIN_VALUE
E. There is no output due to an error.

```
int g = Integer.MAX_VALUE;
g *= 2;
out.println(g);
```

QUESTION 20

Which of the following is not a valid java identifier?

- A. final B. hex6gon C. sept_ember D. ca\$hMone3y E. ____

QUESTION 21

Which sorting algorithm is implemented in the code to the right?

- A. rty!! B. Party
C. zza P D. !!!!!
E. There is no output due to an error.

```
String s = "Pizza Party";
s = s.concat("!!".repeat(3));
s = s.substring(8,13);
System.out.println(s);
```

<p>QUESTION 22</p> <p>What is output by the line marked <code>//q22</code> in the code to the right?</p> <p>A. 17 B. 13 C. 22 D. 19 E. 11</p>	
<p>QUESTION 23</p> <p>What is output by the line marked <code>//q23</code> in the code to the right?</p> <p>A. 76 B. 84 C. 77 D. 81 E. 72</p>	<pre>public static int recur(int a, int b) { if(a <= 0) return 1; if(a > b) return b + recur(b, a - b); else return b + recur(a, b - a); }</pre>
<p>QUESTION 24</p> <p>What is output by the line marked <code>//q24</code> in the code to the right?</p> <p>A. 33 B. 44 C. 37 D. 34 E. 23</p>	<pre>////////////////client code//////////////// out.println(recur(6, 12)); //q22 out.println(recur(24, 39)); //q23 out.println(recur(16, 5)); //q24</pre>
<p>QUESTION 25</p> <p>What is output by the line marked <code>//q25</code> in the code to the right?</p> <p>A. 17 B. 9 C. 12 D. 212 E. No output due to an error.</p>	<pre>LinkedList<Integer> lis; lis = new LinkedList<Integer>(); lis.add(9); lis.add(12); lis.add(212); lis.poll(); lis.add(17); lis.addAll(lis); lis.poll(); out.println(lis.poll()); //q25 lis.poll(); lis.pollLast(); out.println(lis); //q26</pre>
<p>QUESTION 26</p> <p>What is output by the line marked <code>//q26</code> in the code to the right?</p> <p>A. [12, 17] B. [9, 12] C. [12, 212] D. [17, 9] E. No output due to an error.</p>	<pre>String a = "J311Y B3AN5"; String b = "Big Boiz!!!"; String t = "\\w+ \\w+"; String r = "" + a.matches(t); t = "[A-z]{3}\\s[^#\$%]+"; r += " " + b.matches(t); out.println(r); //q27</pre>
<p>QUESTION 27</p> <p>What is output by the line marked <code>//q27</code> in the code to the right?</p> <p>A. true false B. false false C. false true D. true true E. No output due to an error.</p>	<pre>a = a.replaceAll("[0-9YN]+", "\\\$"); b = b.replaceAll("\\W*", "\\\$"); out.println((a + b).length()); //q28</pre>
<p>QUESTION 28</p> <p>What is output by the line marked <code>//q28</code> in the code to the right?</p> <p>A. 29 B. 16 C. 24 D. 21 E. No output due to an error.</p>	

QUESTION 29

What is the value of the post order expression to the right?

- A. 25 B. 34
C. 37 D. 26
E. 29

3 7 2 * 5 + 4 + +

QUESTION 30

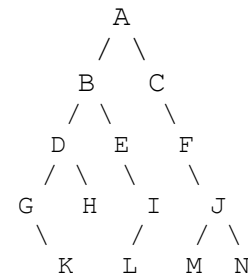
Which of the following data structures has the fastest worst-case big O runtime to remove an item?

- A. Hash Table
B. Binary Search Tree
C. Array
D. Stack
E. Red-Black Tree

QUESTION 31

What is the in-order traversal of the binary tree to the right?

- A. GKDHBELIACFMJN
B. ABDGKHEILCFJMN
C. KGHDLIEBMNJFCA
D. ABCDEFGHIJKLMN
E. CGHIBMKNLEFBAD


QUESTION 32

What could replace **<1*>** in the code to the right?

- A. 5.0f B. 5.0
C. 5 D. A and B.
E. All of the above.

```

PriorityQueue<Double> pq;
pq = new PriorityQueue<Double>();
pq.add(<1*>);
pq.add(4.7);
pq.add(3.9);
pq.add(0.0);
pq.add(4.4);
pq.add(1.5);
pq.add(2.12);
for(int y = 0; y < 3; y++)
    pq.poll();
out.println(pq);

```

QUESTION 33

Assuming **<1*>** is filled correctly with the value 5, what is output by the code to the right?

- A. [3.9, 4.7, 4.4, 5.0]
B. [5.0, 4.7, 4.7, 3.9]
C. [3.9, 4.4, 4.7, 5.0]
D. [5.0, 3.9, 4.4, 4.7]
E. There is no output due to an error.

QUESTION 34

How many swaps will be performed on the following list if a selection sort algorithm is used to sort it in ascending order?

[3, 7, 9, 12, 4, 1, 13, 85]

- A. 7 B. 3
C. 5 D. 6
E. 4

QUESTION 35

Which of the following could replace **<1*>** in the code to the right so that the Jaguar class is instantiated properly?

- A. `age = a;`
 `noise = "Rawr";`
- B. `super("Ra", a);`
 `super.noise = "Rawr";`
- C. `super("Ra", a);`
 `noise = "Rawr";`
- D. B and C.
- E. All of the above.

```
class Cat{
    String noise;
    int age;

    public Cat(String n, int a) {
        age = a;
        noise = n;
    }

    public void prowl() {
        out.println(noise + " " + age);
    }

    public int getOlder(int years) {
        age += years;
        return age;
    }
}
```

QUESTION 36

Assuming **<1*>** is filled correctly, what is output by the line marked //36 in the client code to the right?

- A. 13
- B. 15
- C. 2
- D. There is no output due to a compile error.
- E. There is no output due to a runtime error.

```
class Jaguar extends Cat{
    int spots;

    public Jaguar(int a, int s) {
        <1*>;
        spots = s;
    }
}

//////////client code//////////
Cat c = new Cat("Scratch", 13);
int age = c.getOlder(2);
out.println(age); //q36
```

QUESTION 37

Assuming **<1*>** is filled correctly, what is output by the line marked //37 in the client code to the right?

- A. Rawr
- B. Rawr 12
- C. Rawr 13
- D. There is no output due to a compile error.
- E. There is no output due to a runtime error.

```
Jaguar j = new Jaguar(12, 24);
j.prowl(); //q37
```

QUESTION 38

Assuming **<1*>** is filled correctly, what is output by the line marked //38 in the client code to the right?

- A. Rawr
- B. Rawr 12
- C. Rawr 13
- D. There is no output due to a compile error.
- E. There is no output due to a runtime error.

```
Cat cj = new Jaguar(13, 21);
cj.prowl(); //q38
```

QUESTION 40

© APlus CompSci, LLC