

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 (JDK 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 $32343_9 - 7777_8$?

- A. 17331_{10} B. $44B0_{16}$ C. 41640_8 D. 101243_7 E. 12023_{11}

QUESTION 2

What is output by the code to the right?

- A. 23 B. 25 C. 19 D. 33
E. There is no output due to an error.

```
out.println(1 + 2 * 7 + 4);
```

QUESTION 3

What is output by the code to the right (i's indicate blank spaces)?

- A. iiiHub
B. Hubiii
C. HubbaiBubba
D. iiiHubbaiBubba
E. HubbaiBubbaiii

```
String h = "Hubba Bubba";
out.printf("%6.3s", h);
```

QUESTION 4

What is output by the code to the right?

- A. raf
B. raff
C. Giraffe
D. There is no output due to a compile error.
E. There is no output due to a runtime error.

```
String s = "Giraffe";
s.substring(2, 6);
out.println(s);
```

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 & b | !b & a);
out.println(a);
```

QUESTION 6

What is output by the code to the right?

- A. 17 B. 16.0 C. 17.0 D. 13.0
E. There is no output due to a compile error.

```
int f = 5;
int g = 12;
double h = Math.hypot(f, g);
out.println(h);
```

QUESTION 7

What is output by the code to the right?

- A. 134 B. 13
C. 23 D. 234
E. 1234

```
int d = 4;
if(d++ == 4)
    out.print(1);
if(--d == 5)
    out.print(2);
    out.print(3);
if(d != 4)
    out.print(4);
```

<p>QUESTION 8</p> <p>What is output by the code to the right?</p> <p>A. -6 136 B. -21 131 C. -6 131 D. -21 136 E. -99 136</p>	<pre>int i = 0, j = 100, k = 1; int sum = 0; while(i < j) { sum += k + i++; j -= k; k *= 3; } out.println(j + " " + sum);</pre>
<p>QUESTION 9</p> <p>What is output by the code to the right?</p> <p>A. 256 B. 1434 C. 8602 D. 55 E. There is no output due to an error.</p>	<pre>int i = 89 & 64 + 12 35; i = i ^ 32 / 4 + 21 * 4; out.println(i);</pre>
<p>QUESTION 10</p> <p>What is output by the code to the right?</p> <p>A. 5 B. -6 C. -5 D. 7 E. There is no output due to an error.</p>	<pre>int[] arr = new int[] { 45, 3, -17, 97, -9, 123, 212, 7, 0, 17, 44, 55, -66 }; int h = Arrays.binarySearch(arr, 78); out.println(h);</pre>
<p>QUESTION 11</p> <p>What is output by the code to the right?</p> <p>A. Aty propesny approac B. At propeses qu C. Atpropesizeddestinyquickly D. Atpropeses E. There is no output due to an error.</p>	<pre>String g="At last my\nprophesized"; g+= " destiny\napproaches quickly"; Scanner sc=new Scanner(g); sc.useDelimiter("[g-m]+"); String o = sc.next(); sc.nextLine(); o += sc.next()+sc.next(); sc.nextLine(); sc.next(); o += sc.next(); out.println(o);</pre>
<p>QUESTION 12</p> <p>What is output by the code to the right?</p> <p>A. 21 B. 19 C. 23 D. 22 E. There is no output due to a compile error.</p>	<pre>String s = ""; for(int y = 0; y < 10; y++) s += "" + s.substring(y, s.length()-y); out.println(s.length());</pre>
<p>QUESTION 13</p> <p>What is the order of precedence for the operators to the right ?</p> <p>A. I, IV, III, II B. IV, II, I, III C. I, IV, II, III D. IV, II, III, I E. IV, III, II, I</p>	<p>I. *= II. <= III. & (bitwise) IV. -- (pre)</p>
<p>QUESTION 14</p> <p>What is output by the code to the right?</p> <p>A. 64 B. 16 C. 8 D. 4 E. 32</p>	<pre>out.println(Float.SIZE);</pre>

<p>QUESTION 15</p> <p>What is output by the code to the right?</p> <p>A. [One Time, [this], -212] B. [One Time, (this Collection), -212] C. [One Time, (a), -212] D. There is no output due to a compile error. E. There is no output due to a runtime error.</p>	<pre>ArrayList a = new ArrayList(); a.add("One Time"); a.add(a); a.add(-212); out.println(a);</pre>
<p>QUESTION 16</p> <p>What is output by the code to the right ?</p> <p>A. 2384 B. 11BA C. 47c D. 72E E. There is no output due to an error.</p>	<pre>int i = Integer.parseInt("3456", 8); String s = Integer.toString(i,14); i = Integer.parseInt(s, 11); s = Integer.toHexString(i); out.println(s);</pre>
<p>QUESTION 17</p> <p>What is output by the code to the right ?</p> <p>A. 2147483647 B. 0 C. -2147483648 D. There is no output due to an infinite loop. E. There is no output due to an error.</p>	<pre>int i = Integer.MIN_VALUE; int n = 0; i--; for(; i > 0; i--, n++); out.println(n);</pre>
<p>QUESTION 18</p> <p>What is output by the code to the right ?</p> <p>A. 1 B. 14 C. 12 D. 1234 E. There is no output due to an error.</p>	<pre>switch(5) { case 5:out.print(1); case 1:out.print(2);break; case 3:out.print(3);break; default:out.print(4); }</pre>
<p>QUESTION 19</p> <p>What is output by the code to the right ?</p> <p>A. True B. equal C. true D. false E. There is no output due to an error.</p>	<pre>StringBuffer a,b; a = new StringBuffer("equal"); b = new StringBuffer("equal"); out.println(a.equals(b));</pre>
<p>QUESTION 20</p> <p>What is the worst-case runtime of removing an item from a binary search tree?</p> <p>A. O(N) B. O(NlogN) C. O(logN) D. O(N^2) E. None of the above.</p>	
<p>QUESTION 21</p> <p>What is 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 h = "H3LL0 FR13ND5"; String s1 = "[0-Z]+ \\w+"; String s2 = "(\\d [A-Z]){9,13}"; String o = "" + h.matches(s1); o += " " + h.matches(s2); out.println(o);</pre>

QUESTION 22

Which of the following could replace **<1*>** in the code to the right so that the B constructor works as intended (num is initialized as n times 2)?

- A. `super(n * 2);` B. `super.num = n * 2;`
C. `super(n);` D. A and C.
`num = n * 2;` E. A and B.

QUESTION 23

Which of the following Java concepts is demonstrated by the add methods in the classes to the right?

- A. Overriding B. Overloading
C. Inheritance D. B and C.
E. All of the above.

QUESTION 24

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

- A. 6 18 15
B. 5 18 14
C. 6 18 8
D. 5 18 7
E. There is no output due to an error.

QUESTION 25

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

- A. 6 18 15
B. 6 18 8
C. 6 36 15
D. 6 36 8
E. There is no output due to an error.

QUESTION 26

Assuming that **<1*>** is filled correctly, what is the output by the line marked `//q26` ?

- A. 6 40 18
B. 7 40 18
C. 6 22 18
D. There is no output due to a compile error.
E. There is no output due to a runtime error.

```
class A{

    int num;

    public A(int n) {
        num = n;
    }

    public int add() {
        return num++;
    }

    public int getNum() {
        return num;
    }
}

class B extends A{

    public B(int n) {
        <1*>;
    }

    public int add(int n) {
        num += n;
        return num;
    }

    public int mult(int n) {
        num *= n;
        return num;
    }
}

//////////client code//////////
A a = new A(5);
B b = new B(6);
A c = new B(7);
String o = "";
o += a.add() + " ";
o += b.add(6) + " ";
o += c.add() + " ";
out.println(o); //q24
b.mult(2);
o = "" + a.getNum();
o += " " + b.getNum();
o += " " + c.getNum();
out.println(o); //q25
o = a.add() + " ";
o += b.add(4) + " ";
o += c.add(3) + " ";
out.println(o); //q26
```

QUESTION 27

Which of the following is not a legal Java instantiation?

- A. `Collection a = new LinkedList<String>();`
- B. `List b = new Stack<Integer>();`
- C. `Comparable c = new Float(5);`
- D. `ArrayList<String> d = new ArrayList();`
- E. All of the above are legal instantiations.

QUESTION 28

What is output by the line marked `//q28`?

- A. 212
- B. 278
- C. Output cannot be determined until runtime.
- D. There is no output due to a compile error.
- E. There is no output due to a runtime error.

```
public void inc(int y, int x) {
    for(int i = 0; i < x; i++)
        y += i;
}

public void inc(int[] y, int x) {
    for(int i = 0; i < x; i++)
        y[0] += i;
}
```

QUESTION 29

What is output by the line marked `//q29`?

- A. 136
- B. 0
- C. Output cannot be determined until runtime.
- D. There is no output due to a compile error.
- E. There is no output due to a runtime error.

```
////////////////client code////////////////
int q28 = 212;
inc(q28, 12);
out.println(q28); //q28
int[] q29 = new int[1];
inc(q29, 17);
out.println(q29[0]); //q29
```

QUESTION 30

Which line in the code to the right first causes an error?

- A. `//1`
- B. `//2`
- C. `//3`
- D. `//4`
- E. None of the lines cause an error.

```
short a, b, c;
a = 22;
b = 23;
if(a > 25) //1
    c = a + b; //2
if(a <= 25)
    c = a | b; //3
c += 5; //4
```

QUESTION 31

What is output by the code to the right

- A. `0nIn9l0l9dod129I911odo`
- B. `dod129I911odo51I1`
- C. `odo51I17I5I1I17I5Iodo51I`
- D. `llywood10nIn9l0l9dod129I9`
- E. There is no output due to an error.

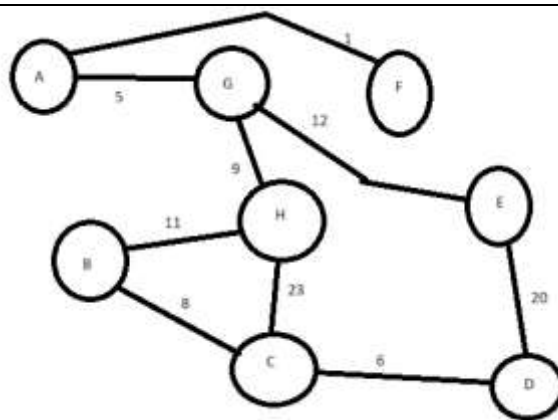
```
String s = "Once_Upon_A_";
s += "Time_In_Hollywood";
for(int y = 0; y < 5; y++) {
    s = s.substring(y + 3);
    s += y + s.charAt(y);
}
String fh = s.substring(s.length()/2);
s = s.substring(0, s.length()/2) + fh;
s = s.substring(1, s.length()-1);
s += s.charAt(s.length()/2 + 1);
s += s.charAt(s.length()/2 - 1);
s += s.charAt(s.length()/2);
}
out.println(s);
```


QUESTION 32	
<p>What could replace <1*> in the code to the right so that the Order interface is the child of the Comparable interface?</p> <p>A. implements B. inherits C. extends D. A or C. E. Any of the above.</p>	<pre>interface Order <1*> Comparable{ //implementation }</pre>
QUESTION 33	
<p>What is output by the code to the right?</p> <p>A. 4 B. 9 C. 6 D. 7 E. 5</p>	<pre>int g = 235; int i = Integer.bitCount(g); out.println(i);</pre>
QUESTION 34	
<p>What is output by the code to the right ?</p> <p>A. 1 1 B. 1 1.0 C. 1 2.0 D. 2.0 2.0 E. There is no output due to an error.</p>	<pre>Object o1, o2; if (true) o1 = new Integer(1); else o1 = new Double(2.0); o2 = true ? new Integer(1) : new Double(2.0); out.print(o1 + " "); out.println(o2);</pre>
QUESTION 35	
<p>What is output by the code to the right ?</p> <p>A. 917 B. 798 C. 812 D. 815 E. There is no output due to an error.</p>	<pre>int sum = 0; for(int i = 0; i < 14; i++) for(int ii = i; ii <= 15; ii++) for(int iii = ii; iii < 16; iii++) sum++; out.println(sum);</pre>
QUESTION 36	
<p>What is the sum of all the values remaining in the queue after the queue pseudocode to the right is complete?</p> <p>A. 20 B. -23 C. 17 D. 23 E. 62</p>	<pre>enqueue 45 enqueue 17 enqueue 212 dequeue X enqueue 78 enqueue 9 enqueue -3 dequeue X dequeue X enqueue 43 dequeue X enqueue -23 dequeue X dequeue X</pre>
QUESTION 37	
<p>What is the last value removed in the queue pseudocode to the right ?</p> <p>A. -3 B. 78 C. 43 D. 9 E. 17</p>	

QUESTION 38

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

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



QUESTION 39

What is the cost of the shortest path from node A to node D in the graph in question 38?

QUESTION 40

What is the 8-bit two's complement value corresponding to the following number?

-17_{10}