

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	
What is $5_7 + 14_5$?	
A. 17_{10} B. 32_4 C. 23_5 D. 210_3 E. 24_7	
QUESTION 2	
What is output by the code to the right?	<code>out.print(4 + 8);</code>
A. 17 B. 18 C. 12 D. 19 E. 13	
QUESTION 3	
What is output by the code to the right?	<code>out.print("ap"+"lus");</code>
A. ap+lus B. ap C. lus D. ap+lus E. No output due to runtime exception	
QUESTION 4	
What is output by the code to the right?	<code>String s = "ap+comp+sci.com"; out.println(s.contains("com"));</code>
A. true B. false	
QUESTION 5	
What is output by the code to the right?	<code>out.print(false false ^ true);</code>
A. true B. false	
QUESTION 6	
What is output by the code to the right?	<code>int b = (int)Math.ceil(4.4); out.println(b);</code>
A. 4 B. 4.4 C. 5 D. 5.0 E. No output due to runtime error	
QUESTION 7	
What is output by the code to the right?	<code>int x = 5; int y = 3; out.print(x + y * x);</code>
A. 40 B. 26 C. 10 D. 20 E. There is no output due to a syntax error.	

QUESTION 8

What is output by the code to the right?

- A. 3
- B. 57
- C. 5
- D. 357
- E. 7

```
int i=2;
int y = 0;
switch(i)
{
    case 2: y = 3;
    case 3: y = 5;
    default: y = 7;
}
out.print(y);
```

QUESTION 9

What is output by the code to the right?

- A. 0123 B. 01234
- C. 1234 D. 012345
- E. 12345

```
for( int x = 0; x < 5; x++ )
    out.print(x);
```

QUESTION 10

What is output by the code to the right?

- A. 3 B. 6
- C. 9 D. 12
- E. 15

```
int[] d = {3,6,9,12,15};
out.println(d[2]);
```

QUESTION 11

What is output by the code to the right?

- A. 2.0 B. 3
- C. 34 D. 1
- E. There is no output due to a runtime exception.

```
Scanner h = new Scanner("1 2 3 4");
h.next();
h.nextDouble();
out.print(h.nextInt());
```

QUESTION 12

What is output by the code to the right?

- A. p B. 21
- C. 17 D. 13
- E. 20

```
int p =0;
for( int x = 2; x < 7; x++ )
    p += x;
out.print(p);
```

QUESTION 13

What is output by the code to the right?

- A. 19 B. 109 C. 316 D. 379
- E. There is no output due to a syntax error.

```
String s = "" + 3 + "7" + 9;
out.println(s);
```

QUESTION 14

What is output by the code to the right?

- A. -2147483648 B. 2147483647
- C. 0 D. -1
- E. 1

```
float g = Float.MAX_VALUE+1;
out.println((int)g);
```

<p>QUESTION 15</p> <p>What is output by the code to the right?</p> <p>A. [3, 2] B. [2, 3]</p> <p>C. [3] D. [2]</p> <p>E. There is no output due to syntax error.</p>	<pre>List<Byte> b; b = new ArrayList<>(); b.add(3); b.add(2); out.print(b);</pre>
<p>QUESTION 16</p> <p>What is output by //1 in the code to the right?</p> <p>A. 4 B. 4.2</p> <p>C. 3 D. 3.1</p> <p>E. null</p>	<pre>Stack<Double> s; s = new Stack<>(); s.push(3.1); s.push(4.2);</pre>
<p>QUESTION 17</p> <p>What is output by //2 in the code to the right?</p> <p>A. 4 B. 4.2</p> <p>C. 3 D. 3.1</p> <p>E. null</p>	<pre>out.println(s.pop()); //1 s.push(7.3); s.push(8.5); out.println(s.size()); //2</pre>
<p>QUESTION 18</p> <p>What is output by //1 in the code to the right?</p> <p>A. 22 B. 2 C. 11 D. 1 E. 99</p>	<pre>Map<Integer, Integer> m; m = new HashMap<>(); m.putIfAbsent(99, 22);</pre>
<p>QUESTION 19</p> <p>What is output by //2 in the code to the right?</p> <p>A. 22 B. 2 C. 11 D. 1 E. 99</p>	<pre>out.print(m.putIfAbsent(99, 11)); //1 out.print(m.size()); //2</pre>
<p>QUESTION 20</p> <p>What is output by the code to the right?</p> <p>A. 0.5 B. -0.0 C. -0.5 D. 0.0 E. -1.0</p>	<pre>out.printf("%1.1f", -5 % 0.9);</pre>
<p>QUESTION 21</p> <p>What is output by the code to the right?</p> <p>A. 10 B. 14 C. 15 D. 12 E. 13</p>	<pre>int i = 27; while (--i + 5 >= 19) { i = ++i - 1; } out.println(i);</pre>
<p>QUESTION 22</p> <p>What is output by the code to the right?</p> <p>A. 6 B. 7 C. 9 D. 30</p> <p>E. No output due to runtime exception</p>	<pre>static int fun2(int a, int b) { if (a % b == 1) { return 6; } else if (b % a == 3) { return 7; } return 9; } //////////////////////////////////// // Client code out.println(fun2(9, 30));</pre>

QUESTION 23

What is output by the code to the right?

- A. hawkhawk
- B. hawkhawkhawk
- C. hawk
- D. hawkhawkhawkhawk
- E. hawkhawkhawkhawkhawk

```
static String fun3(String a, int b)
{
    if (b >= a.length()) {
        return a + a;
    }
    else {
        return a + a + a;
    }
}

////////////////////////////////////
// Client code
out.println(fun3("hawk", 5));
```

QUESTION 24

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

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

```
PriorityQueue<Integer> w;
w = new PriorityQueue<>();

w.add(3);
w.add(2);
w.add(-5);
w.add(33);
w.add(1);
```

QUESTION 25

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

- A. [2, 6, 3, 33]
- B. [2, 3, 33, 6]
- C. [2, 33, 6, 3]
- D. [3, 6, 33]
- E. [2, 3, 6, 33]

```
out.println( w.add(6) ); //1

w.remove();
w.remove();

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

QUESTION 26

Consider the three classes to the right. What is the output from the following code?

```
D x = new F();
x.fun2();
```

- A. HiWhatHuh?
- B. WhatHuh?Hi
- C. WhatHiHuh?
- D. WhatHuh?
- E. Huh?WhatHi

```
class D{
    void fun2() {
        out.print("Hi");
    }
}
class E extends D {
    void fun2() {
        out.print("What");
        super.fun2();
    }
}
class F extends E {
    void fun2() {
        super.fun2();
        out.print("Huh?");
    }
}
```

QUESTION 27

What is output by the code to the right?

- A. [8, 6, 4, 2, 0]
- B. [0, 0, 0, 0, 0]
- C. [0, 2, 4, 6, 8]
- D. [8, 6, 4, 2]
- E. [6, 4, 2, 0]

```
int[] ray = new int[5];
for (int i=ray.length-1; i>=0; --i)
{
    ray[ray.length-1-i]=i*2;
}
out.println(Arrays.toString(ray));
```

QUESTION 28

What is output by the code to the right?

- A. A1
- B. 66
- C. B
- D. 98
- E. 971

```
out.println('A'+1);
```

QUESTION 29

What is output by the code to the right?

- A. 53
- B. 51
- C. 48
- D. 54
- E. 44

```
public static int fun3(int a) {
    if (a < 9)
        return fun3(++a) + a;
    return a;
}

////////////////////////////////////
// client code
out.println(fun3(2));
```

QUESTION 30

What is output by the code to the right?

- A. 27
- B. 19
- C. 25
- D. 21
- E. 15

```
out.print(17^22&9|8);
```

QUESTION 31

What is output by the code to the right?

- A. 16f
- B. 567
- C. 27c
- D. 367
- E. 16

```
int x = Integer.parseInt("16f", 16);
int y = Integer.parseInt("27c", 15);
out.println(Math.max(x, y));
```

QUESTION 32

What is output by the code to the right?

- A. 1001110
- B. 0
- C. 10011100
- D. 100111
- E. 156

```
int x = 10010 >> 7;
out.print(Integer.toBinaryString(x));
```


QUESTION 33

What is the output from the code on the right if the following code replaces **<*** 1 ***>**?

```
out.println(s);
```

- A. [platypus, PLATYpus, chicken, Chicken, cHiCkEn, cHICKEN]
- B. [cHICKEN, cHiCkEn, Chicken, chicken, PLATYpus, platypus]
- C. [Chicken, PLATYpus, cHICKEN, cHiCkEn, chicken, platypus]
- D. [platypus, chicken, cHiCkEn, cHICKEN, PLATYpus, Chicken]
- E. No output due to runtime exception

```
class A implements Comparable<A> {
    private String s;
    public A(String s) {
        this.s = s;
    }
    public int compareTo(A other) {
        if (s.equals(other.s)) return 0;
        return getVal() > other.getVal()
            ? -1 : 1;
    }
    private int getVal() {
        int x = 0;
        for (char c: s.toCharArray())
            x += (int)c;
        return x;
    }
    public String toString() {
        return s;
    }
}
```

```
////////////////////////////////////
// Client Code
```

```
TreeSet<A> s = new TreeSet<>();
s.add(new A("Chicken"));
s.add(new A("chicken"));
s.add(new A("cHICKEN"));
s.add(new A("Chicken"));
s.add(new A("cHiCkEn"));
s.add(new A("platypus"));
s.add(new A("PLATYpus"));
```

<* 1 ***>**

QUESTION 34

What is the output from the code on the right if the following code replaces **<*** 1 ***>**?

```
out.println(s.floor(new A("ChihUaHua")));
```

- A. platypus B. Chicken
- C. cHiCkEn D. PLATYpus
- E. No output due to error

QUESTION 35

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

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

```
Queue<Integer> w;
w = new LinkedList<>();

w.add(3);
w.add(2);
w.add(-5);
w.add(33);
w.add(1);
```

QUESTION 36

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

- A. [-5, 33, 6, 1]
- B. [-5, 6, 1, 33]
- C. [1, -5, 33, 6]
- D. [-5, 6, 33, 1]
- E. [-5, 33, 1, 6]

```
out.println( w.add(6) ); //1

w.remove();
w.remove();

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

QUESTION 37

Which Java bitwise operator will divide a number by 2?

QUESTION 38

What is the ASCII value of lowercase 'd'?

QUESTION 39

Which sorting algorithm does `Collections.sort()` use?

QUESTION 40

What is the worst case runtime to find a value in a non-balanced binary search tree?