

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 343_5 ?

- A. 1201_4 B. 144_8 C. 62_{16} D. 122_9 E. $9A_{11}$

QUESTION 2

What is output by the code to the right?

- A. 3 B. 81
C. 35 D. 0
E. There is no output due to an error.

```
out.println(212 / 7 + 89 / 5 * 3);
```

QUESTION 3

What is output by the code to the right?

- A. AnakinObiWan B. ObiWanAnakin
C. AnakinAnakinObiWan D. ObiWanObiWanAnakin
E. There is no output due to an error.

```
String a = "Anakin";
String b = "ObiWan";
out.printf("%2$s%1$s", a, b);
```

QUESTION 4

What is output by the code to the right?

- A. t does it mea
B. t does it me
C. oes it mean?
D. What does it mean?
E. There is no output due to an error.

```
String s = "What does it mean";
s += s.concat("?");
s = s.substring(3, 15);
out.println(s);
```

QUESTION 5

What is output by the code to the right?

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

```
boolean a = true;
boolean b = true;
boolean c = !!!b;
boolean d = !!!a;
out.println(c ^ d);
```

QUESTION 6

What is output by the code to the right?

- A. 5.0 B. 5 C. 6.3 D. 6.0
E. There is no output due to an error.

```
double a = 6.3;
double b = Math.max(5, a);
out.println(b);
```

QUESTION 7

What is the output by the code to the right?

- A. 123 B. 13
C. 23 D. 3
E. There is no output due to an error.

```
if(6 < 7)
    out.print(1);
else if(5 < 7)
    out.print(2);
out.print(3);
```

| | |
|--|---|
| <p>QUESTION 8</p> <p>What is the output by the code to the right?</p> <p>A. 99 B. 104 C. 88 D. 96 E. There is no output due to an error.</p> | <pre>out.println (('f' - 'z') * ('A' - '9'));</pre> |
| <p>QUESTION 9</p> <p>What is output by the code to the right?</p> <p>A. 9 B. 6 C. 12 D. 19 E. There is no output due to an error.</p> | <pre>int[] i = new int[] {5, 8, 0, 4, 1, 3, 6, 2, 7}; i[i[i[i[3]]]] += ++i[i[2]] + i[i[i[4]]]++; i[i[i[i[7]]]] = i[i[i[4]]]-- - i[i[i[0]]]; i[i[i[4]]] -= i[i[5]]; out.println(i[5] + i[i[5]] - i[i[i[5]]]);</pre> |
| <p>QUESTION 10</p> <p>What is output by the code to the right?</p> <p>A. 78 B. 90 C. 86 D. 91 E. There is no output due to an error.</p> | <pre>int sum = 0; for(int y = 0; y < 7; y++) for(int x = 0; x < y * y; x++) sum++; out.println(sum);</pre> |
| <p>QUESTION 11</p> <p>What is the output by the code to the right?</p> <p>A. Nine B. Eight C. NineFive D. There is no output due to a compile error. E. There is no output due to a runtime error.</p> | <pre>String s = "Nine Eight Seven Five"; Scanner sc = new Scanner(s); while(sc.next().length() %2 == 0) out.print(sc.next());</pre> |
| <p>QUESTION 12</p> <p>What is the output by the code to the right ?</p> <p>A. 114 B. 166 C. 35 D. 115 E. There is no output due to an error.</p> | <pre>out.println(212 ^ 167 84 & 122);</pre> |
| <p>QUESTION 13</p> <p>What is the correct order of precedence for the operators to the right ?</p> <p>A. II, III, IV, I B. I, II, III, IV C. III, IV, I, II D. I, III, II, IV E. III, II, IV, I</p> | <p>I. + II. >> III. < IV. !=</p> |
| <p>QUESTION 14</p> <p>What is the output by the code to the right ?</p> <p>A. 4 B. 32 C. 64 D. 8 E. 16</p> | <pre>out.println(Short.SIZE);</pre> |

QUESTION 15

Which of the following is not a valid java identifier?

- A. Th3n B. lower C. \$\$\$\$ D. one_two E. f33d5

QUESTION 16

What is output by the code to the right?

- A. 127 128 65533 130 131
B. 127 128 129 130 131
C. 127 8364 65533 8218 402
D. 127 128 129 130 131 132
E. There is no output due to an error.

```
byte bytes[] = new byte[5];
for(int i = 127; i < 132; i++)
    bytes[i - 127] = (byte)i;
String str = new String(bytes);
for(int i = 0; i < 5; i++)
    out.print((int)str.charAt(i) + " ");
```

QUESTION 17

What is output by the code to the right?

- A. 1274 B. 1
C. 1146 D. 0
E. There is no output due to an error.

```
int sum = 1;
for(int y = 0; y < 56; y++)
    for(int x = 23; x < y; x += 3)
        for(int z = 1; z < x + y; z *= 2)
            sum = sum++;
out.println(sum);
```

QUESTION 18

What is output by the code to the right?

- A. Cat == Dog? true
B. Cat == Dog? false
C. true
D. false
E. There is no output due to an error.

```
String cat = "length: 10";
String dog = "length: " + cat.length();
out.println("Cat == Dog? " + cat == dog);
```

QUESTION 19

What is output by the code to the right?

- A. *u\$Ba\$*Hu\$mb\$
B. u\$mbu\$Ba\$\$*H
C. \$*Hu\$mbu\$\$Ba\$
D. u\$mbu\$Ba\$\$*H%
E. There is no output due to an error.

```
String s = "Bah Humbug";
for(int c = 0; c < 6; c++) {
    int g = s.length() + c / 3;
    s = '$' + s.substring(c) +
                                   s.substring(0, c);

    s += s;
    s = s.substring(s.length()/2) +
                                   s.substring(0, s.length()/2);
    s = s.substring(c, s.length() - 2) + '%';
    s = s.substring(0, g);
}
out.println(s.replaceAll("\\s+", ""));
```

QUESTION 20

What is the best case runtime of a quick sort algorithm?

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

QUESTION 21

What is output by the code to the right?

- A. 29 B. 27
C. 8 D. 28
E. There is no output due to an error.

```
int vals[] =
    {321, 234, 543, 456, 032, 154, 230, 32};
Set<Integer> diffs = new
    HashSet<Integer>();
for (int i = 0; i < vals.length; i++)
    for (int j = i; j < vals.length; j++)
        diffs.add(vals[i] - vals[j]);
out.println(diffs.size());
```

QUESTION 22

What could replace the **<1*>** in the code to the right so that the Lizard class works as intended?

- A. `weight = w; name = "Lizard"; age = 0;`
- B. `super(w, "Lizard")`
- C. `Reptile(w, "Lizard")`
- D. `super()`
- E. A and B.

QUESTION 23

What could replace the **<2*>** in the code to the right so that the `getOld` method of the Snake class works as intended, adding two times the value of `w` to the `weight`, and one to the `age` variables?

- A. `super.getOld(w * 2)`
- B. `return super.getOld(w * 2)`
- C. `weight += w * 2`
- D. `return getOld(w * 2)`
- E. More than one of the above.

QUESTION 24

Assuming **<1*>** and **<2*>** are filled in correctly, what is output by the line marked `//q24` in the code to the right?

- A. 2:2
- B. 1:1
- C. 1:2
- D. 2:1
- E. There is no output due to an error.

QUESTION 25

Assuming **<1*>** and **<2*>** are filled in correctly, what is output by the line marked `//q25` in the code to the right?

- A. Lizard:73:7 Snake:128:7
- B. Lizard:73:7 Snake:68:7
- C. Lizard:73:14 Snake:68:7
- D. Lizard:73:14 Snake:128:7
- E. There is no output due to an error.

QUESTION 26

Assuming **<1*>** and **<2*>** are filled in correctly, what is output by the line marked `//q26` in the code to the right?

- A. `lizard noise idklizard noise idk`
- B. `hisshiss`
- C. `lizard noise idkhiss`
- D. `hisslizard noise idk`
- E. There is no output due to an error.

```
abstract class Reptile{
    int age, weight;
    String name;
    public Reptile(int w, String n) {
        weight = w;
        name = n;
        age = 0;
    }
    public int getOld(int w) {
        weight += w;
        return ++age;
    }
    public String toString() {
        return name+": "+weight+": "+age;
    }
    abstract String sound();
}
class Lizard extends Reptile{
    public Lizard(int w) {
        <1*>;
    }
    public String sound() {
        return "lizard noise idk";
    }
    public int getOld(int w) {
        age++;
        return super.getOld(w);
    }
}
class Snake extends Reptile{
    public Snake(int w) {
        super(w, "Snake");
    }
    public String sound() {
        return "hiss";
    }
    public int getOld(int w) {
        <2*>;
    }
}
//////////client code//////////
Lizard a = new Lizard(12);
Reptile b = new Snake(8);
out.println
    (a.getOld(4)+":"+b.getOld(3)); //q24
for(int y = 2; y < 20; y += 3) {
    a.getOld(y);
    b.getOld(y);
}
out.println(a+" "+b); //q25
out.println(a.sound()+b.sound()); //q26
```

QUESTION 27

What is output by the code to the right?

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

```
String a = "ABC ABC ABC ABC";
String b = "ABC ABD ABE ABF";
String m =
    "\\w\\w\\w (\\1) (\\1) (\\1)";
out.println
    (a.matches(m) + " " + b.matches(m));
```

QUESTION 28

What is output by the code to the right?

- A. 0
- B. The maximum value of an integer.
- C. There is no output due to an infinite loop.
- D. There is no output due to a runtime error.
- E. There is no output due to a syntax error.

```
int i = 0;
while (-1 << i != 0)
    i++;
out.println(i);
```

QUESTION 29

What is the minimum number of levels in a binary search tree with 112 distinct elements?

- A. 9
- B. 7
- C. 8
- D. 5
- E. 6

QUESTION 30

What is output by the line marked //q30 in the code to the right?

- A. 31
- B. 37
- C. 45
- D. 32
- E. There is no output due to an error.

```
public int recur(int a, int b) {
    if(a < 0 || b < 0) return Math.abs(a-b);
    if(a % 3 == 0) {
        if(b % 5 == 2)
            return 4 + recur(b - 2, a - 1);
        else
            return 2 * recur(a - 3, b - 4);
    }
    else if(b % 2 == 0) {
        if(a % 2 == 0)
            return 10 + recur(a - 5, b);
        else
            return 3 * recur(a, b - 5);
    }
    return 1 + recur(a - 1, b - 1);
}
//////////client code//////////
out.println(recur(10, 11)); //q30
out.println(recur(20, 23)); //q31
```

QUESTION 31

What is output by the line marked //q31 in the code to the right?

- A. 397
- B. 494
- C. 448
- D. 415
- E. There is no output due to an error.

| | |
|--|--|
| <p>QUESTION 32</p> <p>What is output by the code to the right?</p> <p>A. 3 B. 4 C. 5 D. 6 E. There is no output due to an error.</p> | <pre>static boolean mystery(int number) { int i, m = 0, flag = 0; m = number/2; if(number == 0 number == 1) return false; else{ for(i = 2; i <= m ;i++) if(number%i == 0){ flag=1; return false; } if(flag == 0) return true; } return false; } public int hmmm(int a, int b) { int count = 0; for(int n = a; n <= b - 2; n++) { int m = n + 2; if(mystery(m) && mystery(n)) count++; } return count; } //////////client code////////// out.println(hmmm(1, 42));</pre> |
| <p>QUESTION 33</p> <p>What is the purpose of the hmmm method to the right?</p> <p>A. Determine if the difference between a and b is a prime number. B. Determine the number of primes between a and b. C. Determine the number of twin primes between a and b. D. Determine if a and b are twin primes. E. Determine how many number between a and b are not prime.</p> | <pre>static boolean mystery(int number) { int i, m = 0, flag = 0; m = number/2; if(number == 0 number == 1) return false; else{ for(i = 2; i <= m ;i++) if(number%i == 0){ flag=1; return false; } if(flag == 0) return true; } return false; } public int hmmm(int a, int b) { int count = 0; for(int n = a; n <= b - 2; n++) { int m = n + 2; if(mystery(m) && mystery(n)) count++; } return count; } //////////client code////////// out.println(hmmm(1, 42));</pre> |
| <p>QUESTION 34</p> <p>What is output by the code to the right?</p> <p>A. 6565 B. A65 C. 65A D. AA E. There is no output due to an error.</p> | <pre>char A = 'A'; int B = 0; out.print(true ? A : 0); out.print(false ? B : A);</pre> |
| <p>QUESTION 35</p> <p>What is output by the code to the right?</p> <p>A. 0 B. 0.8999999999999999 C. 0.9 D. There is no output due to a compile error. E. There is no output due to a runtime error.</p> | <pre>out.println(2.00 - 1.10);</pre> |
| <p>QUESTION 36</p> <p>What is output by the code to the right?</p> <p>A. 0 B. 99 C. 73 D. 48 E. There is no output due to an error.</p> | <pre>int num = 0; for(int y = 0; y < 100; y++) { for(int x = y; x > num; x--) if(y % x == 0) num++; num--; } out.println(num);</pre> |

| | |
|---|--|
| <p>QUESTION 37</p> <p>What is output by the code to the right?</p> <p>A. false false</p> <p>B. false true</p> <p>C. true false</p> <p>D. true true</p> <p>E. There is no output due to an error.</p> | <pre>String h = "Hello Darkness"; String g = "My Old Friend."; String s = "\\w{2,6} \\S+.*"; out.println (h.matches(s)+" "+g.matches(s));</pre> |
| <p>QUESTION 38</p> <p>What is output by the code to the right?</p> <p>A. eekeSkywalkerLukkkkkkekkeeeke</p> <p>B. eekeSkywalkerLukkkkkkekkeeekeeeeeekeSeeekeSk</p> <p>C. eekeSkywalkerLukkkkkkekkeeekeeeeeekeS</p> <p>D. SeeeSkywalkerLueeeeeeeeeeeeeeeSeeeeS</p> <p>E. There is no output due to an error.</p> | <pre>String s = "SkywalkerLuke"; for(int i = 0; i < 7; i++) { int j = s.length(); int k = j / 2; s += s; s += s.charAt(j); s = s.substring(k) + s.substring(0, k); s = s.substring(k, j + k + i); } out.println(s);</pre> |
| <p>QUESTION 39</p> <p>What is the sum of all the popped items in the following stack pseudocode?</p> <pre>Push 5 Push 4 Push 9 Pop X Push 11 Pop X Pop X Push 12 Pop X Push 8 Push 9 Pop X</pre> | |
| <p>QUESTION 40</p> <p>What is the value of the following post-fix expression?</p> <pre>212 120 - 3 * 4 / 5 6 + * 10 /</pre> | |