

University Interscholastic League

Computer Science Competition

Number 137 (Invitational A - 2013)

General Directions:

- 1) DO NOT OPEN EXAM UNTIL TOLD TO DO SO.
- 2) **NO CALCULATOR OF ANY KIND MAY BE USED.**
- 3) There are 40 questions on this contest exam. 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 45 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. 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, but not on the answer sheet or Scantron card which are 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. Ignore any typographical errors and assume any undefined variables are defined as used.**
- 9) A reference to commonly used Java classes is provided at the end of the test, and you may use this reference sheet 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 packages and classes (e.g. `.util`, `ArrayList`, etc.) are included in any programs or code segments that refer to methods from these classes and 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 an incorrect answer.

QUESTION 1

What is the sum of 267_8 and 342_8 ?

- A. 630_8 B. 610_8 C. 531_8 D. 611_8 E. 631_8

QUESTION 2

What is output by the code to the right?

- A. aaa B. bbb C. aaaa
D. bbbb E. abaab

```
System.out.print(
    "abaab".substring(2).replace('b','a'));
```

QUESTION 3

How many combinations of values for the boolean variables p , q , and r will result in s being set to true?

- A. 6 B. 10 C. 3
D. 7 E. 4

```
boolean p, q, r;
//code to initialize p, q, and r

boolean s = p || !(q && r);
```

QUESTION 4

What is the value of the expression to the right?

- A. h B. i C. n
D. f E. g

```
"huffington".charAt(5)
```

QUESTION 5

What is output by the code to the right?

- A. 1 B. 3.0 C. 4
D. 3 E. 0

```
System.out.print(25%3/2*2);
```

QUESTION 6

What is output by the code to the right?

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

```
int y = 7;
int x = 5;
if(x < 5 || y > 8)
    x++;
if(y <= 7)
    y += 2;
else
    x += 3;
System.out.print(x + " " + y);
```

<p>QUESTION 7</p> <p>What is output by the code to the right?</p> <p>A. 1 B. 339</p> <p>C. 3345 D. 1239</p> <p>E. There is no output due to a syntax error in the client code.</p>	<pre>System.out.print(1+2+"3"+4+5);</pre>
<p>QUESTION 8</p> <p>What is output by the code to the right?</p> <p>A. 6 B. 127.5 C. 14</p> <p>D. 127 E. 15</p>	<pre>int x = 85; int y = 2; x /= y*3; System.out.print(x);</pre>
<p>QUESTION 9</p> <p>What is output by the code to the right, where e represents a blank space in the output below?</p> <p>A. ee12abc 450 B. ee12abc ee450 C. 12eeabc ee450</p> <p>D. 12eeabc 450 E. 12eeabcee450</p>	<pre>int x = 12; int y = 450; System.out.printf("%-4d%3s%n%5d", x, "abc", y);</pre>
<p>QUESTION 10</p> <p>What is output by the code to the right?</p> <p>A. 12 B. 0 C. 8</p> <p>D. 13 E. 7</p>	<pre>int x = 0; for(int i = -3; i < 10; i += 2) x++; System.out.print(x);</pre>
<p>QUESTION 11</p> <p>What is output by the following code?</p> <pre>System.out.print(Math.min(30, Math.pow(2, 5)));</pre> <p>A. 30 B. 25 C. 32 D. 30.0 E. 32.0</p>	

GO ON TO THE NEXT PAGE.

QUESTION 12

What is output by the code to the right?

- A. 7 14 B. 6 15 C. 3 30
 7 14 7 14 6 15
- D. 3 30 E. 7 14
 3 30 6 15

```
public class Stuff {
    public static void main(String[] args) {
        int x = 6;
        int y = 15;
        one(y, x);
        System.out.println(x + " " + y);
    }

    public static void one(int x, int y) {
        x = x/2;
        y = two(x);
        System.out.println(x + " " + y);
    }

    public static int two(int x) {
        x *= 2;
        return x;
    }
}
```

QUESTION 13

What is output by the code to the right?

- A. 0 B. 1 C. 0.0
- D. 1.0 E. There is no output due to a runtime error.

```
int[] arr = new int[10];
arr[1]++;
System.out.println(arr[1]);
```

QUESTION 14

What is output by the statement labeled line 1 in the client code to the right? Assume that the statement labeled line 2 has been removed.

- A. hihi B. hellohello
- C. hi D. There is no output due to a compilation error.
- E. hello

```
public static String met(String s, int y){
    String t = "";
    for(int i = 1; i <= y; i++) {
        t += s;
    }
    return t;
}
```

Assume that the statement labeled line1 has been removed.

QUESTION 15

What is output by the statement labeled line 2 in the client code to the right?

- A. hihi B. Hellohello
- C. hi D. There is no output due to a runtime error.
- E. hello

```
// Client code
String str = "hello";
String s = "hi";
System.out.println(str.met(s, 2)); // line1
System.out.println(met(s, 2)); // line 2
```

QUESTION 16

What is output by the statement labeled line 1 in the client code to the right?

- A. 0 B. 1 C. 2
- D. 3 E. The output will vary from one run of the program to the next.

QUESTION 17

What is output by the statement labeled line 2 in the client code to the right?

- A. 0 B. 1 C. 2
- D. 3 E. The output will vary from one run of the program to the next.

```
public class Animal {
    public static final int MAMMAL = 0;
    public static final int AMPHIB = 1;
    public static final int BIRD = 2;

    private int anim;

    public int change(int x) {
        if (x >= 0)
            anim = x%3;
        return anim;
    }
}

public class Fox extends Animal {
    public int change(int x) {
        return Animal.MAMMAL;
    }
}

// client code
Animal x = new Animal();
Animal y = new Fox();
System.out.print(x.change(5)); // line 1
System.out.print(y.change(5)); // line 2
```

QUESTION 18

What is returned by the method call `stringStuff("abcdef")`?

- A. abcdef B. ace C. fdb
- D. fedcba E. b

```
public static String stringStuff(String s)
{
    if(s == null || s.length() == 1)
        return s;
    else if(s.length() <= 2)
        return "" + s.charAt(s.length()-1);
    else return s.charAt(s.length()-1) +
        stringStuff(
            s.substring(0,s.length()-2));
}
```

GO ON TO THE NEXT PAGE.

<p>QUESTION 19</p> <p>If <code>n=a.length</code> in the method <code>met</code> to the right, what is the big O of the <code>met</code> method when the argument array <code>a</code> is sorted? Give the most restrictive correct answer.</p> <p>A. $O(n)$ B. $O(n\log n)$</p> <p>C. $O(n^2)$ D. $O(\log n)$</p> <p>E. $O(1)$</p>	<pre>public static int met(int key, int[] a) { int low = 0; int high = a.length - 1; int mid; while (low <= high) { mid = low + (high-low)/2; if(key < a[mid]) high = mid - 1; else if (key > a[mid]) low = mid + 1; else return mid; } return -1; }</pre>
<p>QUESTION 20</p> <p>What algorithm does method <code>met</code> implement?</p> <p>A. linear search B. selection sort</p> <p>C. binary search D. insertion sort</p> <p>E. heap sort.</p>	
<p>QUESTION 21</p> <p>What is output by the code to the right?</p> <p>A. 2 B. 47 C. 45</p> <p>D. 0 E. 13</p>	<pre>int a = 15 & (~34); System.out.print(a);</pre>
<p>QUESTION 22</p> <p>What is output by the code to the right?</p> <p>A. Molly B. Tony C. Amy</p> <p>D. Nell E. Anna</p>	<pre>LinkedList<String> s = new LinkedList<String>(); s.addLast("Molly"); s.addLast("Tony"); s.addLast("Anna"); ListIterator<String> it = s.listIterator(); it.add("Amy"); it.add("Nell"); it.next(); it.remove(); System.out.print(it.next());</pre>
<p>QUESTION 23</p> <p>What is output by the code to the right?</p> <p>A. 5 B. 3 C. 9</p> <p>D. 4 E. The output will vary from one run of the program to the next.</p>	<pre>Stack<Integer> x = new Stack<Integer>(); x.push(5); x.push(3); x.push(9); x.push(4); while(x.size() > 0) { if(x.size() == 1) System.out.print(x.pop()); else x.pop(); }</pre>

<p>QUESTION 24</p> <p>What is output by the code to the right?</p> <p>A. 5 B. 3 C. 5 3 5 5</p> <p>D. 3 E. 5.0 3 3.0</p>	<pre>public class Test { public static void strange(int x) { x = 3; System.out.println(x); } public static void main(String[] args) { int x = 5; strange(x); System.out.println(x); } }</pre>
<p>QUESTION 25</p> <p>What is the array <code>arr</code> after the code to the right has been executed?</p> <p>A. {0, 1, 2} B. {0, 0, 0}</p> <p>C. {1, 1, 1} D. {1, 2, 3}</p> <p>E. There is no output due to a compilation error.</p>	<pre>int[] arr = new int[3]; int y = 0; for(int x : arr) { x = y+1; y = x; }</pre>
<p>QUESTION 26</p> <p>What is output by the method call <code>count(451)</code>?</p> <p>A. 0 B. 451 C. 10</p> <p>D. 46 E. 154</p>	<pre>public int count(int n) { if (n == 0) return 0; else { return n%10 + count(n/10); } }</pre>

GO ON TO THE NEXT PAGE.

QUESTION 27

Which of the following can be used to replace <*1> in the code to the right to allow the code segment to compile without error? Assume <*2> is chosen correctly.

- I. throws Exception
- II. throw Exception
- III. throw IOException
- IV. throws IOException

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

Assume <*1*> is filled in correctly.

QUESTION 28

Which of the following can be used to replace <*2> in the code to the right to allow the code segment to compile without error?

- I. throw new
- II. throws
- III. throw IOException

- A. I only B. II only C. I or II
D. III only E. I or III

```
private void mu(Scanner scan) <*1> {
    if(!scan.hasNextInt()){
        <*2> Exception(
            "Different type expected");
    }
    int x = scan.nextInt();
}
```

QUESTION 29

Assume that Tractor and Car are subclasses of the class Vehicle, and that all three classes have a no-argument constructor. What choices for <*1> will allow the following statement to compile?

```
Vehicle x = <*1>();
```

- I. new Vehicle
- II. Vehicle
- III. new Car
- IV. new Tractor
- V. Car
- VI. Tractor

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

QUESTION 30

What is the value of total after the code to the right has been executed?

- A. 25 B. 5 C. 10
D. 20 E. 15

```
int total = 0;
for(int i = 1; i <= 5; i++) {
    for(int j = 1; j <= i; j++)
        total = total + 1;
}
```


<p>QUESTION 31</p> <p>What is the big O of the method to the right? Give the most restrictive correct answer.</p> <p>A. $O(n^2)$ B. $O(n)$ C. $O(n \log n)$</p> <p>D. $O(1)$ E. $O(n^3)$</p>	<pre>public void theta(int n) { int total = 0; for(int i = 1; i <= n; i++) { for(int j = 1; j <= i; j++) total = total + 1; } }</pre>
<p>QUESTION 32</p> <p>Assume that Square is a subclass of the class Rectangle, and assume the code to the right is executed. Which of the following statements would then be legal?</p> <p>I. s = r; II. r = s; III. s = (Square) r;</p> <p>A. I and II B. II only C. III only</p> <p>D. II and III E. I only</p>	<pre>Rectangle r = new Square(); Square s = new Square();</pre>
<p>QUESTION 33</p> <p>Which of the following can replace <*1> in the code to the right so that the code segment compiles without error? Assume that <*2> is chosen correctly.</p> <p>I. abstract class II. abstract III. interface IV. class</p> <p>A. I and III B. II only C. III only</p> <p>D. IV only E. I only</p>	<pre>public <*1> Stuff { double one(); String two(int x); void three(); } public <*2> Stuff2 implements Stuff {</pre>
<p>Assume <*1*> is filled in correctly.</p>	
<p>QUESTION 34</p> <p>Which of the following can replace <*2> in the code to the right so that the code segment compiles without error?</p> <p>I. abstract class II. abstract III. interface IV. class</p> <p>A. I and II B. II only C. III only</p> <p>D. IV only E. I only</p>	<pre>public abstract double one(); public abstract String two(int x); public void three() { System.out.print(3); } }</pre>

QUESTION 35

In the code to the right, what statement can replace **<*1>** in the code so that the code compiles and so that line 1 in the client code prints -30.0?

- I. this(initial)
- II. balance = initial
- III. this.balance = initial

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

```
public class Account {
    private double balance;
    private String status;

    public Account(double initial){
        balance = initial;
    }
    public Account(double initial, String s){
        status = s;
        <*1>;
    }
    public double getBalance() {
        return balance;
    }
}

// Client Code
Account a = new Account(-30, "Overdrawn");
System.out.print(a.getBalance()); //line 1
```

QUESTION 36

The parameter of the main method in a Java program is `String[] args`. What values does this array contain?

- A. None – this parameter serves no purpose
- B. Names of files from the `java.lang` package
- C. Names of files from the `java.util` package
- D. command line arguments
- E. Strings from the BIOS

QUESTION 37

What is output by the code to the right?

- A. -5 B. 1 C. 8
- D. 4 E. 6

```
int[] arr = {1, -5, 6, 4, 2, 8};
int m = arr[0];
for(int i = 1; i < arr.length-1; i++) {
    if(m < arr[i]) m = arr[i];
}
System.out.print(m);
```

QUESTION 38

What does array `t` contain after the statement labeled line 1 is executed?

- A. `t` is null
- B. `{"a", "e you he", "e"}`
- C. `{"ar", "e you her", "e"}`
- D. `{"a"}`
- E. `{"ar"}`

```
public class Question38 {
    public static void main(String[] args) {
        String s = "are you here";
        String[] t = s.split("r"); // line 1
    }
}
```

QUESTION 39

After the while loop to the right terminates, what expression is guaranteed to be true?

- A. `val > 0`
- B. `val > 0 || (x < 100 && y > 10)`
- C. `val <= 0 && (x >= 100 || y <= 10)`
- D. `!(x < 100 || val > 0)`
- E. `x >= 100 && y <= 10`

```
// Assume that variables have been declared
// and initialized.
while ((val > 0) || (x < 100 && y > 10)) {
    <some statements here>
}
```

QUESTION 40

Which of the following choices can be used to replace `<*1*>` in the code to the right to generate the following output:

```
123456
12345
1234
123
12
1
```

- A. `j <= 6-i+1`
- B. `j <= i`
- C. `j <= 6`
- D. `i > j`
- E. `j <= i-1`

```
for(int i = 1; i <=6; i++) {
    for(int j = 1;<*1*>; j++) {
        System.out.print(j);
    }
    System.out.println();
}
```

No Test Material on This Page

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)

```

Computer Science Answer Key

UIL Invitational A 2013

1. E	11. D	21. E	31. A
2. A	12. E	22. B	32. D
3. D	13. B	23. A	33. C
4. C	14. D	24. B	34. E
5. E	15. A	25. B	35. E
6. C	16. C	26. C	36. D
7. C	17. A	27. A	37. E
8. C	18. C	28. A	38. B
9. C	19. D	29. D	39. C
10. E	20. C	30. E	40. A

Notes:

Questions 19 and 31: The clause "Choose the most restrictive correct answer." is necessary because per the formal definition of Big O, an algorithm that is $O(n^2)$ is also $O(n^3)$, $O(n^4)$, and so forth.