

Computer Science  
Tompkins High School  
January 11<sup>th</sup> 2014

**Directions:**

1. DO NOT OPEN THE EXAM UNTIL INSTRUCTED TO DO SO.
2. NO CALCULATORS or calculation devices may be used during the exam.
3. You will have 45 minutes to complete the exam.
4. When time is called you may finish writing down a letter if it is already started.
5. When you are finished with your exam wait quietly.

1. What is the result of the following expression? $50_{10} + 17_8$ A. $57_{10}$ B. $65_{10}$ C. $111_8$ D. $67_{10}$ E. $107_8$	
2. What is output by the code to the right? A. 51 B. 153 C. 54 D. 9 E. Syntax Error	<pre>int x = 3; int y = 17; int z = x*y +3;  System.out.println(z);</pre>
3. What is output by the code to the right? A. 7 12 1 B. 7 12 4 C. 6 13 1 D. 6 13 0 E. 5 13 0	<pre>int a = 7; int b = 12; int z =55;  z %= b++*--a/14; System.out.println(a + " " + b + " " + z);</pre>
4. How many times does the code to the right print the letter 'A'? A. 41 B. 36 C. 25 D. 42 E. infinite	<pre>for(int a = 0; a&lt;=5; a++) {     System.out.print("A");     for(int b = 0; b&lt;=5; b++)     {         System.out.print("A");     } }</pre>
5. What is output by the code to the right? A. jo so co B. joisocoo C. ojo so D. jo is cool E. Runtime Exception	<pre>String text = "Jojo is so cool"; String newText = text.substring(2,5)+ text.substring(7,13);  System.out.println(newText);</pre>
6. What is output by the code to the right? A. 1 3 0 4 2 B. 4 2 3 0 1 C. 1 0 2 3 4 D. 4 3 2 1 0 E. 0 1 4 2 3	<pre>int[] a = {4,2,3,0,1};  for(int i=0; i&lt; 5; i++)     System.out.print(a[a[i]] + " ");</pre>
7. What values of a would make z false? A. $\{(-\infty,100], [105, \infty)\}$ B. $\{(-\infty,100), [105, \infty)\}$ C. $\{(-\infty,100], (105, \infty)\}$ D. $\{(-\infty,100), (105, \infty)\}$ E. $\{(-\infty,99), [105, \infty)\}$	<pre>boolean z = (a&gt;100 &amp;&amp; a&lt;=105)</pre>
8. What is output by the code to the right? A. BD B. BCD C. BF D. B E. BDF	<pre>int grade = 82; if(grade &gt;=90)     System.out.print("A"); if(grade &gt;=80)     System.out.print("B"); else if(grade &gt;=70)     System.out.print("C"); if(grade &gt;=60)     System.out.print("D"); else     System.out.print("F");</pre>
9. What is output by the code to the right? A. This question is \"really\" hard.\\ B. This question is "really" hard.\\ C. This question is \"really\" hard.\ D. This question is "really" hard.\ E. Syntax Error	<pre>System.out.println("This question is \"really\" hard.\\");</pre>

<p>10. What is output by the code to the right?</p> <p>A. true false false false  B. true false true true  C. true false false true  D. true false true false  E. false false true false</p>	<pre>public class ClassA {     int a;     public ClassA(int a)     {         this.a = a     } }  public class MainClass {     public static void main(String[] args)     {         ClassA a = new ClassA(5);         ClassA b = new ClassA(6);         ClassA c = new ClassA(5);         System.out.print(a.equals(a));         System.out.print(" "+a.equals(b));         System.out.print(" "+a.equals(c));         System.out.print(" "+b.equals(c));     } }</pre>
<p>11. What is output by the code to the right?</p> <p>A. 6            B. 19            C. 8  D. 15           E. Runtime Exception</p>	<pre>System.out.println(Math.max(Math.min(8,15), Math.min(6,19)));</pre>
<p>12. What is output by the code to the right?</p> <p>A. *bob * 78.24*  B. *bob * 78.23*  C. * bob*78.24 *  D. * bob*78.23 *  E. *bob*78.23*</p>	<pre>String name = "bob"; double gpa = 78.23585;  System.out.printf(""%5s"%-8.2f",name,gpa);</pre>
<p>13. What is output by the code to the right?</p> <p>A. 9            B. 5            C. 8  D. 11           E. Runtime Exception</p>	<pre>int[][] nums = { {1,2,3},                  {4,5,6},                  {7,8,9},                  {10,11,12}};  System.out.println(nums[5/3][6-5]);</pre>
<p>14. What is output by the code to the right?</p> <p>A. 22           B. 15           C. 18  D. Syntax Error      E. Runtime Exception</p>	<pre>int[][] nums = { {8,2,3},                  {4,5,6},                  {7,8,9}};  int sum = 0; for(int a=0; a&lt;nums.length; a++)     sum += nums[a][nums.length-a-1];  System.out.println(sum);</pre>
<p>15. What is output by the code to the right?</p> <p>A. 3 4           B. 1 3 6           C. 0 1 2 5 6  D. 0 1 2 5      E. 0 1 2 5 6 7</p>	<pre>for(int x = 0; x &lt; 6; x++) {     if(x == 3    x == 4 )         continue;     System.out.print(x + " "); }</pre>
<p>16. What is the result of the expression to the right?</p> <p>A. 4            B. 1            C. 5  D. 0            E. 3</p>	<p>5&gt;&gt;2&lt;&lt;2</p>
<p>17. What is output by the code to the right?</p> <p>A. 45           B. 17           C. 57  D. 32           E. 39</p>	<pre>int x = 023 ^ 0b10101010; System.out.println(Integer.toString(x,16));</pre>

<p>18. What is output by the code to the right?</p> <p>A. [B,A,M]  B. [M,C,A]  C. [A,M,C]  D. [B,M,C]  E. Syntax Error</p>	<pre>ArrayList&lt;Character&gt; letters = new ArrayList&lt;Character&gt;(); letters.add('A'); letters.add('B'); letters.add(1,'M'); letters.add('C'); letters.remove(2); System.out.println(letters);</pre>
<p>19. What is output by the code to the right?</p> <p>A. No Output  B. 8675309  C. 8675967  D. 8888888  E. 8686868</p>	<pre>int x = 0;  for(int c=1;c&lt;=6;c++)     switch(x)     {         case 0:         case 1:         case 2:              System.out.print(x=8);             break;          case 3:             System.out.print(0);             x=10;             break;          case 4:         case 5:             x=3;             System.out.print(x);             break;          case 6:             System.out.print(++x);             x=1;          case 7:             System.out.print(x=5);             break;          case 8:             x--;             System.out.print(--x);             break;          case 9:         default:             System.out.print(9);     } }</pre>
<p>20. When does the program print true?</p> <p>A. When a is positive  B. When a is negative  C. When a is prime  D. When a is not prime  E. When a is less than 20</p>	<pre>int a = 5; int m = 2; while(m&lt;a &amp;&amp; a%m!=0)     m++; System.out.println(!(m&lt;a));</pre>
<p>21. What is output by the code to the right?</p> <p>A. 45                      B. 46                      C. 64  D. 68                      E. The loop never terminates</p>	<pre>int test = 2; do {     test = test*2; }while(test &lt;45); System.out.println(test);</pre>
<p>22. What would be result of calling hammer("Bilbo",2)?</p> <p>A. Bilbob                      B. ilboBb                      C. boBilb  D. lboBib                      E. Bilb</p>	<pre>public static String hammer(String s, int a) {     String r = "R";     if(s.length() %2 !=0)         return s.substring(a)+""+s.substring(0,a)+""+s.charAt(a);     for(int x = 0; x&lt;a;x++)         r += s.charAt(x%s.length()) + ""+ r.charAt((r.length()/2+x)%r.length());     return r; }</pre>
<p>23. What would be the result of calling hammer("Cake",4)?</p> <p>A. RCRaRkReR                      B. RCake                      C. Rake  D. RCRaRkaReR                      E. Raker</p>	

<p>24. What replaces &lt;*1&gt; in the code to the right so that data is correctly sorted after the following call?</p> <p>sortA(data,0,data.size());</p> <p>A. sortA (data,from-1,p-1);  B. sortA (data,from+1,p-1);  C. sortA (data,from,p+1);  D. sortA (data,from,p);  E. sortA (data,from,p-1);</p>	<pre>public static void sortA (int[] data, int from, int to) {     if(from &gt;= to)         return;      int p = (from +to)/2;      int i = from;     int j = to;     while(i&lt;=j)     {         if(data[i] &lt;= data[p])             i++;         else if(data[j] &gt;= data[p])             j--;         else         {             swap(data,i,j);             i++;             j--;         }     }      if(p &lt; j)     {         swap(data, j,p);         p = j;     }      else if(p &gt; i)     {         swap(data, i,p);         p = i;     }      &lt;*1&gt;     sortA (data,p+1, to); }</pre>
<p>25. What type of sort does the code to the right implement?</p> <p>A. Insertion    B. Quick    C. Merge</p> <p>D. Selection    E. Heap</p>	
<p>26. What is the value stored in result after the code to right is run?</p> <p>A. 1                      B. 6                      C. 16  D. 0                      E. 7</p>	<pre>int result; result = 3 * (7 - 2) % 4 - 9 / 4;</pre>
<p>27. What is output by the code to the right?</p> <p>A. 5 4 2 1  B. 6 5 4 3 2 1  C. 1 2 3 4 5 6  D. 3 2 1 6 5 4  E. 2 1 5 4</p>	<pre>int[][] nums = {     {1,2,3},                    {4,5,6}}; for(int r=0; r&lt;nums.length; r++) {     for(int c=nums.length-1; c&gt;=0; c--)         System.out.print(nums[r][c]+" "); }</pre>
<p>28. What is output by the code to the right?</p> <p>A. ACDB    B. DACA                      C. DACB</p> <p>D. ABCD    E. DABA</p>	<pre>LinkedList&lt;String&gt; ll = new LinkedList&lt;String&gt;(); ll.addLast("B"); ll.addFirst("A"); ll.addLast("C"); ll.addFirst("D"); System.out.print(ll.removeFirst()); System.out.print(ll.getFirst()); System.out.print(ll.removeLast()); System.out.print(ll.removeFirst());</pre>
<p>29. What is output by the code to the right?</p> <p>A. [1, 6, 7, 7, 8]    B. [A, B, P, P, Q]                      C. [A, B, P, Q]</p> <p>D. [1, 6, 7, 8]                      E. [1, 6, 7, 7, 8, A, B, P, P, Q]</p>	<pre>Map&lt;Integer,String&gt; map = new HashMap&lt;Integer,String&gt;(); map.put(7,"P"); map.put(8,"P"); map.put(1,"A"); map.put(7,"Q"); map.put(6,"B"); System.out.print(map.keySet());</pre>

<p>30. What is the output of the code to the right?</p> <p>A. 7                      B. 11                      C. 15 D. 6                      E. 2</p>	<pre>int a = 7; a = (a&gt;10)?a^5:(a&lt;10)?a 12:a&amp;6; System.out.println(a);</pre>
<p>31. Given the code to the right, what replaces &lt;*1&gt; so the add works properly for the doubly circular linked list?</p> <p>A. temp.getNext().setPrev(insert); B. temp.setData(insert) C. insert.setPrev(temp.getPrev()); D. insert.getNext().setPrev(temp); E. The code already works. Nothing needs to be added.</p>	<pre>public class DLLNode&lt;E&gt;{     private E data;     private DLLNode&lt;E&gt; next;     private DLLNode&lt;E&gt; prev;      public DLLNode(E data){         this.data = data;         next = prev = null;     }     // accessors and mutators not shown }</pre>
<p>32. Given the code to the right, what replaces &lt;*2&gt; so the size works properly for the doubly circular linked list?</p> <p>A. first B. temp C. null D. temp.getPrev() E. last</p>	<pre>public class DoublyCircularLinkedList&lt;E&gt;{     DLLNode&lt;E&gt; first = null;     DLLNode&lt;E&gt; last = null;      public DoublyCircularLinkedList()     { first = last = null; }      public void add(int x, E data){         if(x==0)             addFirst(data);         else if(x==size())             addLast(data);         else if(x&gt;0 &amp;&amp; x&lt;size()){             DLLNode&lt;E&gt; temp = first;             int count =1;             while(temp.getNext() != null &amp;&amp; count &lt;x){                 count++;                 temp = temp.getNext();             }             DLLNode&lt;E&gt; insert = new DLLNode&lt;E&gt;(data);             insert.setNext(temp.getNext());             insert.setPrev(temp);             &lt;*1&gt;             temp.setNext(insert);         }     }      public int size(){         if(first ==null)             return 0;         else{             DLLNode&lt;E&gt; temp = first;             int count =1;             while(temp != &lt;*2&gt;){                 count++;                 temp = temp.getNext();             }             return count;         }     } }</pre>

<p>33. What is the result of the truck(1,3,2)?</p> <p>A. 36      B. 33      C. 45 D. 17      E. 18</p>	<pre>public static int truck(int x, int y, int m) {     if(x&gt;m    y&lt;m)         return x + y +2;     else         return truck(x+1,y, m) + truck(x, y-1, m); }</pre>
<p>34. What is the result of the truck(1,5,3)?</p> <p>A. 247      B. 230      C. 160 D. 96      E. 125</p>	
<p>35. What is output by the code to the right?</p> <p>A. A17 B20 B. A17 B28 C. A17 B34 D. A17 B17 E. A34 B34</p>	<pre>public static void dataBomb(int[] data) {     data = new int[5];     for(int i=0;i&lt;data.length; i++)         data[i] = i*2;     data[1] =data[2]+data[3];     data[2] =data[0]+data[1]; }  public static void main(String[] args) {     int[] data = {3,8,6};      int total =0;     for(int a: data)         total+=a;     System.out.print("A"+total);      dataBomb(data);      total =0;     for(int a: data)         total+=a;     System.out.print(" B"+total); }</pre>
<p>36. What would the result of a pre-order print after the code the right is run for a binary search tree?</p> <p>A. 7 3 1 2 4 8 9 B. 1 2 3 4 7 8 9 C. 2 1 4 3 9 8 7 D. 9 8 7 4 3 2 1 E. 2 9 1 4 3 8 7</p>	<pre>BinarySearchTree&lt;String&gt; bst = new BinarySearchTree&lt;String&gt;();  bst.add(7); bst.add(3); bst.add(4); bst.add(1); bst.add(2); bst.add(8); bst.add(9);</pre>
<p>37. What would the result of a post-order print after the code the right is run for a binary search tree?</p> <p>A. 7 3 1 2 4 8 9 B. 1 2 3 4 7 8 9 C. 2 1 4 3 9 8 7 D. 9 8 7 4 3 2 1 E. 2 9 1 4 3 8 7</p>	
<p>38. What is output by the code to the right?</p> <p>A. bmt*st\$(s&amp;yh6* B. b^^*^^\$^^&amp;yh6*^ C. b^*st\$^&amp;yh6*^ D. ^b^*st\$^&amp;yh6*^ E. Runtime Exception</p>	<pre>String s = "bmt*st\$(s&amp;yh6*"; String[] broken = s.split("[m()][st]"); for(String a: broken)     System.out.print(a+"^");</pre>
<p>39. What is output by the code to the right?</p> <p>A. 'e'      B. -4      C. 4 D. 15      E. 'l'</p>	<pre>String a = "apples and pickles"; String b = "applications"; System.out.println(a.compareTo(b));</pre>
<p>40. What is output by the code to the right?</p> <p>A. -8      B. large positive number      C. 0 D. 7      E. large negative number</p>	<pre>int a = -5; a=a&gt;&gt;&gt;8; System.out.println(a);</pre>

# Standard Classes and Interfaces — Supplemental Reference

(Accessed From: UIL COMPUTER SCIENCE DISTRICT 2 2011)

## 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.lang.Exception

## 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)