

UIL COMPUTER SCIENCE

S-19 Practice Packet

Written by
Kirby Rankin

Edited by Nancy Barnard

Author Kirby Rankin brings over 25 years of teaching experience and has coached Computer Science for most of those years. His successes include three individual champions and six 2A team champions, and these were in a row from 2008 through 2013. He had many, many more competitors qualify for region and state over his years.

We are a small company that listens! If you have any questions or if there is an area that you would like fully explored, let us hear from you. We hope you enjoy this product and stay in contact with us throughout your academic journey.

~ President Hexco Inc., Linda Tarrant

HEXCO ACADEMIC

www.hexco.com

P.O. Box 199 • Hunt, Texas 78024

Phone: 800.391.2891 • Fax: 830.367.3824

Email: hexco@hexco.com

Copyright © 2019 by Hexco Academic. All rights reserved. Reproduction or translation of any part of this work beyond that permitted by Section 107 or 108 of the 1976 *United States Copyright Act* without the permission of the copyright owner is unlawful. The purchaser of this product is responsible for adhering to this law which prohibits the sharing or reselling of copyrighted material with anyone. This precludes sharing with coaches or students from other schools via mail, fax, email, or simply "passing along." Hexco materials may not be posted online. Exception/permission for photocopies granted by Hexco Academic is only applicable for *Practice Packets* which may be copied expressly for the purchaser's group or classroom at the same physical location.

IF YOU LIKE THIS PRODUCT, WE ALSO RECOMMEND:

Computer Science Practice Packet F17-F19

Computer Science Region / State Practice Packet

Computer Science Concepts - Hands On Element - The First Steps

Computer Science Concepts – The First 15

Computer Science Concepts – The Next 25

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)
- o static int parseInt (String s, int radix)

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> 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 ()

```
int moon=1;
String str="hello"+ (moon>1?" goodbye "+moon+" yes ":" no "+moon+"
maybe ");
out.print(str);
```

34. What is the output of the code segment shown above?

- A. hello no 1 maybe
- B. hello goodbye 1 yes
- C. no 1 maybe
- D. goodbye 1 yes
- E. hello no maybe

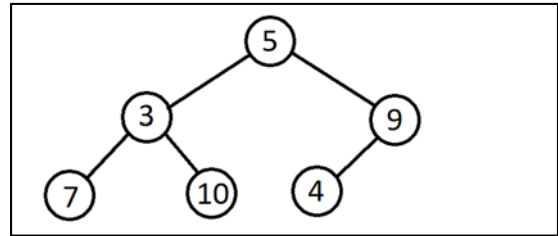
35. Which of the following cannot replace **<code>** in the code segment shown on the right?

- A. int
- B. Integer
- C. double
- D. ArrayList
- E. More than one of the above.

```
ArrayList<Integer> list=new
ArrayList<Integer>();
list.add(1);list.add(2);
list.add(5);list.add(6);
for(<code> i:list)
    out.println(i);
```

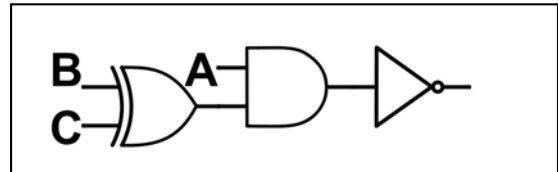
36. Which of the following is true about the data structure diagrammed on the right?

- A. It is a binary search tree.
- B. It is a max heap.
- C. It is a complete tree.
- D. It is a min heap.
- E. It is a complete graph.



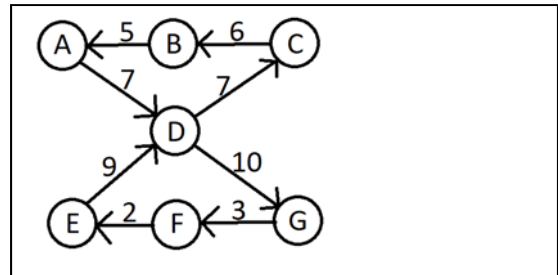
37. Which of the following Boolean expressions is equivalent to the diagram shown on the right?

- A. $\bar{A} * B \oplus C$
- B. $\overline{A * B \oplus C}$
- C. $A + \overline{B * C}$
- D. $\overline{A * C} \oplus B$
- E. $\overline{A * B + C}$



38. The graph on the right is _____.

- A. undirected, weighted and complete
- B. directed, weighted and complete
- C. directed and unweighted
- D. connected, simple and complete
- E. directed and weighted



27. What is printed by the main method in the class Array2D shown on the right?

- A. [0, 1, 2]
[2, 3, 5]
[6, 8, 9]
- B. [1, 2, 8]
[5, 6, 9]
[0, 2, 3]
- C. [0, 2, 1]
[6, 3, 2]
[8, 5, 9]
- D. [0, 2, 6]
[1, 3, 8]
[2, 5, 9]
- E. [8, 2, 1]
[6, 5, 9]
[0, 3, 2]

```
public class Array2D {
    public static void main(String[] args) {
        int[][] mat= {{8,2,1},
                      {6,5,9},
                      {0,3,2}};
        int[][] tam=mtd(mat);
        for(int[] r:tam)
            out.println(Arrays.toString(r));
    }
    public static int[][] mtd(int[][] m) {
        int x=m[0].length;
        for(int i=0;i<x;i++) {
            int[] t=new int[m.length];
            for(int k=0;k<m.length;k++)
                t[k]=m[k][i];
            Arrays.sort(t);
            for(int k=0;k<m.length;k++)
                m[k][i]=t[k];
        }
        return m;
    }
}
```

28. What is the output of the code segment shown on the right?

- A. 39
- B. 105
- C. 223
- D. 367
- E. 609

```
int s=0;
for(int x=19;x<=39;x++)
{
    if(x%3==0)continue;
    if(x%8==0)break;
    s+=x;
}
out.print(s);
```

29. What the output of **line #1** in the code segment shown to the right?

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

30. What the output of **line #2** in the code segment shown to the right?

- A. [8]
- B. [1, 3]
- C. [3, 1]
- D. [1, 2, 3, 4, 8]
- E. [4, 2]

```
//Use the following code segment to answer
//questions 29 and 30.
Set<Integer> s1=new TreeSet<Integer>();
Set<Integer> s2=new TreeSet<Integer>();
s1.add(4);s1.add(2);s1.add(8);s1.add(4);
s2.add(4);s2.add(3);s2.add(1);s2.add(2);
out.println(s1);//line #1
s1.removeAll(s2);
out.println(s1);//line #2
```

```
char chr[] = {'r','e','u','a','o','n','g'};
int m=0,n=1;
while(chr[m]!='o') {
    n=m;m++;
}
```

10. At the conclusion of the while loop, which of the following statements is true about the code segment shown above?

- A. n stores the index value of the character 'o'
- B. m and n are equal
- C. n stores the index value of the character 'a'
- D. m stores the index value of the character 'a'
- E. n and m both store the index value of the character 'o'

```
public static void main(String[] args) throws IOException{
    File f=new File("input.dat");
    Scanner scan=<code>;
    int c=scan.nextInt();
    for(int x=1;x<=c;x++)
        out.print(scan.nextInt()+" ");
    scan.close();
}
```

11. Which of the following must replace <code> in the main method shown above to ensure that scan is a Scanner object associated with a file named **input.dat**? Assume that all necessary classes have been imported.

- A. new Scanner()
- B. new Scanner(f)
- C. new Scanner("input.dat");
- D. "input.dat"
- E. f

12. What is printed by the code segment listed on the right?

- A. eoetm
- B. mteoe
- C. loht
- D. elpoohehtttom
- E. There is no output. Throws an ArrayIndexOutOfBoundsException.

```
String s1="";
String s2="mottthehoople";
for(int i=s2.length()-1;i>=0;i-=3)
    s1+=s2.charAt(i);
out.print(s1);
```

13. What is the output of the code segment shown on the right?

- A. 17 17 7
- B. 20 12 17
- C. 17 17 17
- D. 8 -5 17
- E. 20 17 17

```
int x=8,y=-5,z=7;
x=y=z+10;
out.print(x+" "+y+" "+z);
```