

Computer Science Region-State 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 throughout his years. Kirby currently serves as the author of the written portion of the UIL Computer Science contest.

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 © 2017 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 F16 & S17

Computer Science Concepts - Hands On Element - The First Steps

Computer Science Concepts – The First 15 – 2nd Edition

Computer Science Concepts – The Next 25

23. Which of the following classes is immutable?

<p>A.</p> <pre>import java.util.*; public class ImmutableClass { private int x,y; private ArrayList<Integer> a; public ImmutableClass(int x, int y, ArrayList<Integer> a){ this.x = x; this.y = y; this.a = a;} public int getX() {return x;} public int getY() {return y;} public ArrayList<Integer> getA(){ return a; }</pre>	<p>B.</p> <pre>import java.util.*; public class ImmutableClass { public int x,y; public ArrayList<Integer> a; public ImmutableClass(int x, int y, ArrayList<Integer> a){ this.x = x; this.y = y; this.a = a;} public int getX() {return x;} public int getY() {return y;} public ArrayList<Integer> getA(){ ArrayList<Integer> temp=new ArrayList<Integer>(a); return temp; }</pre>
<p>C.</p> <pre>import java.util.*; public class ImmutableClass { private int x,y; private ArrayList<Integer> a; public ImmutableClass(int x, int y, ArrayList<Integer> a){ this.x = x; this.y = y; this.a = a;} public int getX() {return x;} public int getY() {return y;} public ArrayList<Integer> getA(){ ArrayList<Integer> temp=new ArrayList<Integer>(a); return temp; } public void setX(int x) {this.x = x;} public void setY(int y) {this.y = y;} public void setA(ArrayList<Integer> a) {this.a = a;}</pre>	<p>D.</p> <pre>import java.util.*; public final class ImmutableClass { private final int x,y; private final ArrayList<Integer> a; public ImmutableClass(int x, int y, ArrayList<Integer> a){ this.x = x; this.y = y; this.a = a;} public int getX() {return x;} public int getY() {return y;} public ArrayList<Integer> getA(){ ArrayList<Integer> temp=new ArrayList<Integer>(a); return temp; }</pre>
<p>E. More than one of the classes shown above is immutable.</p>	

Explanations For Written Exam Answers

1. D $333_8 = 219_{10}$; $11011001_2 = 217_{10}$; $DA_{16} = 218_{10}$; $22011_3 = 220_{10}$; $220_{10} > 219_{10}$; therefore $22011_3 > 333_8$.
2. C $-42\%9-5-(-12\%5) = -42\%9-5-(-2) = -42\%9-5+2 = -6-5+2 = -11+2 = -9$
3. A `%10.1f` is the format specifier. It specifies that the value 23.18954 should be printed right justified in 10 spaces and rounded to one decimal place.
4. B The `replace` method replaces every occurrence of the first parameter, in this case `aeiou`, with the second parameter, in this case `X`. The string `aeiou` does not occur in the string `s` so nothing is replaced.
5. B $9 == 5 \ \&\& \ 9 \leq 1 \wedge 5 < 1$
False $\&\&$ False \wedge False
False $\&\&$ False
False
6. D The data type designation `f` must be present in the first line to assign the value to a float type variable. The `round` method returns a long type value rounded to the nearest whole number.
7. E The expression `i * (d+1) - f` evaluates to a double type value and therefore cannot be assigned to a long type variable.
8. A For each iteration of the for loop, the substring `ss` is a pair of letters from the original string `s`. They are `ab`, `br`, `ra`, `ac`, `ca`, `ad`, `da`, `ab`, `br`, and `ra`. For every occurrence of `ab` or `br` add one to `x`. For every occurrence of `ra` add two to `x`. For every other pair add three to `x`.
9. B Since the increment operator (`++`) precedes the loop control variable `q` it is executed before the comparison is made. So, on the first iteration `q` get 1 and then the loop body is executed.
10. C Arrays are objects. `b[] = a` means that both `a` and `b` point at the same array. Here is the content of both arrays as each statement is executed:

 $[4, 1, 5, 0, 2, 3]$
 $[6, 1, 5, 0, 2, 3]$
 $[6, 1, 7, 0, 2, 3]$
 $[6, 8, 7, 0, 2, 3]$
 $[6, 8, 7, 0, 2, 9]$
11. D The argument for `Integer.parseInt` must be a string that represents a whole number. Three of the values in the file are not whole numbers.
12. B Choice A does not correctly calculate x^0 . Choice C does stops one iteration too soon and does not correctly calculate x^0 . Choice D returns y^y . Choice E does not correctly calculate x^0 .

Problem #5 - Apartment

Judges Data

125 5/3/17 17:15 G M Nail is sticking out of cabinet.
218 5/3/17 14:30 G M Air conditioner is not cool enough.
38 5/3/17 6:05 E E Smoke coming out of electrical outlet.
385 5/3/17 10:15 G E No electricity in bathroom electrical outlet.
212 5/3/17 6:05 G M Draft under front door.
4 5/2/17 9:30 G P Leaky faucet.
300 5/2/17 9:29 G P Leaky faucet.
15 5/3/17 15:16 E P Overflowing toilette.
301 5/3/17 15:15 E P Gushing water under sink.
215 5/2/17 6:05 E E Sparks coming from the ceiling.
54 5/3/17 6:00 E E No electricity in apartment.
1 5/3/17 22:30 E O Skunk lose in apartment.
19 5/2/17 8:02 G O Child marked on walls with crayon.
368 5/2/17 17:59 G O Hole in sheet rock.
305 5/4/17 0:15 E P Water falling from ceiling.
205 5/3/17 0:15 E P Broken water line.

Judges Output

215 Sparks coming from the ceiling.
205 Broken water line.
54 No electricity in apartment.
38 Smoke coming out of electrical outlet.
301 Gushing water under sink.
15 Overflowing toilette.
1 Skunk lose in apartment.
305 Water falling from ceiling.
300 Leaky faucet.
4 Leaky faucet.
385 No electricity in bathroom electrical outlet.
212 Draft under front door.
218 Air conditioner is not cool enough.
125 Nail is sticking out of cabinet.
19 Child marked on walls with crayon.
368 Hole in sheet rock.

Problem #2 - Average

```

import java.util.*;
import java.io.*;
import static java.lang.System.*;
public class Average {

    public static void main(String[] args) throws IOException{
        Scanner f=new Scanner(new File("average.dat"));
        //Read how many students are in the class.
        int n=Integer.parseInt(f.nextLine());
        //Total will store the total inches for all students.
        double total=0;
        for(int x=1;x<=n;x++)
        {
            //Create a new scanner object for each line of data.
            Scanner s=new Scanner(f.nextLine());
            double feet=s.nextDouble(); //Read the feet.
            double inches=0;
            s.next(); //Dispose of the word "feet".
            //If there are inches, read that value.
            if(s.hasNext())
                inches=s.nextDouble();
            //Calculate the total number of inches for this classmate and add it to the
            //total for the class.
            total+=feet*12+inches;
        }
        //Calculate the average height in inches.
        double avg=total/n;
        //Calculate the whole number value for the number of feet in the average height.
        int avgFeet=(int)avg/12;
        //Calculate the number of inches in the average and round that value.
        long avgInches=Math.round(avg%12);
        out.println(avgFeet+" feet "+avgInches+" inches");
    }
}

```

Problem #3 - Bingo

```

import java.util.*;
import java.io.*;
import static java.lang.System.*;
public class Bingo {

    public static void main(String[] args) throws IOException{
        Scanner f=new Scanner(new File("bingo.dat"));

        //Create an empty card then read the card numbers from the data file.

        String[][] card=new String[5][5];
        for(int row=0;row<5;row++)
            for(int col=0;col<5;col++)
                card[row][col]=f.next();

        //Discard the blank line.

        f.nextLine();f.nextLine();

        //Read the bingo numbers that are called by the "caller" into a string
        //then use that string to create a Scanner object.

        String p=f.nextLine();
        Scanner plays=new Scanner(p);

        //As long as the card has not become a winner and there are still
        //numbers to be called, read bingo numbers from the Scanner object plays.

        boolean winner=false;
        while(plays.hasNext()&&!winner){
            //letnum contains something like B4 or I21.
            String letnum=plays.next();
            char letter=letnum.charAt(0);//Parse out the letter.
            int col=-1;

            //The letters BINGO each represent a column in the array. This switch
            //converts the letter to the appropriate column number.

            switch(letter){
                case 'B':col=0;break;
                case 'I':col=1;break;
                case 'N':col=2;break;
                case 'G':col=3;break;
                case 'O':col=4;break;
            }
            String number=letnum.substring(1);//Parse out the number.

            //Check to see if this number is present in the designated column.
            //If so, replace the number with a capital X.

```