ICS4UI Midterm 1 28 October, 2016

Write an if-statement that prints "yes" if the point (*x, y*) is either on the positive *y*-axis or on the positive y-axis, and prints "no" otherwise. Assume that *x* and *y* already have values.

|  |
| --- |
| if ((x == 0 && y > 0 ) || (y == 0 && x > 0)) {  println(“yes”);  }  else {  println(“no”);  } |

Declare a 1-D array named *drSeussBooks* that can store 50 Strings.

|  |
| --- |
| String[] drSeussBooks = new String[50]; |

Write some code that fills the first two items in *drSeussBooks* with the Strings "The Lorax" and   
"Green Eggs & Ham", and then fills the last item in the array with "Red Fish Blue Fish".

|  |
| --- |
| drSeussBooks[0] = “The Lorax”;  drSeussBooks[1] = “Green Eggs & Ham”;  drSeussBooks[49] = “Red Fish Blue Fish”; |

Declare a 2-D array named *myData* that can store a 10 x 8 table of decimal values.

|  |
| --- |
| float[][] myData = new float[10][8]; |

Write code that fills *myData* with random values between 0 and 1.

|  |
| --- |
| for (int i = 0; i < 10; i++) {  for (int j = 0; j < 8; j++) {  myData[i][j] = random(0, 1);  }  } |

Write code that counts how many items in the last 3 columns of *myData* are larger than 0.75.

|  |
| --- |
| int total = 0;  for (int i = 0; i < 10; i++) {  for (int j = 5; j < 8; j++) {  if (myData[i][j] > 0.75) {  total ++;  }  }  }  println(str(total) + “ items in the last 3 columns are larger than 0.75”); |

Declare an empty ArrayList of Strings named *myHobbies*.

|  |
| --- |
| ArrayList<String> myHobbies = new ArrayList<String>(); |

Write code that adds the words "chess" and "dance" to *myHobbies*, and then prints the statement,  
"The list now has \_\_\_ items in it", where the \_\_ is filled with the correct value*.*

|  |
| --- |
| myHobbies.add(“chess”);  myHobbies.add(“dance”);  int numHobbies = myHobbies.size();  println(“The list now has ” + str(numHobbies) + “ items in it”); |

Complete these method headers with the appropriate data types

|  |
| --- |
| boolean **isPrimeNumber**(int x )  float **getArea**( Polygon p )  void **printLastName**(String fullName )  String **getQuadrant**(float x, float y ) //Example: getQuadrant(3, -5.6) returns "Quadrant IV"  void **drawMandelbrotSet**()  int **numFailures**(float[] marksArray ) //Example: numFailures( {60, 49, 90, 35, 80}) returns 2  String[] **get3Rhymes**( String word ) //Example: get3Rhymes("pat") returns the array {"bat", "cat", "sat"}  Fraction **getReciprocal**() //Example: Fraction f = new Fraction(2, 5); Fraction g = f.getReciprocal(); |

Write a function named *getTotalSeconds* that takes a stopwatch time as a parameter and returns the total number of seconds represented by that time. The stopwatch time should be a String.   
Examples: getTotalSeconds( "3:40" ) returns 220, and getTotalSeconds( "0:05" ) returns 5.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

int getTotalSeconds( String t ) {

int iColon = t.indexOf(“:”);

String sMinutes = t.substring(0, iColon);

String sSeconds = t.substring(iColon+1);

int minutes = parseInt(sMinutes);

int seconds = parseInt(sSeconds);

return minutes \* 60 + seconds;

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Write some code that reads a list of stopwatch times from a file named *SJAM Junior Girls CC Meet.txt* and calls the function getTotalSeconds on each one, and outputs the results to a second file named *SJAM Times In Seconds.txt.*

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

String[] lines = loadStrings(“SJAM Junior Girls CC Meet.txt”);

PrintWriter fOut = createWriter(“SJAM Times In Seconds.txt”);

for (int i = 0; i < lines.length; i++) {

int totalSeconds = getTotalSeconds(lines[i]);

fOut.println(totalSeconds);

}

fOut.close();

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Write the constructor for this class. Assume that a movie has not been watched when it is first created.

class Movie {

String title;

int starRating;

int numTimesWatched;

//Constructor goes here

Movie( String title, int starRating) {

this.title = title;

this.starRating = starRating;

this.numTimesWatched = 0;

}

//don’t code these yet

void watch() {…}

void displayInfo() {…}

}

Write a line of code that creates a Movie object named *theMartian* with title "The Martian" and a rating of four stars.

|  |
| --- |
| Movie theMartian = new Movie(“The Martian”, 4); |

Suppose you watch The Martian twice. Write some code to indicate this.

|  |
| --- |
| theMartian.watch();  theMartian.watch(); |

Write some code that creates an empty ArrayList named *bestMovies* of type Movie, and then adds the Movie object you created above to the ArrayList.

|  |
| --- |
| ArrayList<Movie> bestMovies = ArrayList<Movie>();  bestMovies.add(theMartian); |

Suppose the ArrayList *bestMovies* now contains several Movie objects , and that some of the movies in it have been watched and others haven't. Write some code that counts how many movies in *bestMovies* have been watched.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

int totalWatched = 0;

for (int i = 0; i < bestMovies.size(); i++) {

if (bestMovies.get(i).numTimesWatched > 0) {

totalWatched++;

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Final task! Are you up for the challenge? Assume the ComplexNumber class has already been written. Write a setup() program that:

* Creates two empty arrays of ComplexNumber objects named listA and listB
* Fills both arrays with random complex numbers
* Multiplies every number in listA with every number in listB, and stores the results in a 2-D array named productsOfAB

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

ComplexNumber[] listA = new ComplexNumber[10];

ComplexNumber[] listB = new ComplexNumber[10];

for (int i = 0; i < 10; i++) {

float a = random(-10, 10);

float b = random(-10, 10);

listA[i] = ComplexNumber(a, b);

}

ComplexNumber[][] productsOfAB = new ComplexNumber[10][10];

for (int a = 0; a < listA.length(); a++) {

for (int b = 0; b < listB.length(); b++) {

ComplexNumber prod = listA[a].

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***