Java1: Lesson 4 – Lab Project

1. Add **summary** at the top, **documentation** at the bottom and **comments** where necessary.
   - Recordings and interviews are an exception, but they should be included in your submission code.
2. **Cite** the source of your information.
3. Refrain from including anything **you don't fully understand**.

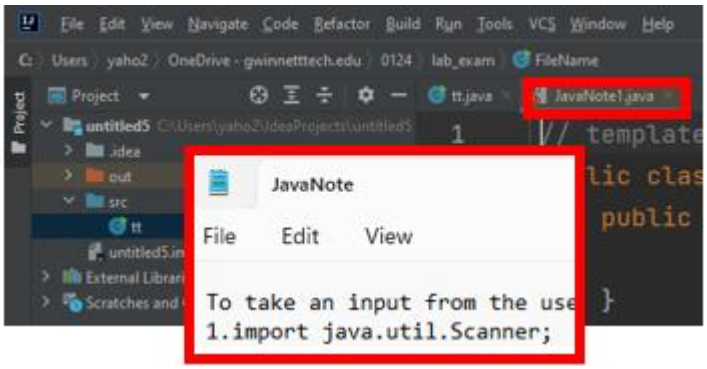
2

50 points

Update your **javaNote** to include the following topics (but not limited to):

1. print/ println/ printf
2. taking user input
3. if statements
4. switch statements
5. random method (for generating numbers or characters)

Take a screenshot demonstrating your proficiency in using **javaNote** while coding (.java/ .txt, etc.)



File(s) to submit	javaNote.txt/ .docx/ .java JavaNote.png

Make sure your code implements the following methods as needed.

`Math.PI`
`Math.E`
`sin()`
`cos()`
`tan()`
`acos()`
`asin()`
`atan()`
`toRadians()`
	`exp()`
	`log()`
	`log10()`
	`pow()`
	`sqrt()`
		`ceil()`
		`floor()`
		`rint()`
		`int round()`
		`long round()`
		
		`max()`
		`min()`
		`abs()`
		`random()`

`isDigit()`
`isLetter()`
`isLetterOfDigit()`
`isLowerCase()`
`isUpperCase()`
`toLowerCase()`
`toUpperCase()`
	`equals()`
	`equalsIgnoreCase()`
	`compareTo ()`
	`log()`
	`compareToIgnoreCase ()`
	`startsWith ()`
	`endsWith ()`
		`length()`
		`charAt()`
		`concat()`
		`toUpperCase()`
		`toLowerCase()`
		`trim()`
		`substring(n)`
		`substring(n1,n2)`
			`indexOf(ch)`
			`indexOf(ch,i)`
			`indexOf(s)`
			`indexOf(s,i)`
			`lastIndesOf(ch)`
			`lastIndesOf(ch,i)`
			`lastIndesOf(s)`
			`lastIndesOf(s,i)`

`parseInt()` `parseDouble()`	`printf("%-10d%-10.4f%-10.4f%-10.4f%-10.4f\n", a,b,c,d,e)`

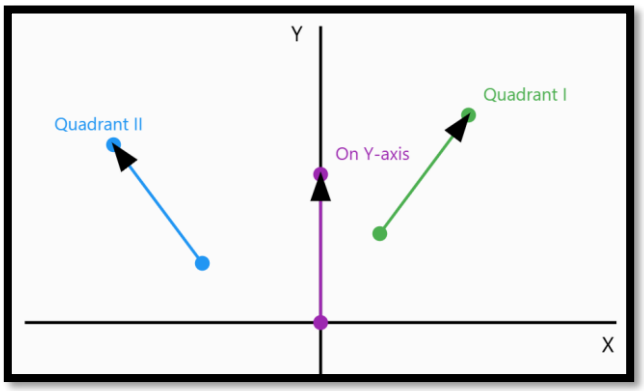
<table>
<tr><td>3</td><td>50 points
Write a program that extracts the user ID from an email address.</td></tr>
<tr><td></td><td>Sample output
? abc123@gmail.com
User ID: abc123</td></tr>
<tr><td></td><td>File(s) to submit
J104_3.java
J104_3.png</td></tr>
<tr><td>4</td><td>100 points
Write a program that takes a positive number as input and converts it into either:
1. Years, months, and days (if the input represents days)
2. Hours, minutes, and seconds (if the input represents seconds)
Conversion Rules:
• If the user selects Day (1), the input represents days, and the program should:
○ Convert the given days into years, months, and days, assuming:
▪ 1 year = 365 days
▪ 1 month = 30 days
• If the user selects Time (2), the input represents seconds, and the program should:
○ Convert the given seconds into hours, minutes, and seconds, using:
▪ 1 hour = 3600 seconds
▪ 1 minute = 60 seconds</td></tr>
<tr><td></td><td>Sample output
Day(1) or Time(2)? 1
? 1234567
Your day is **years ** months and ** days

Day(1) or Time(2)? 2
? 1234567
Your time is **hours ** minutes and ** seconds</td></tr>
<tr><td></td><td>File(s) to submit
J104_4.java
J104_4.png</td></tr>
<tr><td>5</td><td>100 points
Write a Java program that prompts the user to enter two alphabet characters. The program will then generate a random sequence of five uppercase alphabet characters and compare the user's inputs as follows:
• The first input will be compared to the first character of the randomly generated sequence.
• The second input will be compared to the last character of the randomly generated sequence.

Display whether each input is correct or incorrect and show the generated sequence for reference.</td></tr>
<tr><td></td><td>Sample output
? JV
Both are wrong
Generated sequence: WDOKP ← you should compare J with W, V with P
? AL
The first one is correct.
Generated sequence is ADPBP ← you should compare A with A, L with P</td></tr>
<tr><td></td><td>File(s) to submit
J104_5.java
J104_5.png</td></tr>
<tr><td>6</td><td>200 points
Write a program to calculate the distance between two points in 2D space represented by their (x, y) coordinates. The inputs should be (x1,y1) and (x2,y2)</td></tr>
</table>

Refer to https://youtu.be/WqhlG3Vakw8
https://bit.ly/3qbVxNn

$$\boldsymbol{Distance\ (in\ 2D) = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}}$$



1. **Distance Calculation:**
   - Display the result rounded **up** to two decimal places.
   - Show only the integer part of the distance **in parentheses**.
2. **Quadrant Determination:**
   - Identify which quadrant (x2,y2) is in relative to (x1,y1) using the following cases:
     - **Quadrant I**: (+x, +y)
     - **Quadrant II**: (-x, +y)
     - **Quadrant III**: (-x, -y)
     - **Quadrant IV**: (+x, -y)
     - If (x2,y2) lies on an axis, indicate that instead.
3. **Distance Categorization:**
   - **Short Distance**: Less than 5 units
   - **Medium Distance**: 5 to 15 units
   - **Long Distance**: More than 15 units
4. **Verification:**
   - Use this online calculator to confirm your results and attach a screenshot.
     https://www.calculator.net/distance-calculator.html

Sample output	x1: 2 ← enter inputs y1: 3 x2: 5 y2: 7 ========== Distance between *** and *** : 123.45 (123) Integer part: (5) Category: Medium Distance Point (x2, y2) is in Quadrant III
File(s) to submit	J104_6.java J104_6.png J104_6check.png