Java 9/19-21

java 9-12

Java Notes:

JDK→ JRE → JVM

Java Development kit → Java Runtime Environment → Java Virtual Machine

IDE:

eclipse

jDeveloper

Android studios

Ardino

4 types of variables

1. instance variables - non static fields
2. class variables - static fields

Literals

1. boolean (can have two values true or false)
2. integer (numeric values → w/o fractional or exponents—> we will be using decimal numbers) (short , long and double)
3. Floating points (represents a fractional form —- float myFloat = 3.4F; ← end floats with either F/f)

4. character → (char = ‘C’ ;)
5. string → (string myName = “ jennifer j “ ;)

Operators:

Arithemetic Operators → Additional , subtraction, mulitplication, division, make sure floating point is used

assignment operators →

= → a = b

+= — a+=b →

relational operators

== → equal to

!= → not equal to

> → Greater Than

< → less than equal

logical operators

&& (and)

|| (or)

! (not) (bang)

true

true

true

true

true

Unary operators

return 1++ → 1

return 1- → error

return ++23 →

return -65 → -65

int x = 0 →

print (x); 0

print (x—) ; 0 ——————- x now equals -1

print (++x); increment x then print result, which is 0

print (—x); decrement x, which is -1, then print the result

Java output

to output values in console we often use the below options:

- System.out.println(); ← the value inside () goes on its own line
- System.out.print(); ← prints value inside ()

ex: println (’My Age : “ + “ “ + 6);

prints (My Age : 6)

Java Input

- get input from user using a scanner object
- first import the package using
  - import java.util.Scanner;
- then create an object of Scanner;
  - Scanner input = new Scanner(System.in)
- when done, close the scanner object
  - input.close();

Look and do research for input

Java Expression

variables, operators, literals

Java Statements

what is the difference between expressions and statements

Java Blocks

a group of statements enclosed in curly brackets

Java comments

comments are a portion of the program that are completely ignored by the Java complier. they are mainly used to help programmers

Multiline comments

Assignment

take the assignemt we worked on earlier …. expand on it by using

data types, operators, input, output, and comments

submit the link of your github repo…. use comments! comments! comments!!!

Java 9/20

Flow Control

how the flow of a senerero goes

using conditionals

if…else statement

```
if(number > 0){
//code
}
else {
//code
}

//code aftwer
```

if statement

else statement

if else statement …

else if (conditional)….

(you can have as many else if statements)

you do need to close the if statement with an else

think about the else as a default

nested if …else statements

a nested if…else statement happens when there's an if

Switch Statement

allows us to execute a block of code among many alternatives

acts like an if, else, if statements

syntax

break in switch statements

- allows the code to end at the case statement however if you do not insert the break it'll run all of the statements

default → but break after default → kind of like an else statement under if statement

Loops

are used to repeat a block of code. 3 types of loops

1. for
2. while
3. do…while

For loop

used to run a block of code for a certain number if time

//extra points if you can find]

find the max of the two then find the min of the two and position the loop where it reads the smaller value first to the max.

we need to make the first integer the smaller number

Infinite for loops

STAY AWAY FROM THESE!!

For each loop

the java for loop has an alternative syntax that makes it easy to iterate through arrays and collections

While loop

is used to run a specific code until a certain conditions is meet

Do while loop

is similar to the while loop

when to use the for/while loop

When you know the number of iterations needed then you use the for loop.

When you don’t know the number of iterations needed then you use the while loop.

break statement

continue statement

sometimes you may want to skip some statements

```
import java.util.Scanner;

public class UserInputExample {
    public static void main (String [] args){
        Scanner scanner = new Scanner (System.in);
        System.out.print("Enter a first number: ");
        double number1 = scanner.nextDouble();
        //using double data type to store the user's input
        System.out.print("Enter the second number: ");
        double number2 = scanner.nextDouble();
        System.out.print("Enter the third number: ");
        double number3 = scanner.nextDouble();
        System.out.print("Enter the fourth number: ");
        double number4 = scanner.nextDouble();


        scanner.close();
        //remember to close scanner to prevent resource leak!
        //below I am using arithmetic operators
        double sum = number1 + number2 + number3 + number4;
        double difference = number1 - number2 - number3 - number4;
        double product = number1 * number2 * number3 * number4;
        double quotient = number1 / number2 / number3 / number4;
        double modulo = number1 % number2 % number3 % number4;
        // below i am using logical operators
        boolean isEven = (number1 % 2 == 0);
        boolean isPositive = (number1 > 0);

        //displaying my results to the user below
        System.out.println("\nArithmetic Operators:");
```

```
        System.out.println("Sum: " + sum);
        System.out.println("Difference: " + difference);
        System.out.println("Product: " + product);
        System.out.println("Quotient: " + quotient);
        System.out.println("Modulo: " + modulo);

        System.out.println("\nLogical Operators:");
        System.out.println("Is the first number you entered even?..." + isEven);
        System.out.println("Is the the first number you entered positive?..." + isPositive);

    }
}
```

```
import java.util.Scanner;

public class UserInputExample {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter an integer: ");
        int integerInput = scanner.nextInt();

        System.out.print("Enter a floating-point number: ");
        double doubleInput = scanner.nextDouble();

        System.out.print("Enter a character: ");
        char charInput = scanner.next().charAt(0);

        System.out.print("Enter a string: ");
        String stringInput = scanner.next();

        System.out.print("Enter a relational operator (e.g., >, <, ==): ");
        String relationOperator = scanner.next();

        System.out.print("Enter a logical operator (e.g., &&, ||): ");
        String logicalOperator = scanner.next();

        System.out.print("Enter a unary operator (e.g., ++, --, !): ");
        String unaryOperator = scanner.next();

        scanner.close();

        // Evaluate a unary expression
        double unaryResult = 0;
        if (unaryOperator.equals("++")) {
            unaryResult = integerInput + 1;
        } else if (unaryOperator.equals("--")) {
```

```
            unaryResult = integerInput - 1;
        } else if (unaryOperator.equals("!")) {
            unaryResult = (doubleInput != 0) ? 0 : 1;
        }

        // Evaluate a relational expression
        boolean relationResult = false;
        switch (relationOperator) {
            case ">":
                relationResult = integerInput > doubleInput;
                break;
            case "<":
                relationResult = integerInput < doubleInput;
                break;
            case "==":
                relationResult = integerInput == doubleInput;
                break;
            // Add more cases for other relational operators if needed
        }

        // Evaluate a logical expression
        boolean logicalResult = false;
        if (logicalOperator.equals("&&")) {
            logicalResult = (integerInput > 0) && (doubleInput > 0);
        } else if (logicalOperator.equals("||")) {
            logicalResult = (integerInput > 0) || (doubleInput > 0);
        }

        // Displaying the new inputs and the results
        System.out.println("\nUser Inputs:");
        System.out.println("Integer Input: " + integerInput);
        System.out.println("Double Input: " + doubleInput);
        System.out.println("Character Input: " + charInput);
        System.out.println("String Input: " + stringInput);
        System.out.println("Relational Operator: " + relationOperator);
        System.out.println("Logical Operator: " + logicalOperator);
        System.out.println("Unary Operator: " + unaryOperator);

        // Display the results of unary, relational, and logical expressions
        System.out.println("\nResults:");
        System.out.println("Result of Unary Expression: " + unaryResult);
        System.out.println("Result of Relational Expression: " + relationResult);
        System.out.println("Result of Logical Expression: " + logicalResult);
    }
}
```