TOPICS FOR FINAL & THEIR EXAMPLE CODES:

String & Char

In Java, String is a datatype used to define the words or sentences to a variable. We can take string inputs from the user and can also perform multiple different tasks with String methods too.

Syntax: String Name = “Shehraz”;

For Input Single Word: String Name = sc.next();

For Input Multiple Words or Sentence: String Name = sc.nextLine();

In Java, Char is a datatype used to declare only a single character.

Syntax: String A = ‘A’;

For Input only single Character we use charAt(0): String A = sc.next().charAt(0);

To print the first character of the string name (in this case, the first S from "Shehraz"), you can use the charAt() method. Here's how:

String name = "Shehraz";

System.out.println(name.charAt(0)); // Prints 'S'

String Methods:

1. concat() Method:

The concat() method is used to combine two strings. It returns a new string that is the concatenation of the original string and the specified string.

Syntax:

String result = string1.concat(string2);

Example:

String firstName = "Shehraz ";

String lastName = "Sarwar";

String fullName = firstName.concat(lastName);

System.out.println(fullName); // Output: Shehraz Sarwar

2. length() Method:

The length() method is used to find the number of characters in a string, including spaces.

Syntax:

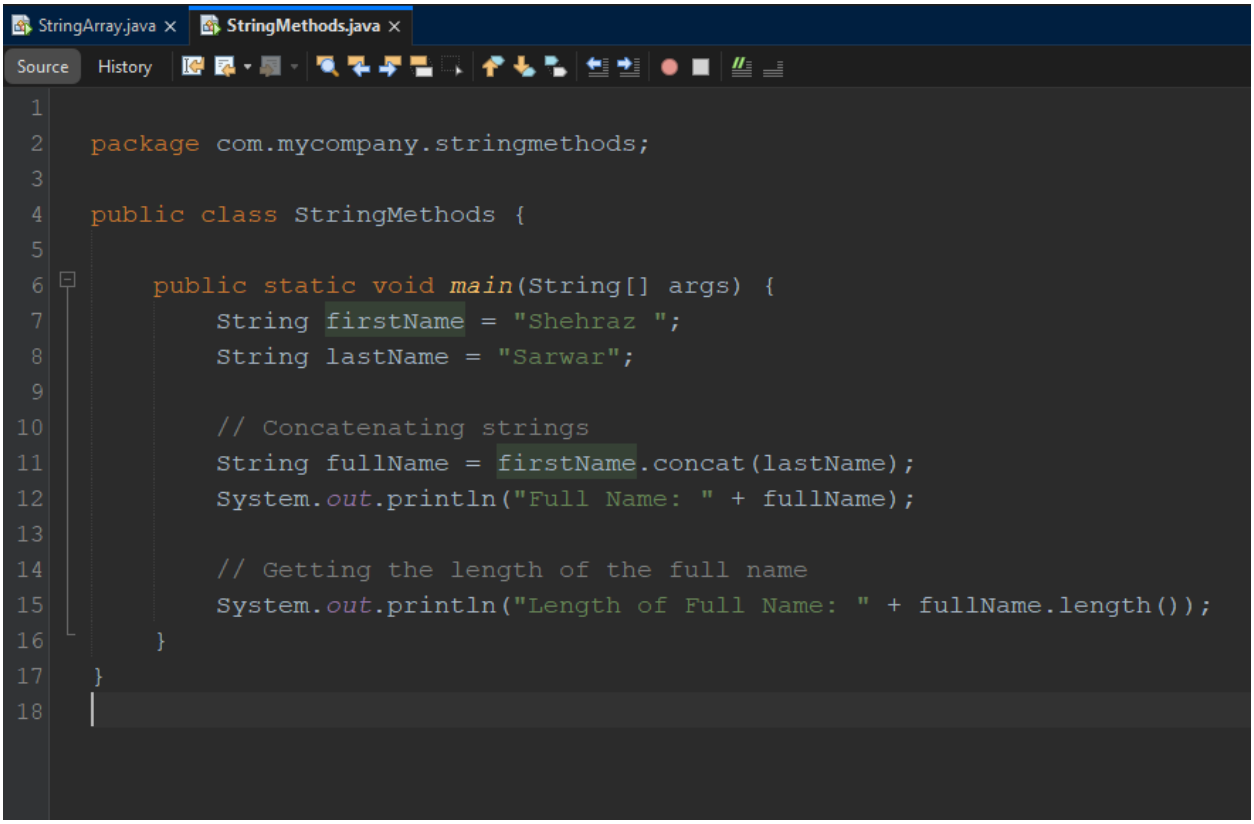
int length = string.length();

Example:

String name = "Shehraz";

System.out.println("Length of the name is: " + name.length()); // Output: 7

Q. Write a program to combine to strings using concat() method and also find the total length of it after combining it using .length() method.



***OUTPUT*:**

Full Name: Shehraz Sarwar

Length of Full Name: 14

Enum

Enum:

In Java, an enum is a special class that represents a group of constants (unchangeable variables). Enums can be used to define a fixed set of related constants, making code more readable and type-safe.

Syntax: public enum class_name { }

Q) Write a program to demonstrate the use of Enum.

The Below program is used to declare the Enum class for books.

Enum Class Code:

```
3
4   public enum EnumClass {
5       DSA("John",200),
6       PF("Mick",100),
7       OOP("Calab",50);
8
9       String author;
10      int price;
11
12      EnumClass(String author, int price){
13          this.author = author;
14          this.price = price;
15      }
16  }
```

Main Class Code:

```java

package com.mycompany.enums;

public class Main {
    public static void main(String args[]){

        System.out.println("Printing Only One Book Data: ");
        System.out.println("Book Name: " + EnumClass.DSA);
        System.out.println("Author: " + EnumClass.DSA.author);
        System.out.println("Price: " +EnumClass.DSA.price);

        System.out.println("\nUsing Loop To Print All: ");
        for(EnumClass i: EnumClass.values()){
            System.out.println("Book Name: "+ i + "\nAuthor: "+ i.author + "\nPrice: "+ i.price);
        }
}
}
```

OUTPUT:

```
Output - Run (Main)
cd G:\Sheeraz Documents\Study Materials\Java\F
Printing Only One Book Data:
Book Name: DSA
Author: John
Price: 200

Using Loop To Print All:
Book Name: DSA
Author: John
Price: 200
Book Name: PF
Author: Mick
Price: 100
Book Name: OOP
Author: Calab
Price: 50
```

Arrays

In Java, arrays are used to store a collection of values in a single variable, rather than having multiple individual variables for each value. An array allows you to store multiple values of the same type in a contiguous memory block, making it more efficient for working with collections of data.

Key Points about Arrays in Java:

1. Fixed Size: Once an array is created, its size cannot be changed. The size is defined when the array is initialized.
2. Homogeneous Data: Arrays store elements of the same data type (e.g., int[], String[]).
3. Indexing: Arrays use zero-based indexing, meaning the first element is at index 0.

1 Dimensional Array:

1) for hardcode values:

Syntax: int[ ] ArrayName = {1,2,3,4,5};

2) for user input:

Syntax: int[ ] ArrayName = new int[5];

ArrayName[0] = sc.nextInt();

We can use A Loop to get all values from the user.

```
for(int i = 0; i < 5; i++){
        ArrayName[i] = sc.nextInt();
}
```

Q) Write a program to demonstrate the use of 1-D Array.

The Below program is first take integer inputs from the user and then find the location.

https://drive.google.com/file/d/1gLP4DQZl5jWqgYReaP1Ula8RaXOy2cAx/view?usp=sharing

Similarly For String Here:

1) for hardcode values:

Syntax: String[] ArrayName = {"Sheeraz Sarwar", "Xyz", "Abc", "ETC"};

2) for user input:

Syntax: String[ ] ArrayName = new String[3];

ArrayName[0] = sc.next();

We can use A Loop to get all values from the user.

```
for (int i = 0; i < ArrayName.length; i++) {
    ArrayName[i] = sc.next();
    // ArrayName[i] = sc.nextLine(); // Use it to get the whole sentence as an input
}
```

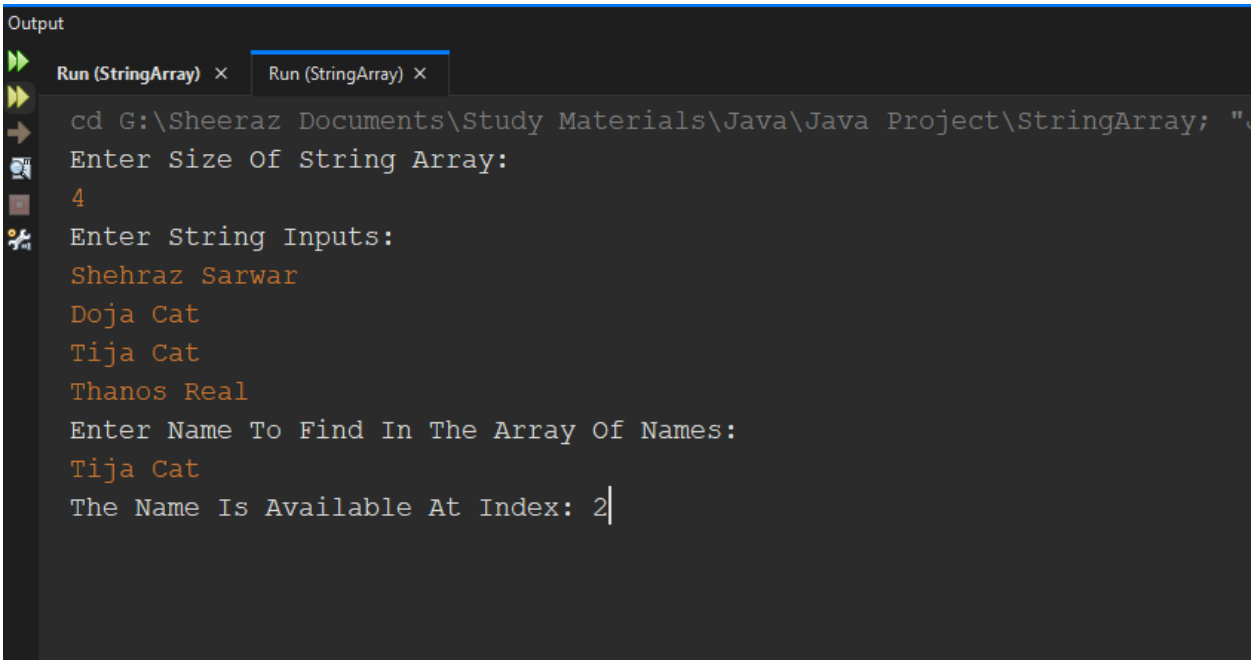
The Below program is first take String full name inputs from the user and then find the location.

Main Class Code:

```
import java.util.Scanner;


public class StringArray {


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


        System.out.println("Enter Size Of String Array: ");
        int size = sc.nextInt();
        sc.nextLine(); // Consume the leftover newline Which is "enter" only need to write it here if we are taking full names from the user
```

```
        System.out.println("Enter String Inputs: ");

        String[] Names = new String[size];

        for(int i = 0; i < size; i++){
            Names[i] = sc.nextLine();
        }

        System.out.println("Enter Name To Find In The Array Of Names: ");
        String Target = sc.nextLine();

        int found = 0;
        for(int i = 0; i < Names.length; i++){
            if(Names[i].equals(Target)){
                System.out.print("The Name Is Available At Index: "+ i);
                found++;
            }
        }

        if(found == 0){
            System.out.print("The Name Is Not Available");
        }
    }
}
```

OUTPUT:



The Below program is used to take single name inputs without spaces and rest of the code logic is same as above.

Main Class Code:

```
import java.util.Scanner;


public class StringArray {


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


        System.out.println("Enter Size Of String Array: ");
        int size = sc.nextInt();


        String[] Names = new String[size];
```

```
        System.out.println("Enter String Inputs: ");

        for(int i = 0; i < size; i++){
            Names[i] = sc.next();
        }

        System.out.println("Enter Name To Find In The Array Of Names: ");
        String Target = sc.next();

        int found = 0;
        for(int i = 0; i < Names.length; i++){
            if(Names[i].equals(Target)){
                System.out.print("The Name Is Available At Index: "+ i);
                found++;
            }
        }

        if(found == 0){
            System.out.print("The Name Is Not Available");
        }
    }
}
```

***OUTPUT*:**

Output will be the same as before but this time it will only take first names as an input not full name.

Q) Write a program to sort the array using Bubble Sort.

Main Class Code:

```
import java.util.Scanner;

public class BubbleSort {

    public static void printarray(int[] arr){
        for(int i = 0; i < arr.length; i++){
            System.out.print(arr[i] + " ");
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Size Of Array: ");
        int n = sc.nextInt();

        int[] arr = new int[n];

        System.out.println("Enter Elements Of Array: ");
        for(int i = 0; i < n; i++){
            arr[i] = sc.nextInt();
        }

        //Time Complexity = O(n^2)
        //Bubble Sort
        for(int i = 0; i < n-1; i++){
```

```
            for(int j = 0; j < n-i-1; j++){
                if(arr[j] > arr[j+1]){
                    //Swapping here
                    int temp = arr[j];
                    arr[j] = arr[j+1];
                    arr[j+1] = temp;
                }
            }
        }
        System.out.println("Sorted Array: ");
        printarray(arr); //I am using function here to print the array

    }
}
```

OUTPUT:

```
cd G:\Sheeraz Documents\Study Mate
Enter Size Of Array: 5
Enter Elements Of Array:
7 8 3 1 2
Sorted Array:
1 2 3 7 8
```

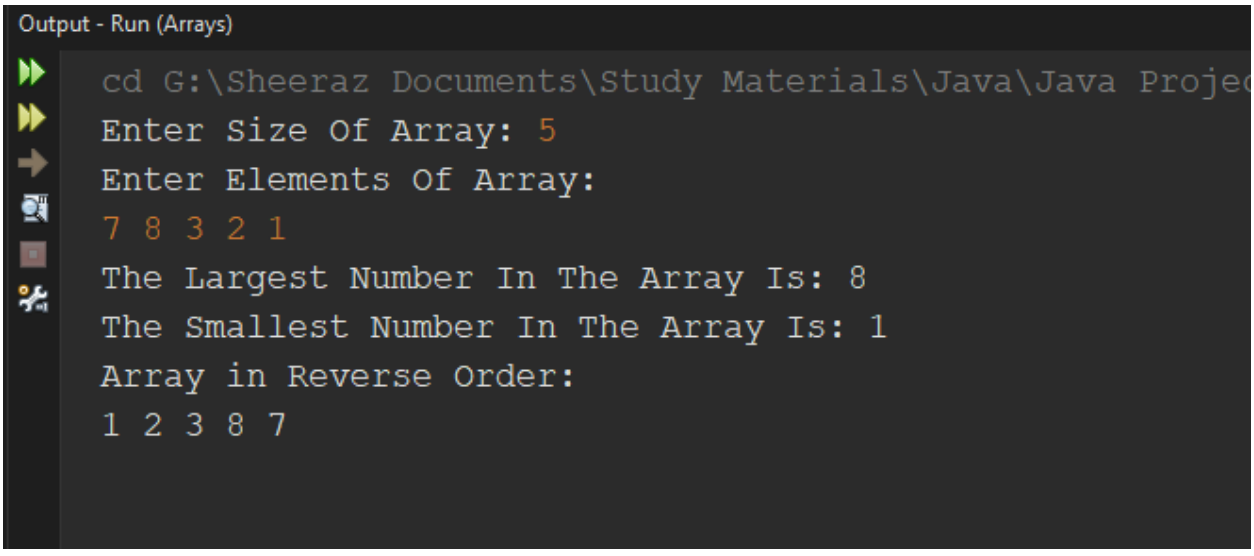
Q) Write a program to find the largest and smallest element in an array and also print the array in reverse order in the end.

Main Class Code:

```
import java.util.Scanner;


public class Arrays {


    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Size Of Array: ");
        int n = sc.nextInt();


        int[] arr = new int[n];


        System.out.println("Enter Elements Of Array: ");
        for(int i = 0; i < n; i++){
            arr[i] = sc.nextInt();
        }


        int largest = arr[0];
        int smallest = arr[0];


        for(int i = 1; i < arr.length; i++){
            if(arr[i] > largest){
                largest = arr[i];
            }
            if(arr[i] < smallest){
```

```
            smallest = arr[i];
        }
    }

    System.out.println("The Largest Number In The Array Is: " + largest);
    System.out.println("The Smallest Number In The Array Is: " + smallest);

    System.out.println("Array in Reverse Order: ");
    //arr.length - 1 because .length method gives us the size of the array which is 5 but index always starts from 0
    for(int i = arr.length-1; i >= 0; i--){
        System.out.print(arr[i] + " ");
    }
  }
}
```

***OUTPUT*:**



Vector

In Java, a Vector is a class in the java.util package that implements a dynamic array. It can grow or shrink in size as needed to accommodate elements. Like arrays, Vectors are used to store objects, but they are more versatile since they dynamically resize.

To use vector must implement this first.

import java.util.Vector;

To use Iterator with Vector must implement this first.

import java.util.Iterator;

- **Here How You Can Declare Vectors in Java:**

The Below program use all Vector Methods. Run and analyze this program on your code editor and you are done with vector 😊

Main Class Code:

```
import java.util.Scanner;
import java.util.Vector;
import java.util.Iterator;

public class Vectors {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        Vector<String> vec = new Vector<>();
        vec.add("Shehraz");  //or vec.addElement() does the same thing
        vec.addElement("Sarwar");
        System.out.println(vec);
        System.out.println(vec.size());        //printing size of vector (no of elements in it)
        System.out.println(vec.capacity());   //printing capacity of vector (by default it's always 10)

        Vector<String> vec2 = new Vector<>();
        vec2.addAll(vec);      //copying all elements of vec into vec2
        vec2.add("Thanos");
        System.out.println(vec2);

        //Adding new string Jake in place of 0 index
```

```
vec.add(0,"Jake");      //this will place Jake at 0 and shift other two string to the 1 and 2 index (shift right)

System.out.println(vec);


//this will do the same thing like vec.add(0,"Jake")

vec.insertElementAt("Mike", 1); //only syntax and order is change rest it does the same work like vec.add(0,"Jake") done

System.out.println(vec);


//here how you can get position of any element

System.out.println(vec.indexOf("Shehraz"));


//here how you can get any element using its index

System.out.println(vec.get(2));


//here how you can check if the element exists in the vecotr or not

System.out.println(vec.contains("Shehraz")); //its a boolean funtion it will return only true or false


//here how you can replace any element in the vector

vec.set(2,"Lisa");

System.out.println(vec);


//this will remove element at index 2

vec.remove(2);  //or removeElementAt(2) do the same thing

System.out.println(vec);


//this will clear the whole vector

vec.clear();   // or vec.removeAll() do the same thing
```

```
System.out.println(vec);

//these two will print the first element and last element of the second vecotr (vec2)
//since vec is clear we cant print its elements because there are no elements are left now in it
System.out.println(vec2.firstElement());
System.out.println(vec2.lastElement());

//Below are the different ways to loop through the vectors but it's recommended to use 1st and 3rd way as its easier

//1)
//below loop will print all index and there specified elements both
for(int i = 0; i < vec2.size(); i++){
    System.out.println(i +") "+ vec2.get(i));
}

//2)
//below loop will print all elements of the vector through Iterator but without index
Iterator<String> iterate = vec2.iterator();
while(iterate.hasNext()){
   System.out.println(iterate.next());
}

//3)
//below loop will print all elements of the vector but without index
//it is also work as iterator but with less complex syntax so its recommended to use this
for(String i : vec2){   //int i: vec2 if vec2 is of type Integer
   System.out.println(i);
}
```

```
        //Here how you can take input from the user into vector
        System.out.print("Enter Size Of Vector: ");
        int size = sc.nextInt();

        Vector<String> newVec = new Vector <>(size);

        System.out.println("Enter Names: ");
        for(int i = 0; i < size; i++){
           newVec.add(i,sc.next());
           /* OR
           String names = sc.next();
           newVec.add(i,names);
           */
        }

        System.out.println("New Vector Created: " + newVec);

    }
}
```

***OUTPUT*:**

```
[Shehraz, Sarwar]
2
10
[Shehraz, Sarwar, Thanos]
[Jake, Shehraz, Sarwar]
[Jake, Mike, Shehraz, Sarwar]
```

2

Shehraz

true

[Jake, Mike, Lisa, Sarwar]

[Jake, Mike, Sarwar]

[]

Shehraz

Thanos

0) Shehraz

1) Sarwar

2) Thanos

Shehraz

Sarwar

Thanos

Shehraz

Sarwar

Thanos

Enter Size Of Vector: 2

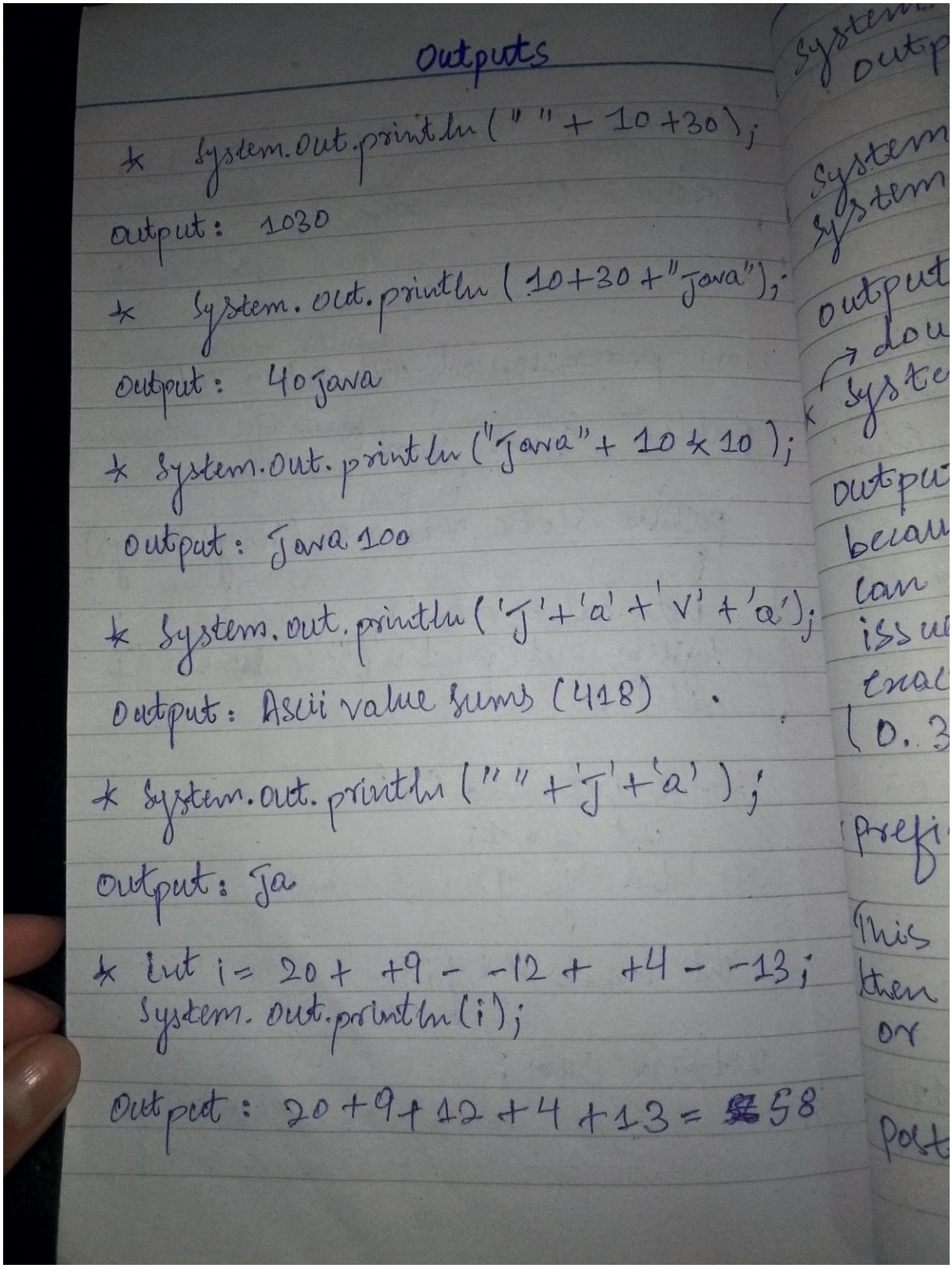
Enter Names:

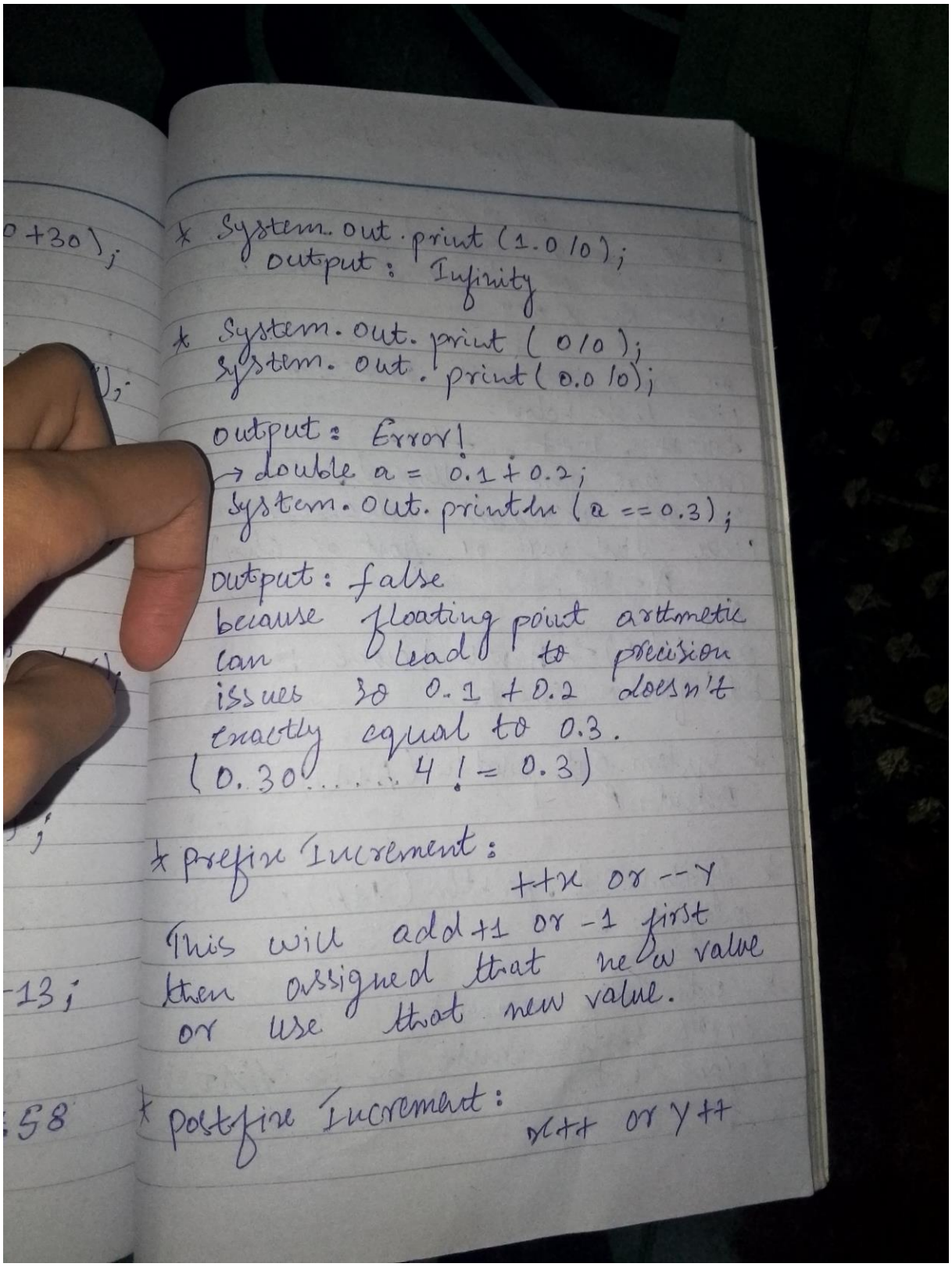
Shehraz

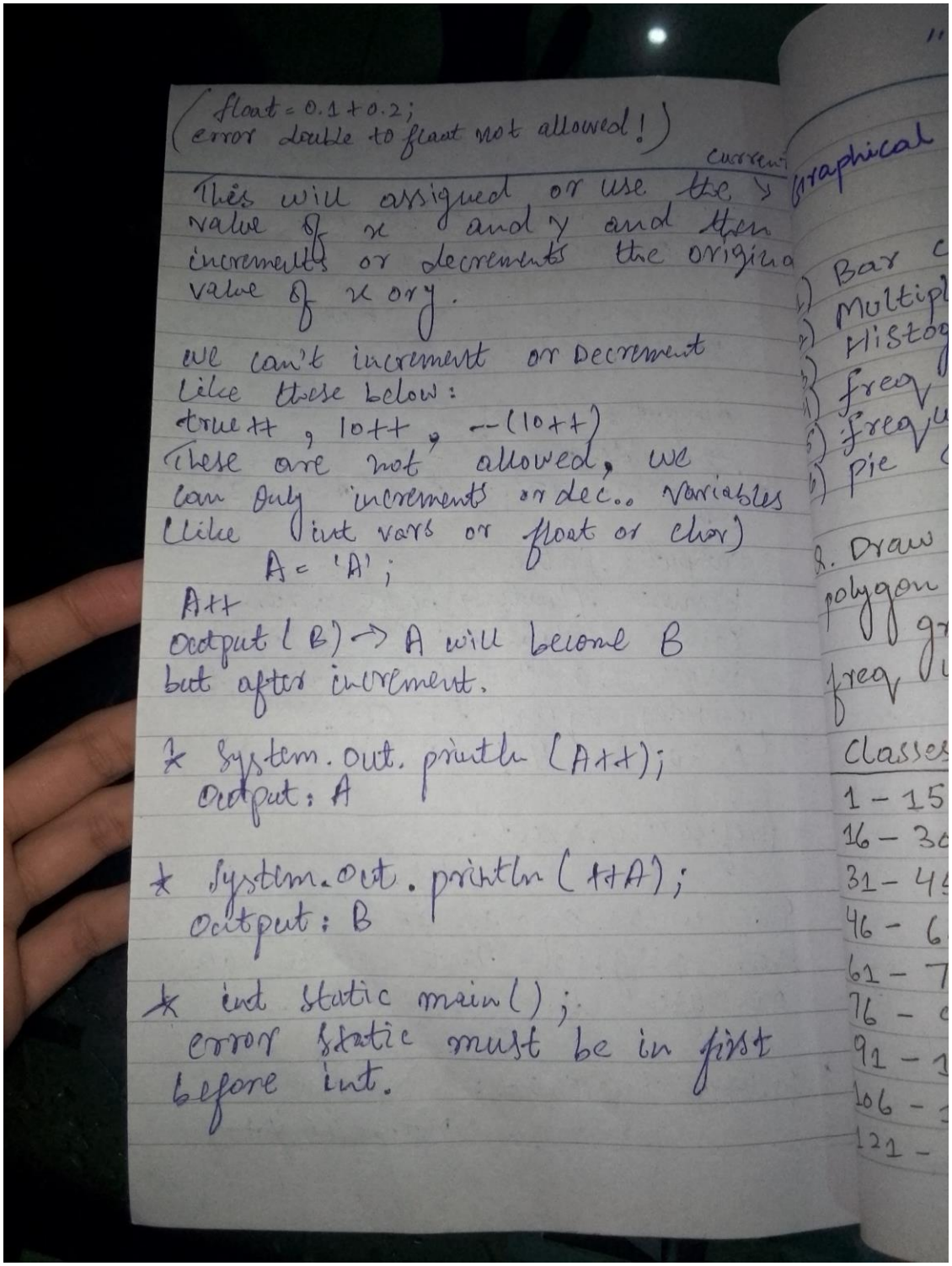
Sarwar

New Vector Created: [Shehraz, Sarwar]

Some Important Outputs:




* System.out.print(1.0/0);
output: Infinity
* System.out.print(0/0);
System.out.print(0.0/0);
output: Error!
→ double a = 0.1+0.2;
System.out.println(a==0.3);
output: false
because floating point arthmetic can lead to precision issues so 0.1+0.2 doesn't exactly equal to 0.3.
(0.30000000000000004 != 0.3)
* prefix Increment:
++x or --y
This will add +1 or -1 first then assigned that new value or use that new value.
* postfix Increment:
x++ or y++


(float = 0.1 + 0.2;
(error Double to float not allowed!)

This will assigned or use the current value of x and y and then increments or decrements the original value of x or y.

we can't increment or Decrement like these below:
true++ , 10++ , --(10++)
These are not allowed, we can only increments or dec.. variables (like int vars or float or char)
A = 'A';
A++
output (B) → A will become B but after increment.

* System.out.println(A++);
output: A

* System.out.println(++A);
output: B

* int static main();
error static must be in first before int.

graphical

1) Bar c
2) Multipl
3) Histog
4) freq
5) frequ
6) pie

8. Draw
polygon
freq

Classes
1 - 15
16 - 30
31 - 45
46 - 6
61 - 7
76 - 9
91 - 1
106 -
121 -

Written By Shehraz Sarwar
https://www.linkedin.com/in/shehraz-sarwar-321394247/