NATIONAL UNIVERSITY
551 M.F. JHOCSON ST., SAMPALOC, MANILA

COMPUTER ENGINEERING
CPOOPG2L: Object Oriented Programming

NATIONAL UNIVERSITY
COLLEGE OF ENGINEERING

Name/Section	**Seth Marcus Martin** **COE211**	**Rating**
Date performed	**December 16, 2022**	
Date submitted		
Professor	**Eng. Marlon Bagara**	

ACTIVITY NO. 4
ARRAY

I. OBJECTIVES:

This activity aims to:

1. Discuss the concept of array and make an analogy in real life situation and specify the size, index, and elements of an array.
2. Recognize the syntax on how to use one and two dimensional array and how to access data to and from it
3. Declare two different arrays as parallel.

LEARNING OUTCOMES (LOs) *At the end of the activity, the students should be able to:*	**COURSE LEARNING OUTCOMES (CLOs)**				
	1	2	3	4	5
1. Write a program that will store data in a one dimensional and two dimensional arrays.		•	•		
2. Manipulate data stored in the array.		•	•		

COURSE LEARNING OUTCOMES (CLOs)

1. Understand the fundamental concept of OOP through Java programming
2. Write programs using console and dialog box.
3. Apply the concept of iterative, control, and array structure programming..
4. Construct classes, objects, methods and constructor.
5. Write programs in GUI environment

II. SOFTWARE/HARDWARE/EQUIPMENT NEEDED:

1. Eclipse
2. Computer unit

III. SAFETY GUIDELINES:

1. Make sure you have both an adjustable table and chair so that ergonomic accommodations can be made for each person using the computer.
2. The computer screen should be front and centre so neck turning is unnecessary.
3. Keep your lab space clean and organized.
4. Clean your lab bench and equipment, and lock the door before you leave the laboratory.
5. Never eat, drink, or smoke while working in the laboratory.
6. DO NOT TOUCH ANYTHING WITH WHICH YOU ARE NOT COMPLETELY FAMILIAR!!! It is always better to ask questions to laboratory technicians or your instructors than to risk harm to yourself or damage to the equipment.

IV. THEORY

An array is an object that is a collection of fixed number of components, called elements, where all the components are of the same data type. Array elements are accessed through array index.

Syntax: one-dimensional array

```
int [ ] arrayName = new int [ arraysize];
```

example:

```
int arrayNum = new int[5];
String arrayName = new String[5];
```

Syntax: two-dimensional array

```
int [ ] arrayName = new int [ number of rows] [number of columns];
```

example:

```
int arrayNum = new int[3][4];
String arrayName = new String[3][3];
```

V. PROCEDURE:

1. Write the program below which is to store and display numbers in the array. Write comments after each line to indicate the operation of the statement..

NATIONAL UNIVERSITY
551 M.F. JHOCSON ST., SAMPALOC, MANILA

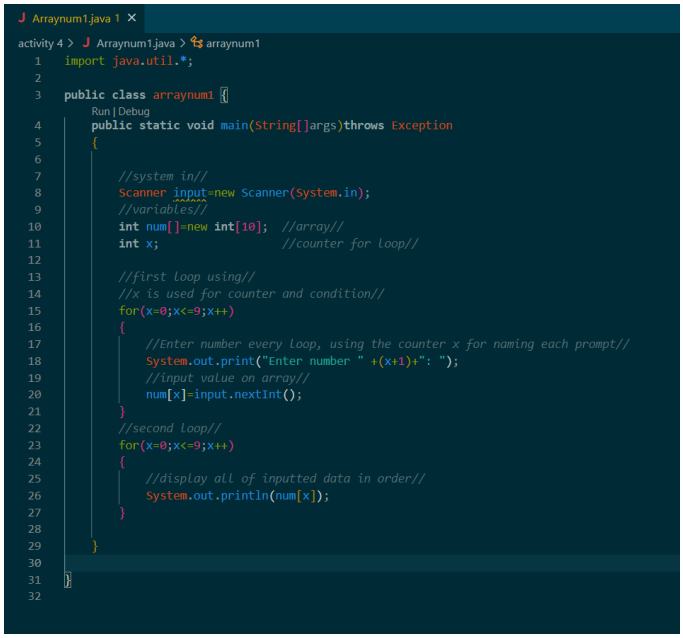
COMPUTER ENGINEERING
CPOOPG2L: Object Oriented Programming

NATIONAL UNIVERSITY
COLLEGE OF ENGINEERING

Program1:

```
import java.io.*;
import java.util.*;

public class ArrayNum2 {
      public static void main(String[]args)throws Exception
      {
            Scanner input=new Scanner(System.in);
            int num[]=new int[10];
            int x;

            for(x=0;x<=9;x++)
            {
                  System.out.print("Enter    number"
+(x+1)+": ");
                  num[x]=input.nextInt();
            }
            for(x=0;x<=9;x++)
            {
                  System.out.println(num[x]);
            }

      }

}
```

Output



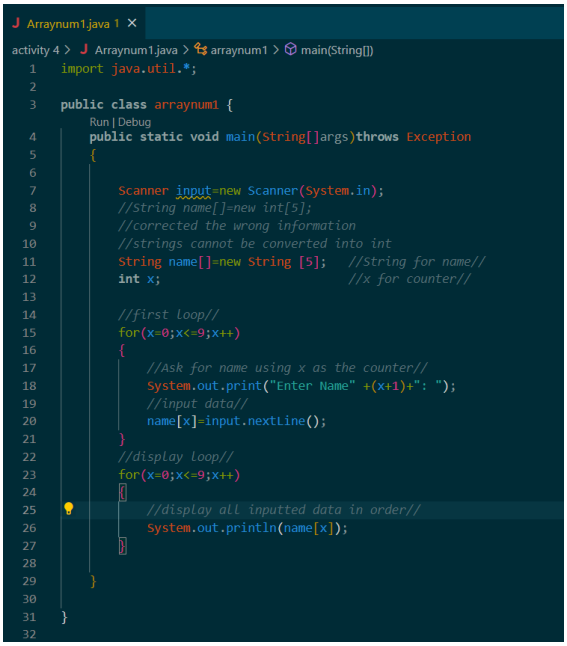

```
Enter number 1: 10
Enter number 2: 30
Enter number 3: 20
Enter number 4: 40
Enter number 5: 60
Enter number 6: 50
Enter number 7: 80
Enter number 8: 90
Enter number 9: 40
Enter number 10: 30
10
30
20
40
60
50
80
90
40
30
```

2. A program that store and display names in an array. Write comments after each line to indicate the operation of the statement.

Program2:

```
import java.io.*;
import java.util.*;

public class ArrayName {
	public static void main(String[]args)throws Exception
	{
		Scanner input=new Scanner(System.in);
		String name[]=new int[5];
		int x;
		for(x=0;x<=9;x++)
		{
			System.out.print("Enter Name" +(x+1)+": ");
			name[x]=input.nextLine();
		}
		for(x=0;x<=9;x++)
		{
			System.out.println(name[x]);
		}

	}

}
```

Output




```
Enter Name1: Seth
Enter Name2: Marcus
Enter Name3: Martin
Enter Name4: Marlon
Enter Name5: Bagara
Enter Name6: John
Enter Name7: Jacob
Enter Name8: Jayson
Enter Name9: Jennifer
Enter Name10: Justin
Seth
Marcus
Martin
Marlon
Bagara
John
Jacob
Jayson
Jennifer
Justin
```

3. Illustrate the output for the given program. Write comments after each line to indicate the operation of the statement.

Program3

```
Import java.io.*;
import java.util.*;

public class ArrayMulti{
          public static void main(String[]args)throws Exception
          {
                    Scanner input=new Scanner(System.in);
                    int num[][]=newint[3][4];
                    int i,j;


                    for(i=0;i<3;i++)
                    {
                              for(j=0;j<=3;j++)
                              {
                                        System.out.print("Enter number: ");
                                        num[i][j]=input.nextInt();
                              }
                    }
                    for(i=0;i<3;i++)
                    {
                              for(j=0;j<=3;j++)
                              {
                                        System.out.print(num[i][j]);

                              }
                    }

          }
}
```

```
Arraynum1.java 1
activity 4 > Arraynum1.java > arraynum1 > main(String[])
  2   import java.util.*;
  3
  4   public class arraynum1{
          Run | Debug
          public static void main(String[]args)throws Exception
  5       {
  6           Scanner input=new Scanner(System.in);
  7           //variables//
  8           int num[][]=new int[3][4]; //one variable with 2 dependent array//
  9           //two counter for nested loops//
 10           int i,j;
 11
 12
 13           //first loop//
 14           for(i=0;i<3;i++)
 15           {
 16               //first nested loop//
 17               for(j=0;j<=3;j++)
 18               {
 19                   //ask for prompt from user//
 20                   System.out.print(s: "Enter number: ");
 21                   //input data for num//
 22                   //num has 2 array which is dependent from each other//
 23                   //the first loop value is num[0][0] which is counted as one array//
 24                   num[i][j]=input.nextInt();
 25               }
 26           }
 27           //second loop//
 28           for(i=0;i<3;i++)
 29           {
 30               //second nested loop//
 31               for(j=0;j<=3;j++)
 32               {
 33                   //display the data//
 34                   System.out.print(num[i][j]);
 35
 36               }
 37           }
 38
```

```
Enter number: 2
Enter number: 4
Enter number: 5
Enter number: 6
Enter number: 7
Enter number: 8
Enter number: 9
Enter number: 8
Enter number: 7
Enter number: 6
123456789876
PS C:\Users\Sky\Desktop\codes>
```

4. Illustrate the output for the given program. Write comments after each line to indicate the operation of the statement.

Program4

```
importjava.io.*;
import java.util.*;

publicclass ArrayMultiString{
          publicstaticvoid main(String[]args)throws Exception
          {
                    Scanner input=new Scanner(System.in);
                    String name[][]=new String[2][3];
                    int i,j;


                    for(i=0;i<2;i++)
                    {
                              for(j=0;j<=2;j++)
                              {
                                        System.out.print("Enter name: ");
                                        name[i][j]=input.nextLine();
                              }
                    }
                    for(i=0;i<2;i++)
                    {
                              for(j=0;j<=2;j++)
                              {
                                        System.out.print(name[i][j]);

                              }
                    }

          }

}
```

Output:

```
Arraynum1.java 1
activity 4 > Arraynum1.java > arraynum1 > main(String[])
1   import java.io.*;
2   import java.util.*;
3
4   public class arraynum1{
        Run | Debug
5       public static void main(String[]args)throws Exception
6       {
7           Scanner input=new Scanner(System.in);
8           String name[][]=new String[2][3];       //one variable for string with 2 array//
9           int i,j;                                //counter//
10
11
12          //first loop//
13          for(i=0;i<2;i++)
14          {
15              //first nested loop//
16              for(j=0;j<=2;j++)
17              {
18                  //input data//
19                  System.out.print(s: "Enter name: ");
20                  //dependent array, read as one variable//
21                  name[i][j]=input.nextLine();
22              }
23          }
24          //second loop//
25          for(i=0;i<2;i++)
26          {
27              //second nested loop//
28              for(j=0;j<=2;j++)
29              {
30                  //display all inputted data//
31                  System.out.println(name[i][j]);
32              }
33          }
34
35      }
36
37  }
```

```
PROBLEMS 1   OUTPUT   DEBUG CONSOLE   TERMINAL

Enter name: Seth
Enter name: Marcus
Enter name: Martin
Enter name: B
Enter name: C
Enter name: D
Seth
Marcus
Martin
B
C
D
PS C:\Users\Sky\Desktop\codes>
```

VI. PROBLEMS/QUESTIONS:

1. Write a program to determine and display the names of the number of PASSED and FAILED out of 10 inputted names with the corresponding grade. Restrict the inputted grade for each name in the range of 50 to 100, in which 75 is the passing grade. If the inputted grade is not on the range, the program will remind the user of a message for the accepted range for the grade and allows the user to re-enter a grade.
   Note:
   a. Use an array in the program.
   b. Display the names of passed and failed with corresponding grade and remark (Remarks is either Passed or Failed)

2. Write a complete program to execute what is asked in each item below:
   a. Input 7 numbers and store it into an array and compute the sum of the 7 inputted numbers at the same time.
   b. Display all the contents of the array.
   c. Display the 1^{st} , 3^{rd}, 5^{th} , and the 7^{th} elements in the array using for loop.
   d. Compute the sum of the 1^{st} , 3^{rd}, 5^{th} , and the 7^{th} elements in the array using for loop.

3. Write a complete program to execute what is asked in each item below:
   a. Input 8 different names and store them into an array
   b. Display all the contents of the array.
   c. Display the 2^{nd} , 4^{th}, 6^{th} , and the 8^{th} elements in the array using for loop.
   d. Compute the sum of the 2^{nd} , 4^{th}, 6^{th} , and the 8^{th} elements in the array using for loop

4. Write a complete program to execute what is asked in each item below:
   a. Using a parallel array, Input 5 different names with the corresponding grade.
   b. Display the names with grade (not sorted).

5. Use a multidimensional array to construct a multiplication table with 5 rows and 10 columns.

VII. INTERPRETATION/ANALYSIS OF DATA:

activity4no1.java | activity4no2.java | activity4no3.java | activity4no4.java | activity4no5.java

activity 4 > activity4no1.java > activity4no1 > main(String[])

```
do{
    //input grade of student//
    System.out.print(s: "Input grade of student: "); y[x]=cl.nextInt();

    //counter for pass or fail//
    if(y[x]<75 && y[x]>=50)
    {
        //if student failed, add count to fail//
        fail=fail+1;
        grade[x]= "FAILED";
        //display student's name if failed//
        System.out.println(name[x]+" failed the exam");
    }

    else if(y[x]>=75&&y[x]<=100)
    {
        //if student passed, add count to pass//
        pass=pass+1;
        grade[x]= "PASSED";
        //display student's name if passed//
        System.out.println(name[x]+" passed the exam");
    }

    //if else statement, nested//
    if (y[x]<50 || y[x]>100)
    {
        //if user input below 50 or above 100//
        System.out.print(s: "Please input grades 50-100\n");
    }
    else{
        //if user input 50-100, proceed with loop//
        System.out.println(x: "\n\nPress Enter to continue");
        //Press enter to proceed with the code//
        try{System.in.read();}
        catch(Exception e){}
        //clear screen to erase previous displayed data//
        clearScreen();
    }
```

Number 1 has a lot of nested codes. It also a reused previous code where I only re-edited that are needed to be added. I also added an extra code that I've learned where I can press enter before continuing, and used one of my previous sub code to clear the screen every prompt used.

I also added the information needed which it tells if the student passed or not.

activity4no1.java | activity4no2.java | activity4no3.java | activity4no4.java | activity4no5.java

activity 4 > activity4no2.java > activity4no2 > main(String[])

```
12      //ask for input for 7 numbers
13      System.out.print(s: "Input a seven numbers: \n");
14      //loop for input//
15      for (x=1;x<=7;x++){
16          System.out.print(x+".Input a numbers: ");inp[x]=cl.nextInt();
17      }
18
19      //formula for sum of seven//
20      inp[9] = inp[1] + inp[2] + inp[3] + inp[4] + inp[5] + inp[6] + inp[7];
21
22      System.out.println(x: "\n-------------------------------------------------\n");
23      //display sum of seven number//
24      System.out.print("Sum of all seven numbers: "+inp[9]);
25      System.out.println(x: "\n\n-------------------------------------------------\n");
26
27
28      //display seven numbers//
29      System.out.print(s: "Display all seven numbers: \n");
30      for (x=1;x<=7;x++){
31          System.out.println(" Number " +x+": "+inp[x]);
32      }
33      System.out.println(x: "\n-------------------------------------------------\n");
34
35      //display specified numbers//
36      System.out.print(s: "Display 1st, 3rd, 5th, 7th numbers: \n");
37      for (x=1;x<=7;x=x+2){
38          System.out.println(" Number " +x+": "+inp[x]);
39      }
40      System.out.println(x: "\n-------------------------------------------------\n");
41
42      //formula for specified numbers//
43      for(x=1;x<=7;x=x+2){
44          inp[10] = inp[10] + inp[x];
45      }
46      System.out.print("Sum of the four numbers: "+ inp[10]);
47      System.out.println(x: "\n\n-------------------------------------------------\n");
48
```

Number 2 is straight forward coding which I just followed the information given to me

There are a lot of alternative coding can be done and having a lot of for loop inside a code is quite tedious, creating a better alternative is much appreciated.

activity4no1.java | activity4no2.java | activity4no3.java | activity4no4.java | activity4no5.java

activity 4 > activity4no3.java > activity4no3 > main(String[])

```java
public class activity4no3 {
    static Scanner cl = new Scanner (System.in);
    Run | Debug
    public static void main(String args[]){

        clearScreen();
        //variables//
        String inp[] = new String[11];
        int x;

        //ask for input for 8 names
        System.out.print(s: "Ask for names: \n");
        //loop for input//
        for (x=1;x<=8;x++){
            System.out.print(x+".Input a Name: ");inp[x]=cl.nextLine();
        }

        System.out.println(x: "\n-----------------------------------------------\n");


        //display eight names//
        System.out.print(s: "Display all eight names: \n");
        for (x=1;x<=8;x++){
            System.out.println("Name " +x+": "+inp[x]);
        }
        System.out.println(x: "\n-----------------------------------------------\n");

        //display specified name//
        System.out.print(s: "Display 2nd, 4th, 6th, 8th Name: \n");
        for (x=2;x<=8;x=x+2){
            System.out.println("Name " +x+": "+inp[x]);
        }
        System.out.println(x: "\n-----------------------------------------------\n");


    }
    //code for clearscreen//
    public static void clearScreen() {
```

Same goes with number 3, just add an extra number for the array, difference is we cannot compute the sum for names since it's not an integer

NATIONAL UNIVERSITY
551 M.F. JHOCSON ST., SAMPALOC, MANILA

NATIONAL UNIVERSITY
COLLEGE OF ENGINEERING

COMPUTER ENGINEERING
CPOOPG2L: Object Oriented Programming

Parallel array is just two different array of integer and strings.

Using the loop function we can ask for the string and integer

And displaying them out using the next loop.

A lot of data used in this code, especially the global variable

Having a global variable that can be used by the main and sub code trees

I've used the sub-code to calculate the needed data for the multiplication table.

Using the main code to display the gathered data for the multi-array

I've also used the function printf so I can manage the formatting for the spacing

So it can have same spacing throughout the output

VIII. CONCLUSION/RECOMMENDATION:

The exercise improved my proficiency in the language at hand and trained my brain to understand assignments and problems more quickly. It was like relearning the languages I had previously acquired. The exercise also made me consider how I could make codes more aesthetically pleasant for both myself and my future coworkers. They would comprehend it better if it is simpler for me to read. Both the concept and the execution of a better code structure are required. Because of so, debugging is simpler.

IX. REFERENCES:

Dale, Nell. (2018). Object-oriented data structures using Java.
Litvin, Maria. (2015). Java methods: : Object-oriented programming and data structures. 3rd Ed.
Baesens, Bart. (2015). Beginning java programming: : The object-oriented approach.
Gaddis, Tony. (2013).Starting out with Java: from control structures through objects. 5TH ed. Boston: Pearson

NATIONAL UNIVERSITY
551 M.F. JHOCSON ST., SAMPALOC, MANILA

COMPUTER ENGINEERING
CPOOPG2L: Object Oriented Programming

NATIONAL UNIVERSITY
COLLEGE OF ENGINEERING

Name/Section	Seth Marcus Martin COE211	Rating
Date performed	December 16, 2022	
Date submitted		
Professor	Eng. Marlon Bagara	

ACTIVITY NO. 5
ARRAY SORT

I. OBJECTIVES:

This activity aims to:

1. Explain the concept of bubble sort.
2. Roleplay the operation of bubble sort.
3. Illustrate the sorting of one-dimensional parallel array

LEARNING OUTCOMES (LOs) *At the end of the activity, the students should be able to:*	COURSE LEARNING OUTCOMES (CLOs)				
	1	2	3	4	5
1. Construct a program to arrange the contents of parallel array		•	•		

COURSE LEARNING OUTCOMES (CLOs)

1. Understand the fundamental concept of OOP through Java programming
2. Write programs using console and dialog box.
3. Apply the concept of iterative, control, and array structure programming.
4. Construct classes, objects, methods and constructor.
5. Write programs in GUI environment

II. SOFTWARE/HARDWARE/EQUIPMENT NEEDED:

1. Eclipse
2. Computer unit

III. SAFETY GUIDELINES:

1. Make sure you have both an adjustable table and chair so that ergonomic accommodations can be made for each person using the computer.
2. The computer screen should be front and center so neck turning is unnecessary.
3. Keep your lab space clean and organized.
4. Clean your lab bench and equipment, and lock the door before you leave the laboratory.

5. Never eat, drink, or smoke while working in the laboratory.
6. DO NOT TOUCH ANYTHING WITH WHICH YOU ARE NOT COMPLETELY FAMILIAR!!! It is always better to ask questions to laboratory technicians or to your instructors than to risk harm to yourself or damage to the equipment.

IV. THEORY

The bubble sort performs pairwise comparisons and to exchange the positions of the pair if they are out of order.

Properties of bubble sort:

1. After one pass through the array, the largest element will be at the end of the array.
2. During the pass if no pair of consecutive entries is out of order.

Implementing bubble sort:

```
for (b=0; b<=9;b++)
{
        for (a=b+1; a<=9;a++)
        {
                if (num[b]>num[a])
                {

                        tempT=num[b];
                        num[b]=num[a];
                        num[a]=tempT;


                }
        }
}
```

V. PROCEDURE:

1. Write the program below to store and display numbers in the array. And sort the elements in ascending order. Write comments after each line to indicate the operation of the statement.

Program1:

```
importjava.io.BufferedReader;
importjava.io.*;
import java.util.*;

publicclass ArrayNum2 {
        publicstaticvoid main(String[]args)throws Exception
        {
```

```
                Scanner input=new Scanner(System.in);
                int num[]=newint[10];
                int x,b,a,tempT;


                for(x=0;x<=9;x++)
                {
                        System.out.print("Enter number" +(x+1)+": ");
                        num[x]=input.nextInt();
                }
                for(x=0;x<=9;x++)
                {
                        System.out.println(num[x]);
                }

                //sort the numbers
                for (b=0; b<=9;b++)
                {
                        for (a=b+1; a<=9;a++)
                        {
                                if (num[b]>num[a])
                                {
                                        tempT=num[b];
                                        num[b]=num[a];
                                        num[a]=tempT;
                                }
                        }
                }

                System.out.print("\n\narranged number\n" );
                for(x=0;x<=9;x++)
                {
                        System.out.println(num[x]);
                }

        }

}
```

Output


```
2
3
52
1
63
45
85
88
43
65


arranged number
1
2
3
43
45
52
63
65
85
88
```

VI. PROBLEMS/QUESTIONS:

1. Write a complete program to execute what is asked in each item below:
    a. Input 7 numbers and store them into an array and compute the sum of the 7 inputted numbers.
    b. Display the sorted elements in descending order

2. Write a complete program to execute what is asked in each item below:
    a. Input 5 different names and store them into an array
    b. Display the sorted elements in descending order

3. Write a complete program to execute what is asked in each item below:
    a. Using a parallel array, Input 5 different names with the corresponding grade.
    b. Display the names with grade (not sorted).
    c. Sort the names according to their grade in ascending order.
    d. Display the names with grade (sorted)

VII. INTERPRETATION/ANALYSIS OF DATA:

```java
//input number Loop//
for(x=1;x<=7;x++){
    System.out.print(x+ ". Input number: ");
    no[x]=cl.nextInt();
}

//sum formula//
no[8] = no[2] + no[3] + no[4] + no[5] + no[1] + no[6] + no[7];

System.out.println(x: "\n\n-----------------------------------------------\n\n");

//display sum//
System.out.print("Sum of all numbers: "+no[8]);

//loop for arrangement//
for (x=1; x<=7;x++)
{
    for (y=x+1; y<=7;y++)
    {
        if (no[x]>no[y])
        {
            //temp variable use for sorting lowest to highest//
            z=no[x];
            no[x]=no[y];
            no[y]=z;
        }
    }
}

System.out.println();
System.out.print(s: "Arranged numbers (Descending): ");
System.out.println();
for(x=7;x>=1;x--){
    System.out.println(x+ ". Number: "+no[x]);
}

System.out.println(x: "\n\n-----------------------------------------------\n\n");
```

It reused the previous codes, it's pretty much straightforward

I also referenced the code for arranging the codes

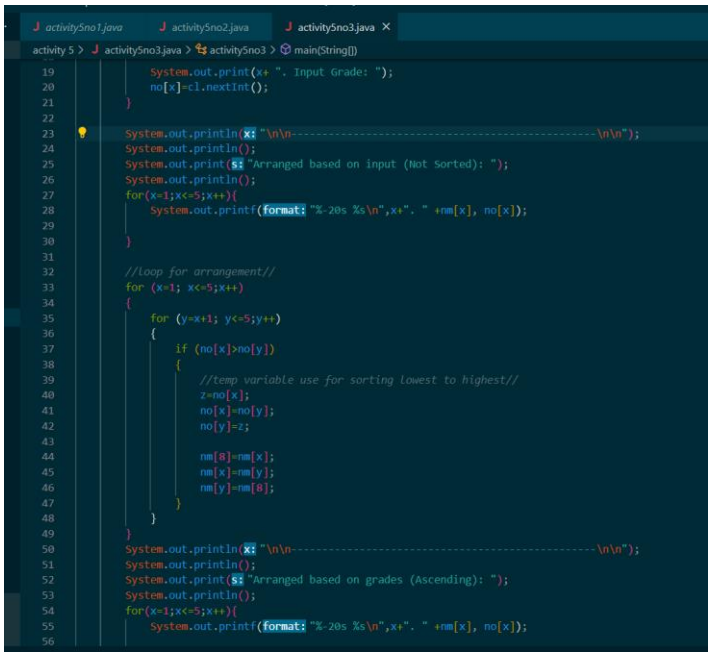
I used the counting of x-- so I can reverse the arrangement of ascending, but I can also change the condition inside the if statement with the same outcome.

activity5no1.java | activity5no2.java | activity5no3.java

activity 5 > activity5no2.java > activity5no2 > main(String[])

```
10          String z;
11          int x,y;
12
13          //input name loop//
14          for(x=1;x<=5;x++){
15              System.out.print(x+ ". Input Name: ");
16              no[x]=cl.nextLine();
17          }
18
19          System.out.println(x: "\n\n-----------------------------------------------\n\n");
20
21          //loop for arrangement//
22          for (x=1; x<=5;x++)
23          {
24              for (y=x+1; y<=5;y++)
25              {
26                  if (no[x].compareTo(no[y])>0)
27                  {
28                      //temp variable use for sorting lowest to highest//
29                      z=no[x];
30                      no[x]=no[y];
31                      no[y]=z;
32                  }
33              }
34          }
35
36          System.out.println();
37          System.out.print(s: "Arranged names (Descending): ");
38          System.out.println();
39          for(x=5;x>=1;x--){
40              System.out.println(x+ ". "+no[x]);
41          }
42
43          System.out.println(x: "\n\n-----------------------------------------------\n\n");
44
45      }
46
47      //code for clearscreen//
```

Same goes with the names, the difference is I have to use a string for the temporary variable so I can store the data inside one of the array to the temporary variable



```
            System.out.print(x+ ". Input Grade: ");
            no[x]=cl.nextInt();
        }

        System.out.println(x: "\n\n-----------------------------------------------\n\n");
        System.out.println();
        System.out.print(s: "Arranged based on input (Not Sorted): ");
        System.out.println();
        for(x=1;x<=5;x++){
            System.out.printf(format: "%-20s %s\n",x+". " +nm[x], no[x]);

        }

        //loop for arrangement//
        for (x=1; x<=5;x++)
        {
            for (y=x+1; y<=5;y++)
            {
                if (no[x]>no[y])
                {
                    //temp variable use for sorting lowest to highest//
                    z=no[x];
                    no[x]=no[y];
                    no[y]=z;

                    nm[8]=nm[x];
                    nm[x]=nm[y];
                    nm[y]=nm[8];
                }
            }
        }
        System.out.println(x: "\n\n-----------------------------------------------\n\n");
        System.out.println();
        System.out.print(s: "Arranged based on grades (Ascending): ");
        System.out.println();
        for(x=1;x<=5;x++){
            System.out.printf(format: "%-20s %s\n",x+". " +nm[x], no[x]);
```

Since I have a parallel array at hand and the needed sorting is relying to the integer, I simultaneously added the loop for names inside the loop for the integer so once the numbers are arranged the name will be arranged with it

VIII. CONCLUSION/RECOMMENDATION:

The exercise increased my fluency with the language at hand; it was similar to relearning past languages; it also trained my brain to assess assignments and problems more fluently. The task also prompted my thoughts on codes and how I may make them more visually beautiful, not just for myself but also for my future coworkers. It is simpler for them to comprehend if it is easier for me to read. The concept of a better code structure is as important as how it can be run. Debugging is much easier as a result.

IX. REFERENCES:

Dale, Nell. (2018). Object-oriented data structures using Java.
Litvin, Maria. (2015). Java methods: : Object-oriented programming and data structures. 3rd Ed.
Baesens, Bart. (2015). Beginning java programming: : The object-oriented approach.
Gaddis, Tony. (2013).Starting out with Java: from control structures through objects. 5TH ed. Boston: Pearson