



Al-Mustaqbal University College (MUC)

Intelligent Medical Systems Department

Computer Programing II

Lecturer: Maytham N. Meqdad

Lecture 3

A program in Java using Array to calculate students' grades "failing" "pass" "average" "good" "very good" "excellent"

```
import java.util.Scanner;

public class Grades {
    public static void main(String[] args) {
        int[][] grades = new int[4][7];
        Scanner input = new Scanner(System.in);

        // loop through each student
        for (int i = 0; i < grades.length; i++) {
            System.out.println("Enter grades for Student " + (i+1) + ":");

            // loop through each grade for the current student
            for (int j = 0; j < grades[i].length; j++) {
                System.out.print("Grade " + (j+1) + ": ");
                grades[i][j] = input.nextInt();
            }
        }

        // display grades
        for (int i = 0; i < grades.length; i++) {
            System.out.print("Student " + (i+1) + " grades: ");
            for (int j = 0; j < grades[i].length; j++) {
                if (grades[i][j] < 50) {
                    System.out.print("Fail ");
                } else if (grades[i][j] >= 50 && grades[i][j] <= 59) {
                    System.out.print("Pass ");
                } else if (grades[i][j] >= 60 && grades[i][j] <= 69) {
                    System.out.print("Madiem ");
                } else if (grades[i][j] >= 70 && grades[i][j] <= 79) {
                    System.out.print("Good ");
                } else if (grades[i][j] >= 80 && grades[i][j] <= 89) {
                    System.out.print("Very Good ");
                } else if (grades[i][j] >= 90 && grades[i][j] <= 100) {
                    System.out.print("Excellent ");
                } else {
                    System.out.print("Error ");
                }
            }
            System.out.println();
        }
    }
}
```

08:02

Sure, here's an example Java code using a multidimensional array to print some Iraqi cities:

```
public class IraqiCities {
    public static void main(String[] args) {
        String[][] cities = {
            {"Baghdad", "Najaf", "Karbala"},
            {"Erbil", "Duhok", "Sulaymaniyah"},
            {"Basra", "Nasiriyah", "Amara"}
        };

        for (int i = 0; i < cities.length; i++) {
            System.out.print("Cities in " + getRegionName(i) + ": ");
            for (int j = 0; j < cities[i].length; j++) {
                System.out.print(cities[i][j] + ", ");
            }
            System.out.println();
        }
    }

    public static String getRegionName(int regionIndex) {
        switch (regionIndex) {
            case 0: return "Central";
            case 1: return "Kurdistan";
            case 2: return "South";
            default: return "Unknown";
        }
    }
}
```

This code defines a 2D array `cities` that contains the names of some Iraqi cities grouped by region. It then loops through the array and prints each region's name followed by the cities in that region.

Here's the output of running this code:

Cities in Central: Baghdad, Najaf, Karbala

Cities in Kurdistan: Erbil, Duhok, Sulaymaniyah

Cities in South: Basra, Nasiriyah, Amara

Example: print the stars by using Array

```
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *
```

```
1 public class StarSquare {  
2     public static void main(String[] args) {  
3         int size = 5;  
4         char[][] square = new char[size][size];  
5  
6         // fill the array with stars  
7         for (int i = 0; i < size; i++) {  
8             for (int j = 0; j < size; j++) {  
9                 square[i][j] = '*';  
10            }  
11        }  
12  
13        // print the array  
14        for (int i = 0; i < size; i++) {  
15            for (int j = 0; j < size; j++) {  
16                System.out.print(square[i][j] + " ");  
17            }  
18            System.out.println();  
19        }  
20    }  
21 }
```

Here is a sample code in Java to calculate the sum of multiple numbers:

```
import java.util.Scanner;

public class SumOfNumbers {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int n, sum = 0;
        System.out.print("Enter the number of integers you want to add: ");
        n = input.nextInt();
        System.out.print("Enter " + n + " integers: ");
        for(int i = 0; i < n; i++) {
            int num = input.nextInt();
            sum += num;
        }
        System.out.println("The sum of the entered integers is: " + sum);
    }
}
```

This code prompts the user to enter the number of integers they want to add, and then asks them to enter each integer. It adds each integer to the sum variable using a for loop, and then prints out the final sum.

10/27/2024

Example program in Java that calculates the subtraction of multiple numbers:

```
import java.util.Scanner;

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

        System.out.print("Enter the number of values to subtract: ");
        int count = input.nextInt();

        System.out.print("Enter the first value: ");
        int result = input.nextInt();

        for (int i = 1; i < count; i++) {
            System.out.print("Enter the next value: ");
            int nextValue = input.nextInt();
            result -= nextValue;
        }

        System.out.println("The result of the subtraction is: " + result);
    }
}
```

This program prompts the user to enter the number of values to subtract, and then asks for each value one by one. It subtracts each value from the previous result, and finally displays the final result.

Example output:

```
Enter the number of values to subtract: 4
Enter the first value: 10
Enter the next value: 3
Enter the next value: 2
Enter the next value: 5
The result of the subtraction is: 0
```

Example program in Java that calculates the multiplication of multiple numbers:

```
import java.util.Scanner;

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

        System.out.print("Enter the number of values to multiply: ");
        int count = input.nextInt();

        System.out.print("Enter the first value: ");
        int result = input.nextInt();

        for (int i = 1; i < count; i++) {
            System.out.print("Enter the next value: ");
            int nextValue = input.nextInt();
            result *= nextValue;
        }

        System.out.println("The result of the multiplication is: " + result);
    }
}
```

This program prompts the user to enter the number of values to multiply, and then asks for each value one by one. It multiplies each value with the previous result, and finally displays the final result.

Example output:

...

Enter the number of values to multiply: 4

Enter the first value: 3

Enter the next value: 4

Enter the next value: 2

Enter the next value: 5

The result of the multiplication is: 120

Example program in Java that calculates the division of multiple numbers:

```
import java.util.Scanner;

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

        System.out.print("Enter the number of values to divide: ");
        int count = input.nextInt();

        System.out.print("Enter the first value: ");
        double result = input.nextDouble();

        for (int i = 1; i < count; i++) {
            System.out.print("Enter the next value: ");
            double nextValue = input.nextDouble();
            result /= nextValue;
        }

        System.out.println("The result of the division is: " + result);
    }
}
```

This program prompts the user to enter the number of values to divide, and then asks for each value one by one. It divides each value with the previous result, and finally displays the final result

Example output

Enter the number of values to divide: 3

Enter the first value: 36

Enter the next value: 4

Enter the next value: 2

The result of the division is: 4.5

Example program in Java that uses a multidimensional array to print the name of a color based on a user input number:


```

import java.util.Scanner;

public class ColorArray {
    public static void main(String[] args) {
        String[][] colors = {
            {"0", "red"},
            {"1", "orange"},
            {"2", "yellow"},
            {"3", "green"},
            {"4", "blue"},
            {"5", "indigo"},
            {"6", "violet"}
        };

        Scanner input = new Scanner(System.in);

        System.out.print("Enter a number between 0 and 6: ");
        String number = input.nextLine();

        boolean found = false;
        for (int i = 0; i < colors.length; i++) {
            if (colors[i][0].equals(number)) {
                System.out.println("The color is: " + colors[i][1]);
                found = true;
                break;
            }
        }

        if (!found) {
            System.out.println("Invalid number entered.");
        }
    }
}

```

This program initializes a multidimensional array called `colors` with the number and name of each color. It prompts the user to enter a number between 0 and 6, and then searches the `colors` array for the corresponding color name. If the number is found, it prints the color name. If the number is not found, it prints an error message

Example output:

Enter a number between 0 and 6: 3

The color is: green

Example program in Java that uses an array to print the human temperature based on a user input number:

```
import java.util.Scanner;

public class TemperatureArray {
    public static void main(String[] args) {
        int[] temps = {35, 36, 37, 38, 39, 40};
        String[] messages = {"normal", "The temperature is a little high",
"The temperature is very high", "The temperature is very high", "The
temperature is very high", "The temperature is very high"};

        Scanner input = new Scanner(System.in);

        System.out.print("Enter the human temperature: ");
        int temp = input.nextInt();

        boolean found = false;
        for (int i = 0; i < temps.length; i++) {
            if (temps[i] == temp) {
                System.out.println(messages[i]);
                found = true;
                break;
            }
        }

        if (!found) {
            System.out.println("Number is error.");
        }
    }
}
```

This program initializes two arrays: `temps` with the possible human temperatures, and `messages` with the corresponding messages to print for each temperature. It prompts the user to enter a temperature, and then searches the

`temps` array for the corresponding message. If the temperature is found, it prints the message. If the temperature is not found, it prints an error message.

Example output:

Enter the human temperature: 36

Normal

- Write a Java or C++ program that reads from user 4x3 matrix (A) and 3x5 Matrix (B) and produces the matrix (C) with size 4x5 such that $C = A * B$

```
import java.util.Scanner;
public class Main
{
    public static void main (String args[])
    {

        Scanner sc=new Scanner(System.in);

        int row_A = 4;        //taking row as input
        int col_A = 3;        //taking column as input

        int row_B = 3;
        int col_B = 5;

        // Declaring the two-dimensional matrix (A)
        int matrix_A[][] = new int[row_A][col_A];
        int matrix_B[][] = new int[row_B][col_B];

        // Read the matrix (A) values
        System.out.println ("Enter the elements of the matrix A: ");
        //loop for row
        for (int i = 0; i < row_A; i++)
        //inner for loop for column
            for (int j = 0; j < col_A; j++)
                matrix_A[i][j] = sc.nextInt ();

        // Read the matrix (B) values
        System.out.println ("Enter the elements of the matrix B: ");

        for (int i = 0; i < row_B; i++)
            for (int j = 0; j < col_B; j++)
                matrix_B[i][j] = sc.nextInt ();

        // Mutliplying Two matrices
        int[][] C = new int[row_A][col_B];
        for(int i = 0; i < row_A; i++) {
            for (int j = 0; j < col_B; j++) {
                for (int k = 0; k < col_A; k++) {
                    C[i][j] += matrix_A[i][k] * matrix_B[k][j];
                }
            }
        }
    }
}
```

```

//accessing array elements
    System.out.println ("Elements of the array_A are: ");
    for (int i = 0; i < row_A; i++)
    {
        for (int j = 0; j < col_A; j++)
//prints the array elements
            System.out.print (matrix_A[i][j] + " ");
//throws the cursor to the next line
            System.out.println ();
        }

        System.out.println ("Elements of the array_B are: ");
        for (int i = 0; i < row_B; i++)
        {
            for (int j = 0; j < col_B; j++)
//prints the array elements B
                System.out.print (matrix_B[i][j] + " ");
                System.out.println ();
            }

            // Displaying the result
            System.out.println("Multiplication of two matrices is: ");
            for(int[] row : C) {
                for (int column : row) {
                    System.out.print(column + " ");
                }
                System.out.println();
            }
        }
    }
}

```

Here is a sample Java code using arrays to calculate integrals:

```

import java.util.Scanner;

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

        System.out.println("Enter the number of intervals:");
        int n = scanner.nextInt();

        double[] x = new double[n + 1];
        double[] y = new double[n + 1];

        System.out.println("Enter the lower limit:");
        double a = scanner.nextDouble();

        System.out.println("Enter the upper limit:");
        double b = scanner.nextDouble();

        double h = (b - a) / n;

        for (int i = 0; i <= n; i++) {
            x[i] = a + i * h;
            y[i] = f(x[i]);
        }

        double sum = 0;

        for (int i = 1; i < n; i++) {
            sum += y[i];
        }
    }
}

```

```

        double integral = h * ((y[0] + y[n]) / 2 + sum);

        System.out.println("The value of the integral is " + integral);
    }

    public static double f(double x) {
        // Define your function here
        return Math.sin(x);
    }
}

```

This code prompts the user to enter the number of intervals, lower and upper limits of integration, and uses the trapezoidal rule to calculate the integral of a given function (in this case, $\sin(x)$). The x and y values are stored in arrays for convenience.

Here is a sample Java code using arrays to add two matrices of odd size:

```

import java.util.Scanner;

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

        System.out.println("Enter the size of the matrices (odd
number:");
        int size = scanner.nextInt();

        if (size % 2 == 0) {
            System.out.println("Size must be odd!");
            return;
        }

        int[][] matrix1 = new int[size][size];
        int[][] matrix2 = new int[size][size];
        int[][] result = new int[size][size];

        System.out.println("Enter the elements of the first matrix:");

        for (int i = 0; i < size; i++) {
            for (int j = 0; j < size; j++) {
                matrix1[i][j] = scanner.nextInt();
            }
        }
    }
}

```

```

for (int i = 0; i < size; i++) {
    for (int j = 0; j < size; j++) {
        result[i][j] = matrix1[i][j] + matrix2[i][j];
    }
}

System.out.println("The sum of the matrices is:");

for (int i = 0; i < size; i++) {
    for (int j = 0; j < size; j++) {
        System.out.print(result[i][j] + " ");
    }
    System.out.println();
}
}
}

```

This code prompts the user to enter the size of the matrices (which must be odd), and the elements of two matrices of that size. The matrices are stored in arrays, and the sum is calculated by adding the corresponding elements of each matrix. The result is then printed to the console.

- Example code snippet that demonstrates how to calculate the limit of a function using arrays in Java:

```

public class LimitCalculator {
    public static void main(String[] args) {
        // Define the input array
        double[] x = { 1.0, 1.5, 1.9, 1.99, 1.999 };

        // Define the function being evaluated
        double f(double x) {
            return Math.pow(x, 2) - 1;
        }

        // Evaluate the function for each input value
        double[] y = new double[x.length];
        for (int i = 0; i < x.length; i++) {
            y[i] = f(x[i]);
        }

        // Print the output array
        System.out.println(Arrays.toString(y));

        // Analyze the output array to determine the limit
        double limit = y[y.length - 1];
        System.out.println("The limit is " + limit);
    }
}

```

In this example, the input array `x` contains a sequence of numbers that approach the value at which the limit is being calculated. The function being evaluated is defined using a lambda expression, which takes an input value `x` and returns the output of the function. The loop iterates over the elements of the input array, calling the function for each element and storing the results in the output array `y`. Finally, the limit is determined by analyzing the output array, which in this case is simply the last element of the array.

- Example code snippet that demonstrates how to calculate the derivatives of a function using arrays in Java:

```
public class DerivativeCalculator {
    public static void main(String[] args) {
        // Define the input array
        double[] x = { 1.0, 1.5, 1.9, 1.99, 1.999 };

        // Define the function being evaluated
        double f(double x) {
            return Math.pow(x, 2) - 1;
        }

        // Calculate the first derivative using the forward difference
        method
        double[] dydx = new double[x.length];
        for (int i = 0; i < x.length - 1; i++) {
            dydx[i] = (f(x[i + 1]) - f(x[i])) / (x[i + 1] - x[i]);
        }
        dydx[x.length - 1] = dydx[x.length - 2];

        // Print the output array
        System.out.println(Arrays.toString(dydx));

        // Calculate the second derivative using the central difference
        method
        double[] d2ydx2 = new double[x.length];
        for (int i = 1; i < x.length - 1; i++) {
            d2ydx2[i] = (f(x[i + 1]) - 2 * f(x[i]) + f(x[i - 1])) / Math.pow((x[i + 1]
- x[i]), 2);
        }
        d2ydx2[0] = d2ydx2[1];
        d2ydx2[x.length - 1] = d2ydx2[x.length - 2];
    }
}
```

```

// Print the output array
System.out.println(Arrays.toString(dydx));

// Calculate the second derivative using the central difference
method
double[] d2ydx2 = new double[x.length];
for (int i = 1; i < x.length - 1; i++) {
    d2ydx2[i] = (f(x[i + 1]) - 2 * f(x[i]) + f(x[i - 1])) / Math.pow((x[i + 1]
- x[i]), 2);
}
d2ydx2[0] = d2ydx2[1];
d2ydx2[x.length - 1] = d2ydx2[x.length - 2];

// Print the output array
System.out.println(Arrays.toString(d2ydx2));
}
}

```

- In this example, the input array `x` contains a sequence of numbers at which the derivatives of the function are being calculated. The function being evaluated is defined using a lambda expression, which takes an input value `x` and returns the output of the function. The first derivative is calculated using the forward difference method, which approximates the derivative by dividing the difference in function values at adjacent points by the difference in input values. The second derivative is calculated using the central difference method, which approximates the derivative by dividing the difference in first derivative values at adjacent points by the square of the difference in input values. The output arrays `dydx` and `d2ydx2` contain the values of the first and second derivatives, respectively, at each point in the input array.
- =====