

Week 3 - Level 2 - 10 Practice Problems[Arrays]

1. Create a program to find the bonus of 10 employees based on their years of service and the total bonus amount the company Zara has to pay, along with the old and new salary.

Hint =>

- a. Zara decides to give a bonus of 5% to employees whose year of service is more than 5 years or 2% if less than 5 years
- b. Define a double array to save salary and years of service for each of the 10 employees
- c. Also define a double array to save the new salary and the bonus amount as well as variables to save the total bonus, total old salary, and new salary
- d. Define a loop to take input from the user. If salary or year of service is an invalid number then ask the use to enter again. Note in this case you will have to decrement the index counter
- e. Define another loop to calculate the bonus of 10 employees based on their years of service. Save the bonus in the array, compute the new salary, and save in the array. Also, the total bonus and total old and new salary can be calculated in the loop
- f. Print the total bonus payout as well as the total old and new salary of all the employees

Ans) Code:

```
//import java utility scanner
import java.util.Scanner;
//declare class
public class EmployeeBonus {
    public static void main(String[] args) {
        //declare required variables
        final int empcount = 10;
        double[] salaries = new double[empcount];
        double[] yearsOfService = new double[empcount];
        double totalBonus = 0, totalOldSalary = 0, totalNewSalary = 0;
        //declare scanner object
        Scanner myobj = new Scanner(System.in);
        //loop to take array input for salaries and years of service
        for (int i = 0; i < empcount; i++) {
            System.out.print("Enter salary for employee " + (i + 1) + ": ");
            salaries[i] = myobj.nextDouble();
            System.out.print("Enter years of service for employee " + (i + 1) + ": ");
            yearsOfService[i] = myobj.nextDouble();
            //ternery operator to check given conditions for years of service
            double bonus = yearsOfService[i] > 5 ? salaries[i] * 0.05 : salaries[i] * 0.02;
            //modify the required index of array
            totalBonus += bonus;
            totalOldSalary += salaries[i];
        }
    }
}
```

```

        totalNewSalary += salaries[i] + bonus;
    }

    //print the required data
    System.out.println("\nTotal bonus: " + totalBonus);
    System.out.println("Total old salary: " + totalOldSalary);
    System.out.println("Total new salary: " + totalNewSalary);
}
}

```

Output Verification:

```

C:\Windows\System32\cmd.exe
C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week3\Level2>java EmployeeBonus
Enter salary for employee 1: 1
Enter years of service for employee 1: 5
Enter salary for employee 2: 2
Enter years of service for employee 2: 3
Enter salary for employee 3: 4
Enter years of service for employee 3: 8
Enter salary for employee 4: 5
Enter years of service for employee 4: 9
Enter salary for employee 5: 6
Enter years of service for employee 5: 4
Enter salary for employee 6: 3
Enter years of service for employee 6: 8
Enter salary for employee 7: 7
Enter years of service for employee 7: 8
Enter salary for employee 8: 9
Enter years of service for employee 8: 9
Enter salary for employee 9: 8
Enter years of service for employee 9: 3
Enter salary for employee 10: 10
Enter years of service for employee 10: 10

Total bonus: 2.24
Total old salary: 55.0
Total new salary: 57.239999999999995

```

2. Create a program to find the youngest friends among 3 Amar, Akbar, and Anthony based on their ages and the tallest among the friends based on their heights

Hint =>

- Take user input for age and height for the 3 friends and store it in two arrays each to store the values for age and height of the 3 friends
- Loop through the array and find the youngest of the 3 friends and the tallest of the 3 friends
- Finally display the youngest and tallest of the 3 friends

Ans) Code:

```

//import java utility scanner
import java.util.Scanner;

//declare class
public class YoungestTallest {

```

```

public static void main(String[] args) {
    //declare arrays and variables
    int friendcount = 3;
    String[] names = {"Amar", "Akbar", "Anthony"};
    //initialise array
    int[] ages = new int[friendcount];
    double[] heights = new double[friendcount];
    Scanner myobj = new Scanner(System.in);
    // Input ages and heights
    for (int i = 0; i < friendcount; i++) {
        System.out.print("Enter the age of " + names[i] + ": ");
        ages[i] = myobj.nextInt();
        System.out.print("Enter the height (in cm) of " + names[i] + ": ");
        heights[i] = myobj.nextDouble();
    }
    // Find the youngest
    int youngestIndex = 0;
    for (int i = 1; i < friendcount; i++) {
        if (ages[i] < ages[youngestIndex]) {
            youngestIndex = i;
        }
    }
    // Find the tallest
    int tallestIndex = 0;
    for (int i = 1; i < friendcount; i++) {
        if (heights[i] > heights[tallestIndex]) {
            tallestIndex = i;
        }
    }
    // print the results
    System.out.println("\nThe youngest friend is " + names[youngestIndex] + " with an
age of " + ages[youngestIndex] + " years.");
}

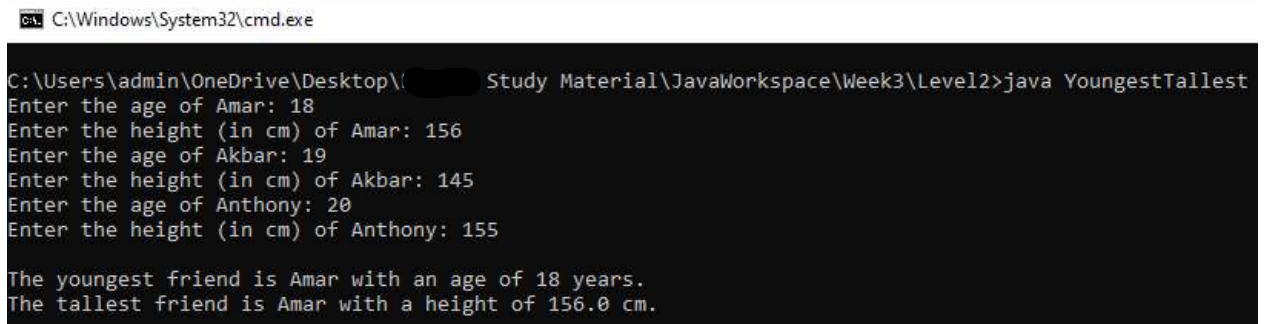
```

```

        System.out.println("The tallest friend is " + names[tallestIndex] + " with a height of "
+ heights[tallestIndex] + " cm.");
    }
}

```

Output Verification:



```

C:\Windows\System32\cmd.exe
C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week3\Level2>java YoungestTallest
Enter the age of Amar: 18
Enter the height (in cm) of Amar: 156
Enter the age of Akbar: 19
Enter the height (in cm) of Akbar: 145
Enter the age of Anthony: 20
Enter the height (in cm) of Anthony: 155

The youngest friend is Amar with an age of 18 years.
The tallest friend is Amar with a height of 156.0 cm.

```

3.Create a program to store the digits of the number in an array and find the largest and second largest element of the array.

Hint =>

- Create a number variable and Take user input.
- Define an array to store the digits. Set the size of the array to maxDigit variable initially set to 10
- Create an integer variable index with the value 0 to reflect the array index.
- Use a loop to iterate until the number is not equal to 0.
- Remove the last digit from the number in each iteration and add it to the array.
- Increment the index by 1 in each iteration and if the index count equals maxDigit then break out of the loop and the remaining digits are not added to the array
- Define variable to store largest and second largest digit and initialize it to zero
- Loop through the array and use conditional statements to find the largest and second largest number in the array
- Finally display the largest and second-largest number

Ans) Code:

```

//import java utility scanner
import java.util.Scanner;

//declare class
public class LargestDigit {

    public static void main(String[] args) {

        final int maxDigit = 10; // maximum number of digits
        int[] digits = new int[maxDigit];
    }
}

```

```

int index = 0; // to track array index
    //declare scanner object
Scanner myobj = new Scanner(System.in);
// take user input
System.out.print("Enter a number: ");
int number = myobj.nextInt();
// extract digits and store them in the array
while (number != 0) {
    if (index == maxDigit) {
        break; // stop if maxDigit is reached
    }
    digits[index++] = number % 10; // store the last digit
    number /= 10; // remove the last digit
}
// find the largest and second largest digits
int largest = 0, secondLargest = 0;
for (int i = 0; i < index; i++) {
    if (digits[i] > largest) {
        secondLargest = largest; // update second largest
        largest = digits[i]; // update largest
    } else if (digits[i] > secondLargest && digits[i] != largest) {
        secondLargest = digits[i]; // update second largest
    }
}
// print output
System.out.println("Largest digit: " + largest);
System.out.println("Second largest digit: " + secondLargest);
}
}

```

Output Verification:

```
C:\Windows\System32\cmd.exe
C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week3\Level2>java LargestDigit
Enter a number: 15846
Largest digit: 8
Second largest digit: 6
```

4. Rework the program 2, especially the **Hint f** where if index equals maxDigit, we break from the loop. Here we want to modify to Increase the size of the array i.e maxDigit by 10 if the index is equal to maxDigit. This is done to consider all digits to find the largest and second-largest number

Hint =>

- In Hint f inside the loop if the index is equal to maxDigit, increase maxDigit and make digits array to store more elements.
- To do this, we need to create a new temp array of size maxDigit, copy from the current digits array the digits into the temp array, and assign the current digits array to the temp array
- Now the digits array will be able to store all digits of the number in the array and then find the largest and second largest number

Ans) Code:

```
//import java utility scanner
import java.util.Scanner;

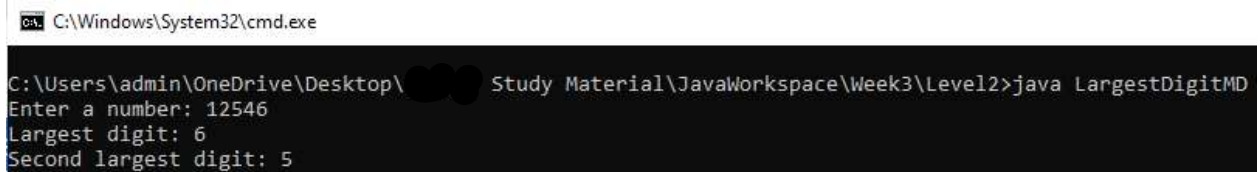
//declare class
public class LargestDigitMD {
    public static void main(String[] args) {
        Scanner myobj = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = myobj.nextInt();
        int maxDigit = 10; // Initial size of the array
        int[] digits = new int[maxDigit]; // Array to store digits
        int index = 0;
        // Extract digits and store in the array
        while (number != 0) {
            if (index == maxDigit) {
                // Increase array size by 10
                maxDigit += 10;
            }
            digits[index] = number % 10;
            number /= 10;
            index++;
        }
        // Find largest and second largest digit
        int largest = -1;
        int secondLargest = -1;
        for (int i = 0; i < index; i++) {
            if (digits[i] > largest) {
                secondLargest = largest;
                largest = digits[i];
            } else if (digits[i] > secondLargest && digits[i] != largest) {
                secondLargest = digits[i];
            }
        }
        System.out.println("Largest digit: " + largest);
        System.out.println("Second largest digit: " + secondLargest);
    }
}
```

```

        int[] temp = new int[maxDigit];
        System.arraycopy(digits, 0, temp, 0, digits.length);
        digits = temp;
    }
    digits[index++] = number % 10; // Store the last digit in the array
    number /= 10; // Remove the last digit from the number
}
// Finding the largest and second largest digit
int largest = 0, secondLargest = 0;
for (int i = 0; i < index; i++) {
    if (digits[i] > largest) {
        secondLargest = largest;
        largest = digits[i];
    } else if (digits[i] > secondLargest && digits[i] != largest) {
        secondLargest = digits[i];
    }
}
// Display results
System.out.println("Largest digit: " + largest);
System.out.println("Second largest digit: " + secondLargest);
}
}

```

Output Verification:



```

C:\Windows\System32\cmd.exe
C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week3\Level2>java LargestDigitMD
Enter a number: 12546
Largest digit: 6
Second largest digit: 5

```

5. Create a program to take a number as input and reverse the number. To do this, store the digits of the number in an array and display the array in reverse order

Hint =>

- a. Take user input for a number.
- b. Find the count of digits in the number.
- c. Find the digits in the number and save them in an array
- d. Create an array to store the elements of the digits array in reverse order
- e. Finally, display the elements of the array in reverse order

Ans) Code:

```
//import java utility scanner
import java.util.Scanner;

//declare class
public class ReverseNumber {

    public static void main(String[] args) {

        Scanner myobj = new Scanner(System.in);

        // Take user input
        System.out.print("Enter a number: ");
        int number = myobj.nextInt();

        // Find count of digits
        int temp = number;
        int digitCount = 0;
        while (temp != 0) {
            digitCount++;
            temp /= 10;
        }

        // Store digits in an array
        int[] digits = new int[digitCount];
        temp = number; // Reset temp to the input number
        for (int i = 0; i < digitCount; i++) {
            digits[i] = temp % 10; // Extract the last digit
            temp /= 10; // Remove the last digit
        }

        // Display the digits in reverse order
```

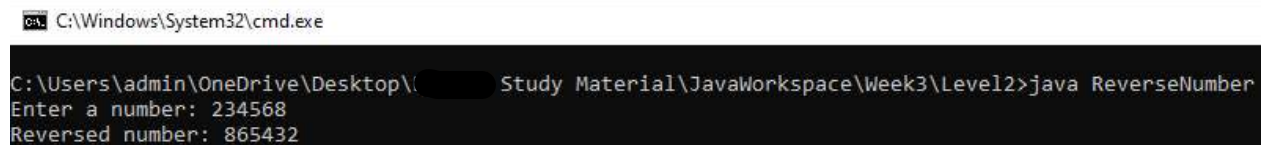


```

        System.out.print("Reversed number: ");
        for (int i = 0; i < digitCount; i++) {
            System.out.print(digits[i]);
        }
    }
}

```

Output Verification:



C:\Windows\System32\cmd.exe

```

C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week3\Level2>java ReverseNumber
Enter a number: 234568
Reversed number: 865432

```

6.

An organization took up an exercise to find the Body Mass Index (BMI) of all the persons in the team. For this create a program to find the BMI and display the height, weight, BMI and status of each individual

Hint =>

- Take input for a number of persons
- Create arrays to store the weight, height, BMI, and weight status of the persons
- Take input for the weight and height of the persons
- Calculate the BMI of all the persons and store them in an array and also find the weight status of the persons
- Display the height, weight, BMI, and weight status of each person
- Use the table to determine the weight status of the person

BMI	Status
≤ 18.4	Underweight
18.5 - 24.9	Normal
25.0 - 39.9	Overweight
≥ 40.0	Obese

Ans) Code:

```

//import java utility scanner
import java.util.Scanner;
//declare class

```

```

public class BMI {
    public static void main(String[] args) {
        //declare scanner object
        Scanner myobj = new Scanner(System.in);
        // Input number of persons
        System.out.print("Enter the number of persons: ");
        int numPersons = myobj.nextInt();
        // Arrays to store data
        double[] heights = new double[numPersons];
        double[] weights = new double[numPersons];
        double[] bmis = new double[numPersons];
        String[] statuses = new String[numPersons];
        // Input height and weight for each person
        for (int i = 0; i < numPersons; i++) {
            System.out.print("Enter height (in meters) for person " + (i + 1) + ": ");
            heights[i] = myobj.nextDouble();
            System.out.print("Enter weight (in kilograms) for person " + (i + 1) + ": ");
            weights[i] = myobj.nextDouble();
            // Calculate BMI
            bmis[i] = weights[i] / (heights[i] * heights[i]);
            // Determine weight status
            if (bmis[i] <= 18.4) {
                statuses[i] = "Underweight";
            } else if (bmis[i] >= 18.5 && bmis[i] <= 24.9) {
                statuses[i] = "Normal";
            } else if (bmis[i] >= 25.0 && bmis[i] <= 39.9) {
                statuses[i] = "Overweight";
            } else {
                statuses[i] = "Obese";
            }
        }
    }
}

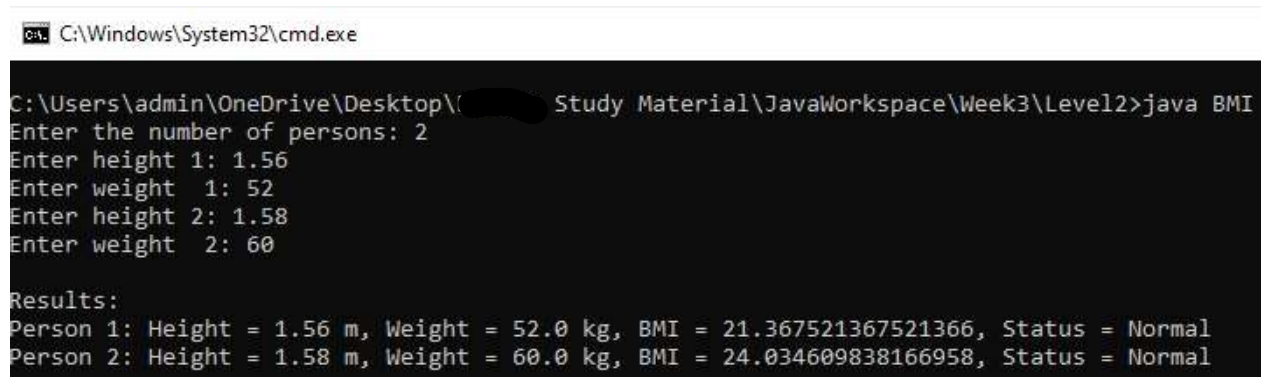
```

```

// Display results
System.out.println("\nResults:");
for (int i = 0; i < numPersons; i++) {
    System.out.println("Person " + (i + 1) + ": Height = " + heights[i] + " m, Weight = " +
weights[i] + " kg, BMI = " + bmis[i] + ", Status = " + statuses[i]);
}
}
}

```

Output Verification:



```

C:\Windows\System32\cmd.exe

C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week3\Level2>java BMI
Enter the number of persons: 2
Enter height 1: 1.56
Enter weight 1: 52
Enter height 2: 1.58
Enter weight 2: 60

Results:
Person 1: Height = 1.56 m, Weight = 52.0 kg, BMI = 21.367521367521366, Status = Normal
Person 2: Height = 1.58 m, Weight = 60.0 kg, BMI = 24.034609838166958, Status = Normal

```

7. Rewrite the above program using multi-dimensional array to store height, weight, and BMI in 2D array for all the persons

Hint =>

- Take input for a number of persons
- Create a multi-dimensional array to store weight, height and BMI. Also create an to store the weight status of the persons

```

double[][] personData = new double[number][3];
String[] weightStatus = new String[number];

```

- Take input for weight and height of the persons and for negative values, ask the user to enter positive values
- Calculate BMI of all the persons and store them in the personData array and also find the weight status and put them in the weightStatus array
- Display the height, weight, BMI and status of each person

Ans) Code:

```

//import java utility
import java.util.Scanner;

//declare class

```

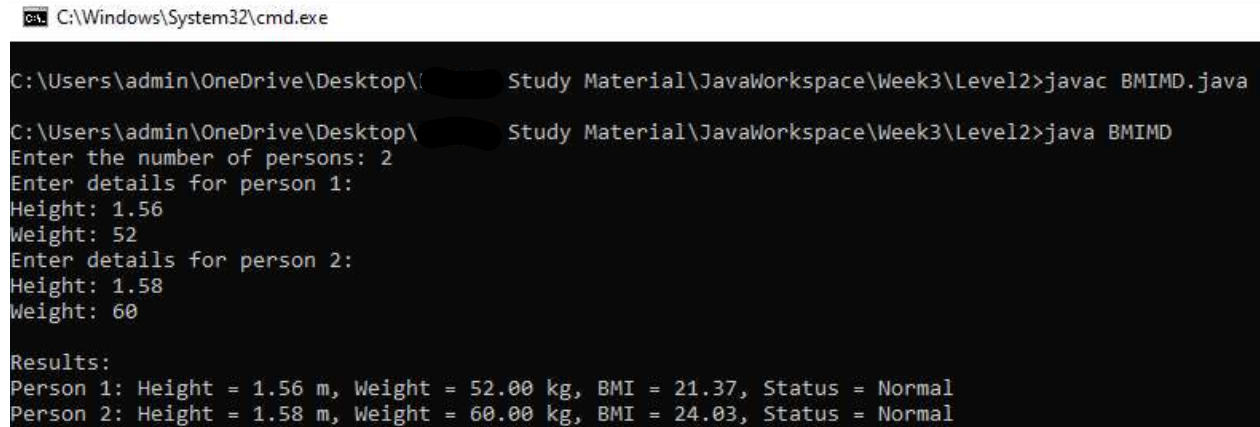
```

public class BMIMD {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // Input number of persons
        System.out.print("Enter the number of persons: ");
        int numPersons = scanner.nextInt();
        // Multi-dimensional array to store height, weight, and BMI
        double[][] personData = new double[numPersons][3]; // [height, weight, BMI]
        String[] weightStatus = new String[numPersons];
        // Input height and weight for each person
        for (int i = 0; i < numPersons; i++) {
            System.out.println("Enter details for person " + (i + 1) + " :");
            // Input height and validate
            System.out.print("Height: ");
            personData[i][0] = scanner.nextDouble();
            // Input weight and validate
            System.out.print("Weight: ");
            personData[i][1] = scanner.nextDouble();
            // Calculate BMI
            personData[i][2] = personData[i][1] / (personData[i][0] * personData[i][0]);
            // Determine weight status
            if (personData[i][2] <= 18.4) {
                weightStatus[i] = "Underweight";
            } else if (personData[i][2] >= 18.5 && personData[i][2] <= 24.9) {
                weightStatus[i] = "Normal";
            } else if (personData[i][2] >= 25.0 && personData[i][2] <= 39.9) {
                weightStatus[i] = "Overweight";
            } else {
                weightStatus[i] = "Obese";
            }
        }
    }
}

```

```
// Display results
System.out.println("\nResults:");
for (int i = 0; i < numPersons; i++) {
    System.out.printf("Person %d: Height = %.2f m, Weight = %.2f kg, BMI = %.2f, Status = %s\n",
        (i + 1), personData[i][0], personData[i][1], personData[i][2], weightStatus[i]);
}
}
```

Output Verification:



```
C:\Windows\System32\cmd.exe
C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week3\Level2>javac BMIMD.java
C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week3\Level2>java BMIMD
Enter the number of persons: 2
Enter details for person 1:
Height: 1.56
Weight: 52
Enter details for person 2:
Height: 1.58
Weight: 60

Results:
Person 1: Height = 1.56 m, Weight = 52.00 kg, BMI = 21.37, Status = Normal
Person 2: Height = 1.58 m, Weight = 60.00 kg, BMI = 24.03, Status = Normal
```

8.

Create a program to take input marks of students in 3 subjects physics, chemistry, and maths. Compute the percentage and then calculate the grade as per the following guidelines

Grade	Remarks	Marks
A	(Level 4, above agency-normalized standards)	80% and above
B	(Level 3, at agency-normalized standards)	70-79%
C	(Level 2, below, but approaching agency-normalized standards)	60-69%
D	(Level 1, well below agency-normalized standards)	50-59%
E	(Level 1- , too below agency-normalized standards)	40-49%
R	(Remedial standards)	39% and below

Hint =>

- Take input for the number of students
- Create arrays to store marks, percentages, and grades of the students
- Take input for marks of students in physics, chemistry, and maths. If the marks are negative, ask the user to enter positive values and decrement the index
- Calculate the percentage and grade of the students based on the percentage
- Display the marks, percentages, and grades of each student

Ans) Code:

```
//import java utility scanner
import java.util.Scanner;
//declare class
public class StudentGrades {
    public static void main(String[] args) {
        //declare scanner object
        Scanner scanner = new Scanner(System.in);
        // Input number of students
        System.out.print("Enter the number of students: ");
        int numStudents = scanner.nextInt();
        // Arrays to store marks, percentages, and grades
        int[][] marks = new int[numStudents][3]; // 3 subjects: Physics, Chemistry, Maths
        double[] percentages = new double[numStudents];
        String[] grades = new String[numStudents];
```

```

// Array to define subject names
String[] subjects = {"Physics", "Chemistry", "Maths"};
// Input marks for each student
for (int i = 0; i < numStudents; i++) {
    System.out.println("Enter marks for student " + (i + 1) + ":");
    // Input marks for Physics, Chemistry, and Maths using nested loop
    for (int j = 0; j < 3; j++) {
        System.out.print(subjects[j] + " marks (0-100): ");
        marks[i][j] = scanner.nextInt();
    }
    // Calculate percentage
    percentages[i] = (marks[i][0] + marks[i][1] + marks[i][2]) / 3.0;
    // Assign grade based on percentage
    if (percentages[i] >= 80) {
        grades[i] = "A";
    } else if (percentages[i] >= 70) {
        grades[i] = "B";
    } else if (percentages[i] >= 60) {
        grades[i] = "C";
    } else if (percentages[i] >= 50) {
        grades[i] = "D";
    } else if (percentages[i] >= 40) {
        grades[i] = "E";
    } else {
        grades[i] = "R";
    }
}
// Display results
System.out.println("\nResults:");
for (int i = 0; i < numStudents; i++) {
    System.out.printf("Student %d: Physics = %d, Chemistry = %d, Maths = %d, Percentage
= %.2f, Grade = %s\n",

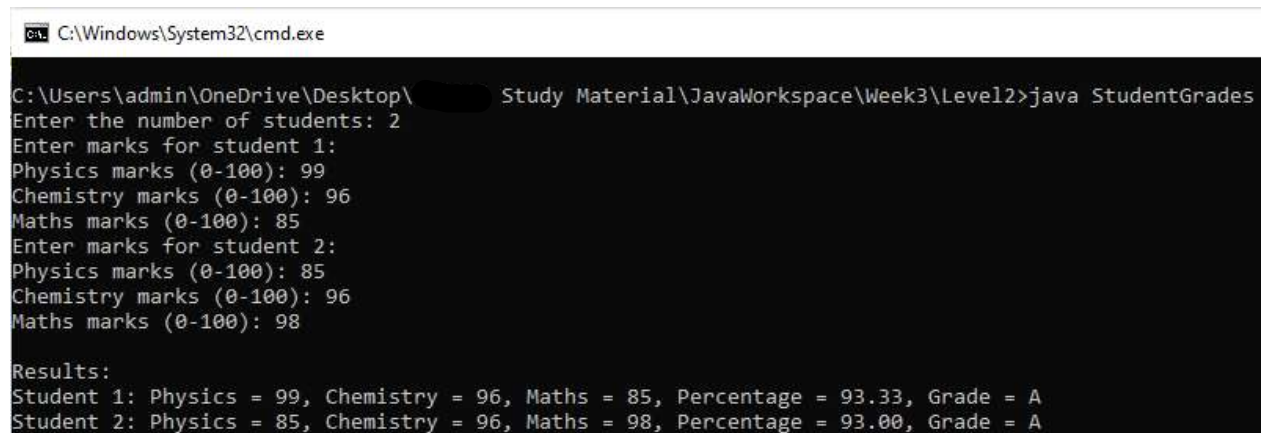
```

```

        (i + 1), marks[i][0], marks[i][1], marks[i][2], percentages[i], grades[i]);
    }
}
}

```

Output Verification:



```

C:\Windows\System32\cmd.exe
C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week3\Level2>java StudentGrades
Enter the number of students: 2
Enter marks for student 1:
Physics marks (0-100): 99
Chemistry marks (0-100): 96
Maths marks (0-100): 85
Enter marks for student 2:
Physics marks (0-100): 85
Chemistry marks (0-100): 96
Maths marks (0-100): 98
Results:
Student 1: Physics = 99, Chemistry = 96, Maths = 85, Percentage = 93.33, Grade = A
Student 2: Physics = 85, Chemistry = 96, Maths = 98, Percentage = 93.00, Grade = A

```

9. Rewrite the above program to store the marks of the students in physics, chemistry, and maths in a 2D array and then compute the percentage and grade

Hint =>

- All the steps are the same as the problem 8 except the marks are stored in a 2D array
- Use the 2D array to calculate the percentages, and grades of the students

Ans) Code:

```

//import java utility scanner
import java.util.Scanner;
//declare class
public class StudentGrades2D {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // Input number of students
        System.out.print("Enter the number of students: ");
        int numStudents = scanner.nextInt();
        // 2D array to store marks for Physics, Chemistry, and Maths
        int[][] marks = new int[numStudents][3]; // 3 subjects: Physics, Chemistry, Maths
        double[] percentages = new double[numStudents];
        String[] grades = new String[numStudents];
        // Input marks for each student
        for (int i = 0; i < numStudents; i++) {
            System.out.println("Enter marks for student " + (i + 1) + ":");

```



```

// Input marks for Physics, Chemistry, and Maths
for (int j = 0; j < 3; j++) {
    String subject = j == 0 ? "Physics" : j == 1 ? "Chemistry" : "Maths";
    System.out.print(subject + " marks (0-100): ");
    marks[i][j] = scanner.nextInt();
}
// Calculate percentage
percentages[i] = (marks[i][0] + marks[i][1] + marks[i][2]) / 3.0;
// Assign grade based on percentage
if (percentages[i] >= 80) {
    grades[i] = "A";
} else if (percentages[i] >= 70) {
    grades[i] = "B";
} else if (percentages[i] >= 60) {
    grades[i] = "C";
} else if (percentages[i] >= 50) {
    grades[i] = "D";
} else if (percentages[i] >= 40) {
    grades[i] = "E";
} else {
    grades[i] = "R";
}
}
// Display results
System.out.println("\nResults:");
for (int i = 0; i < numStudents; i++) {
    System.out.printf("Student %d: Physics = %d, Chemistry = %d, Maths = %d, Percentage
= %.2f, Grade = %s\n",
        (i + 1), marks[i][0], marks[i][1], marks[i][2], percentages[i], grades[i]);
}
}
}

```

Output Verification:

```
C:\Windows\System32\cmd.exe
C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week3\Level2>java StudentGrades2D
Enter the number of students: 2
Enter marks for student 1:
Physics marks (0-100): 99
Chemistry marks (0-100): 96
Maths marks (0-100): 93
Enter marks for student 2:
Physics marks (0-100): 98
Chemistry marks (0-100): 95
Maths marks (0-100): 99

Results:
Student 1: Physics = 99, Chemistry = 96, Maths = 93, Percentage = 96.00, Grade = A
Student 2: Physics = 98, Chemistry = 95, Maths = 99, Percentage = 97.33, Grade = A
```

10. Create a program to take a number as input find the frequency of each digit in the number using an array and display the frequency of each digit

Hint =>

- Take the input for a number
- Find the count of digits in the number
- Find the digits in the number and save them in an array
- Find the frequency of each digit in the number. For this define a frequency array of size 10, Loop through the digits array, and increase the frequency of each digit
- Display the frequency of each digit in the number

Ans) Code:

```
//import java utility scanner
import java.util.Scanner;
//declare class
public class Frequency {
    public static void main(String[] args) {
        //declare scanner object
        Scanner scanner = new Scanner(System.in);
        // Take input for the number
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        // Create an array to store the frequency of each digit (0-9)
        int[] frequency = new int[10];
        // Process each digit in the number
        while (number != 0) {
            int digit = number % 10; // Extract the last digit
            frequency[digit]++; // Increase the frequency of the digit
            number /= 10; // Remove the last digit
        }
        // Display the frequency of each digit
        System.out.println("Digit frequencies in the number:");
```

```
        for (int i = 0; i < 10; i++) {  
            if (frequency[i] > 0) { // Only show digits that appear in the number  
                System.out.println("Digit " + i + ": " + frequency[i]);  
            }  
        }  
    }  
}
```

Output Verification:

cmd C:\Windows\System32\cmd.exe

```
C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week3\Level2>java Frequency  
Enter a number: 1125446  
Digit frequencies in the number:  
Digit 1: 2  
Digit 2: 1  
Digit 4: 2  
Digit 5: 1  
Digit 6: 1
```