CodInClub

Powered By BridgeLabz

Best Programming Practice

1. All values as variables including Fixed, User Inputs, and Results
2. Avoid Hard Coding of variables wherever possible
3. Proper naming conventions for all variables
4. Proper Program Name and Class Name
5. Follow proper indentation
6. Give comments for every step or logical block like a variable declaration or conditional and loop blocks
7. For every user input validate the user input, if invalid, state the error either exit the program or ask user to enter again
8. Use Array ***length*** property while using ***for*** loop

1. **Sample Program 1 -** Create a program to find the sum of all the digits of a number given by a user using an array and display the sum.

   Hint =>

   a. Take the input for a number and validate, if failed state and exit the program
   b. Find the count of digits in the number
   c. Find the digits in the number and save them in an array
   d. Find the sum of the digits of the number and display the sum

Java

```java
// Create SumOfDigit Class to compute the sum of all digits of a number using
// an array
import java.util.Scanner;

class SumOfDigits {
   public static void main(String[] args) {
      // Create a Scanner Object
      Scanner input = new Scanner(System.in);

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

      // Validate the user input number, if negative state invalid and exit
      if (number < 0) {
            System.err.println("Invalid Number.");
            System.exit(0);
      }
      // Find the count of digits in the number
      int count = 0;
      int temp = number;
      while (temp > 0) {
            count++;
            temp /= 10;
      }

      // Find the digits in the number and save them in an array
      int[] digits = new int[count];
      for (int i = 0; i < count; i++) {
            digits[i] = number % 10;
            number /= 10;
      }

      // Find the sum of the digits of the number
      int sum = 0;
      for (int i = 0; i < count; i++) {
            sum += digits[i];
      }

      // Display the sum of the digits of the number
      System.out.println("\nSum of Digits: " + sum);

      // Close the Scanner Object
      input.close();
   }
}
```

2. **Sample Program 2 -** Working with Multi-Dimensional Arrays. Write a Java program to create a 2 Dimensional (2D) array (matrix) of integers, initialize it with values, and print the sum of all elements in the matrix

   Hint =>

   a. Take the input for a number of rows and columns
   b. Create a 2D array (matrix) of integers
   c. Take the input for the elements of the matrix
   d. Calculate the sum of all elements in the matrix and display the sum
   e. Also, Display the matrix

Java

```java
// Program to create a 2D array, display the elements and calculate the sum of
// the elements of the array
import java.util.Scanner;

class 2DArray {
   public static void main(String[] args) {
      // Create a Scanner Object
      Scanner input = new Scanner(System.in);

      // Declare the 2D Array
      int[][] arr = new int[3][3];

      // Input the elements of the 2D Array
      System.out.println("Enter the elements of the 2D Array: ");
      for (int i = 0; i < 3; i++) {
         for (int j = 0; j < 3; j++) {
            arr[i][j] = input.nextInt();
         }
      }

      // Display the elements of the 2D Array and calculate the sum of the
      // elements of the 2D Array
      int sum = 0;
      System.out.println("The elements of the 2D Array are: ");
      for (int i = 0; i < 3; i++) {
         for (int j = 0; j < 3; j++) {
            System.out.print(arr[i][j] + " ");
            sum += arr[i][j];
         }
         System.out.println();
      }

      // Display the sum of the elements of the 2D Array
      System.out.println("The sum of the elements of the 2D Array is: " + sum);

      // Close the Scanner Object
      input.close();
   }
}
```

Level 2 Practice Programs

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

```java
import java.util.Scanner;

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

        // creating Arrays to hold salaries and years of service
        double[] salaries = new double[10];
        double[] yearsOfService = new double[10];

        // creating Arrays to hold new salary and bonus amounts
        double[] newSalaries = new double[10];
        double[] bonuses = new double[10];

        // creating Variables to calculate totals
        double totalBonus = 0;
        double totalOldSalary = 0;
        double totalNewSalary = 0;

        // Input loop
        for (int i = 0; i < 10; i++) {
            System.out.println("Enter the salary for employee " + (i + 1) +
": ");
```

```
            double salary = input.nextDouble();

            System.out.println("Enter the years of service for employee " +
(i + 1) + ": ");
            double years = input.nextDouble();

            // Validate input
            if (salary <= 0 || years < 0) {
                System.out.println("Invalid input. Salary must be positive
and years of service cannot be negative. Try again.");
                i--; // Decrement index to retry for this employee
                continue;
            }

            // Save input into arrays
            salaries[i] = salary;
            yearsOfService[i] = years;
        }

        // Calculating loop
        for (int i = 0; i < 10; i++) {
            double bonusPercentage = yearsOfService[i] > 5 ? 0.05 : 0.02;
            double bonus = salaries[i] * bonusPercentage;
            double newSalary = salaries[i] + bonus;

            // Save calculated values
            bonuses[i] = bonus;
            newSalaries[i] = newSalary;

            // calculating totals
            totalBonus += bonus;
            totalOldSalary += salaries[i];
            totalNewSalary += newSalary;
        }

        // Output results
        System.out.println("\nEmployee-wise Details:");
        for (int i = 0; i < 10; i++) {
            System.out.printf("Employee %d: Old Salary = %.2f, Bonus =
%.2f, New Salary = %.2f\n",
                              i + 1, salaries[i], bonuses[i],
newSalaries[i]);
        }

        System.out.printf("\nTotal Bonus Payout: %.2f\n", totalBonus);
        System.out.printf("Total Old Salary: %.2f\n", totalOldSalary);
        System.out.printf("Total New Salary: %.2f\n", totalNewSalary);
    }
}
```

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 =>

   a. 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
   b. Loop through the array and find the youngest of the 3 friends and the tallest of the 3 friends
   c. Finally display the youngest and tallest of the 3 friends

```
import java.util.Scanner;

public class HeightComparison2 {
    public static void main(String[] args) {
        // Initialize Scanner object with the name input
        Scanner input = new Scanner(System.in);

        // Arrays to store the ages and heights of the 3 friends
        String[] friends = {"Amar", "Akbar", "Anthony"};
        int[] ages = new int[3];
        double[] heights = new double[3];

        // Taking user input for ages and heights
        for (int i = 0; i < 3; i++) {
            System.out.print("Enter the age of " + friends[i] + ": ");
            ages[i] = input.nextInt();

            System.out.print("Enter the height (in cm) of " + friends[i] +
": ");
            heights[i] = input.nextDouble();
        }
        // Variables to find the youngest and tallest
        int youngestIndex = 0;
        int minAge = ages[0];

        int tallestIndex = 0;
        double maxHeight = heights[0];

        // Loop through the arrays to find the youngest and tallest
        for (int i = 1; i < 3; i++) {
            if (ages[i] < minAge) {
                        // Check if the current friend is younger
                minAge = ages[i];
                youngestIndex = i;
            }
            if (heights[i] > maxHeight) {
                        // Check if the current friend is taller
                maxHeight = heights[i];
                tallestIndex = i;
            }
        }

        // Display the results
        System.out.println("\nThe youngest friend is " +
friends[youngestIndex] + " with age " + minAge + " years.");
        System.out.println("The tallest friend is " + friends[tallestIndex]
+ " with height " + maxHeight + " cm.");

        // Close the input object
        input.close();
    }
}
```

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 =>

   a. Create a number variable and Take user input.
   b. Define an array to store the digits. Set the size of the array to maxDigit variable initially set to 10
   c. Create an integer variable index with the value 0 to reflect the array index.

d. Use a loop to iterate until the number is not equal to 0.
e. Remove the last digit from the number in each iteration and add it to the array.
f. 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
g. Define variable to store largest and second largest digit and initialize it to zero
h. Loop through the array and use conditional statements to find the largest and second largest number in the array
i. Finally display the largest and second-largest number

```java
import java.util.Scanner;

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

        // Taking number as user input
        System.out.print("Enter a number to know the largest and second
largest digit: ");
        int number = input.nextInt();

        // creating an array to store the digits, with max size set to
maxDigit
        int maxDigit = 10;
        int[] digits = new int[maxDigit];
        int index = 0;

        // Extracting digits from the number and store in the array
        while (number != 0) {
            if (index == maxDigit) {
                // Stop adding digits if the array size is reached
                break;
            }
                   // Extracting the last digit
            digits[index] = number % 10;
            // Removing the last digit from the number
                  number /= 10;
            // increasing index
                  index++;
        }

        // creating Variables to store the largest and second largest
digits
```

```
        int largest = 0;
        int secondLargest = 0;

        // Loop through the array to find the largest and second largest digits
        for (int i = 0; i < index; i++) {
            if (digits[i] > largest) {
                // Updating second largest before updating largest
                secondLargest = largest;
                largest = digits[i];
            } else if (digits[i] > secondLargest && digits[i] != largest) {
                // Updating second largest if current digit is less than largest but greater than second largest
                secondLargest = digits[i];
            }
        }

        // Displaying the results
        System.out.println("The largest digit is: " + largest);
        System.out.println("The second largest digit is: " + secondLargest);

        // Closing input object
        input.close();
    }
}
```

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 =>

   a. In Hint f inside the loop if the index is equal to maxDigit, increase maxDigit and make digits array to store more elements.
   b. 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
   c. 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

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

```
import java.util.Scanner;

public class LargestTwoDigits2 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // Taking number as user input
        System.out.print("Enter a number to know the largest and second
largest digit: ");
        int number = input.nextInt();

        // creating an array to store the digits, with max size set to
maxDigit
        int maxDigit = 10;
        int[] digits = new int[maxDigit];
        int index = 0;

        // Extracting digits from the number and store in the array
        while (number != 0) {

            if (index == maxDigit) {
                if (index == maxDigit) {
                              // Increase the size of the array
                              maxDigit += 10;
                              // Creating a new temporary array
                              int[] temp = new int[maxDigit];
                              // Copying existing array
                              System.arraycopy(digits, 0, temp, 0,
digits.length);
                              // Reassigning digits array to the temp array
                              digits = temp;
            }

            }
                   // Extracting the last digit
            digits[index] = number % 10;
            // Removing the last digit from the number
                  number /= 10;
            // increasing index
                  index++;
        }

        // creating Variables to store the largest and second largest
digits
        int largest = 0;
        int secondLargest = 0;

        // Loop through the array to find the largest and second largest
digits
        for (int i = 0; i < index; i++) {
            if (digits[i] > largest) {
                // Updating second largest before updating largest
                secondLargest = largest;
                largest = digits[i];
            } else if (digits[i] > secondLargest && digits[i] != largest) {
                // Updating second largest if current digit is less than
largest but greater than second largest
                secondLargest = digits[i];
            }
        }

        // Displaying the results
        System.out.println("The largest digit is: " + largest);
        System.out.println("The second largest digit is: " +
secondLargest);

        // Closing input object
        input.close();
    }
}
```

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 =>

   a. Take input for a number of persons
   b. Create arrays to store the weight, height, BMI, and weight status of the persons
   c. Take input for the weight and height of the persons
   d. Calculate the BMI of all the persons and store them in an array and also find the weight status of the persons
   e. Display the height, weight, BMI, and weight status of each person
   f. 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

```
import java.util.Scanner;

public class BMICalculator2 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // Take input for the number of persons
        System.out.print("Enter the number of persons: ");
        int n = input.nextInt();

        // Arrays to store height, weight, BMI, and status
        double[] heights = new double[n];
        double[] weights = new double[n];
        double[] bmis = new double[n];
        String[] statuses = new String[n];

        // Taking input for height and weight
        for (int i = 0; i < n; i++) {
            System.out.println("Person " + (i + 1) + ":");
            System.out.print("Enter height (meters): ");
            heights[i] = input.nextDouble();
            System.out.print("Enter weight (kg): ");
            weights[i] = input.nextDouble();

            // Validating inputs
            if (heights[i] <= 0 || weights[i] <= 0) {
                System.out.println("Invalid input. Try again.");
                i--; // Retry for the current person
                continue;
            }

            // Calculate BMI
            bmis[i] = weights[i] / (heights[i] * heights[i]);

            //assigning weight status
            if (bmis[i] <= 18.4) {
                statuses[i] = "Underweight";
            } else if (bmis[i] <= 24.9) {
                statuses[i] = "Normal";
            } else if (bmis[i] <= 39.9) {
                statuses[i] = "Overweight";
            } else {
                statuses[i] = "Obese";
            }
        }

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

            //closing the input
        input.close();
    }
}
```

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

Hint =>

a. Take input for a number of persons

b. 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];
```

c. Take input for weight and height of the persons and for negative values, ask the user to enter positive values

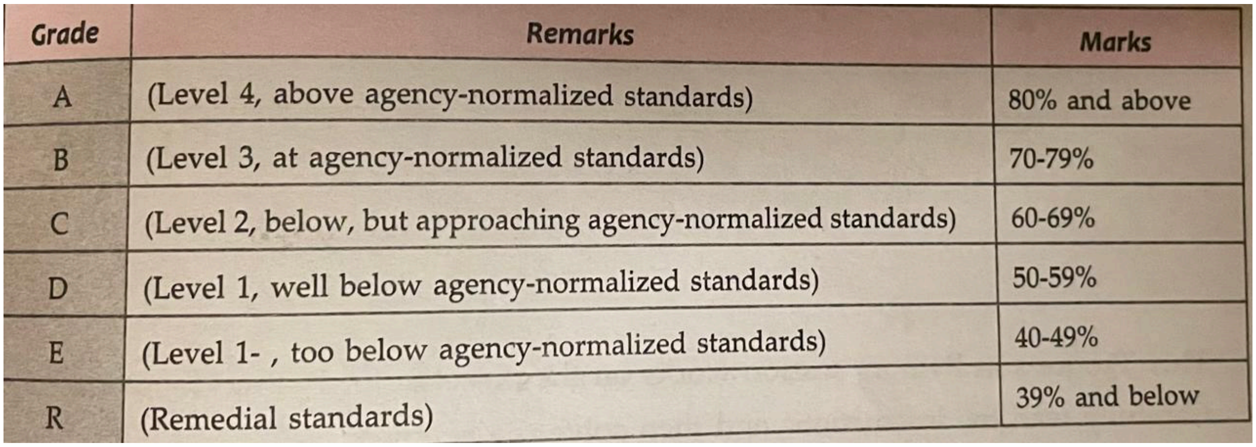
d. Calculate BMI of all the persons and store them in the personData array and also find the weight status and put them in the weightStatus array

e. Display the height, weight, BMI and status of each person

```
import java.util.Scanner;

public class BMICalculator2D {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // Take input for the number of persons
        System.out.print("Enter the number of persons: ");
        int number = input.nextInt();

        // Create a 2D array to store height, weight, and BMI
        double[][] personData = new double[number][3];
        String[] weightStatus = new String[number];

        // Take input for height and weight, calculate BMI, and determine
weight status
        for (int i = 0; i < number; i++) {
            System.out.println("Person " + (i + 1) + ":");

            // Input height and validate
            do {
                System.out.print("Enter height (meters): ");
                personData[i][0] = input.nextDouble();
                if (personData[i][0] <= 0) {
                    System.out.println("Height must be positive. Try
again.");
```

```
                }
            } while (personData[i][0] <= 0);

            // Input weight and validate
            do {
                System.out.print("Enter weight (kg): ");
                personData[i][1] = input.nextDouble();
                if (personData[i][1] <= 0) {
                    System.out.println("Weight must be positive. Try 
again.");
                }
            } while (personData[i][1] <= 0);

            // Calculate BMI and store it in the array
            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] <= 24.9) {
                weightStatus[i] = "Normal";
            } else if (personData[i][2] <= 39.9) {
                weightStatus[i] = "Overweight";
            } else {
                weightStatus[i] = "Obese";
            }
        }

        // Display results
        System.out.println("\nPerson Details:");

System.out.println("-------------------------------------------------------
---");
        for (int i = 0; i < number; i++) {
            System.out.println("Person " + (i + 1) + ":");
            System.out.println("Height: " + personData[i][0] + " meters");
            System.out.println("Weight: " + personData[i][1] + " kg");
            System.out.println("BMI: " + personData[i][2]);
            System.out.println("Status: " + weightStatus[i]);
System.out.println("----------------------------------------------------------
---");
        }
            //closing input
        input.close();
    }
}
```

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

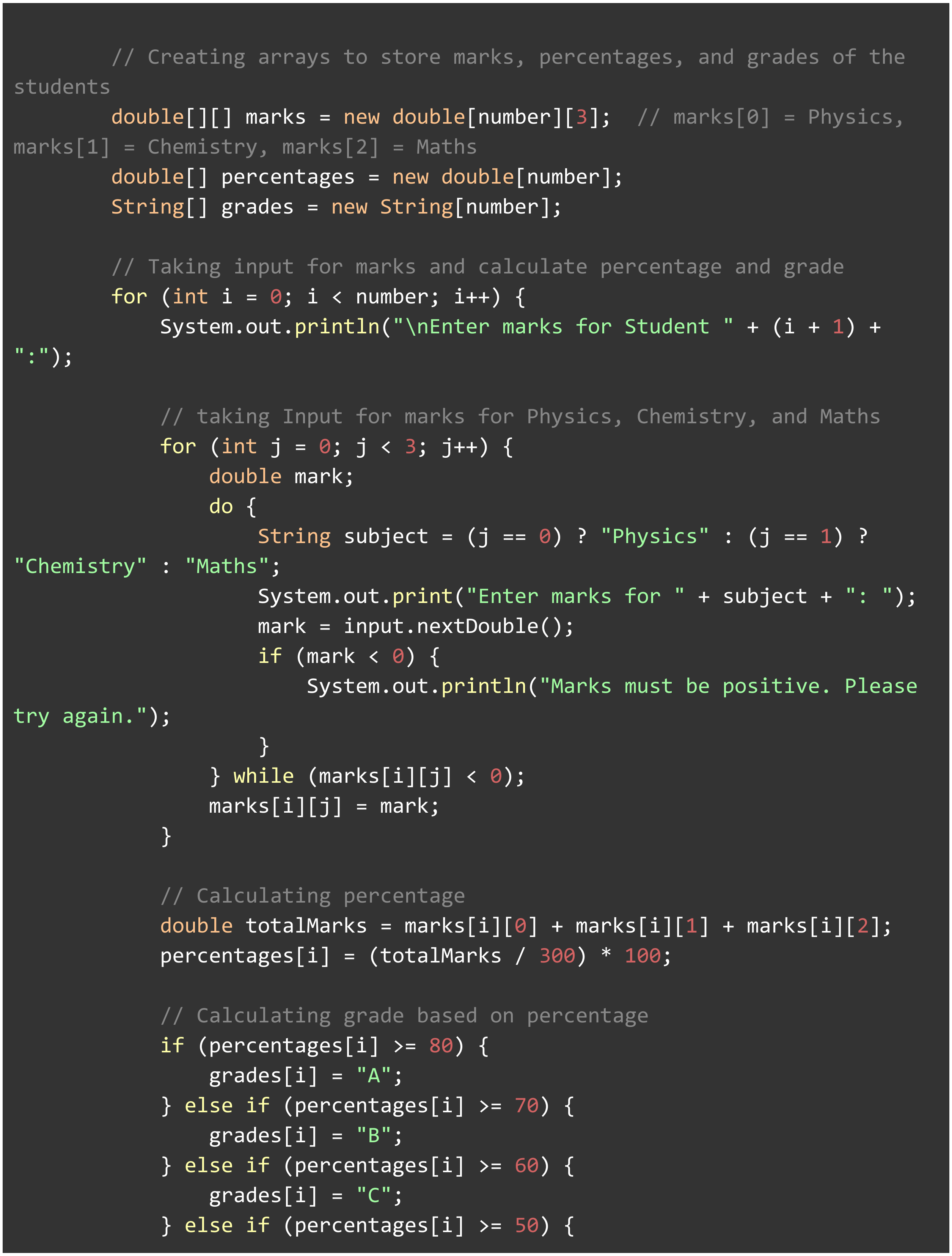


Grade	Remarks	Marks
A	(Level 4, above agency-normalized standards)	80% and above
B	(Level 3, at agency-normalized standards)	70-79%
C	(Level 2, below, but approaching agency-normalized standards)	60-69%
D	(Level 1, well below agency-normalized standards)	50-59%
E	(Level 1- , too below agency-normalized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

Hint =>

a. Take input for the number of students
b. Create arrays to store marks, percentages, and grades of the students
c. 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
d. Calculate the percentage and grade of the students based on the percentage
e. Display the marks, percentages, and grades of each student

```
import java.util.Scanner;

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

        // Taking input for the number of students
        System.out.print("Enter the number of students: ");
        int number = input.nextInt();
```



```
        // Creating arrays to store marks, percentages, and grades of the students
        double[][] marks = new double[number][3];  // marks[0] = Physics, marks[1] = Chemistry, marks[2] = Maths
        double[] percentages = new double[number];
        String[] grades = new String[number];

        // Taking input for marks and calculate percentage and grade
        for (int i = 0; i < number; i++) {
            System.out.println("\nEnter marks for Student " + (i + 1) + ":");

            // taking Input for marks for Physics, Chemistry, and Maths
            for (int j = 0; j < 3; j++) {
                double mark;
                do {
                    String subject = (j == 0) ? "Physics" : (j == 1) ? "Chemistry" : "Maths";
                    System.out.print("Enter marks for " + subject + ": ");
                    mark = input.nextDouble();
                    if (mark < 0) {
                        System.out.println("Marks must be positive. Please try again.");
                    }
                } while (marks[i][j] < 0);
                marks[i][j] = mark;
            }

            // Calculating percentage
            double totalMarks = marks[i][0] + marks[i][1] + marks[i][2];
            percentages[i] = (totalMarks / 300) * 100;

            // Calculating 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";
            }
        }

        // Displaying the marks, percentages, and grades for each student
        System.out.println("\nStudent Results:");

System.out.println("-------------------------------------------------------
-----");
        System.out.println("Student  Physics  Chemistry  Maths  Percentage
Grade");

System.out.println("-------------------------------------------------------
-----");
        for (int i = 0; i < number; i++) {
            System.out.println("Student " + (i + 1) + "   " + marks[i][0] +
"      " + marks[i][1] + "       "
                               + marks[i][2] + "     " + percentages[i] +
"%     " + grades[i]);
        }

            //closing the input
        input.close();
    }
}
```

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 =>

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

```
import java.util.Scanner;

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

        // Take input for the number of students
        System.out.print("Enter the number of students: ");
        int number = input.nextInt();

        // Create a 2D array to store marks for Physics, Chemistry, and
Maths for each student
        double[][] marks = new double[number][3];  // marks[0] = Physics,
marks[1] = Chemistry, marks[2] = Maths
        double[] percentages = new double[number];
        String[] grades = new String[number];

        // Take input for marks and calculate percentage and grade
        for (int i = 0; i < number; i++) {
            System.out.println("\nEnter marks for Student " + (i + 1) +
":");

            // Input marks for Physics, Chemistry, and Maths
            for (int j = 0; j < 3; j++) {
                double mark;
                do {
                    String subject = (j == 0) ? "Physics" : (j == 1) ?
"Chemistry" : "Maths";
                    System.out.print("Enter marks for " + subject + ": ");
                    mark = input.nextDouble();
                    if (mark < 0) {
                        System.out.println("Marks must be positive. Please
try again.");
                    }
                } while (mark < 0);
                marks[i][j] = mark;
            }

            // Calculate percentage
            double totalMarks = marks[i][0] + marks[i][1] + marks[i][2];
            percentages[i] = (totalMarks / 300) * 100;

            // Calculate 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 the marks, percentages, and grades for each student
        System.out.println("\nStudent Results:");

System.out.println("-------------------------------------------------------
-----");
        System.out.println("Student  Physics  Chemistry  Maths  Percentage
Grade");

System.out.println("-------------------------------------------------------
-----");
        for (int i = 0; i < number; i++) {
            System.out.println("Student " + (i + 1) + "   " + marks[i][0] +
"      " + marks[i][1] + "       "
                               + marks[i][2] + "     " + percentages[i] +
"%      " + grades[i]);
        }

        input.close();
    }
}
```

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 =>

a. Take the 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. 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
e. Display the frequency of each digit in the number

```
import java.util.Scanner;

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

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

        // Find the number of digits and store the digits in an array
        int temp = number;
        int count = 0;

        // Count the number of digits in the number
        while (temp != 0) {
            count++;
            temp /= 10;
        }

        // Create an array to store digits
        int[] digits = new int[count];
        temp = number;
        for (int i = count - 1; i >= 0; i--) {
            digits[i] = temp % 10;
            temp /= 10;
        }

        // Create an array to store the frequency of each digit (0-9)
        int[] frequency = new int[10];

        // Loop through the digits array and count the frequency of each
digit
        for (int i = 0; i < count; i++) {
            frequency[digits[i]]++;
        }
        // Display the frequency of each digit
        System.out.println("\nDigit Frequency:");
        for (int i = 0; i < 10; i++) {
            if (frequency[i] > 0) {
                System.out.println("Digit " + i + ": " + frequency[i] + " times");
            }
        }

        input.close();
    }
}
```