

Level 3 Practice Programs

1. Write a LeapYear program that takes a year as input and outputs the Year is a Leap Year or not a Leap Year.

Hint =>

- a. The LeapYear program only works for year ≥ 1582 , corresponding to a year in the Gregorian calendar. So ensure to check for the same.
- b. Further, the Leap Year is a Year divisible by 4 and not 100 unless it is divisible by 400. E.g. 1800 is not a Leap Year and 2000 is a Leap Year.
- c. Write code having multiple **if else** statements based on conditions provided above and a second part having only one if statement and multiple logical

Ans) Code:

```
//import java utility scanner
import java.util.Scanner;
//declare class
public class LeapYearIfElse {
    public static void main(String[] args) {
        // Create a Scanner object to take user input
        Scanner scanner = new Scanner(System.in);
        // Prompt user to enter a year
        System.out.print("Enter a year (>=1582): ");
        int year = scanner.nextInt();
        // Check if the year is valid (Gregorian calendar starts from 1582)
        if (year < 1582) {
            System.out.println("Invalid input! The program works for years >= 1582.");
        } else {
            //Using multiple if-else statements
            if (year % 4 == 0) { // Check if the year is divisible by 4
                if (year % 100 == 0) { // If divisible by 100, check further
                    if (year % 400 == 0) { // If divisible by 400, it's a Leap Year
                        System.out.println(year + " is a Leap Year.");
                    } else { // If divisible by 100 but not by 400, it's NOT a Leap Year
                        System.out.println(year + " is NOT a Leap Year.");
                    }
                }
            } else { // If divisible by 4 but not by 100, it's a Leap Year
                System.out.println(year + " is a Leap Year.");
            }
        } else { // If not divisible by 4, it's NOT a Leap Year
            System.out.println(year + " is NOT a Leap Year.");
        }
    }
}
```

```
}
```

Output Verification:

```
C:\Windows\System32\cmd.exe
C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week2\Level3>java LeapYearIfElse
Enter a year (>=1582): 1583
1583 is NOT a Leap Year.
```

2. Rewrite program 1 to determine Leap Year with single if condition using logical and **&&** and or **||** operators

Ans) Code:

// Import the Scanner class from the java.util package

```
import java.util.Scanner;
```

//declare class

```
public class LeapYear {
```

```
    public static void main(String[] args) {
```

```
        // Create a Scanner object to take user input
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        // Prompt user to enter a year
```

```
        System.out.print("Enter a year (>=1582): ");
```

```
        int year = scanner.nextInt();
```

```
        // Check if the year is valid (Gregorian calendar starts from 1582)
```

```
        if (year < 1582) {
```

```
            System.out.println("Invalid input! The program works for years >= 1582.");
```

```
            //Using a single if statement with logical operators
```

```
            if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
```

```
                System.out.println("(Single If) " + year + " is a Leap Year.");
```

```
            } else {
```

```
                System.out.println("(Single If) " + year + " is NOT a Leap Year.");
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

Output Verification:

```
C:\Windows\System32\cmd.exe
C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week2\Level3>java LeapYear
Enter year to be checked:
1583
Not a leap year !!
```

3.

Write a program to input marks and 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. Ensure the Output clearly shows the Average Mark as well as the Grade and Remarks

Ans) //Input java utility scanner

```
import java.util.Scanner;
```

```
//declare class
```

```
public class GradeCalculator {
```

```
    public static void main(String[] args) {
```

```
        // Create a Scanner object to take user input
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        // Taking input for three subjects
```

```
        System.out.print("Enter marks for Physics: ");
```

```
        int physics = scanner.nextInt(); // Input Physics marks
```

```
        System.out.print("Enter marks for Chemistry: ");
```

```
        int chemistry = scanner.nextInt(); // Input Chemistry marks
```

```
        System.out.print("Enter marks for Maths: ");
```

```
        int maths = scanner.nextInt(); // Input Maths marks
```

```
        // Calculate average marks
```

```
        double average = (physics + chemistry + maths) / 3.0;
```

```
        // Variables to store the Grade and Remarks
```

```
        String grade, remarks;
```

```
        // Determine the grade and remarks based on the given grading scale
```

```
        if (average >= 80) {
```

```
            grade = "A";
```

```
            remarks = "Level 4, above agency-normalized standards";
```

```
        } else if (average >= 70) {
```

```
            grade = "B";
```

```
            remarks = "Level 3, at agency-normalized standards";
```

```
        } else if (average >= 60) {
```

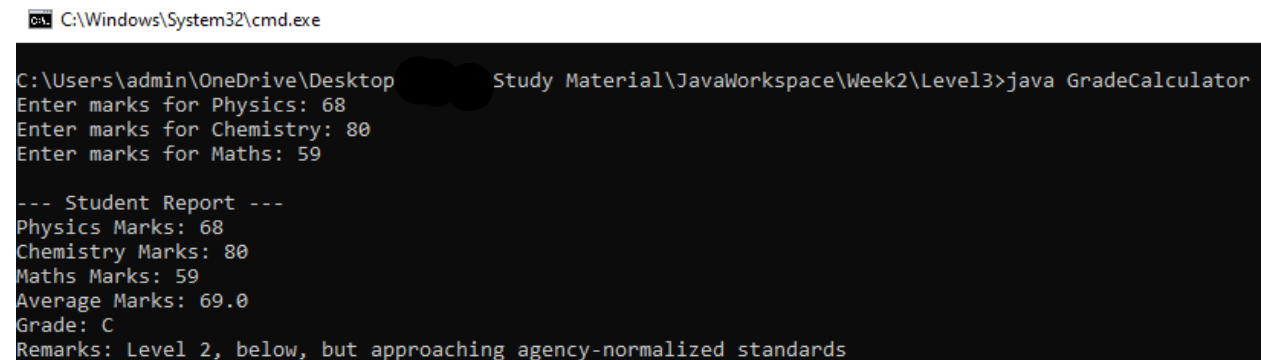
```
            grade = "C";
```

```

        remarks = "Level 2, below, but approaching agency-normalized standards";
    } else if (average >= 50) {
        grade = "D";
        remarks = "Level 1, well below agency-normalized standards";
    } else if (average >= 40) {
        grade = "E";
        remarks = "Level 1-, too below agency-normalized standards";
    } else {
        grade = "R";
        remarks = "Remedial standards";
    }
    // Display the Student Report with all relevant details
    System.out.println("\n--- Student Report ---");
    System.out.println("Physics Marks: " + physics);
    System.out.println("Chemistry Marks: " + chemistry);
    System.out.println("Maths Marks: " + maths);
    System.out.println("Average Marks: " + average);
    System.out.println("Grade: " + grade);
    System.out.println("Remarks: " + remarks);
}
}

```

Output Verification:



```

C:\Windows\System32\cmd.exe
C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week2\Level3>java GradeCalculator
Enter marks for Physics: 68
Enter marks for Chemistry: 80
Enter marks for Maths: 59

--- Student Report ---
Physics Marks: 68
Chemistry Marks: 80
Maths Marks: 59
Average Marks: 69.0
Grade: C
Remarks: Level 2, below, but approaching agency-normalized standards

```

4. Write a Program to check if the given number is a prime number or not

Hint =>

- A number that can be divided exactly only by itself and 1 are Prime Numbers,
- Prime Numbers checks are done for number greater than 1
- Loop through all the numbers from 2 to the user input number and check if the remainder is zero. If the remainder is zero break out from the loop as the number is divisible by some other number and is not a prime number.
- Use isPrime boolean variable to store the result

Ans) Code:

```
//import java utility scanner
```

```
import java.util.Scanner;
//declare class
public class IsPrime {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in); //declare scanner object
        //Get the user input for the number.
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        //Create a boolean variable isPrime and initialize it as true.
        boolean isPrime = true;
        // Prime numbers are greater than 1.
        if (number <= 1) {
            isPrime = false;
        } else {
            //Loop through all the numbers from 2 to the user input number.
            for (int i = 2; i <= number / 2; i++) {
                // Check if the remainder is zero.
                if (number % i == 0) {
                    // If the remainder is zero, the number is not a prime number.
                    isPrime = false;
                    break; // Break out from the loop.
                }
            }
        }
        //Display the result.
        if (isPrime) {
            System.out.println(number + " is a prime number.");
        } else {
            System.out.println(number + " is not a prime number.");
        }
    }
}
```

}

Output verification:

```
C:\Windows\System32\cmd.exe
C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week2\Level3>java IsPrime
Enter a number: 586
586 is not a prime number.
```

5. Create a program to check if a number is armstrong or not. Use the hints to show the steps clearly in the code

Hint =>

- Armstrong Number is a number whose Sum of cubes of each digit results in the original number as in for e.g. $153 = 1^3 + 5^3 + 3^3$
- Get an integer input and store it in the number variable and define sum variable, initialize it to zero and originalNumber variable and assign it to input number variable
- Use the **while** loop till the originalNumber is not equal to zero
- In the **while** loop find the remainder number by using the modulus operator as in **number % 10**. Find the cube of the number and add it to the **sum** variable
- Again in while loop find the quotient of the number and assign it to the original number using number / 10 expression. This removes the last digit of the original number.
- Finally check if the number and the sum are the same, if same its an Armstrong number else not. So display accordingly

Ans) Code:

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

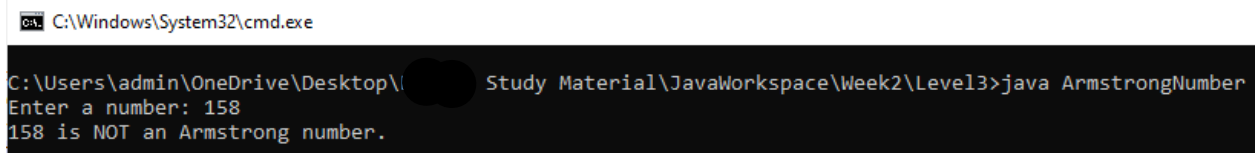
//declare class
public class ArmstrongNumber {
    public static void main(String[] args) {
        int number,sum=0,originalNumber;//declare variable
        Scanner scanner = new Scanner(System.in);
        // Get user input
        System.out.print("Enter a number: ");
        int number = scanner.nextInt(); // Store input number
        originalNumber = number; // Store the original number for comparison
        // check each digit
        while (originalNumber != 0) {
```

```

        int digit = originalNumber % 10; // Get last digit
        sum += (digit * digit * digit); // Add cube of the digit to sum
        originalNumber /= 10; // Remove last digit
    }
    // Check if the sum matches the original number
    if (sum == number) {
        System.out.println(number + " is an Armstrong number.");
    } else {
        System.out.println(number + " is NOT an Armstrong number.");
    }
}
}
}

```

Output Verification:



```

C:\Windows\System32\cmd.exe
C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week2\Level3>java ArmstrongNumber
Enter a number: 158
158 is NOT an Armstrong number.

```

6. Create a program to count the number of digits in an integer.

Hint =>

- Get an integer input for the number variable.
- Create an integer variable count with value 0.
- Use a loop to iterate until number is not equal to 0.
- Remove the last digit from number in each iteration
- Increase count by 1 in each iteration.
- Finally display the count to show the number of digits

Ans) Code:

```

//import java utility scanner
import java.util.Scanner;
//declare class
public class DigitCounter {
    public static void main(String[] args) {
        //Get an integer input for the number variable.
        Scanner scanner = new Scanner(System.in);
    }
}

```

```
System.out.print("Enter an integer: ");
int number = scanner.nextInt();
//Create an integer variable count with the value 0.
int count = 0;
//Use a loop
while (number != 0) {
    //Remove the last digit from the number in each iteration.
    number = number / 10;
    //Increase the count by 1 in each iteration.
    count++;
}
//display the count to show the number of digits.
System.out.println("Number of digits: " + count);
}
```

Output Verification:

```
C:\Windows\System32\cmd.exe
C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week2\Level3>java DigitCounter
Enter an integer: 5864
Number of digits: 4
```


7.

Create a program to find the BMI of a person

Hint =>

- Take user input in double for the weight (in kg) of the person and height (in cm) for the person and store it in the corresponding variable.
- Use the formula $BMI = \text{weight} / (\text{height} * \text{height})$. Note unit is kg/m^2 . For this convert cm to meter
- 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 the Scanner class for user input
import java.util.Scanner;

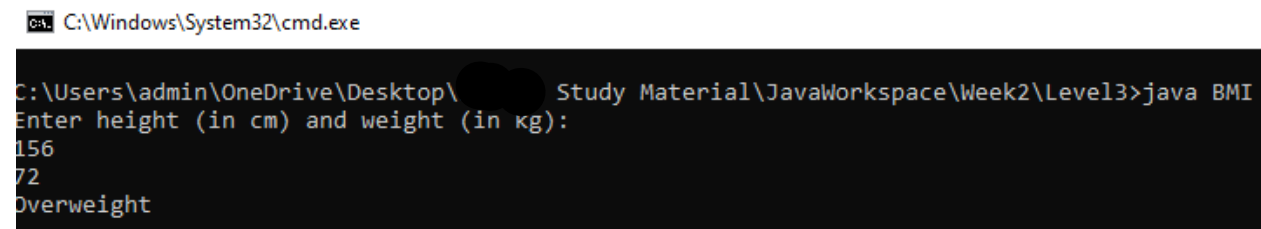
// declare class
public class BMI {
    public static void main(String[] args) {
        // Declare weight and height as integers
        int weight, height;
        double bmi,hm; // BMI should be a double
        // Create Scanner object for input
        Scanner myobj = new Scanner(System.in);
        // Prompt user for input
        System.out.println("Enter height (in cm) and weight (in kg):");
        height = myobj.nextInt(); // Read height in cm
        weight = myobj.nextInt(); // Read weight in kg
        // Convert height from cm to meters
        hm = height * 0.01;
```

```

// Calculate BMI using correct formula: BMI = weight / (height in meters)^2
bmi = weight / (hm * hm);
// Determine BMI category using correct syntax for if-else statements
if (bmi <= 18.4) {
    System.out.println("Underweight");
} else if (bmi >= 18.5 && bmi <= 24.9) {
    System.out.println("Normal");
} else if (bmi >= 25.0 && bmi <= 39.9) {
    System.out.println("Overweight");
} else { // No need for "if" here, as it's the only remaining condition
    System.out.println("Obese");
}
}
}

```

Output Verification:



The screenshot shows a Windows command prompt window with the title bar "C:\Windows\System32\cmd.exe". The command prompt shows the following text:

```

C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week2\Level3>java BMI
Enter height (in cm) and weight (in kg):
156
72
Overweight

```

8. Create a program to check if a number taken from the user is a Harshad Number.

Hint =>

- A Harshad number is an integer which is divisible by the sum of its digits.
For example, 21 which is perfectly divided by 3 (sum of digits: 2 + 1).
- Get an integer input for the number variable.
- Create an integer variable sum with initial value 0.
- Create a while loop to access each digit of the number.
- Inside the loop, add each digit of the number to sum.
- Check if the number is perfectly divisible by the sum.
- If the number is divisible by the sum, print Harshad Number. Otherwise, print Not a Harshad Number.

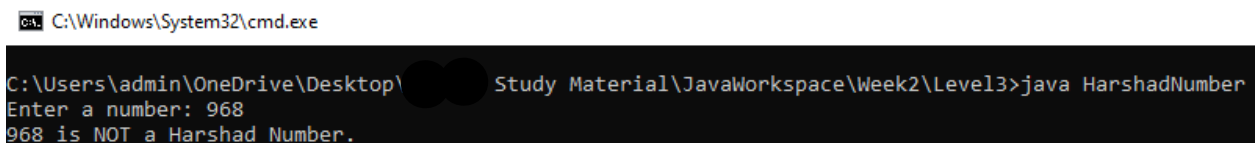
Ans) Code:

```

import java.util.Scanner; // Import Scanner for user input
//declare class
public class HarshadNumber {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in); // Create Scanner object
        int number,temp,sum=0;//declare variables
        // Get user input
        System.out.print("Enter a number: ");
        number = scanner.nextInt(); // Store input
        sum = 0; // To store the sum of digits
        temp = number; // Use a temporary variable to extract digits
        // Extract each digit and compute the sum
        while (temp > 0) {
            int digit = temp % 10; // Get the last digit
            sum += digit; // Add the digit to sum
            temp /= 10; // Remove the last digit by dividing by 10
        }
        // Check if the number is divisible by the sum of its digits
        if (number % sum == 0) {
            System.out.println(number + " is a Harshad Number.");
        } else {
            System.out.println(number + " is NOT a Harshad Number.");
        }
    }
}

```

Output Verification:



```

C:\Windows\System32\cmd.exe
C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week2\Level3>java HarshadNumber
Enter a number: 968
968 is NOT a Harshad Number.

```

9. Create a program to check if a number is an Abundant Number.

Hint =>

- a. An abundant number is an integer in which the sum of all the divisors of the number is greater than the number itself. For example,

Divisor of 12: 1, 2, 3, 4, 6

Sum of divisor: $1 + 2 + 3 + 4 + 6 = 16 > 12$

- b. Get an integer input for the number variable.
- c. Create an integer variable sum with initial value 0.
- d. Run a for loop from $i = 1$ to $i < \text{number}$.
- e. Inside the loop, check if number is divisible by i .
- f. If true, add i to sum.
- g. Outside the loop Check if sum is greater than number.
- h. If the sum is greater than the number, print Abundant Number. Otherwise, print Not an Abundant Number.

Ans) Code:

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

//declare class
public class AbundantNumber{

    public static void main(String[] args) {

        int number,sum;//declare variable

        Scanner scanner = new Scanner(System.in);//declare scanner object

        //Get an integer input for the number variable
        System.out.print("Enter a positive integer: ");
        number = scanner.nextInt();

        //Create an integer variable sum with an initial value of 0
        sum = 0;

        //Run a for loop from i = 1 to i < number
        for (int i = 1; i < number; i++) {

            // Step 5: Inside the loop, check if the number is divisible by i
            if (number % i == 0) {

                // If true, add i to the sum
                sum += i;

            }

        }

    }

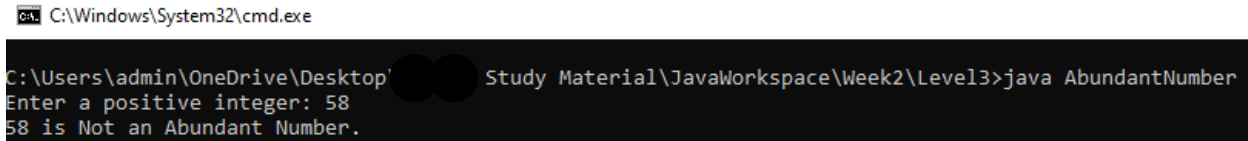
}
```

```

    }
    //Outside the loop, check if the sum is greater than the number
    if (sum > number) {
        //If the sum is greater than the number, print Abundant Number
        System.out.println(number + " is an Abundant Number.");
    } else {
        // Otherwise, print Not an Abundant Number
        System.out.println(number + " is Not an Abundant Number.");
    }
}
}
}

```

Output Verification:



```

C:\Windows\System32\cmd.exe
C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week2\Level3>java AbundantNumber
Enter a positive integer: 58
58 is Not an Abundant Number.

```

10. Write a program to create a calculator using **switch...case**.

Hint =>

- Create two double variables named first and second and a String variable named op.
- Get input values for all variables.
- The input for the operator can only be one of the four values: "+", "-", "*", or "/".
- Run a for loop from i = 1 to i < number.
- Based on the input value of the op, perform specific operations using the **switch...case** statement and print the result.
- If op is +, perform addition between first and second; if it is -, perform subtraction and so on.
- If op is neither of those 4 values, print Invalid Operator.

Ans) Code:

```

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

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

```

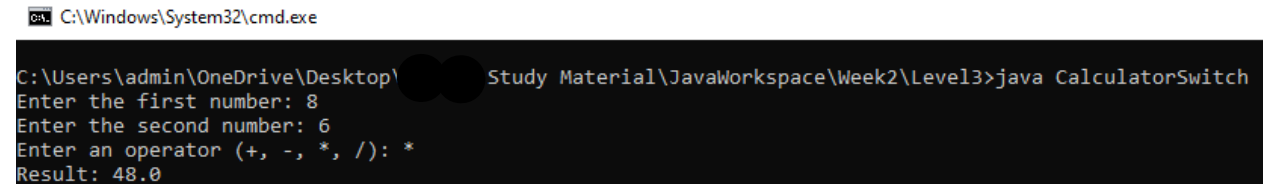
```
//Create two double variables named first and second and a String variable named op.
double first, second;
String op;
//Get input values for all variables.
System.out.print("Enter the first number: ");
first = scanner.nextDouble();
System.out.print("Enter the second number: ");
second = scanner.nextDouble();
System.out.print("Enter an operator (+, -, *, /): ");
op = scanner.next();
//Perform specific operations using switch...case statement and print the result.
switch (op) {
    case "+":
        System.out.println("Result: " + (first + second));
        break;
    case "-":
        System.out.println("Result: " + (first - second));
        break;
    case "*":
        System.out.println("Result: " + (first * second));
        break;
    case "/":
        //prevent division by zero
        if (second != 0) {
            System.out.println("Result: " + (first / second));
        } else {
            System.out.println("Error! Division by zero is not allowed.");
        }
        break;
    default:
        //op is neither of those 4 values, print Invalid Operator.
```

```

        System.out.println("Invalid Operator!");
        break;
    }
}
}

```

Output Verification:



```

C:\Windows\System32\cmd.exe
C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week2\Level3>java CalculatorSwitch
Enter the first number: 8
Enter the second number: 6
Enter an operator (+, -, *, /): *
Result: 48.0

```

11. Write a program **DayOfWeek** that takes a date as input and prints the day of the week that the date falls on. Your program should take three command-line arguments: m (month), d (day), and y (year). For m use 1 for January, 2 for February, and so forth. For output print 0 for Sunday, 1 for Monday, 2 for Tuesday, and so forth. Use the following formulas, for the Gregorian calendar (where / denotes integer division):

$$y_0 = y - (14 - m) / 12$$

$$x = y_0 + y_0/4 - y_0/100 + y_0/400$$

$$m_0 = m + 12 \times ((14 - m) / 12) - 2$$

$$d_0 = (d + x + 31m_0 / 12) \bmod 7$$

Ans) Code:

```

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

//declare class
public class DayOfWeek {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in); //declare scanner object
        // User input for month, day, and year
        System.out.print("Enter month (1-12): ");
        int m = scanner.nextInt();
        System.out.print("Enter day: ");
    }
}

```

```
int d = scanner.nextInt();
System.out.print("Enter year: ");
int y = scanner.nextInt();
// Apply the formula
int y0 = y - (14 - m) / 12;
int x = y0 + y0 / 4 - y0 / 100 + y0 / 400;
int m0 = m + 12 * ((14 - m) / 12) - 2;
int dayOfWeek = (d + x + (31 * m0) / 12) % 7;
// Print the day of the week
System.out.println("Day of the week: " + dayOfWeek);
}
}
```

Output Verification:

```
C:\Windows\System32\cmd.exe
C:\Users\admin\OneDrive\Desktop\Study Material\JavaWorkspace\Week2\Level3>java DayOfWeek
Enter month (1-12): 5
Enter day: 8
Enter year: 2021
Day of the week: 6
```