**Conditional Statements - Level 3 - 11 Practice Problems**

Soham Roy,

RA2411033010148,

Z2 Section,

CSE w/s SWE

**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 =>

1. The LeapYear program only works for year >= 1582, corresponding to a year in the Gregorian calendar. So ensure to check for the same.
2. 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.
3. Write code having multiple *if else* statements based on conditions provided above and a second part having only one if statement and multiple logical

**Program:**

/\*\*A LeapYear program with multiple if-else statements that takes a year as input and outputs the Year is a Leap Year or not a Leap Year\*/

package step\_program;

import java.util.\*; //Importing java.util.\*

public class LeapYearWithoutLogical

{

public static void main(String args[])

{

int year; //Initializing the variables

boolean flag = false;

Scanner sc=new Scanner(System.*in*); //Initializing Scanner object

System.*out*.println("Enter the year:");

year = sc.nextInt(); //Inputting the year from the user

if(year >= 1582) //Checking first condition

{

if(year % 4 == 0) //Checking second condition

{

if(year % 100 == 0) //Checking third condition

{

if(year % 400 == 0) //Checking fourth condition

flag = true; //If all Condition matches

} //Close If Block

else

flag = true; //If all Condition matches

} //Close If Block

} //Close If Block

if(flag) //Final Result

System.*out*.println(year + " is a Leap Year"); //Displaying Final Output

else

System.*out*.println(year + " is not a Leap Year"); //Displaying Final Output

} //End Method

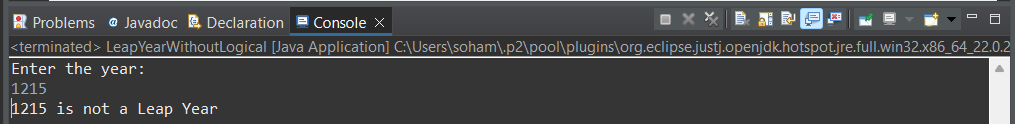
} //End Class

**Output:**

I/P:

1215

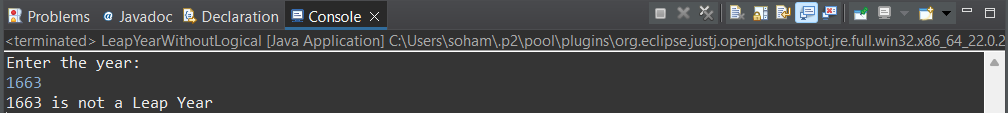
O/P:



I/P:

1663

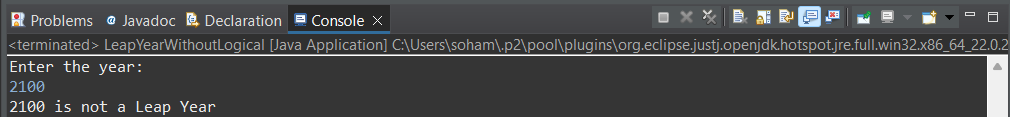
O/P:



I/P:

2100

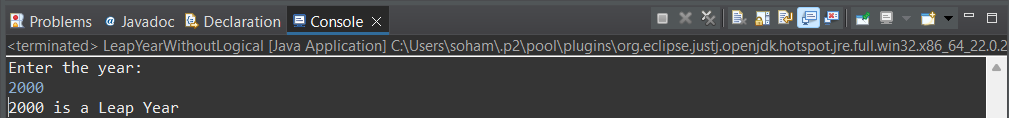
O/P:



I/P:

2000

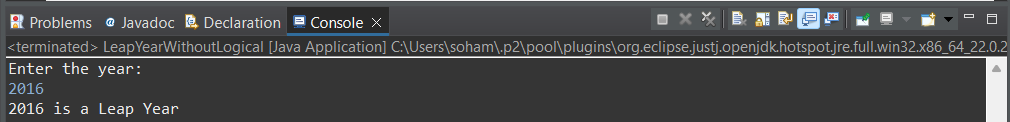
O/P:



I/P:

2016

O/P:



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

**Program:**

/\*\*A LeapYear program with logical operators that takes a year as input and outputs the Year is a Leap Year or not a Leap Year\*/

package step\_program;

import java.util.\*; //Importing java.util.\*

public class LeapYearWithLogical

{

public static void main(String args[])

{

int year; //Initializing the variables

boolean flag = false;

Scanner sc=new Scanner(System.***in***); //Initializing Scanner object

System.***out***.println("Enter the year:");

year = sc.nextInt(); //Inputting the year from the user

if(year >= 1582 && year % 4 == 0 && (year % 400 == 0 || year % 100 != 0)) //Checking all the Conditions at once using logical operators

System.***out***.println(year + " is a Leap Year"); //Displaying Final Output

else

System.***out***.println(year + " is not a Leap Year"); //Displaying Final Output

} //End Method

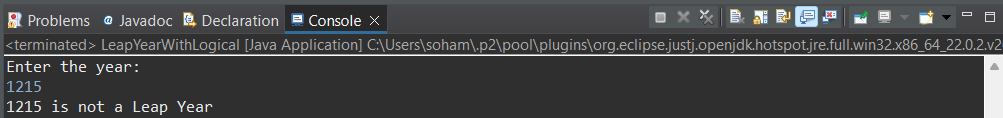
} //End Class

**Output:**

I/P:

1215

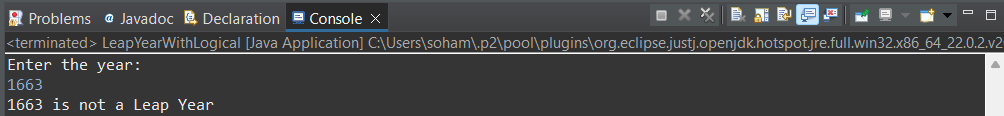
O/P:



I/P:

1663

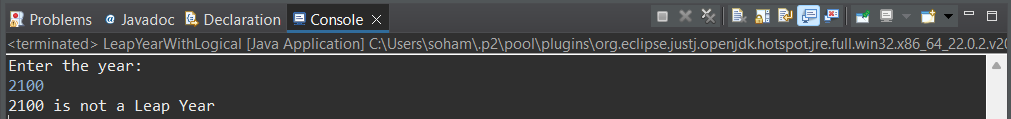
O/P:



I/P:

2100

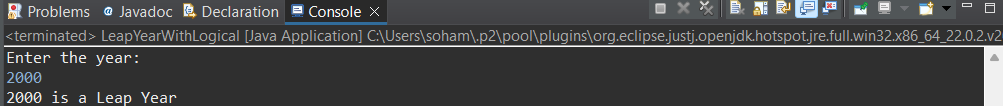
O/P:



I/P:

2000

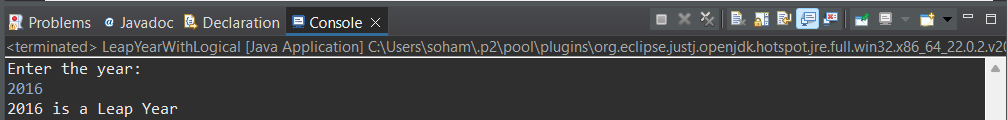
O/P:



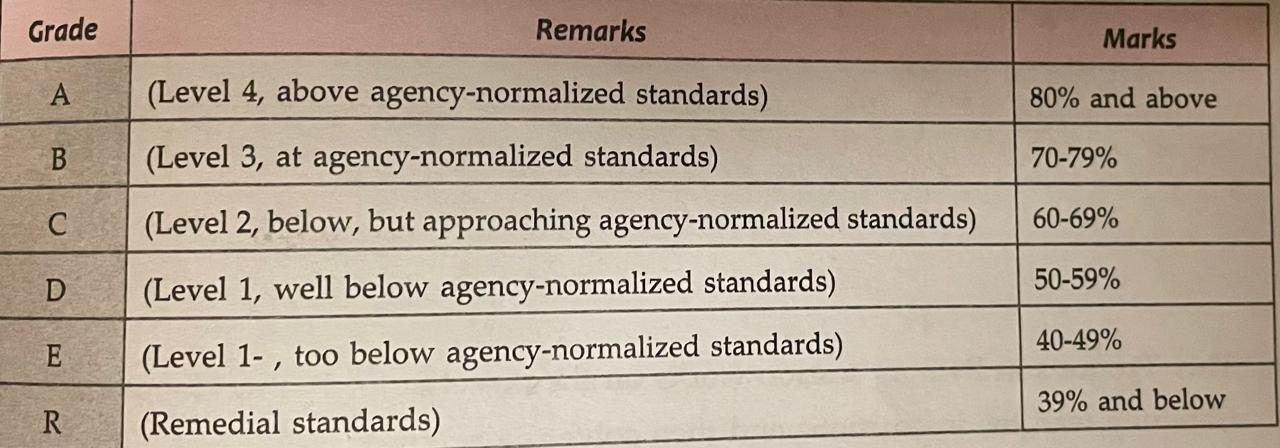
I/P:

2016

O/P:



**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



Hint =>

1. Ensure the Output clearly shows the Average Mark as well as the Grade and Remarks

**Program:**

/\*\*A program to input marks and 3 subjects physics, chemistry and maths and compute the percentage and calculate the grade \*/

package step\_program;

import java.util.\*; //Importing java.util.\* for Scanner class

public class MarksGrade

{

public static void main(String args[])

{

int physics, chemistry, maths; //Initializing variables

double percentage;

Scanner sc=new Scanner(System.***in***); //Initializing Scanner object

System.***out***.println("Enter the marks in Physics, Chemistry and Maths:");

physics = sc.nextInt(); //Inputting the Physics marks from the user

chemistry = sc.nextInt(); //Inputting the Chemistry marks from the user

maths = sc.nextInt(); //Inputting the Maths marks from the user

percentage = (physics + chemistry + maths) / 3.0; //Calculating the percentage

System.***out***.println("Average Marks = " + percentage); //Displaying Final Output

if(percentage >= 80) //Checking which Grade the student will get

System.***out***.println("Grade A : Level 4, above agency-normalized standards"); //Displaying Final Output

else if(percentage >= 70)

System.***out***.println("Grade B : Level 3, at agency-normalized standards"); //Displaying Final Output

else if(percentage >= 60)

System.***out***.println("Grade C : Level 2, below, but approaching agency-normalized standards"); //Displaying Final Output

else if(percentage >= 50)

System.***out***.println("Grade D : Level 1, well below agency-normalized standards"); //Displaying Final Output

else if(percentage >= 40)

System.***out***.println("Grade E : Level 1-, too below agency-normalized standards"); //Displaying Final Output

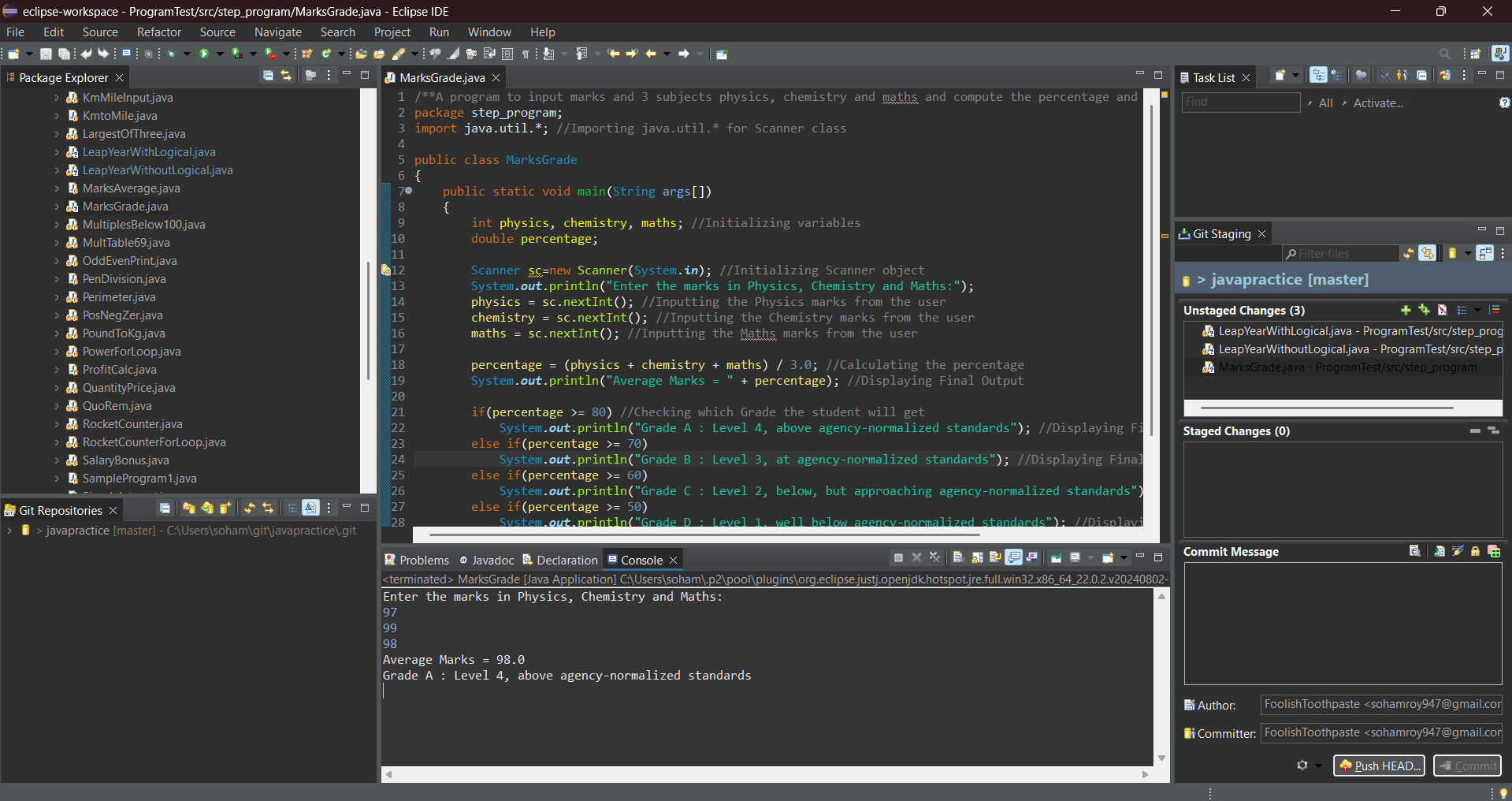
else //Below 40, end of if else ladder

System.***out***.println("Grade R : Remedial status"); //Displaying Final Output

} //End Method

} // End Class

**Output:**

****

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

Hint =>

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

**Program:**

/\*\*A program to check if the given number is a prime number or not\*/

package step\_program;

import java.util.\*; //Importing java.util.\* for Scanner class

public class Prime

{

public static void main(String args[])

{

int number; //Initializing the variables

boolean isPrime = true;

Scanner sc=new Scanner(System.***in***); //Initializing Scanner class

System.***out***.println("Enter the number:");

number = sc.nextInt(); //Inputting the number from the user

if(number > 1) //Checking if the input number is greater than 1

{

for(int i = 2; i < number; i++) //For Loop

{

if(number % i == 0) //Checking if i is a factor of number

{

isPrime = false; //Concluding that the number is not Prime

break; //Breaking out of loop

} //End If Block

} //End Loop

}

else

isPrime = false; //Concluding that the number is not Prime as it is less than 1

if(isPrime) //Final Check

System.***out***.println(number + " is a prime number."); //Displaying Final Output

else

System.***out***.println(number + " is not a prime number."); //Displaying Final Output

} //End Method

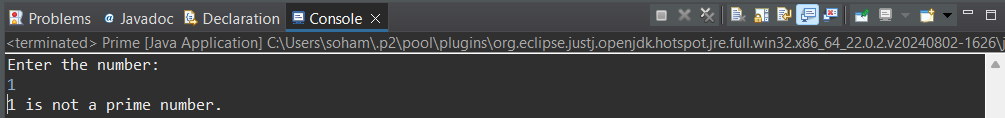
} //End Class

**Output**:

I/P:

1

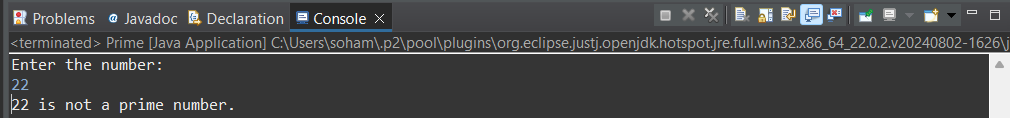
O/P:



I/P:

22

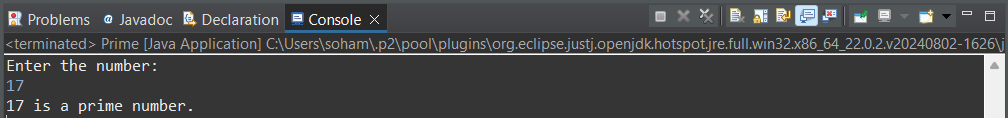
O/P:



I/P:

17

O/P:



**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 =>

1. 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
2. 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
3. Use the *while* loop till the originalNumber is not equal to zero
4. In the *while* loop find the reminder number by using the modulus operator as in  *number % 10*. Find the cube of the number and add it to the *sum* variable
5. 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.
6. Finally check if the number and the sum are the same, if same its an Armstrong number else not. So display accordingly

**Program:**

/\*\*A program to check if a number is Armstrong or not\*/

package step\_program;

import java.util.\*; //Importing java.util.\* for Scanner class

public class Armstrong

{

public static void main(String args[])

{

int number, originalNumber, remainder, cube, sum = 0; //Initializing variables

Scanner sc=new Scanner(System.***in***); //Initializing Scanner object

System.***out***.println("Enter the number:");

number = sc.nextInt(); //Inputting the number from the user

originalNumber = number; //Storing the value of input number in separate variable

while(number > 0) //While Loop

{

remainder = number % 10; //Extracting the last digit of the number

cube = remainder \* remainder \* remainder; //Calculating the cube of the last digit

sum+= cube; //Adding the cube of the last digit to the sum

number/= 10; //Removing the last digit from the number

} //End While Loop

if(sum == originalNumber) //Checking if the sum of the cubes of the digits is equal to the original number

System.***out***.println(originalNumber + " is an Armstrong number"); //Displaying Final Output

else

System.***out***.println(originalNumber + " is not an Armstrong number"); //Displaying Final Output

} //End Method

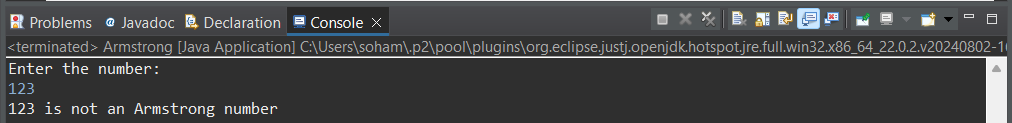
} //End Class

**Output:**

I/P:

123

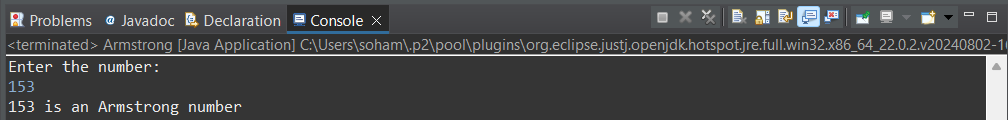
O/P:



I/P:

153

O/P:



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

Hint =>

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

**Program:**

/\*\*A program to count the number of digits in an integer\*/

package step\_program;

import java.util.\*; //Importing java.util.\* for Scanner class

public class DigiCount

{

public static void main(String args[])

{

int number, number2, count = 0; //Initializing variables

Scanner sc=new Scanner(System.***in***); //Initializing Scanner object

System.***out***.println("Enter the number:");

number = sc.nextInt(); //Entering the number from the user

number2 = number; //Storing the value of the original number

while(number > 0) //While Loop

{

count++; //Incrementing the count of the digits

number/= 10; //Removing the last digit from the number

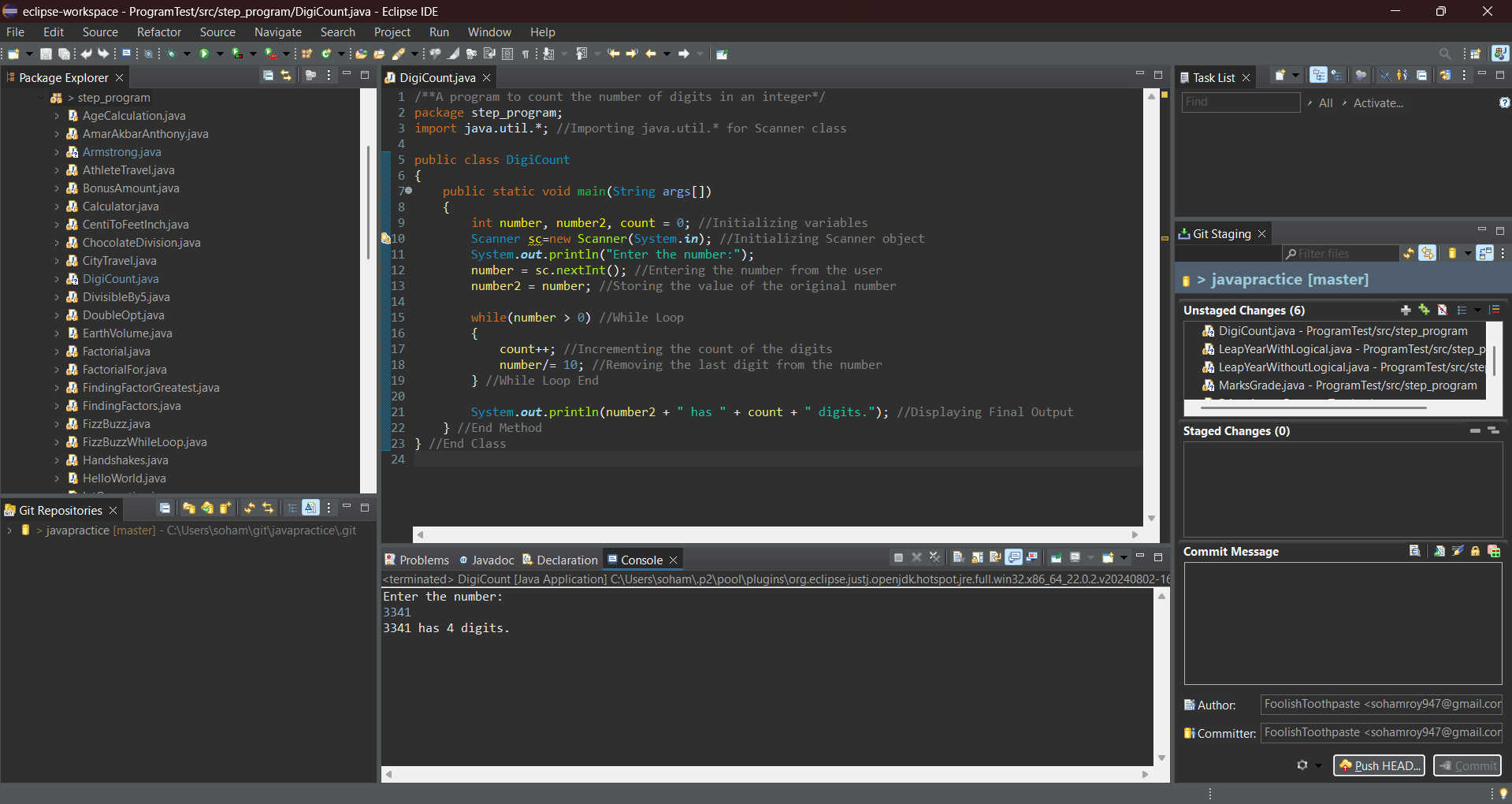
} //While Loop End

System.***out***.println(number2 + " has " + count + " digits."); //Displaying Final Output

} //End Method

} //End Class

**Output:**

****

**7.** Create a program to find the BMI of a person

Hint =>

1. 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.
2. Use the formula BMI = weight / (height \* height). Note unit is kg/m^2. For this convert cm to meter
3. Use the table to determine the weight status of the person

**Program:**

/\*A program to find the BMI of a person\*\*/

package step\_program;

import java.util.\*; //Importing java.util.\* for Scanner class

public class BMI

{

public static void main(String args[])

{

double weight, height, heightMeter, BMI; //Initializing variables

Scanner sc=new Scanner(System.***in***); //Initializing Scanner class

System.***out***.println("Enter the weight (in kg) and height (in cm):");

weight = sc.nextDouble(); //Inputting the weight of the person in kg from the user

height = sc.nextDouble(); //Inputting the height of the person in cm from the user

heightMeter = height / 100; //Converting the height from centimeter to meter

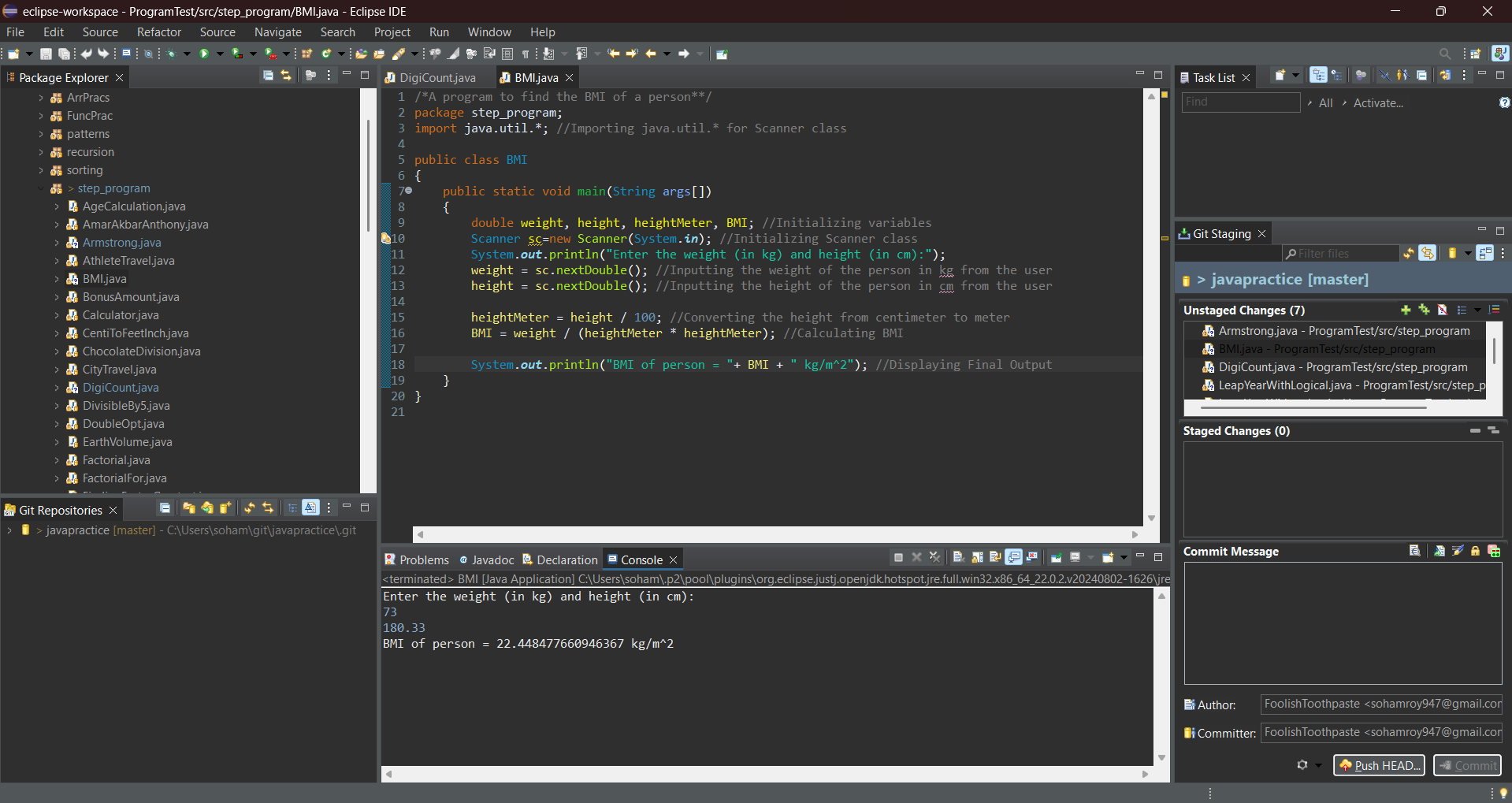
BMI = weight / (heightMeter \* heightMeter); //Calculating BMI

System.***out***.println("BMI of person = "+ BMI + " kg/m^2"); //Displaying Final Output

}

}

**Output:**

****

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

Hint =>

1. 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).

1. Get an integer input for the number variable.
2. Create an integer variable sum with initial value 0.
3. Create a while loop to access each digit of the number.
4. Inside the loop, add each digit of the number to sum.
5. Check if the number is perfectly divisible by the sum.
6. If the number is divisible by the sum, print Harshad Number. Otherwise, print Not a Harshad Number.

**Program:**

/\*\*A program to check if a number taken from the user is a Harshad Number\*/

package step\_program;

import java.util.\*; //Importing java.util.\* for Scanner class

public class HarshadNumber

{

public static void main(String args[])

{

int number, numberOriginal, sum = 0, digit; //Initializing variables

Scanner sc=new Scanner(System.***in***); //Initializing Scanner object

System.***out***.println("Enter the number:");

number = sc.nextInt(); //Inputting the number from the user

numberOriginal = number; //Storing the value of the number

while(number > 0) //While Loop

{

digit = number % 10; //Extracting the last digit of the number

sum+= digit; //Adding last digit of the number to sum

number/= 10; //Removing the last digit of the number

} //End While Loop

if(numberOriginal % sum == 0) //Checking if the number is perfectly divisible by the sum of its digits

System.***out***.println("Harshad Number"); //Displaying Final Output

else

System.***out***.println("Not a Harshad Number"); //Displaying Final Output

} //End Method

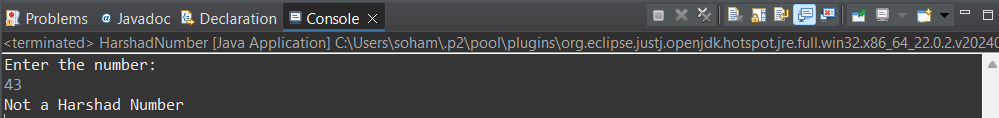
} //End Class

**Output:**

I/P:

43

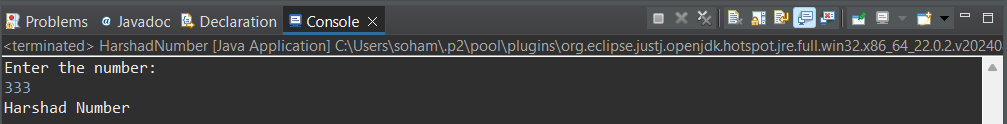
O/P:



I/P:

333

O/P:



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

Hint =>

1. 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

1. Get an integer input for the number variable.
2. Create an integer variable sum with initial value 0.
3. Run a for loop from i = 1 to i < number.
4. Inside the loop, check if number is divisible by i.
5. If true, add i to sum.
6. Outside the loop Check if sum is greater than number.
7. If the sum is greater than the number, print Abundant Number. Otherwise, print Not an Abundant Number.

**Program:**

/\*\*A program to check if a number is an Abundant Number\*/

package step\_program;

import java.util.\*; //Importing java.util.\* for Scanner class

public class AbundantNumber

{

public static void main(String args[])

{

int number, sum = 0; //Initializing variables

Scanner sc=new Scanner(System.***in***); //Initializing Scanner object

System.***out***.println("Enter the number:");

number = sc.nextInt(); //Inputting the number from the user

for(int i = 1; i < number; i++) //For Loop

{

if(number % i == 0) //Checking if i perfectly divides the number

sum+= i; //Adding i to the sum

} //End For Loop

if(sum > number) //Checking if the sum of the divisors is larger than the number

System.***out***.println("Abundant Number"); //Displaying Final Output

else

System.***out***.println("Not an Abundant Number"); //Displaying Final Output

} //End Method

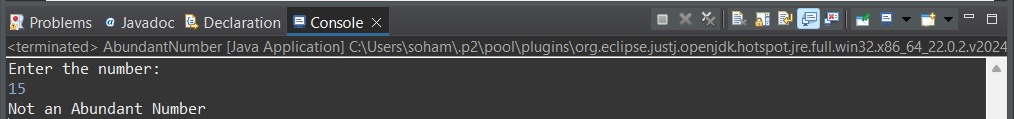
} //End Class

**Output:**

I/P:

15

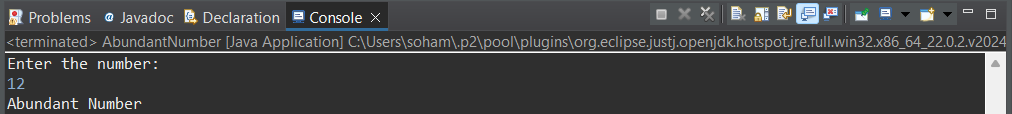
O/P:



I/P:

12

O/P:



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

Hint =>

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

**Program:**

/\*\*A program to create a calculator using switch...case\*/

package step\_program;

import java.util.\*; //Importing java.util.\* for Scanner class

public class CalculatorSwitchCase

{

public static void main(String args[])

{

double first, second; //Initializing the number variables

String op; //Initializing operator variable

Scanner sc=new Scanner(System.***in***); //Initializing Scanner object

System.***out***.println("Enter the two numbers to perform the operation on:");

first = sc.nextDouble(); //Inputting the first number from the user

second = sc.nextDouble(); //Inputting the second number from the user

sc.nextLine(); //nextLine to avoid input stream error

System.***out***.println("Enter the operator: + or - or \* or /");

op = sc.nextLine(); //Inputting the operator from the user

switch(op) //Switch block

{

case "+": //If operator is +

System.***out***.println("Sum = " + (first + second)); //Displaying Final Output

break;

case "-": //If operator is -

System.***out***.println("Difference = " + (first - second)); //Displaying Final Output

break;

case "\*": //If operator is \*

System.***out***.println("Product = " + (first \* second)); //Displaying Final Output

break;

case "/": //If operator is /

System.***out***.println("Division = " + (first / second)); //Displaying Final Output

break;

default: //If operator is neither of the available options

System.***out***.println("Invalid Operator"); //Displaying Final Output

} //End Switch Block

} //End Method

} //End Class

**Output:**

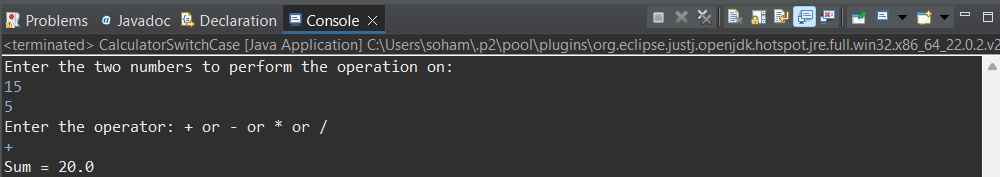
I/P:

15

5

+

O/P:



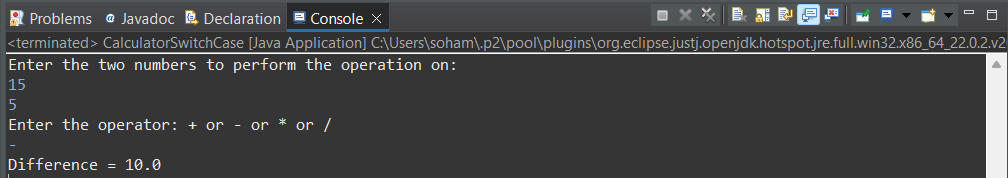
I/P:

15

5

-

O/P:



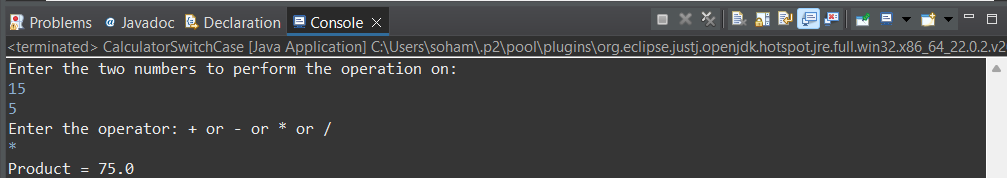
I/P:

15

5

\*

O/P:



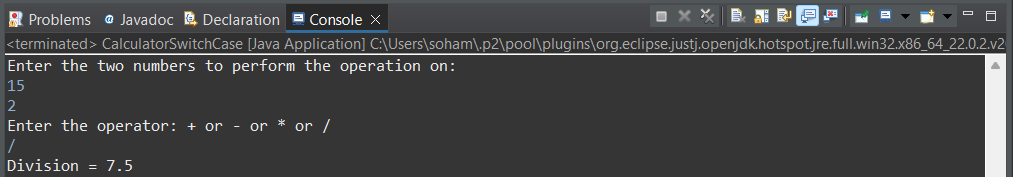
I/P:

15

2

/

O/P:



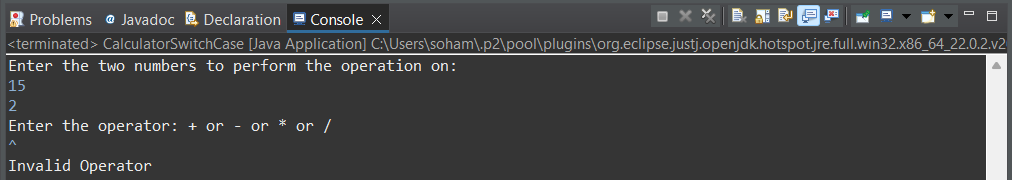
I/P:

15

2

^

O/P:



**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 × ((14 − *m*) / 12) − 2

*d*0 = (*d* + *x* + 31*m*0 / 12) mod 7

**Program:**

/\*\*A program DayOfWeek that takes a date as input and prints the day of the week that the date falls on\*/

package step\_program;

import java.util.\*; //Importing java.util.\* for Scanner class

public class DayofWeek

{

public static void main(String args[])

{

int m, d, y, y0, x, m0, d0; //Initializing the variables

Scanner sc=new Scanner(System.***in***); //initializing Scanner object

System.***out***.println("Enter the date in the following format: Month, Day, Year");

m = sc.nextInt(); //Inputting the month value from the user

d = sc.nextInt(); //Inputting the day value from the user

y = sc.nextInt(); //Inputting the year value from the user

//Equations for finding out the day of the week

y0 = y - (14 - m) / 12;

x = y0 + y0/4 - y0/100 + y0/400;

m0 = m + 12 \* ((14 - m) / 12) - 2;

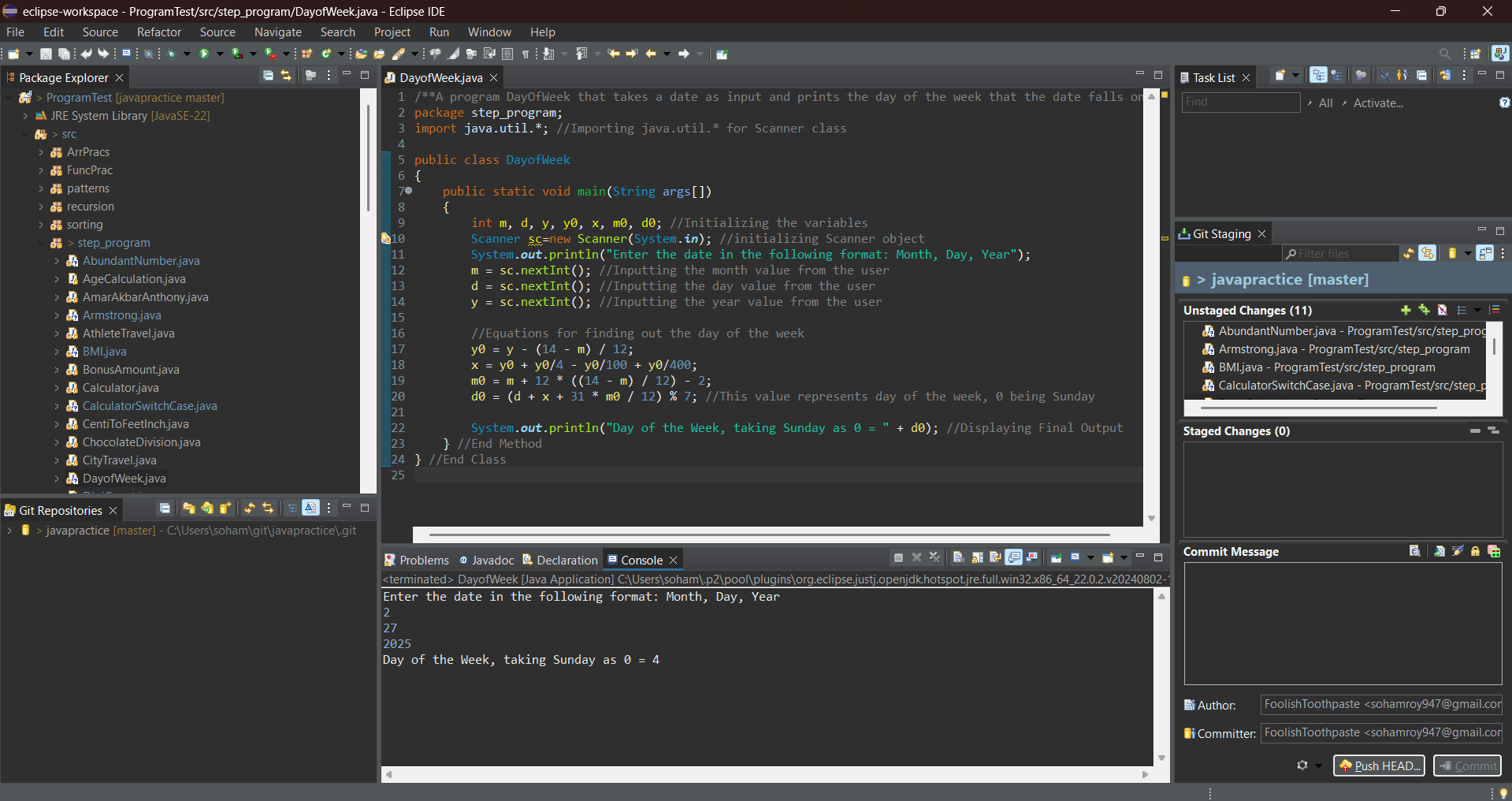
d0 = (d + x + 31 \* m0 / 12) % 7; //This value represents day of the week, 0 being Sunday

System.***out***.println("Day of the Week, taking Sunday as 0 = " + d0); //Displaying Final Output

} //End Method

} //End Class

**Output:**

****