**Batch: A2 Roll No.: 16010123032**

**Experiment / assignment / tutorial No.1**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **TITLE : Armstrong number** |

**AIM:**

Write a Java program to display armstrong numbers in the given range (Make use of a function).

Variations :

Implementation of Program with One class

Accessibility with static and non-static methods within class and outside class.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected OUTCOME of Experiment:**

CO1:Apply the features of object oriented programming languages. (C++ and

Java)

CO2:Explore arrays, vectors, classes and objects in C++ and Java **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. E. Balagurusamy, “Programming with Java”, McGraw-Hill.
2. E. Balagurusamy, “Object Oriented Programming with C++”, McGraw-Hill.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Pre Lab/ Prior Concepts:**

The Scanner class is a class in java.util, which allows the user to read values of various types. There are far more methods in class Scanner than you will need in this course. We only cover a small useful subset, ones that allow us to read in numeric values from either the keyboard or file without having to convert them from strings and determine if there are more values to be read.

Scanner in = new Scanner(System.in);  // System.in is an InputStream

 Numeric and String Methods

|  |  |
| --- | --- |
| **Method** | **Returns** |
| int nextInt() | Returns the next token as an int. If the next token is not an integer,InputMismatchException is thrown. |
| long nextLong() | Returns the next token as a long. If the next token is not an integer,InputMismatchException is thrown. |
| float nextFloat() | Returns the next token as a float. If the next token is not a float or is out of range, InputMismatchException is thrown. |
| double nextDouble() | Returns the next token as a long. If the next token is not a float or is out of range, InputMismatchException is thrown. |
| String next() | Finds and returns the next complete token from this scanner and returns it as a string; a token is usually ended by whitespace such as a blank or line break. If not token exists,NoSuchElementException is thrown. |
| String nextLine() | Returns the rest of the current line, excluding any line separator at the end. |
| void close() | Closes the scanner. |

The Scanner looks for tokens in the input. A token is a series of characters that ends with what Java calls whitespace. A whitespace character can be a blank, a tab character, a carriage return. Thus, if we read a line that has a series of numbers separated by blanks, the scanner will take each number as a separate token. .

The numeric values may all be on one line with blanks between each value or may be on separate lines.   Whitespace characters (blanks or carriage returns) act as separators.  The next method returns the next input value as a string, regardless of what is keyed.  For example, given the following code segment and data

* int number = in.nextInt();
* float real = in.nextFloat();
* long number2 = in.nextLong();
* double real2 = in.nextDouble();
* String string = in.next();

**Algorithm:**

* **Input**: Obtain the positive integer that you want to check.
* **Count the Digits**:
* **Calculate the Sum of Digits Raised to the Power of the Number of Digits**:
* Extract each digit from the number. You can do this by repeatedly dividing the number by 10 and taking the remainder (i.e., the last digit).
* Raise each digit to the power of the total number of digits.
* Sum up these powered digits.
* **Compare the Sum with the Original Number**:
* If the sum of the powered digits equals the original number, then the number is an Armstrong number.
* Otherwise, it is not.

**Implementation details:**

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        int count=0;

        int number;

        System.out.println("Enter a Number: ");

        Scanner sc = new Scanner(System.in);

        number = sc.nextInt();

        int n = number;

        int num = number;

        while(n!=0)

        {

            n = n/10;

            count++;

        }

        int armstrong=0;

        while(num!=0)

        {

            int digit = num % 10;

            armstrong+=(int)(Math.pow(digit,count));

            num/=10;

        }

        if(armstrong == number)

        {

            System.out.println("Given Number is an Armstrong number");

        }

        else

        {

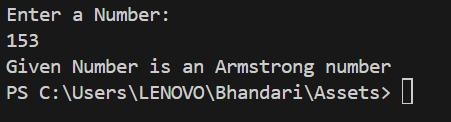
            System.out.println("Given Number is Not Armstrong number");

        }

    }

}

**Output:**



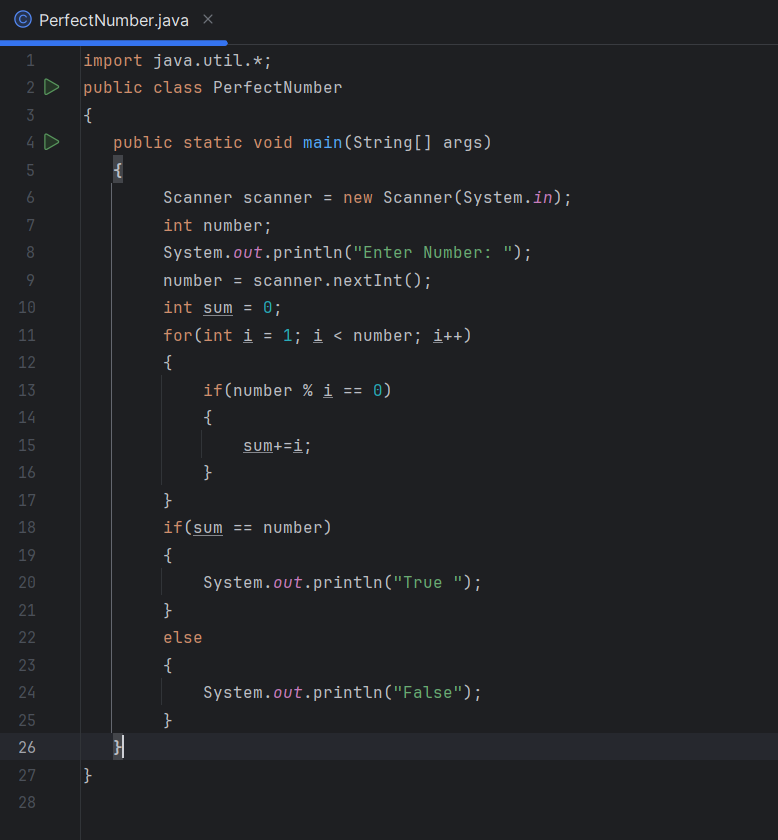
**Conclusion:**

Implemented a Java program to display armstrong numbers.

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_ Signature of faculty in-charge**

**Post Lab Descriptive Questions:**

Q.1 Write a program to find the perfect numbers between the range.



Q.2 Write a program to check whether the entered year is a leap year or not.

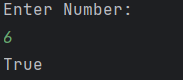


Q.3 Write a program to find gcd and lcm of two numbers(find gcd using recursive function).

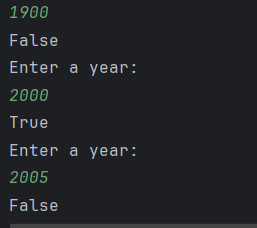


**Output:**

1)



2)



3)

