**Batch: C4 Roll No.: 16010123217**

**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:**

1. Start
2. Input an integer number from the user.
3. Initialize a counter to zero.
4. Use a loop to repeatedly divide the number by 10 until it becomes zero.
5. Increment the counter for each division.
6. Print the total number of digits
7. Initialize sum to zero and save the original number.
8. For each digit in the number:
   1. Extract the last digit using modulo operation (num % 10).
   2. Raise the digit to the power of the total number of digits using pow function from Math library
   3. Add the result to sum
   4. Remove the last digit by integer division (num /= 10).
9. After processing all digits, compare the sum with the original number.
10. If sum equals the original number, print "Yes, the given number is an Armstrong Number".
11. Otherwise, print "No, the given number is not an Armstrong Number".
12. End

**Implementation details:**

*//Version 1 basic version without any function*

import java.util.*\**;

class Armstrong{

    public static void main(String [] args){

        Scanner s1 = new Scanner (System.in);

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

        int num = s1.nextInt();

        int count = 0, last\_digit, sum =0, og\_num = num;

        while(num>0){

            count++;

            num /=10;

        }

        num = og\_num;

        System.out.println("Number of digits = "+count);

        for(int i = 0; i<count; i++){

            last\_digit = num%10;

            num /=10;

            sum += (int)Math.pow(last\_digit, count);

        }

        if (sum == og\_num) {

            System.out.println("Yes, the given number is an Armstrong Number");

        }

        else{

             System.out.println("No, the given number is not an Armstrong Number");

        }

    }

}

*//Version 2 Static method in same class*

import java.util.*\**;

class Armstrong{

    public static void main(String [] args){

        Scanner s1 = new Scanner (System.in);

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

        int num = s1.nextInt();

        int count = 0, og\_num = num;

        while(num>0){

            count++;

            num /=10;

        }

        System.out.println("Number of digits = "+count);

        if(checkArmstrong(og\_num, count)) {

            System.out.println("Yes, the given number is an Armstrong Number");

        }

        else{

            System.out.println("No, the given number is not an Armstrong Number");

        }

        s1.close();

    }

    static boolean checkArmstrong(int num, int count){

        int last\_digit = 0, sum = 0, og\_num = num;

        for(int i = 0; i<count; i++){

            last\_digit = num%10;

            num /=10;

            sum += (int)Math.pow(last\_digit, count);

        }

        if (sum == og\_num) {

            return true;

        }

        else{

             return false;

        }

    }

}

*//Version 3*

*// Creating 2 different classes*

import java.util.*\**;

class Armstrong{

    public static void main(String [] args){

        Scanner s1 = new Scanner (System.in);

        ArmstrongChecker checker = new ArmstrongChecker();

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

        int num = s1.nextInt();

        int count = 0, og\_num = num;

        while(num>0){

            count++;

            num /=10;

        }

        System.out.println("Number of digits = "+count);

        if(checker.checkArmstrong(og\_num, count)) {

            System.out.println("Yes, the given number is an Armstrong Number");

        }

        else{

            System.out.println("No, the given number is not an Armstrong Number");

        }

        s1.close();

    }

}

class ArmstrongChecker{

    boolean checkArmstrong(int num, int count){

        int last\_digit = 0, sum = 0, og\_num = num;

        for(int i = 0; i<count; i++){

            last\_digit = num%10;

            num /=10;

            sum += (int)Math.pow(last\_digit, count);

        }

        if (sum == og\_num) {

            return true;

        }

        else{

             return false;

        }

    }

}

*//Version 4*

*//Creating 2 different classes but creating not creating an object for other function to call in main function*

import java.util.*\**;

class Armstrong{

    public static void main(String [] args){

        Scanner s1 = new Scanner (System.in);

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

        int num = s1.nextInt();

        int count = 0, og\_num = num;

        while(num>0){

            count++;

            num /=10;

        }

        System.out.println("Number of digits = "+count);

        if(ArmstrongChecker.checkArmstrong(og\_num, count)) {

            System.out.println("Yes, the given number is an Armstrong Number");

        }

        else{

            System.out.println("No, the given number is not an Armstrong Number");

        }

        s1.close();

    }

}

class ArmstrongChecker{

    public static boolean checkArmstrong(int num, int count){

        int last\_digit = 0, sum = 0, og\_num = num;

        for(int i = 0; i<count; i++){

            last\_digit = num%10;

            num /=10;

            sum += (int)Math.pow(last\_digit, count);

        }

        if (sum == og\_num) {

            return true;

        }

        else{

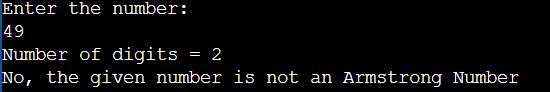
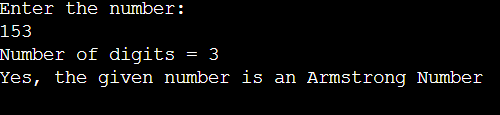
             return false;

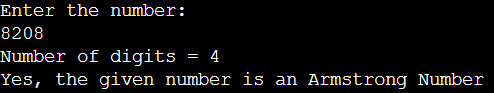
        }

    }

}

**Output:**





**Conclusion:** In this experiment, we explored the classes and objects in oops by performing same problem with various versions. Also we got familiar with JAVA by taking input using Scanner class and solving Armstrong problem with post lab questions.

**Date: 01/08/24 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).

Ans.

Q.1)

import java.util.*\**;

class PerfectNumbers {

  public static void main(String[] args) {

    System.out.println("Enter the starting number");

    System.out.println("Enter the ending number");

    Scanner s1 = new Scanner(System.in);

    int start = s1.nextInt();

    int end = s1.nextInt();

    for (int i = start; i <= end; i++) {

      PerfectNumber(i);

    }

    s1.close();

  }

  static void PerfectNumber(int num) {

    int sum = 0;

    for (int i = 1; i < num; i++) {

      if (num % i == 0) {

        sum += i;

      }

    }

    if (sum == num) {

      System.out.println(num);

    }

  }

}

Q.2)

import java.util.*\**;

class Leap{

  public static void main(String [] args){

    Scanner s1 = new Scanner (System.in);

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

    int year = s1.nextInt();

    if(checkLeap(year)){

      System.out.println("Yes, it is a leap year");

    }

    else{

      System.out.println("No, it is not a leap year");

    }

    s1.close();

  }

  static boolean checkLeap(int year){

    if(year%4 == 0){

      if(year%100 == 0){

        if(year%400 == 0){

          return true;

        }

        else{

          return false;

        }

      }

      else{

        return true;

      }

    }

    else{

      return false;

    }

  }

}

Q.3)

import java.util.*\**;

class GCDandLCM{

  public static void main(String [] args){

    Scanner s1 = new Scanner (System.in);

    System.out.println("Enter the first number:\t");

    int num1 = s1.nextInt();

    System.out.println("Enter the second number:\t");

    int num2 = s1.nextInt();

    int gcd = GCD(num1, num2);

    System.out.println("GCD = " + gcd);

    System.out.println("LCM = " + (num1 \* num2) / gcd);

    s1.close();

  }

  static int GCD(int num1, int num2){

    if (num2 != 0) {

      return GCD(num2, num1 % num2);

    } else {

      return num1;

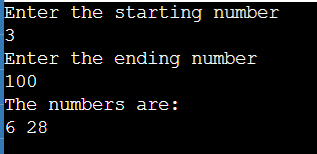
    }

  }

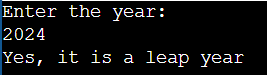
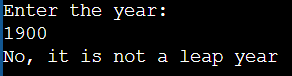
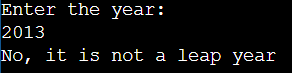
}

**Output:**

Q.1)



Q.2)



Q.3)

