**TITLE : Perfect Number**

**AIM:** Define a class Perfect which accepts the range of numbers from the user. Create a static function check\_per , which checks if the number is a perfect number or not and sends the result back to the main function which counts and displays the perfect numbers within that range.

**Batch:**

**Roll No.:**

**Experiment / assignment / tutorial No.01**

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

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

**16010122074**

**A3**

Variations :

Implementation of Program with One class

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

# Expected OUTCOME of Experiment:

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

# Books/ Journals/ Websites referred:

1. E. Balagurusamy , “Programming with Java” McGraw-Hill.
2. Sachin Malhotra, Saurabh Choudhary, “Programming in Java”, Oxford Publications.

# 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. |

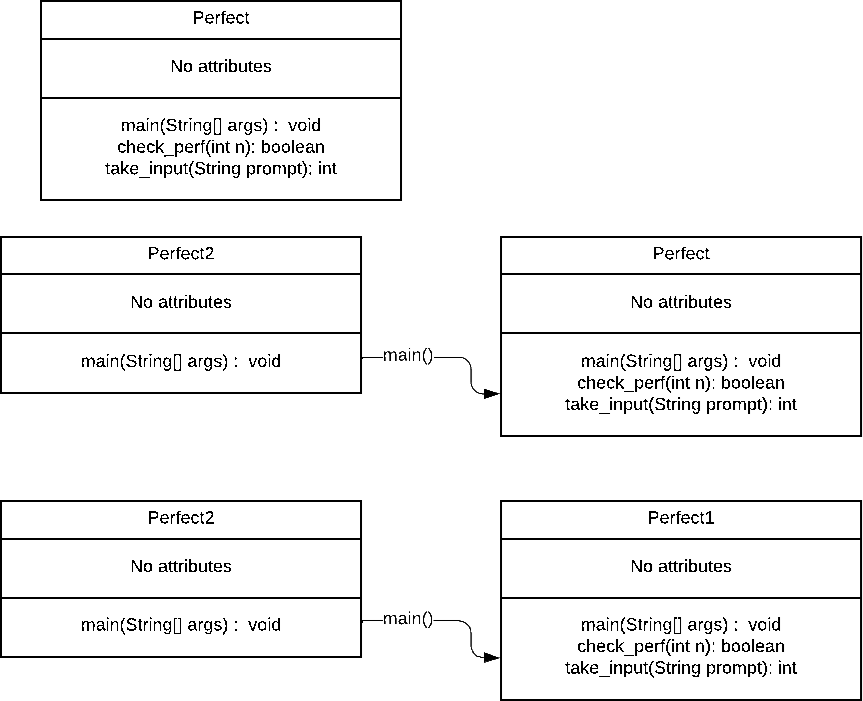
The Scanner looks for tokens in the input. A token is a series of characters that ends with what Java calls whitespace. A whitespace character can be a blank, a tab character, a carriage return. Thus, if we read a line that has a series of numbers separated by blanks, the scanner will take each number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

# Class Diagram:



**Algorithm:**

1. Display "This program will find all perfect numbers within a given range."
2. Initialize variables "lower," "upper," and "count."
3. Start an indefinite loop.
4. Prompt the user to enter the lower limit "lower" of the range.
5. Prompt the user to enter the upper limit "upper" of the range.
6. Set "count" to 0.
7. For each number "num" in the range [lower, upper]: a. Calculate the sum of divisors "sum" of "num." b. If "sum" is equal to "num," display "num" as a perfect number and increment "count" by 1.
8. Display "There are " + count + " perfect numbers in the given range."
9. Prompt the user with "Do you want to continue? (1 for yes, 2 for no)."
10. Take user input "choice."
11. If "choice" is 2, exit the loop and end the program.
12. If "choice" is not 1 or 2, display "Invalid choice. Exiting." and exit the loop.

# Implementation details:

import java.util.Scanner;

public class Perfect {

public static void main(String[] args) {

System.out.println("This program will print all perfect numbers in a given range.");

while (true) {

int lower = take\_input("Enter lower limit: "); int upper = take\_input("Enter upper limit: "); int count = 0;

for (int i = lower; i <= upper; i++) { if (is\_perf(i)) {

System.out.println(i); count++;

}

}

System.out.println("There are " + count + " perfect numbers in the given range.");

System.out.println("Do you want to continue? (1 for yes, 2 for no)"); Scanner sc = new Scanner(System.in);

int choice = take\_input("Enter your choice: "); if (choice == 2)

break;

if (choice != 1)

{

System.out.println("Invalid choice. Exiting.");

break;

}

}

}

public static int take\_input(String prompt) { Scanner sc = new Scanner(System.in); while (true) {

System.out.print(prompt); if (sc.hasNextInt()) {

int n = sc.nextInt(); if (n > 0)

return n;

}

System.out.println("Invalid input. Please enter a positive

integer.");

sc.nextLine();

}

}

public static boolean is\_perf(int n) { int sum = 0;

for (int i = 1; i < n; i++) { if (n % i == 0)

sum += i;

}

return sum == n;

}

}

**Variation 1:**

import java.util.Scanner; public class Perfect2 {

public static void main(String args[])

{

Perfect1 obj = new Perfect1(); obj.main();

}

}

public class Perfect1 { public void main() {

System.out.println("This program will print all perfect numbers in a

given range.");

while (true) {

int lower = take\_input("Enter lower limit: "); int upper = take\_input("Enter upper limit: "); int count = 0;

for (int i = lower; i <= upper; i++) { if (is\_perf(i)) {

System.out.println(i); count++;

}

}

System.out.println("There are " + count + " perfect numbers in the given range.");

System.out.println("Do you want to continue? (1 for yes, 2 for no)"); Scanner sc = new Scanner(System.in);

int choice = take\_input("Enter your choice: "); if (choice == 2)

break;

if (choice != 1)

{

System.out.println("Invalid choice. Exiting."); break;

}

}

}

public int take\_input(String prompt) { Scanner sc = new Scanner(System.in); while (true) {

System.out.print(prompt); if (sc.hasNextInt()) {

int n = sc.nextInt(); if (n > 0)

return n;

integer.");

}

}

}

System.out.println("Invalid input. Please enter a positive sc.nextLine();

public boolean is\_perf(int n) { int sum = 0;

for (int i = 1; i < n; i++) { if (n % i == 0)

sum += i;

}

return sum == n;

}

}

**Variation 3:**

import java.util.Scanner;

public class Perfect {

public static void main(String[] args) {

System.out.println("This program will print all perfect numbers in a given range.");

while (true) {

int lower = take\_input("Enter lower limit: "); int upper = take\_input("Enter upper limit: "); int count = 0;

for (int i = lower; i <= upper; i++) { if (is\_perf(i)) {

System.out.println(i); count++;

}

}

System.out.println("There are " + count + " perfect numbers in the given range.");

System.out.println("Do you want to continue? (1 for yes, 2 for no)"); Scanner sc = new Scanner(System.in);

int choice = take\_input("Enter your choice: "); if (choice == 2)

break;

if (choice != 1)

{

System.out.println("Invalid choice. Exiting."); break;

}

}

}

public static int take\_input(String prompt) { Scanner sc = new Scanner(System.in); while (true) {

System.out.print(prompt);

if (sc.hasNextInt()) { int n = sc.nextInt(); if (n > 0)

return n;

}

System.out.println("Invalid input. Please enter a positive

integer.");

sc.nextLine();

}

}

public static boolean is\_perf(int n) { int sum = 0;

for (int i = 1; i < n; i++) { if (n % i == 0)

sum += i;

}

return sum == n;

}

}

public class Perfect2 {

public static void main(String args[])

{

Perfect.main(args);

}

}

# Output:

**All produce the same result:**

This program will print all perfect numbers in a given range. Enter lower limit: 1

Enter upper limit: 10000 6

28

496

8128

There are 4 perfect numbers in the given range. Do you want to continue? (1 for yes, 2 for no) Enter your choice: 1

Enter lower limit: 10 Enter upper limit: 200 28

There are 1 perfect numbers in the given range. Do you want to continue? (1 for yes, 2 for no) Enter your choice: 2

# Conclusion:

We learned the basics of object oriented programming in Java, and also learned about the difference between static and non static objects and the different ways to call a function in out program.

# Date: Signature of faculty in-charge

**Post Lab Descriptive Questions:**

Q.1 Write a program to find the area and circumference of a circle using two classes.

import java.util.Scanner;

public class Circle {

public static void main(String[] args) { Scanner sc = new Scanner(System.in); System.out.print("Enter the radius: "); double radius = sc.nextDouble();

System.out.println("Area: " + Area.area(radius)); System.out.println("Circumference: " +

Circumference.circumference(radius)); sc.close();

}

}

class Area {

public static double area(double radius) {

return Math.round(Math.PI \* radius \* radius \* 10000.0) / 10000.0;

}

}

class Circumference {

public static double circumference(double radius) {

return Math.round(2 \* Math.PI \* radius \* 10000.0) / 10000.0;

}

}

Q2. Write the output of following program

public class BreakExample2 {

public static void main(String[] args) {

// outer loop

for (int i = 1; i <= 3; i++) {

// inner loop

for (int j = 1; j <= 3; j++) { if (i == 2 && j == 2) {

// using break statement inside the inner loop break;

}

System.out.println(i + " " + j);

}

}

}

}

**Output:**

**1 1**

**1 2**

**1 3**

**2 1**

**3 1**

**3 2**

**3 3**

Q3. Why is Java known as a platform independent language?

**Ans**. The java language is known as a platform independent language because any code which is written in java is converted to an intermediate form known as byte code which is independent of the machine on which the program is run or compiled on. It is the same for all platforms. Whenever the program is to be executed, the byte code is converted to machine code by the Java virtual machine (JVM) which converts the java compiled byte code into machine code which is different for each machine.

Q4. Write a recursive static method for calculation of gcd of a number.

public static void gcd(int a, int b) { if (b == 0) {

System.out.println("GCD is " + a);

} else {

gcd(b, a % b);

}

}