1. **Write a program that prints the numbers from 1 to 100. But for multiples of three print “Fizz” instead of the number and for the multiples of five print “Buzz”. For numbers which are multiples of both three and five print “FizzBuzz”.**

**Ans:**

import java.util.stream.IntStream;

public class FizzBuzz

{

public static void main(String[] args)

{

fizzBuzzInJava8(100);

}

private static void fizzBuzzBeforeJava8(int num)

{

for (int i = 1; i <= num; i++)

{

if (((i % 3) == 0) && ((i % 5) == 0))

System.out.println("fizzbuzz");

else if ((i % 3) == 0)

System.out.println("fizz");

else if ((i % 5) == 0)

System.out.println("buzz");

else

System.out.println(i);

}

}

private static void fizzBuzzInJava8(int num) {

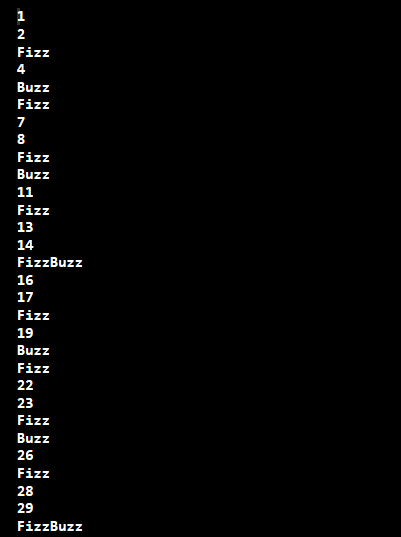
IntStream.rangeClosed(1, 100)

.mapToObj(i -> i % 3 == 0 ? (i % 5 == 0 ? "FizzBuzz" : "Fizz") : (i % 5 == 0 ? "Buzz" : i))

.forEach(System.out::println);

}

}



1. **Write a program that determine whether or not an integer input is a leap year.**

* **Definition of leap year:**
  + **Rule 1: A year is called leap year if it is divisible by 400.**

**Example: 1600, 2000 etc. are leap year while 1500, 1700 are not leap year.**

* + **Rule 2: If year is not divisible by 400 as well as 100 but it is divisible by 4 then that year are also leap year.**

**Example: 2004, 2008, 1012 are leap year.**

**Ans:**

public class LeapYearCalculator

{

public static void main(String args[])

{

int year = 2010;

boolean isLeapYear = false;

if(year % 400 == 0)

{

isLeapYear = true;

}

else if (year % 100 == 0)

{

isLeapYear = false;

}

else if(year % 4 == 0)

{

isLeapYear = true;

}

else

{

isLeapYear = false;

}

if(isLeapYear)

{

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

}

else

{

System.out.println("Year "+year+" is not a Leap Year");

}

}

}



1. **Write a program that produce the following output giving an integer input n.**
   1. **Ans:**

import java.io.\*;

public class Pattern

{

public static void printStars(int n)

{

int i, j;

for(i=0; i<n; i++)

{

for(j=0; j<=i; j++)

{

System.out.print("\* ");

}

System.out.println();

}

}

public static void main(String args[])

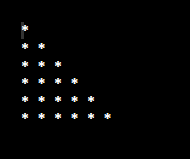
{

int n = 6;

printStars(n);

}

}

****

* 1. **Ans:**

import java.io.\*;

public class Pattern

{

public static void printStars(int n)

{

int i, j;

for(i=0; i<n; i++)

{

for(j=2\*(n-i); j>=0; j--)

{

System.out.print(" ");

}

for(j=0; j<=i; j++)

{

System.out.print("\* ");

}

System.out.println();

}

}

public static void main(String args[])

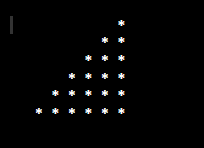
{

int n = 6;

printStars(n);

}

}



* 1. **Ans:**

import java.io.\*;

public class Pattern

{

public static void printTriagle(int n)

{

for (int i=0; i<n; i++)

{

for (int j=n-i; j>1; j--)

{

System.out.print(" ");

}

for (int j=0; j<=i; j++ )

{

System.out.print("\* ");

}

System.out.println();

}

}

public static void main(String args[])

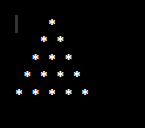
{

int n = 5;

printTriagle(n);

}

}



* 1. **Ans:**

public class Pattern {

   public static void main(String[] args) {

   int n = 5;

   int i,j;

   for(i = 1; i <= n; i++)

   {

      for(j = n-1; j >= i; j--)

      {

      System.out.print(" ");

      }

      System.out.print("\*");

      //inner loop

      for(j = 1; j < (i-1) \* 2; j++)

      {

         System.out.print(" ");

       }

      if(i > 1)

      {

         System.out.print("\*");

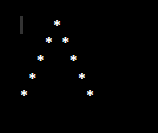
      }

      System.out.print("\n");

      }

   }

}



* 1. **Ans:**

public class Pattern {

public static void main(String[] args) {

int n =5;

for(int i = 1; i <= n; i++)

{

for(int j = 1; j <= n; j++)

{

if(i == j || i+j == n+1)

{

System.out.print(" \*");//1 space

}else

{

System.out.print(" ");//2 spaces

}

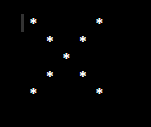
}

System.out.println();

}

}

}



1. **(Python specific) In Python, what is the difference between else and finally in exception handling?**

**Ans:**

In exception handling else statements blocks will be executed when there is no exception occurs.

Finally statements will be execute statements in all the cases whether it is exception or non-exception.

else:

If there is no exception then executes the block

Finally:

This will executes all the statements (exception or no exception)

1. **Write a program that finds all prime numbers up to n for input n.**

**Ans:**

public class GFG

{

static boolean isPrime(int n)

{

if (n <= 1)

return false;

for (int i = 2; i < n; i++)

if (n % i == 0)

return false;

return true;

}

static void printPrime(int n)

{

for (int i = 2; i <= n; i++)

{

if (isPrime(i))

System.out.print(i + " ");

}

}

public static void main(String[] args)

{

int n = 20;

printPrime(n);

}

}

