List of Java Programs

1. **Program1** **–** **WAP** **to** **display** **the** **List** **of** **even** **numbers**
2. **Program2** **-** **Factorial** **of** **a** **number**
3. **Program3** **-** **Compare** **Two** **Numbers** **using** **else-if**
4. **Program4** **-** **Determine** **If** **Year** **Is** **Leap** **Year**
5. **Program5** **-** **Fibonacci** **Series**
6. **Program6** **-** **Palindrome** **Number**
7. **Program7-** **Generate** **prime** **numbers** **between** **1** **&** **given** **number**
8. **Program8-** **Pyramid** **of** **stars** **using** **nested** **for** **loops**
9. **Program9** **-** **Reversed** **pyramid** **using** **for** **loops** **&** **decrement** **operator.**
10. **Program10** **-** **Nested** **Switch**
11. **Program11** **-** **Calculate** **Circle** **Area** **using** **radius**
12. **Program12** **-** **Factorial** **of** **a** **number** **using** **recursion**
13. **Program13** **-** **Pyramid** **of** **numbers** **using** **for** **loops**
14. **Program14** **-** **To** **Find** **Maximum** **of** **Two** **Numbers.**
15. **Program15** **-** **To** **Find** **Minimum** **of** **Two** **Numbers** **using** **conditional** **operator**
16. **Program** **16** **-** **Write** **a** **program** **that** **will** **read** **a** **float** **type** **value** **from** **the** **keyboard** **and** **print** **the** **following** **output.**

**->Small** **Integer** **not** **less** **than** **the** **number.**

**->Given** **Number.**

**->Largest** **Integer** **not** **greater** **than** **the** **number.**

1. **Program** **17** **-** **Write** **a** **program** **to** **generate** **5** **Random** **nos.** **between** **1** **to** **100,** **and** **it** **should** **not** **follow** **with** **decimal** **point.**
2. **Program** **18** **-** **Write** **a** **program** **to** **display** **a** **greet** **message** **according** **to** **Marks** **obtained** **by** **student**
3. **Program** **19** - **Write** **a** **program** **to** **find** **SUM** **AND** **PRODUCT** **of** **a** **given** **Digit.**
4. **Program** **20** - **Write** **a** **program** **to** **find** **sum** **of** **all** **integers** **greater** **than** **100** **and** **less** **than** **200** **that** **are** **divisible** **by** **7**

**Program1** **–** **List** **of** **even** **numbers**

public class ListEvenNumbers {

public static void main(String[] args) {

//define limit int limit = 50;

System.out.println("Printing Even numbers between 1 and " +

limit);

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

**//** **if** **the** **number** **is** **divisible** **by** **2** **then** **it** **is** **even** **if(** **i** **%** **2** **==** **0){**

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

**}**

**}**

**}**

**}**

**/\***

Output of List Even Numbers Java Example would be Printing Even numbers between 1 and 50

**2** **4** **6** **8** **10** **12** **14** **16** **18** **20** **22** **24** **26** **28** **30** **32** **34** **36** **38** **40** **42** **44** **46** **48** **50**

**\*/**

**Program2** **-** **Factorial** **of** **a** **number**

public class NumberFactorial {

public static void main(String[] args) { int number = 5;

**/\***

* **Factorial** **of** **any** **number** **is!** **n.**
* **For** **example,** **factorial** **of** **4** **is** **4\*3\*2\*1.**

**\*/**

int factorial = number;

**for(int** **i** **=(number** **-** **1);** **i** **>** **1;** **i--)**

**{**

factorial = factorial \* i;

**}**

System.out.println("Factorial of a number is " + factorial);

**}**

**}**

**/\***

Output of the Factorial program would be Factorial of a number is 120

**\*/**

**Program3** **-** **Compare** **Two** **Numbers** **using** **else-if**

public class CompareTwoNumbers {

public static void main(String[] args) {

//declare two numbers to compare int num1 = 324;

int num2 = 234;

if(num1 > num2){

System.out.println(num1 + " is greater than " + num2);

**}**

else if(num1 < num2){

System.out.println(num1 + " is less than " + num2);

**}**

else{

System.out.println(num1 + " is equal to " + num2);

**}**

**}**

**}**

**/\***

Output of Compare Two Numbers Java Example would be

324 is greater than 234

**\*/**

**Program4** **-** **Determine** **If** **Year** **Is** **Leap** **Year**

public class DetermineLeapYearExample {

public static void main(String[] args) {

//year we want to check int year = 2004;

**//if** **year** **is** **divisible** **by** **4,** **it** **is** **a** **leap** **year**

**if(year** **%** **400** **==** **0)** **||** **((year** **%** **4** **==** **0)** **&&** **(year** **%** **100** **!=** **0))** **System.out.println("Year** **"** **+** **year** **+** **"** **is** **a** **leap** **year");**

else

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

**}**

**}**

**/\***

Output of the example would be Year 2004 is a leap year

**\*/**

**Program5** **-** **Fibonacci** **Series**

public class JavaFibonacciSeriesExample { public static void main(String[] args) {

//number of elements to generate in a series int limit = 20;

long[] series = new long[limit];

//create first 2 series elements

series[0] = 0;

series[1] = 1;

//create the Fibonacci series and store it in an array for(int i=2; i < limit; i++){

series[i] = series[i-1] + series[i-2];

**}**

//print the Fibonacci series numbers

System.out.println("Fibonacci Series upto " + limit); for(int i=0; i< limit; i++){

System.out.print(series[i] + " ");

**}**

**}**

**}**

**/\***

Output of the Fibonacci Series Java Example would be Fibonacci Series upto 20

**0** **1** **1** **2** **3** **5** **8** **13** **21** **34** **55** **89** **144** **233** **377** **610** **987** **1597** **2584** **4181**

**\*/**

**Program6** **-** **Palindrome** **Number**

public class JavaPalindromeNumberExample { public static void main(String[] args) {

//array of numbers to be checked

int numbers[] = new int[]{121,13,34,11,22,54};

//iterate through the numbers

for(int i=0; i < numbers.length; i++){

int number = numbers[i]; int reversedNumber = 0; int temp=0;

**/\***

* **If** **the** **number** **is** **equal** **to** **it's** **reversed** **number,** **then**
* **the** **given** **number** **is** **a** **palindrome** **number.**

**\***

* **For** **ex,121** **is** **a** **palindrome** **number** **while** **12** **is** **not.**

**\*/**

//reverse the number

while(number > 0){ temp = number % 10; number = number / 10;

reversedNumber = reversedNumber \* 10 + temp;

**}**

if(numbers[i] == reversedNumber) System.out.println(numbers[i] + " is a palindrome"); else

System.out.println(numbers[i] + " not a palindrome ");

**}**

**}**

**}**

**/\***

Output of Java Palindrome Number Example would be

121 is a palindrome number

13 is not a palindrome number

34 is not a palindrome number

11 is a palindrome number

22 is a palindrome number

54 is not a palindrome number

**\*/**

**Program7-** **Generate** **prime** **numbers** **between** **1** **&** **given** **number**

public class GeneratePrimeNumbersExample { public static void main(String[] args) {

//define limit int limit = 100;

System.out.println("Prime numbers between 1 and " + limit);

//loop through the numbers one by one for(int i=1; i < 100; i++){

boolean isPrime = true;

**//check** **to** **see** **if** **the** **number** **is** **prime** **for(int** **j=2;** **j** **<** **i** **;** **j++){**

**if(i** **%** **j** **==** **0){**

isPrime = false; break;

**}**

**}**

// print the number if(isPrime)

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

**}**

**}**

**}**

**/\***

Output of Prime Numbers example would be Prime numbers between 1 and 100

**1** **2** **3** **5** **7** **11** **13** **17** **19** **23** **29** **31** **37** **41** **43** **47** **53** **59** **61** **67** **71** **73** **79** **83** **89** **97**

**\*/**

**Program8-** **Pyramid** **of** **stars** **using** **nested** **for** **loops**

**/\***

Java Pyramid 1 Example

This Java Pyramid example shows how to generate pyramid or triangle like given below using for loop.

**\***

**\*\***

**\*\*\***

**\*\*\*\***

**\*\*\*\*\***

**\*/**

public class JavaPyramid1 {

public static void main(String[] args) { for(int i=1; i<= 5 ;i++){

for(int j=0; j < i; j++){ System.out.print("\*");

**}**

//generate a new line System.out.println("");

**}**

**}**

**}**

**/\***

Output of the above program would be

**\***

**\*\***

**\*\*\***

**\*\*\*\***

**\*\*\*\*\***

**\*/**

**Program9** **–** **Reversed** **pyramid** **using** **for** **loops** **&** **decrement** **operator.**

**/\***

Java Pyramid 5 Example

This Java Pyramid example shows how to generate pyramid or triangle like given below using for loop.

12345

1234

123

12

1

**\*/**

public class JavaPyramid5 {

public static void main(String[] args) { for(int i=5; i>0 ;i--){

for(int j=0; j < i; j++){ System.out.print(j+1);

**}**

System.out.println("");

**}**

**}**

**}**

**/\***

Output of the example would be 12345

1234

123

12

1

**\*/**

**Program10** **-** **Nested** **Switch**

**/\***

Statements Example

This example shows how to use nested switch statements in a java program.

**\*/**

public class NestedSwitchExample {

public static void main(String[] args) {

**/\***

* **Like** **any** **other** **Java** **statements,** **switch** **statements**
* **can** **also** **be** **nested** **in** **each** **other** **as** **given** **in**
* **below** **example.**

**\*/**

**int** **i** **=** **0;** **int** **j** **=** **1;**

switch(i)

**{**

case 0S

switch(j)

**{**

case 0S

System.out.println("i is 0, j is 0"); break;

case 1S

System.out.println("i is 0, j is 1"); break;

defaultS

System.out.println("nested default case!!");

**}**

break;

defaultS

System.out.println("No matching case found!!");

**}**

**}**

**}**

**/\***

**Output** **would** **be,** **i** **is** **0,** **j** **is** **1**

**\*/**

**Program11** **-** **Calculate** **Circle** **Area** **using** **radius**

**/\***

This program shows how to calculate area of circle using it's radius.

**\*/**

import java.io.BufferedReader; import java.io.IOException; import java.io.InputStreamReader;

public class CalculateCircleAreaExample { public static void main(String[] args) {

int radius = 0;

System.out.println("Please enter radius of a circle");

try

**{**

//get the radius from console BufferedReader br = new BufferedReader(new

InputStreamReader(System.in));

radius = Integer.parseInt(br.readLine());

**}**

//if invalid value was entered

catch(NumberFormatException ne)

**{**

System.out.println("Invalid radius value" + ne); System.exit(0);

**}**

catch(IOException ioe)

**{**

System.out.println("IO Error S" + ioe); System.exit(0);

**}**

**/\***

* **Area** **of** **a** **circle** **is**
* **pi** **\*** **r** **\*** **r**
* **where** **r** **is** **a** **radius** **of** **a** **circle.**

**\*/**

//NOTE S use Math.PI constant to get value of pi double area = Math.PI \* radius \* radius;

System.out.println("Area of a circle is " + area);

**}**

**}**

**/\***

Output of Calculate Circle Area using Java Example would be

Please enter radius of a circle 19

Area of a circle is 1134.1149479459152

**\*/**

**Program12** **-** **Factorial** **of** **a** **number** **using** **recursion**

**/\***

This program shows how to calculate

Factorial of a number using recursion function.

**\*/**

import java.io.BufferedReader; import java.io.IOException; import java.io.InputStreamReader;

public class JavaFactorialUsingRecursion {

public static void main(String args[]) throws NumberFormatException, IOException{

System.out.println("Enter the numberS ");

//get input from the user

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

int a = Integer.parseInt(br.readLine());

//call the recursive function to generate factorial int result= fact(a);

System.out.println("Factorial of the number isS " + result);

**}**

static int fact(int b)

**{**

if(b <= 1)

**//if** **the** **number** **is** **1** **then** **return** **1** **return** **1;**

else

**}**

**}**

//else call the same function with the value - 1 return b \* fact(b-1);

**/\***

Output of this Java example would be

Enter the numberS 5

Factorial of the number isS 120

**\*/**

**Program13** **–** **pyramid** **of** **numbers** **using** **for** **loops**

**/\***

Generate Pyramid For a Given Number Example

This Java example shows how to generate a pyramid of numbers for given number using for loop example.

**\*/**

import java.io.BufferedReader; import java.io.InputStreamReader;

public class GeneratePyramidExample {

public static void main (String[] args) throws Exception{

BufferedReader keyboard = new BufferedReader (new InputStreamReader(System.in));

System.out.println("Enter NumberS");

int as= Integer.parseInt (keyboard.readLine()); System.out.println("Enter XS");

int x= Integer.parseInt (keyboard.readLine());

**int** **y** **=** **0;**

for(int i=0; i<= as ;i++){

**for(int** **j=1;** **j** **<=** **i** **;** **j++){** **System.out.print(y** **+** **"\t");** **y** **=** **y** **+** **x;**

**}**

System.out.println("");

**}**

**}**

**}**

**/\***

Output of this example would be

Enter NumberS 5

Enter XS 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **0** |  |  |  | |
| **1** | **2** |  |
| **3** | **4** | **5** |
| **6** | **7** | **8** | **9** |  |
| **10** | **11** | **12** | **13** | **14** |

Enter NumberS 5

Enter XS 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **0** |  |  |  |  |
| **2** | **4** |  |  |  |
| **6** | **8** | **10** |  |  |
| **12** | **14** | **16** | **18** |  |
| **20** | **22** | **24** | **26** | **28** |

Enter NumberS 5

Enter XS 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **0** |  |  |  | |
| **3** | **6** |  |
| **9** | **12** | **15** |
| **18** | **21** | **24** | **27** |  |
| **30** | **33** | **36** | **39** | **42** |
| **\*/** |  |  |  |  |

**Program14** **–** **To** **Find** **Maximum** **of** **Two** **Numbers.**

**/\***

To Find Maximum of 2 Numbers using if else

**\*/**

class Maxoftwo{

public static void main(String args[]){

//taking value as command line argument.

//Converting String format to Integer value int i = Integer.parseInt(args[0]);

int j = Integer.parseInt(args[1]); if(i > j)

System.out.println(i+" is greater than "+j);

else

System.out.println(j+" is greater than "+i);

**}**

**}**

**Program15** **–** **To** **Find** **Minimum** **of** **Two** **Numbers** **using** **conditional** **operator**

**/\***

To find minimum of 2 Numbers using ternary operator

**\*/**

class Minoftwo{

public static void main(String args[]){

//taking value as command line argument.

//Converting String format to Integer value int i = Integer.parseInt(args[0]);

int j = Integer.parseInt(args[1]); int result = (i<j)?iSj;

System.out.println(result+" is a minimum value");

**}**

**}**

**Program** **16**

/\* Write a program that will read a float type value from the keyboard and print the following output.

->Small Integer not less than the number.

->Given Number.

->Largest Integer not greater than the number.

**\*/**

class ValueFormat{

public static void mainoString args[]){ double i = 34.32; //given number

System.out.printlno"Small Integer not greater than the number : "+Math.ceiloi));

System.out.printlno"Given Number : "+i); System.out.printlno"Largest Integer not greater than the number :

"+Math.flooroi));

}

**Program** **17** **-** **Write** **a** **program** **to** **generate** **5** **Random** **nos.** **between** **1** **to** **100,** **and** **it** **should** **not** **follow** **with** **decimal** **point.**

class RandomDemo{

public static void mainoString args[]){ foroint i=1;i<=5;i++){

System.out.printlnooint)oMath.randomo)\*100));

}

}

}

**Program** **18** - **Write** **a** **program** **to** **display** **a** **greet** **message** **according** **to** **Marks** **obtained** **by** **student.**

class SwitchDemo{

public static void mainoString args[]){

int marks = Integer.parseIntoargs[0]); //take marks as command line argument.

switchomarks/10){ case 10:

case 9:

case 8:

System.out.printlno"Excellent"); break;

case 7:

System.out.printlno"Very Good"); break;

case 6:

System.out.printlno"Good"); break;

case 5:

System.out.printlno"Work Hard"); break;

case 4:

System.out.printlno"Poor"); break;

case 3:

case 2:

case 1:

case 0:

System.out.printlno"Very Poor"); break;

default:

System.out.printlno"Invalid value Entered");

}

}

}

**Program** **19** - **Write** **a** **program** **to** **find** **SUM** **AND** **PRODUCT** **of** **a** **given** **Digit.**

class Sum\_Product\_ofDigit{

public static void mainoString args[]){ int num = Integer.parseIntoargs[0]);

//taking value as command line argument. int temp = num,result=0;

//Logic for sum of digit whileotemp>0){

result = result + temp; temp--;

}

System.out.printlno"Sum of Digit for "+num+" is : "+result);

//Logic for product of digit temp = num;

result = 1;

whileotemp > 0){

result = result \* temp; temp--;

}

System.out.printlno"Product of Digit for "+num+" is : "+result);

}

}

**Program** **20** - **Write** **a** **program** **to** **find** **sum** **of** **all** **integers** **greater** **than** **100** **and** **less** **than** **200** **that** **are** **divisible** **by** **7**

class SumOfDigit{

public static void mainoString args[]){ int result=0;

foroint i=100;i<=200;i++){ ifoi%7==0)

result+=i;

}

System.out.printlno"Output of Program is : "+result);

}

}