1.Java Program to find the grade of a student using switch case. Here we will write a Java program find out students grades using switch case statement. The below table shows the grading system.

|  |  |
| --- | --- |
| **Score in subject** | **Grade** |
| >=90 | A |
| 80-89 | B |
| 70-79 | C |
| 60-69 | D |
| 50-59 | E |
| <50 | F |

package com.torryharris

import java.util.Scanner;

public class Student { public static String findGrade(int score)

{

if(score < 0 || score > 100)

return "Invalid score";

switch(score/10) {

case 10:

case 9: return "A";

case 8: return "B";

case 7: return "C";

case 6: return "D";

case 5: return "E";

default: return "F";

}

}

public static void main(String[] args)

{

// Create Scanner class object to get input value

Scanner scan = new Scanner(System.in);

System.out.print("Enter score value: ");

int score = scan.nextInt();

System.out.println("Grade = " + findGrade(score));

scan.close();

}

}

2.Pascal Triangle in Java | Pascal triangle is a triangular array of binomial coefficients. In pascal’s triangle, each number is the sum of the two numbers directly above it.

     1

    1 1

   1 2 1

  1 3 3 1

 1 4 6 4 1

package com.torryharris

import java.util.Scanner;

class GFG {

static void printPascal(int n)

{

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

{

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

System.out.print(binomialCoeff(line, i)+" ");

System.out.println();

}

}

static int binomialCoeff(int n, int k)

{

int res = 1;

if (k > n - k)

k = n - k;

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

{

res \*= (n - i);

res /= (i + 1);

}

return res;

}

public static void main(String args[])

{

int n = 7;

printPascal(n);

}

}

Output:-

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

**3.A number is said to be a magic number if the sum of its digits is calculated till a single digit recursively by adding the sum of the digits after every addition. If the single-digit comes out to be 1, then the number is a magic number.** Example of magic numbers are:- 10, 19, 28, 55, 1234 e.t.c.

package com.util.torryharris;

import java.util.scanner;

class GFG

{

public static boolean isMagic(int n)

{

int sum = 0;

while (n > 0 || sum > 9)

{

if (n == 0)

{

n = sum;

sum = 0;

}

sum += n % 10;

n /= 10;

}

return (sum == 1);

}

public static void main(String args[])

{

int n = 1234;

if (isMagic(n))

System.out.println("Magic Number");

else

System.out.println("Not a magic Number");

}

}

4.) The highest common factor (HCF) of two or more numbers is the greatest number which divides each of them exactly. Greatest Common Measure(GCM) and Greatest Common Divisor(GCD) are the other terms used to refer to HCF.

package com.th;

import java.util.Scanner;

public class Hcf {

public static void main(String args[]){

int hcf = 0;

Scanner sc = new Scanner(System.in);

System.out.println("Enter first number = ");

int a = sc.nextInt();

System.out.println("Enter second number = ");

int b = sc.nextInt();

for(int i = 1; i <= a || i <= b; i++)

{

if( a%i == 0 && b%i == 0 )

hcf = i;

}

System.out.println("HCF of given two numbers is = "+hcf);

}

}

output1:

Enter first number =

625

Enter second number =

125

HCF of given two numbers is = 125

output2:

Enter first number =

27

Enter second number =

678

HCF of given two numbers is = 3

**5.) Palindrome number or not**

package com.th;

import java.util.Scanner;

public class Palindrome {

public static void main(String[] args) {

int remainder,reverse=0,temp;

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

temp=n;

while(n>0){

remainder=n%10; //getting remainder

reverse=(reverse\*10)+remainder;

n=n/10;

}

if(temp == reverse)

System.out.println("palindrome number ");

else

System.out.println("not palindrome");

}

}

output1:

12

not palindrome

output2:

121

palindrome number

output3:

333

palindrome number

**6.)Krishnamurthy number or not**

package com.th;

import java.util.Scanner;

class KrishnamurthyNumber

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

if (isKrishnamurthy(n))

System.out.println(n+" is a krishnamurthy number");

else

System.out.println(n+" not a krishnamurthy number");

}

static int factorial(int n)

{

int fact = 1;

while (n != 0)

{

fact = fact \* n;

n--;

}

return fact;

}

static boolean isKrishnamurthy(int n)

{

int sum = 0;

int temp = n;

while (temp != 0)

{

sum += factorial(temp % 10);

temp = temp / 10;

}

return (sum == n);

}

}

output1:

12

12 not a krishnamurthy number

output2:

145

145 is a krishnamurthy number

output3:

125

125 not a krishnamurthy number

7.**Java program to find the sum of even digits in a given number.**

package torryharris.com.main.pack;

import java.util.Scanner;

public class Main{

public void getEvenDigitSum(int number)

{

int originalNumber = number;

int digit, sum = 0;

while(number != 0)

{

digit = number % 10;

if(digit % 2 == 0)

sum = sum + digit;

number = number / 10;

}

System.out.println("Sum of even digits of number "+originalNumber+ " is: "+sum);

}

public static void main(String args[])

{

Main eds = new Main();

eds.getEvenDigitSum(12345);

eds.getEvenDigitSum(22222);

eds.getEvenDigitSum(13579);

}

}

**8.A number whose sum of its digits powered with their respective position is equal to the original number is called disarium number.** Examples of disarium numbers are- 135, 175, 518 and e.tc.

Package torryharris.com.main.pack;

import java.util.Scanner;

public class Main{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

System.out.print("Input a number:");

int num=sc.nextInt();

int copy=num,d=0,sum=0;

Strings=Integer.toString(num);

int len=s.length();

while(copy>0)

{

d=copy%10;

sum=sum+(int)Math.pow(d,len);

len--;

copy=copy/10;

}

if(sum==num)

System.out.println("Disarium Number.");

else

System.out.println("Not a DisariumNumber.");

}

}

9.Fibonacci series in Java | In the Fibonacci series, the next element will be the sum of the previous two elements. The Fibonacci sequence is a series of numbers where a number is found by adding up the two numbers before it. Starting with 0 and 1, the sequence goes 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, and so on…

package torryharris.com.main.pack;

import java.util.Scanner;

public class Main{

public static void main(String args[])

{

int n1=0,n2=1,n3,i,count=10;

System.out.print(n1+" "+n2);//printing0and1

for(i=2;i<count;i++)

{

n3=n1+n2;

System.out.print(""+n3);

n1=n2;

n2=n3;

}

}

}

10. Conversion from Celsius to Fahrenheit and from Fahrenheit to Celsius has an important role in the conversion of units system. In this post, we will develop a Java program to convert Celsius to Fahrenheit.

The formula used to convert from Celsius to Fahrenheit is given as, ⁰F= (⁰C \* 9/5) + 32

package com.torry.mainpack;

public class Main{

public static void main(String[] args){

float fehrenheit,celcius;

celsius=13;

fahrenheit=((celsius\*9)/5)+32;

System.out.println(“Temperature in Fahrenheit is:”+ fahrenheit);

}

}

**11.Write a Java program to find the largest number in the array.**

Package com.torry.mainpack;

public class Main{

static int array[]={78,56,98,34,167};

static int largest()

{

int i;

int max=array[0];

for(i=1;i<array.length;i++)

if(array[i]>max)

max=array[i];

return max;

}

public static void main(String[] args) {

System.out.println(“Largest in given array is: ”+largest());

}

}

**12.Write a Java program to merge two sorted arrays.**

package com.torry;  
  
public class Main {  
  
public static void main (String[] args) {  
  
int[] arr1 = {11, 34, 66, 75};  
  
int n1 = arr1.length;  
  
int[] arr2 = {1, 5, 19, 50, 89, 100};  
  
int n2 = arr2.length;  
  
int[] merge = new int[n1 + n2];  
  
int i = 0, j = 0, k = 0, x;  
  
System.out.print("Array 1: ");  
  
for (x = 0; x < n1; x++)  
  
System.out.print(arr1[x] + " ");  
  
System.out.print("\nArray 2: ");  
  
for (x = 0; x < n2; x++)  
  
System.out.print(arr2[x] + " ");  
  
while (i < n1 && j < n2) {  
  
if (arr1[i] < arr2[j])  
  
merge[k++] = arr1[i++];  
  
else  
  
merge[k++] = arr2[j++];  
  
}  
  
while (i < n1)  
  
merge[k++] = arr1[i++];  
  
while (j < n2)  
  
merge[k++] = arr2[j++];  
  
System.out.print("\nArray after merging: ");  
  
for (x = 0; x < n1 + n2; x++)  
  
System.out.print(merge[x] + " ");  
  
}  
  
}

**13.)** **How to count repeated elements in an array in Java programming language. If the array is sorted then counting repeated elements in an array will be easy compare to the unsorted array**.

package com.company;

public class Main {

public static void main(String[] args) {

int [] arr =new int[] {10,20,10,30,40,30,50};

int count=0;

System.out.println("duplicate elements in array are ");

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

{

for(int j=i+1;j<arr.length;j++)

{

if(arr[i]==arr[j])

{

count++;

System.out.println(arr[j]);

System.out.println("total duplicate element are "+count);

}

}

}

}

}

duplicate elements in array are

10

total duplicate element are 1

30

total duplicate element are 2

**14.)****An array can contain duplicate elements, and the array may be sorted or unsorted. Let us see different ways to remove duplicates from a given array in Java programming language.**

**Sorted array = {10, 10, 20, 30, 40, 40, 50};  
After removing the duplicates from array = {10, 20, 30, 40, 50};  
Unsorted Array = {30, 50, 20, 50, 10, 20, 30, 10, 10, 40};  
After removing the duplicates from array = {30, 50, 20, 10, 40};**

package com.company;

import java.util.Arrays;

public class Main {

public static int removeDuplicate(int arr[], int n) {

if (n == 0 || n == 1) {

return n;

}

int[] temp = new int[n];

int j = 0;

for( int i = 0; i<n-1;i++)

{

if (arr[i] != arr[i + 1]) {

temp[j++] = arr[i];

}

}

temp[j++]=arr[n-1];

//change in original arr

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

{

arr[i] = temp[i];

}

return j;

}

public static void main(String[] args) {

int [] arr =new int[] {10,20,10,30,40,30,50};

Arrays.sort(arr);

int length = arr.length;

length=removeDuplicate(arr,length);

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

{

System.out.println(arr[i]+" ");

}

}

}

**Output:**

10

20

30

40

50

**15.)****How to Insert an element at a specific position in an Array in Java? Here we will see examples to insert an element to the array at index. To insert an element at the end see:- Add element in array Java.**

**We can insert an element at a specific position in two different ways,  
a) Using a new array  
b) With the help of ArrayList.**

**Example:-  
Array = {10, 20, 30, 40, 50};  
Element to be inserted = 99;  
Position = 3;  
Final result = {10, 20, 30, 99, 40, 50};**

package com.company;

import java.util.Arrays;

public class Arrele {

public static int[] addElement(int [] arr,int element)

{

int temp[] = new int[arr.length+1];

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

{

temp[i]=arr[i];

}

temp[arr.length]=element;

return temp;

}

public static void main(String[]args)

{

int arr[] = {30,50,30,40,10};

int element=99;

System.out.println("original array "+ Arrays.toString(arr));

arr = addElement(arr,element);

System.out.println("new array "+Arrays.toString(arr));

}

}

original array [30, 50, 30, 40, 10]

new array [30, 50, 30, 40, 10, 99]

16.we will develop the Armstrong number program in Java. First, we will develop a java program to check an Armstrong number, and then we will develop a java program for an Armstrong number between 1 to 1000.

**A positive integer is called Armstrong number of order n if**, **abcd…. = an** **+ bn** **+ cn** **+ dn** **+ ….**

**Armstrong.java**

package com.torryharris

import java.util.Scanner;

public class ArmstorngNumber {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

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

int num=scan.nextInt();

if(isArmNUm(num))

{

System.out.println(num + " is armstrong number");

}

else

System.out.println(num + " is not armstrong number");

System.out.println("Armstrong number between 1 to 1000 :");

for(int i=1;i<=1000;i++)

{

if(isArmNUm(i))

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

}

}

public static boolean isArmNUm(int n)

{

int temp=n;

int rem=0,sum=0,s=0;

while(temp>0)

{

temp=temp/10;

s++;

}

temp=n;

while(temp>0)

{

rem= temp%10;

sum=sum + (int)Math.pow(rem,s);

temp=temp/10;

}

if (sum==n)

return true;

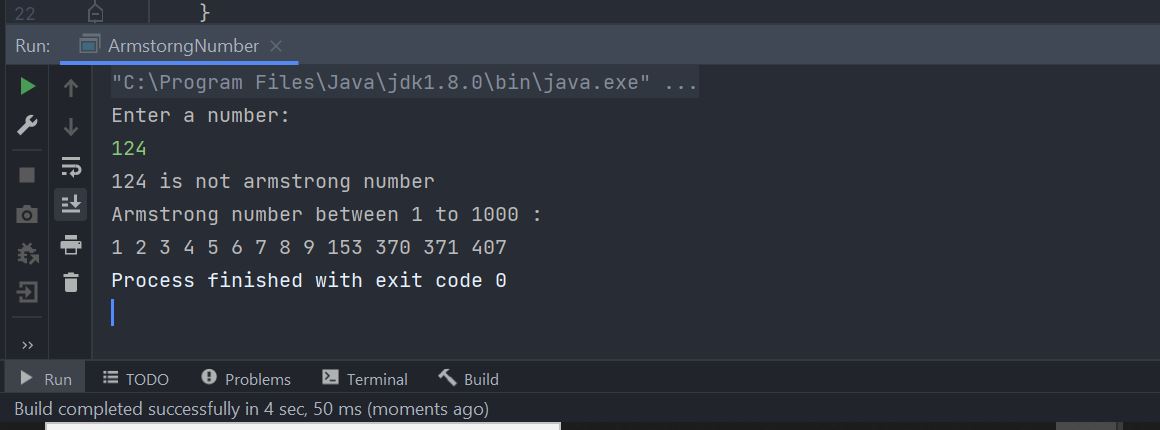
else

return false;

}

}

**Output:**

****

**17.** **In mathematics, a number is called an Automorphic number if the square of the number ends with the same number. Example of Automorphic numbers are:- 5, 6, 25, 76, e.t.c..**

**AutomorphicNum.java**

package com.torryharris;

import java.util.Scanner;

public class AutomorphicNum {

public static void main(String[] args) {

int num;

Scanner scan= new Scanner(System.in);

System.out.println("Enter a number");

num=scan.nextInt();

int ldigit= num%10;

if( ((num\*num)%10) ==ldigit )

System.out.println(num+" is a Automorphic number");

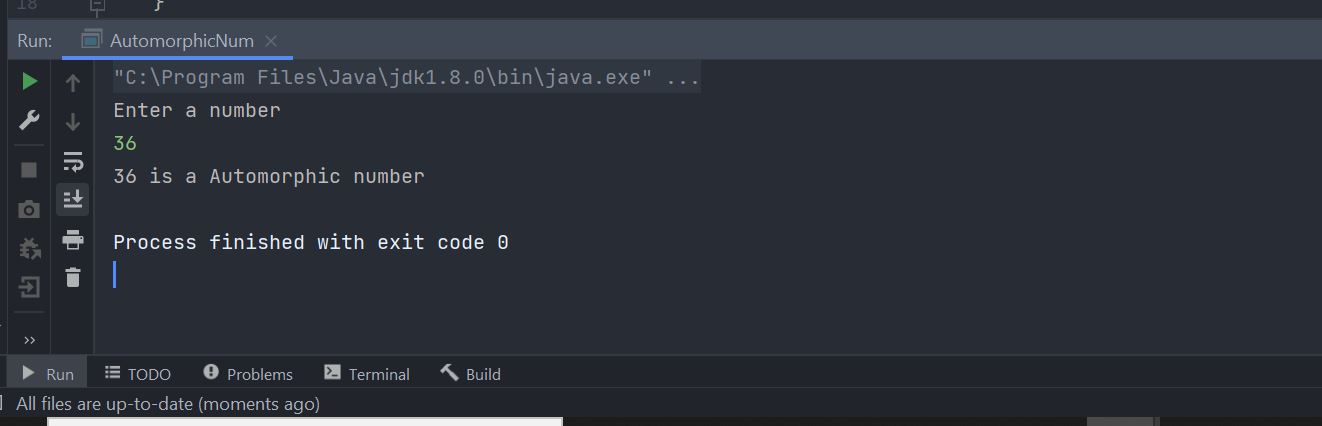
else

System.out.println(num+" is not a Automorphic number");

}

}

**Output:**

****

**18.** **we will develop a Java program for a BMI calculator.**

The BMI stands for Body Mass Index. It is a measure of health based on height and weight. It can be calculated by taking the weight in kilograms and dividing it by the square of your height in meters.

Formula for Calculating BMI in Metric Units,

***BMI = (Weight in Kg) / (Height in Meters \* Height in Meters)***

For example,  
weight = 75 kg, height = 1.5 m  
BMI = 75 / (1.5\*1.5) = 33.33

Using the range of BMI, individuals are classified as underweight, normal or overweight. Its value is in a specific range for a healthy individual. The following table shows the main BMI categories,

|  |  |
| --- | --- |
| ***BMI Range*** | ***Category*** |
| < 18.5 | Thinness |
| 18.5 – 25 | Normal |
| 25 – 30 | Overweight |
| > 30 | Obese |

BMI calculator finds whether the person is underweight, normal, overweight, or obese. The personwho wants to lose their want can use a weight loss calculator which can tell the daily caloric requirements, and many easy tips to lose weight.

**BmiCalc.java**

package com.torryharris;

import java.util.Scanner;

public class BmiCalc {

public static void main(String[] args) {

float bmi,weight,height;

Scanner scan = new Scanner(System.in);

System.out.println("Enter weight in KG:");

weight= scan.nextFloat();

System.out.println("Enter height in meter:");

height = scan.nextFloat();

bmi= weight/(height\*height);

System.out.format("BMI= %.2f\n",bmi);

if(bmi<18.5)

System.out.println("Thinness");

else if(bmi>=8.5&& bmi<=25)

System.out.println("Normal");

else if(bmi>25 && bmi<=30)

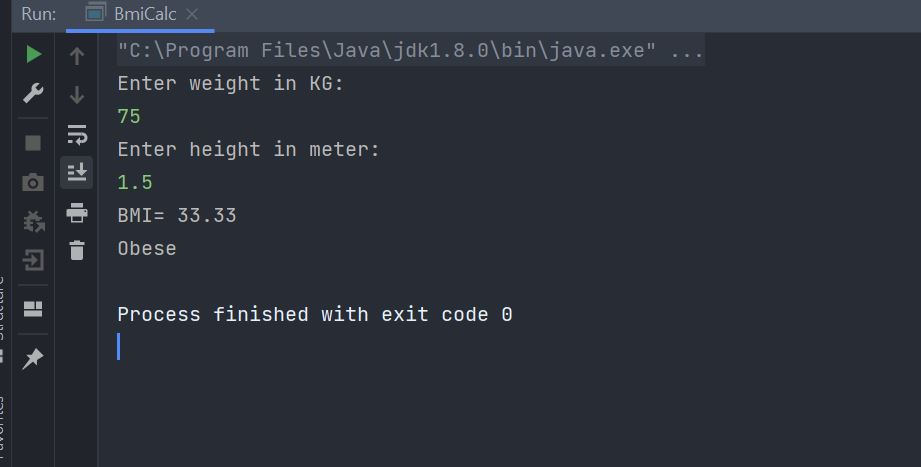
System.out.println("Overweight");

else

System.out.println("Obese");

}

}



**19.Perfect number**:- **A number whose factors sum except itself, is equal to the same number is called a perfect number.**

**PerfectNum.java**  
package com.torryharris;

import java.util.Scanner;

public class PerfectNum {

public static void main(String[] args) {

int sum=0;

Scanner scan = new Scanner(System.in);

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

int n=scan.nextInt();

for(int i=1;i<=n/2;i++)

{

if(n%i==0)

{

sum=sum+i;

}

}

if(sum==n)

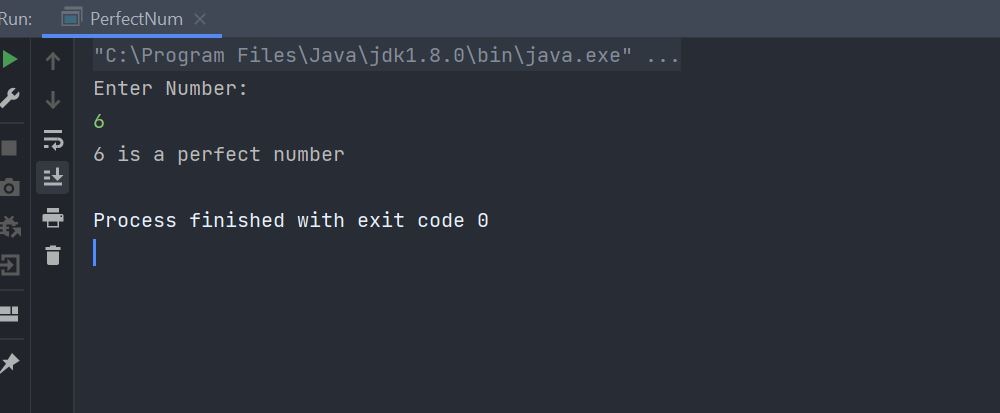
System.out.println(n+" is a perfect number");

else

System.out.println(n+" not a perfect number");

}

}



**20.Write a Java program to find the second largest number in Java.**

**SecondMax.java**

package com.torryharris;

import java.util.Scanner;

public class SecondMax {

public static void main(String[] args) {

int len,max;

System.out.println("Enter number of elements");

Scanner scan = new Scanner(System.in);

len=scan.nextInt();

int arr[]= new int[len];

System.out.println("Enter "+len+" elements:");

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

{

arr[i]= scan.nextInt();

}

for(int i=0;i<len-1;i++)

{

for(int j=0;j<(len-i-1);j++)

{

if(arr[j]<arr[j+1])

{

arr[j]=arr[j]+arr[j+1];

arr[j+1]=arr[j]-arr[j+1];

arr[j]=arr[j]-arr[j+1];

}

}

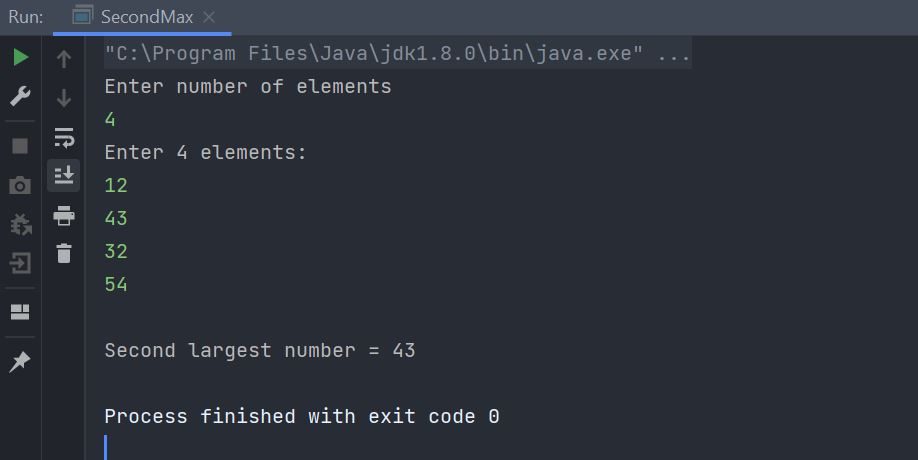
}

System.out.println("\nSecond largest number = "+ arr[1]);

}

}

**Output:**

****