**1)- Take an input string like-**

**String input = "Hello JavaTpoint";**

**You need to write the program to reverse the String?**

**CODE:**

class Solution

{

//Reverse a string

static String reverse(String p)

{

String s=p;

String reverse="";

char[] S=new char[s.length()];

int l=s.length()-1;

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

{

S[l-i]=s.charAt(i);

}

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

{

reverse=reverse+(S[i]);

}

return reverse;

}

}

public class Main

{

public static void main(String[] args) {

Solution d=new Solution();

Scanner sc=new Scanner(System.in);

//REVERSE A STRING

System.out.println("Enter the string to reverse: ");

String s=sc.nextLine();

System.out.println("String after reverse: ");

System.out.println(d.reverse(s));

System.out.println();

}

}

**2->Write a program to print the prime number between 1 to 20.**

**CODE:**

class Solution

{

static int[] prime()

{

int[] primes=new int[100];

int k=0;

for(int i=1;i<=20;i++)

{

int count=0;

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

{

if(i%j==0)

{

count++;

}

}

if(count==1)

{

primes[k]=i;

k++;

}

}

return primes;

}

}

public class Main

{

public static void main(String[] args) {

Solution d=new Solution();

//primes

int[] p=d.prime();

System.out.println("PRIME NUMBER BETWEEN 1 AND 20:");

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

{

if(p[i]!=0)

{

System.out.println(p[i]);}}}}

**3-> Write a program to print the table of 2 to 5 numbers.**

class Solution

{

//print 2 to 5 tables

static void tables(int n)

{

int N=n;

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

{

for (int j=1;j<=10;j++)

{

System.out.println(i + " x " + j +" = " + i\*j);

}

System.out.println();

}

}

}

public class Main

{

public static void main(String[] args) {

Solution d=new Solution();

//TABLES FROM 2 to 5

System.out.println("TABLES");

int n=5;

d.tables(n);

}}

**4-> Write a program to print the even and odd numbers between 1 to 20.**

class Solution

{

/print even and odd numbers between 1 to 20

static void even\_odd(int n)

{

System.out.println("EVEN NUMBERS: ");

int m=n;

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

{

if(i%2==0)

{

System.out.println(i);

}

}

System.out.println();

System.out.println("ODD NUMBERS: ");

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

{

if(i%2!=0)

{

System.out.println(i);

}

}

}}

public class Main

{

public static void main(String[] args) {

Solution d=new Solution();

Scanner sc=new Scanner(System.in);

d.even\_odd(20);

}

}

**5-> Write a program to print the division of Student marks based on percentage.**

**Like: Dictation, first division, second division, third division, fail.**

**Solve the program by if-else and switch statements.**

class Solution

{

//Grading using if else

static void Grading(int percent)

{

if(percent>=90)

{

System.out.println("The student has achieved Dictation Grade");

}

else if(percent<90 && percent>=80)

{

System.out.println("The student has achieved 1st Grade");

}

else if(percent<80 && percent >=60)

{

System.out.println("The student has achieved 2nd Grade");

}

else if(percent<60 && percent>=40)

{

System.out.println("The student has achieved 3rd Grade");

}

else

{

System.out.println("The student is failed");

}

}

//Grading using switch case

static void switch\_grading(int percent)

{

int f=percent/10;

switch(f)

{

case 10:

System.out.println("The student has achieved Dictation Grade");

break;

case 9:

System.out.println("The student has achieved Dictation Grade");

break;

case 8:

System.out.println("The student has achieved 1st Grade");

break;

case 7:

System.out.println("The student has achieved 2nd Grade");

break;

case 6:

System.out.println("The student has achieved 2nd Grade");

break;

case 5:

System.out.println("The student has achieved 3rd Grade");

break;

case 4:

System.out.println("The student has achieved 3rd Grade");

break;

case 3:

System.out.println("The student is failed");

break;

case 2:

System.out.println("The student is failed");

break;

case 1:

System.out.println("The student is failed");

break;

case 0:

System.out.println("The student is failed");

break;

}

}

}

public class Main

{

public static void main(String[] args) {

Solution d=new Solution();

Scanner sc=new Scanner(System.in);

//grade using if-else

d.Grading(59);

//grade using switch

d.switch\_grading(59);

}

}

**6-> Write a program for factorial 8;**

class Solution

{

//Factorial

int factorial(int n)

{

if(n==0)

{

return 1;

}

else

{

return n\*factorial(n-1);

}

}}

public class Main

{

public static void main(String[] args) {

Solution d=new Solution();

Scanner sc=new Scanner(System.in);

//factorial

int fact=d.factorial(8);

System.out.println("Factorial of 8 is: "+fact);

System.out.println();

}}

**7-> Write a program to swap the a and b numbers. Where int a = 5; int b = 10;**

class Solution

{

static void swap(int a,int b)

{

System.out.println("Numbers before swaping: "+"a = "+a+" b = "+ b);

a=a+b;

b=a-b;

a=a-b;

System.out.println("Numbers after swaping: "+"a = "+a+" b = "+ b);

}

}

public class Main

{

public static void main(String[] args) {

Solution d=new Solution();

Scanner sc=new Scanner(System.in);

//swapping Numbers

d.swap(5,10);

}}

**8-> Write a program to print the given number is leap year or not. Where your**

**number is int year = 1990;**

class Solution

{

//leap year or not

static void leap(int n)

{

if(((n%4==0) && (n%100!=0)) || (n%400==0))

{

System.out.println(n+" is a leap year");

}

else

{

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

}

}

}

public class Main

{

public static void main(String[] args) {

Solution d=new Solution();

Scanner sc=new Scanner(System.in);

//leap

d.leap(1990);

}}

**9-> Write a program to Print a solid Diamond Star Pattern.**

**\***

**\* \***

**\* \* \***

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

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

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

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

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

**\* \* \***

**\* \***

**\***

class Solution

{

void pattern\_diamond(int number)

{

int m, n;

// prints the first half diamond

for (m = 1; m <= number; m++)

{

for (n = 1; n <= number - m; n++)

{

System.out.print(" ");

}

for (n = 1; n <= m \* 2 - 1; n++)

{

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

}

System.out.println();

}

for (m = number - 1; m > 0; m--)

{

for (n = 1; n <= number - m; n++)

{

System.out.print(" ");

}

for (n = 1; n <= m \* 2 - 1; n++)

{

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

}

System.out.println();

}

}

**}**

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

Solution S=new Solution();

System.out.println("Enter the integer for pattern");

int n=sc.nextInt();

S.pattern\_diamond(n);

}}

**10->Write a Program to Print the Pant's style Star Pattern.**

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

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

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

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

**\* \***

import java.util.\*;

class Solution{

void pant(int k)

{

int p=k;

int i,j;

int x=p;

int y=p;

for(i=1;i<=p;i++)

{

for(j=1;j<p\*2;j++)

{

if(j>x && j<y)

{

System.out.printf(" ");

}

else

{

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

}

}

x--;

y++;

System.out.println();

}

}

}

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

Solution S=new Solution();

System.out.println("Enter the integer for pattern");

int n=sc.nextInt();

S.pant(n);

}}

**11-> Write a Program to Print the Inverted Pant's Shape Star Pattern.**

**\* \***

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

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

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

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

import java.util.\*;

class Solution{

void reverse\_pant(int k)

{

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

{

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

{

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

}

for(int j=1;j<=2\*(k-i);j++)

{

System.out.print(" ");

}

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

{

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

}

System.out.println();

}

}

}

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

Solution S=new Solution();

System.out.println("Enter the integer for pattern");

int n=sc.nextInt();

S.reverse\_pant(n);

}}

**12-> Find out the 3rd max element from the array with a single loop.**

**int[] arr = {22,100,44,11,22,100,77,44,11};**

class Solution

{

static void thirdLargest(int arr[], int arr\_size)

{

if (arr\_size < 3)

{

System.out.println("Invalid Input");

return;

}

int first = arr[0], second = Integer.MIN\_VALUE,third = Integer.MIN\_VALUE;

for (int i = 1; i < arr\_size; i++)

{

if (arr[i] > first)

{

third = second;

second = first;

first = arr[i];

}

else if (arr[i] > second)

{

third = second;

second = arr[i];

}

else if (arr[i] > third)

{

third = arr[i];

}

}

System.out.println("The Third Largest element is %d\n", third);

}

}

}

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

Solution S=new Solution();

System.out.println("Enter the size of array: ");

int array\_size=sc.nextInt();

int[] arr=new int[array\_size];

System.out.println("Enter the elements of array: ");

for(int i=0;i<array\_size;i++)

{

arr[i]=sc.nextInt();

}

S.thirdLargest(arr,array\_size);

**}}**

**1-> Create 1st class, where a class has one method that takes 2 integers as parameters and returns an integer denoting their sum.**

**Create 2nd class that inherits from a superclass of the first class.**

**Again 3rd class and here print the output of the first class method with the help of 2nd class object.**

class First

{

int sum(int a,int b)

{

return a+b;

}

}

class Second extends First

{

}

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

Second obj=new Second();

System.out.println("Enter two numbers: ");

int a=sc.nextInt();

int b=sc.nextInt();

System.out.println("The sum of "+ a +" and "+ b + " is " +obj.sum(a,b));

}

}

**2-> Create an interface that contains a method, where the method return type is int and the method has one int type parameter. You need to write another class that implements the interface.**

**And your function just takes an integer as input and return the sum of all its divisors. For example divisors of 8 are 1, 2, 4 and 8, so divisor addition should return 15. The value of n will be at most 1000.**

interface ABC

{

int divisors(int n);

}

class NEW implements ABC

{

int sum=0;

public int divisors(int n)

{

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

{

if(n%i==0)

{

sum=sum+i;

}

}

return sum+n;

}

}

public class Main

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

NEW N=new NEW();

System.out.println(N.divisors(n));

}}

**3-> Write a program to sort the given array of String in descending order. Your string value should be the same during the output. Like- .12 printed as .12 and 0.12 printed as 0.12**

**Ex-> String[] str = {“-100”,”50”,”.12”,”0.12”,”0”,”000.000”};**

class Solution

{

public static HashMap<String, Float> sortByValue(HashMap<String, Float> hm)

{

List<Map.Entry<String, Float> > list =

new LinkedList<Map.Entry<String, Float> >(hm.entrySet());

Collections.sort(list, new Comparator<Map.Entry<String, Float> >() {

public int compare(Map.Entry<String, Float> o1,

Map.Entry<String, Float> o2)

{

return (o2.getValue()).compareTo(o1.getValue());

}

});

HashMap<String, Float> temp = new LinkedHashMap<String, Float>();

for (Map.Entry<String, Float> aa : list) {

temp.put(aa.getKey(), aa.getValue());

}

return temp;

}

}

public class Main

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

Solution S=new Solution();

System.out.println("Enter the size of array: ");

int n=sc.nextInt();

System.out.println("Enter elements of array: ");

String[] str=new String[n];

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

{

str[i]=sc.next();

}

HashMap<String,Float> hm=new HashMap<String,Float>();

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

{

hm.put(str[i],Float.parseFloat(str[i]));

}

hm=S.sortByValue(hm);

Iterator <String> it = hm.keySet().iterator();

while(it.hasNext())

{

String key=it.next();

System.out.println(key);

}

}

}

**4-> You will be given two integers a and b as input, you have to compute a/b.**

**If a and b are not 32 bit signed integers or if b is zero,**

**exception will occur and you have to report it.**

**Read sample Input/Output to know what to report in case of exceptions.**

**Sample Input 0:**

**10**

**3**

**Sample Output 0:**

**3**

**Sample Input 1:**

**10**

**Hello**

**Sample Output 1:**

**java.util.InputMismatchException**

**Sample Input 2:**

**10**

**0**

**Sample Output 2:**

**java.lang.ArithmeticException: / by zero**

**Sample Input 3:**

**23.323**

**0**

**Sample Output 3:**

**java.util.InputMismatchException**

**Sample Input 4:**

**2147483648**

**55**

**Sample Output 4:**

**java.util.InputMismatchException**

public class Main

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

/\*

int a,b;

int result;

System.out.println("Enter 2 numbers: ");

try

{

a=sc.nextInt();

b=sc.nextInt();

result=a/b;

System.out.println("Division of number results " + result);

}

catch(InputMismatchException e)

{

System.out.println(e.toString());

}

catch(ArithmeticException e)

{

System.out.println(e.toString());

}

}

}

**5->**

**Write a Java program to create a class known as "SBIBankAccount"**

**with methods called deposit() and withdraw().**

**And create opening account methos openAccount.**

**create sub classes for premiumAccount and jundhanAccount and override the deposit,**

**withdraw, and openAccount in sub class.**

**Where if you open premium account should be start from 5K and your jundhanAccount**

**start from 0 balance.**

**where withdraw() method to prevent withdrawals if the account balance falls**

**below 5K for the premium account and prevent withdrawals if the account balance falls**

**below 0 for the jundhanAccount.**

import java.util.\*;

abstract class SBIBankAccount

{

double balance=0.0d;

public abstract void deposit(double amount);

public abstract void withdrawl(double amount);

public abstract void openAccount();

}

class premiumAccount extends SBIBankAccount

{

premiumAccount()

{

}

public void openAccount()

{

balance=balance+5000;

}

public void withdrawl(double amount)

{

balance=balance-amount;

if(balance<5000)

{

balance=balance+amount;

System.out.println("Your Premium Accounts does not have Rs 5000 after this withdrawl. Try a smaller amount.");

}

else

{

System.out.println("Amount "+ amount + "has been debited from your Premium Account.");

}

}

public void deposit(double amount)

{

balance=balance+amount;

}

}

class jundhanAccount extends SBIBankAccount

{

jundhanAccount()

{

}

public void openAccount()

{

balance=balance+0;

}

public void withdrawl(double amount)

{

balance=balance-amount;

if(balance<0)

{

balance=balance+amount;

System.out.println("Your Jundhan Accounts does not have this amount. Try a smaller amount.");

}

else

{

System.out.println("Amount "+ amount + "has been debited from your Jundhan Account.");

}

}

public void deposit(double amount)

{

balance=balance+amount;

}

}

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

String account\_type;

int n=-1;

int m=0;

premiumAccount PA=new premiumAccount();

jundhanAccount JA=new jundhanAccount();

do

{

System.out.println("Which type of account would you like to have? (Jundhan/Premium):");

account\_type=sc.next();

account\_type=account\_type.toUpperCase();

if(account\_type=="JUNDHAN")

{

JA.openAccount();

}

else if(account\_type=="PREMIUM")

{

PA.openAccount();

}

System.out.println("Do you want to open another account? Enter 1 for yes or 0 for no.");

n=sc.nextInt();

}while(n==1);

do

{

int action;

System.out.println("To withdrawl from Jhandhan-enter 1\nTo deposit in Jhandhan-enter 2\nTo withdrawl from Premium-enter 3\nTo deposit to Premium-enter 4\nTo exit-enter 5");

action=sc.nextInt();

String temp\_type;

double temp\_amount;

switch(action)

{

case 1:

System.out.println("Enter the amount you would like to withdrawl:");

temp\_amount=sc.nextDouble();

JA.withdrawl(temp\_amount);

break;

case 2:

System.out.println("Enter the amount you would like to deposit:");

temp\_amount=sc.nextDouble();

JA.deposit(temp\_amount);

break;

case 3:

System.out.println("Enter the amount you would like to withdrawl:");

temp\_amount=sc.nextDouble();

PA.withdrawl(temp\_amount);

break;

case 4:

System.out.println("Enter the amount you would like to deposit:");

temp\_amount=sc.nextDouble();

PA.deposit(temp\_amount);

break;

case 5:

break;

}

System.out.println("Do you want to do any other action? Enter 1 for Yes and 0 for No.");

m=sc.nextInt();

}while(m==1);

}

}

**16- Write a Java programming to create a banking system with three classes -**

**Bank, Account, SavingsAccount, and CurrentAccount. The bank should have a list of**

**accounts and methods for adding them. Accounts should be an interface with methods**

**to deposit, withdraw, calculate interest, and view balances. SavingsAccount and**

**CurrentAccount should implement the Account interface and have their own unique**

**methods.**

interface Account

{

void deposit(double amount);

void withdrawl(double amount);

double calculateInterest();

double viewBalance();

}

class SavingsAccount implements Account

{

double balance;

double interest;

SavingsAccount()

{

}

SavingsAccount(double balance,double interest)

{

this.balance=balance;

this.interest=interest;

}

public void deposit(double amount)

{

balance=balance+amount;

}

public void withdrawl(double amount)

{

balance=balance-amount;

}

public double calculateInterest()

{

double intrestamount=(balance\*interest)/100;

balance=balance+intrestamount;

return balance;

}

public double viewBalance()

{

return balance;

}

}

class CurrentAccount implements Account

{

double balance;

double interest;

CurrentAccount()

{

}

CurrentAccount(double balance,double interest)

{

this.balance=balance;

this.interest=interest;

}

public void deposit(double amount)

{

balance=balance+amount;

}

public void withdrawl(double amount)

{

balance=balance-amount;

}

public double calculateInterest()

{

double intrestamount=(balance\*interest)/100;

balance=balance+intrestamount;

return balance;

}

public double viewBalance()

{

return balance;

}

}

class Bank

{

ArrayList<String> Accounts\_list;

public Bank()

{

Accounts\_list=new ArrayList<String>();

}

public void Add\_Accounts(String account)

{

Accounts\_list.add(account);

}

public void Remove\_Accounts(String account)

{

Accounts\_list.remove(account);

}

public void deposit(Account account\_obj,double amount)

{

account\_obj.deposit(amount);

}

public void withdrawl(Account account\_obj,double amount)

{

account\_obj.withdrawl(amount);

}

public double viewBalance(Account account\_obj)

{

return account\_obj.viewBalance();

}

public double calculateInterest(Account account\_obj)

{

return account\_obj.calculateInterest();

}

}

public class Main

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

Bank bank=new Bank();

int n=-1;

int m;

double initial\_amount;

String account\_type;

double interest\_savings=1.2d;

double interest\_current=2.5d;

SavingsAccount savings=new SavingsAccount();

CurrentAccount currents=new CurrentAccount();

do

{

System.out.println("Enter the type of account type you want to have (Savings/Current) and initial amount:");

account\_type=sc.next();

initial\_amount=sc.nextDouble();

if(account\_type.equals("SAVING"))

{

savings=new SavingsAccount(initial\_amount,interest\_savings);

}

else if(account\_type.equals("CURRENT"))

{

currents=new CurrentAccount(initial\_amount,interest\_savings);

}

else

{

System.out.println("Enter a valid account type.");

n=1;

}

System.out.println("Do you want to open any other account? Enter 1 for Yes and 0 for No.");

n=sc.nextInt();

if(n==-1)

{

n=sc.nextInt();

}

}while(n==1);

do

{

int action;

System.out.println("To withdrawl-enter 1\nTo deposit-enter 2\nFTo get balance amount details-enter 3\nTo calculate balance after intrest\nTo exit-enter 5");

action=sc.nextInt();

String temp\_type;

double temp\_amount;

switch(action)

{

case 1:

System.out.println("Enter the account type(savings/currents) and amount you would like to withdrawl:");

temp\_type=sc.next();

temp\_amount=sc.nextDouble();

if(temp\_type=="savings")

{

bank.withdrawl(savings,temp\_amount);

}

else

{

bank.withdrawl(currents,temp\_amount);

}

break;

case 2:

System.out.println("Enter the account type(savings/currents) and amount you would like to deposit:");

temp\_type=sc.next();

temp\_amount=sc.nextDouble();

if(temp\_type=="savings")

{

bank.deposit(savings,temp\_amount);

}

else

{

bank.deposit(currents,temp\_amount);

}

break;

case 3:

System.out.println("Enter the account type(savings/currents):");

temp\_type=sc.next();

if(temp\_type=="savings")

{

System.out.println(savings.viewBalance());

}

else

{

System.out.println(currents.viewBalance());

}

break;

case 4:

System.out.println("Enter the account type(savings/currents):");

temp\_type=sc.next();

temp\_amount=sc.nextDouble();

if(temp\_type=="savings")

{

System.out.println(bank.calculateInterest(savings));

}

else

{

System.out.println(bank.calculateInterest(currents));

}

break;

case 5:

break;

}

System.out.println("Do you want to do any other action? Enter 1 for Yes and 0 for No.");

m=sc.nextInt();

}while(m==1);

}

}