**P1.Progaram to print the sum of numbers entered as command line arguements.**

package p1;

public class P1

{

public static void main(String[] args)

{

System.out.println("Progaram to print the sum of numbers entered as command line arguements.");

int a,b,sum;

a=Integer.parseInt(args[0]);

b=Integer.parseInt(args[1]);

sum=a+b;

System.out.println(sum);

}

}

**P2.Program to print the factorial of a given number.**

package p2;

import java.util.Scanner;

public class P2

{

public static void main(String[] args)

{

System.out.println("Program to print the factorial of a given number!");

System.out.println("Please enter a number to compute it's factorial :");

int n;

Scanner inp=new Scanner(System.in);

n=inp.nextInt();

System.out.println(n);

int result=factorial(n);

System.out.println("Factorial of the number "+n+" entered is: "+result);

}

private static int factorial(int n)

{

if(n>0)

return n\*factorial(n-1);

if(n<0)

return 1;

else

return 1;

}

}

**P3.Program to learn the use of single dimension array by defining the array dynamically.**

package p3;

import java.util.Scanner;

public class P3

{

public static void main(String[] args)

{

System.out.println("Program to learn the use of single dimension array by defining the array dynamically.");

System.out.println("Enter a positive number only for the size of the array: ");

Scanner inp=new Scanner(System.in);

int n=inp.nextInt();

while(n<=0)

{

System.out.println("Please enter a positive number only: ");

n=inp.nextInt();

}

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

{

System.out.println(i);

}

int i=1;

while(i<=n)

{

System.out.println(i);

i++;

}

i=1;

do

{

System.out.println(i);

i++;

}

while(i<=n);

int ar[];

ar=new int[n];

System.out.println("Enter Array Elements: ");

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

{

ar[i]=inp.nextInt();

}

System.out.print("Array elements are: [");

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

{

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

}

System.out.print("]");

}

}

**P4.To learn the use of Two Dimensional array by defining the array dynamically.**

package p4;

import java.util.Scanner;

public class P4

{

public static void main(String[] args)

{

System.out.println("To learn the use of Two Dimensional array by defining the array dynamically.");

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

Scanner inp=new Scanner(System.in);

int row=inp.nextInt();

while(row<=0)

{

System.out.print("Please enter a positive number only: ");

row=inp.nextInt();

}

int col=inp.nextInt();

while(col<=0)

{

System.out.print("Please enter a positive number only: ");

col=inp.nextInt();

}

int arr[][];

arr=new int[row][col];

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

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

{

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

{

arr[i][j]=inp.nextInt();

}

}

System.out.println("Array elements are: ");

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

{

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

{

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

}

System.out.println();

}

}

}

**P5.Program to convert a decimal to a binary number.**

package p5;

import java.util.Scanner;

public class P5

{

public static void main(String[] args)

{

System.out.println("Program to convert a decimal to a binary number: ");

System.out.println("Enter the decimal number: ");

Scanner inp=new Scanner(System.in);

int n=inp.nextInt();

int count=0;

String str=" ";

while(n>=0)

{

count=n%2;

str=count+" "+str;

n=n/2;

}

if(n<0)

{

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

}

if(n==0)

System.out.print("It's Binary is: "+str);

}

}

**P6.Program to inout a prime number from the user and check whether its prime or not.**

package p6;

import java.util.Scanner;

public class P6

{

public static void main(String[] args)

{

System.out.println("Program to inout a prime number from the user and check whether its prime or not.");

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

Scanner inp=new Scanner(System.in);

int n=inp.nextInt();

boolean flag=false;

while(n==0 || n==1)

{

System.out.println("Enter a natural number more than 1: ");

n=inp.nextInt();

}

int c;

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

{

if(n%i==0)

{

flag=true;

break;

}

}

if(flag)

{

System.out.println("Number is not prime.");

}

else

System.out.println("Number is prime.");

}

}

**P7.Program to find the sum of any number of integers, entering numbers from the keyboard and entering the number of numbers as command line arguements.**

package p7;

import java.util.Scanner;

public class P7

{

public static void main(String[] args)

{

System.out.println("Program to find the sum of any number of integers, entering numbers from the keyboard and entering the number of numbers as command line arguements.");

int n=Integer.parseInt(args[0]);

System.out.println("Enter "+n+" numbers: ");

int sum=0;

Scanner inp=new Scanner(System.in);

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

{

sum+=inp.nextInt();

}

System.out.println("Their sum is: "+sum);

}

}

**P8.Program to show the working of different functions of String and StringBufferclass like setCharAt(),setLength(),append(),insert(),concat() and equals().**

package p8;

import java.util.Scanner;

public class P8

{

public static void main(String[] args)

{

System.out.println("Program to show the working of different functions of String and StringBufferclass like setCharAt(),setLength(),append(),insert(),concat() and equals().");

Scanner inp=new Scanner(System.in);

System.out.print("Please enter the string:");

StringBuffer str=new StringBuffer(inp.next());

System.out.println("The string entered is: "+str);

System.out.println("Enter your choice for showing the working of the following functions:\n 1. To insert a character at a specified position of the entered string.;\n 2. To show the specified length of the entered string.;\n 3. To add a character or another string at the end of the entered string;\n 4. To insert a character or another string at a specified position of the entered string.;\n 5. To add two strings after entering them and to show the resultant string ;\n 6. To check whether the entered two strings are exactly the same or not.;");

int n=inp.nextInt();

char var='y';

while(var=='y')

{

System.out.println("Enter y to continue and n to exit.");

var=inp.next().charAt(0);

switch(n)

{

case 1:System.out.println("Showing the working of choice 1:");

System.out.print("Enter the position where the new character needs to be replaced: ");

int pos1=inp.nextInt();

System.out.print("Enter the character to be placed at position "+pos1+": ");

char ch=inp.next().charAt(0);

str.setCharAt(pos1, ch);

System.out.println("The corrected string is: "+str);

break;

case 2:System.out.println("Showing the working of choice 2:");

System.out.print("Enter the length of the string to be displayed: ");

int pos2=inp.nextInt();

str.setLength(pos2);

System.out.println("The modified string is: "+str);

break;

case 3:System.out.println("Showing the working of choice 3:");

System.out.print("Enter the string to be appended: ");

StringBuffer strapp=new StringBuffer(inp.next());

str.append(strapp);

System.out.println("Modified String is: "+str );

break;

case 4:System.out.println("Showing the working of choice 4:");

System.out.println("Enter the position at which a character or a string needs to be inserted:");

int pos4=inp.nextInt();

System.out.println("Enter the character or a string to be inserted at position:"+pos4);

StringBuffer strins=new StringBuffer(inp.next());

str.insert(pos4, strins);

System.out.println("Modified String is:"+str);

break;

case 5:System.out.println("Showing the working of choice 5:");

String strcatt1=str.toString();

System.out.print("Enter the second string to be concatenated with the entered string: ");

StringBuffer strcat2=new StringBuffer(inp.next());

String strcatt2=strcat2.toString();

String strcatt3=strcatt1.concat(strcatt2);

System.out.println("Modified String is:"+strcatt3);

break;

case 6:System.out.println("Showing the working of choice 6:");

System.out.println("Checks whether the entered two strings are equal or not.");

String strreq1=str.toString();

System.out.print("Enter the second string :");

StringBuffer streq2=new StringBuffer(inp.next());

String strreq2=streq2.toString();

boolean x=strreq1.equals(strreq2);

if(x)

{

System.out.println("Strings are equal");

}

else

System.out.println("Strings are not equal");

break;

default:System.out.println("Invalid Entry. Enter from {1,2,3,4,5,6}");

break;

}

}

}

}

**P9.Program to create a 'distance' class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer.**

package p9;

import java.util.Scanner;

class distance

{

int inches=0,feet=0;

void calcdist()

{

int inches;

int feet;

Scanner inp=new Scanner(System.in);

System.out.println("Choose to Enter the distance whether in \n 1. cm \n 2. mm \n or 3. inches");

int k=inp.nextInt();

System.out.println("Global variables have the values: feet="+this.feet+" and inches="+this.inches);

switch(k)

{

case 1:System.out.print("Enter the distance in cm:");

float cm=inp.nextFloat();

inches=(int)(cm/2.54);

feet=(int)inches/12;

inches=inches%12;

System.out.println("The distance in feet and inches is:(Local Variables) "+feet+"feet "+inches+"inches");

System.out.println("The distance in feet and inches is:(Global Variables) "+this.feet+"feet "+this.inches+"inches");

break;

case 2:System.out.print("Enter the distance in mm:");

float mm=inp.nextFloat();

inches=(int)(mm/25.4);

feet=(int)inches/12;

inches=inches%12;

System.out.println("The distance in feet and inches is:(Local Variables) "+feet+"feet "+inches+"inches");

System.out.println("The distance in feet and inches is:(Global Variables) "+this.feet+"feet "+this.inches+"inches");

break;

case 3:System.out.print("Enter the distance in inches:");

inches=inp.nextInt();

feet=(int)(inches/12);

inches=inches%12;

System.out.println("The distance in feet and inches is:(Local Variables) "+feet+"feet "+inches+"inches");

System.out.println("The distance in feet and inches is:(Global Variables) "+this.feet+"feet "+this.inches+"inches");

break;

default:System.out.println("Enter a valid choice");

}

}

}

public class P9

{

public static void main(String[] args)

{

System.out.println("Program to create a 'distance' class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer. ");

distance d=new distance();

d.calcdist();

}

}

package p9;

import java.util.Scanner;

class distance

{

int inches=0,feet=0;

void calcdist()

{

int inches;

int feet;

Scanner inp=new Scanner(System.in);

System.out.println("Choose to Enter the distance whether in \n 1. cm \n 2. mm \n or 3. inches");

int k=inp.nextInt();

System.out.println("Global variables have the values: feet="+this.feet+" and inches="+this.inches);

switch(k)

{

case 1:System.out.print("Enter the distance in cm:");

float cm=inp.nextFloat();

inches=(int)(cm/2.54);

feet=(int)inches/12;

inches=inches%12;

System.out.println("The distance in feet and inches is:(Local Variables) "+feet+"feet "+inches+"inches");

System.out.println("The distance in feet and inches is:(Global Variables) "+this.feet+"feet "+this.inches+"inches");

break;

case 2:System.out.print("Enter the distance in mm:");

float mm=inp.nextFloat();

inches=(int)(mm/25.4);

feet=(int)inches/12;

inches=inches%12;

System.out.println("The distance in feet and inches is:(Local Variables) "+feet+"feet "+inches+"inches");

System.out.println("The distance in feet and inches is:(Global Variables) "+this.feet+"feet "+this.inches+"inches");

break;

case 3:System.out.print("Enter the distance in inches:");

inches=inp.nextInt();

feet=(int)(inches/12);

inches=inches%12;

System.out.println("The distance in feet and inches is:(Local Variables) "+feet+"feet "+inches+"inches");

System.out.println("The distance in feet and inches is:(Global Variables) "+this.feet+"feet "+this.inches+"inches");

break;

default:System.out.println("Enter a valid choice");

}

}

}

public class P9

{

public static void main(String[] args)

{

System.out.println("Program to create a 'distance' class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer. ");

distance d=new distance();

d.calcdist();

}

}

***P10.rogram to modify the 'distance' class by creating constructor for assigning values(feet and inches) to the distance object. Create another object and assign second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.***

package p10;

public class P10

{

public static void main(String[] args) throws CloneNotSupportedException

{

System.out.println("Program to modify the 'distance' class by creating constructor for assigning values(feet and inches) to the distance object. Create another object and assign second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.");

distance cobj1=new distance(2,3);

System.out.println("Values of object 1:");

cobj1.display();

distance cobj2=cobj1;

System.out.println("Values of object 2 which is pointing object 1:");

cobj2.display();

cobj2.inches=6;

cobj2.feet=7;

System.out.println("Values of object 1 after it's modification from pointer object 2:");

cobj1.display();

System.out.println("Values of object 2:");

cobj2.display();

distance cobj3=(distance)cobj1.clone();

System.out.println("Values of object 3 after making it's clone from object 1:");

cobj3.display();

cobj3.inches=10;

cobj3.feet=13;

System.out.println("Values of object 3 changed:");

cobj3.display();

System.out.println("Values of object 1 remains same when values of object 3 are shown(Cloning shown):");

cobj1.display();

}

}

class distance implements Cloneable

{

int inches,feet;

public distance(int x,int y)

{

inches=x;

feet=y;

}

void display()

{

System.out.println("Values are:"+inches+"inches and "+feet+"feet");

}

public Object clone() throws CloneNotSupportedException

{

return super.clone();

}

}

***P11.Program to show that during functon overloading, if no matching arguement is formed, then java will apply automatic type conversion(From lower to higher data type).***

package p11;

public class P11

{

public static void main(String[] args)

{

System.out.println("Program to show that during functon overloading, if no matching arguement is formed, then java will apply automatic type conversion(From lower to higher data type).");

int a=3,b=4;

sum(a,b);

}

static void sum(double x, double y)

{

double s=x+y;

System.out.println("Sum is:"+s);

}

static void sum()

{

int a=2,b=3;

int s=a+b;

System.out.println(s);

}

}

**P12.Program to show the difference between public and private access specifiers. The program should also show that the primitive data types are passed by value and objects are passed by reference and to learn the use of final keyword.**

//Done more than P12. Done similar to P17.

package p12;

import newpkg.\*;

import java.util.Scanner;

public class P12

{

public static void main(String[] args)

{

System.out.println("Program to show the difference between public and private access specifiers. The program should also show that the primitive data types are passed by value and objects are passed by reference and to learn the use of final keyword.");

Scanner inp=new Scanner(System.in);

System.out.print("Enter your choice:");

int n=inp.nextInt();

New cobj1=new New();

Access cobj2=new Access();

switch(n)

{

case 1:pubfunct();break;

case 2:prifunct();break;

case 3:profunct();break;

case 4:defaultfunct();break;

case 5:cobj1.pubfunct();break;

case 6:cobj1.prifunct();break;

case 7:cobj1.profunct();break;

case 8:cobj1.defaultfunct();break;

case 9:cobj2.pubfunct();break;

case 10:cobj2.prifunct();break;

case 11:cobj2.profunct();break;

case 12:cobj2.defaultfunct();break;

}

}

public static void pubfunct()

{

System.out.println("Public function of this class.");

}

private static void prifunct()

{

System.out.println("Private function of this class.");

}

protected static void profunct()

{

System.out.println("Protected function of this class.");

}

static void defaultfunct()

{

System.out.println("Default function of this class.");

}

}

class New

{

public static void pubfunct()

{

System.out.println("Public function of New class of same package.");

}

private static void prifunct()

{

System.out.println("Private function of New class of same package.");

}

protected static void profunct()

{

System.out.println("Protected function of New class of same package.");

}

static void defaultfunct()

{

System.out.println("Default function of New class of same package.");

}

}

package newpkg;

public class Access

{

public static void pubfunct()

{

System.out.println("Public function of Access class of different package.");

}

private static void prifunct()

{

System.out.println("Private function of Access class of different package.");

}

protected static void profunct()

{

System.out.println("Protected function of Access class of different package.");

}

static void defaultfunct()

{

System.out.println("Default function of Access class of different package.");

}

}

**P13.Program to show the use of Static Functions and to pass variable length arguements in a function.**

package p13;

public class P13

{

public static void main(String[] args)

{

System.out.println("Program to show the use of Static Functions and to pass variable length arguements in a function.");

P13 cobj=new P13();

System.out.println("Using cobj.normfunct(); ");

cobj.normfunct();

System.out.println("Using statfunct(); ");

statfunct();

System.out.println("Using cobj.statfunct();");

cobj.statfunct();

vararg(200,7,80);

}

static void statfunct()

{

System.out.println("This is a static function.");

}

void normfunct()

{

System.out.println("This is a normal function.");

}

static void vararg(int ...n)

{

int count=1;

System.out.println("Value of the arguements passed are:");

for(int i:n)

{

System.out.println("The arguement "+count+" is:");

System.out.println(i);

count++;

}

}

}

**P14.Program to demonstate the concept of boxing and unboxing.**

package p14;

public class P14

{

public static void main(String[] args)

{

System.out.println("Program to demonstate the concept of boxing and unboxing.");

System.out.println("Using Integer a=new Integer(5);");

Integer a=new Integer(5);

System.out.println("Value of a using +a.intValue()(Without Autoboxing):"+a.intValue());

System.out.println("Value of a using +a(Autoboxing):"+a);

int b=67;

Integer c=b;

System.out.println("Value of c from b using +c(Autoboxing):"+c);

int d=a;

System.out.println("Value of d from a using int d=a; (Unboxing):"+d);

int e=a.intValue();

System.out.println("Value of e from a using int e=a.intValue(); (Without Unboxing):"+e);

c=20;

System.out.println("Changed value of c:"+c);

System.out.println("Changed value of c doesn't change the value of b (Due to autoboxing new c object is created):"+b);

d=7;

System.out.println("Changed value of d:"+d);

System.out.println("Changed value of d doesn't change the value of a (Due to unboxing):"+a);

}

}

***P15.Created a multi-file program where in one file, a string message is taken as input from the user and the function to display the message on the screen is given on another file.***

package p15;

public class P15

{

public static void main(String[] args)

{

System.out.println("Created a multi-file program where in one file, a string message is taken as input from the user and the function to display the message on the screen is given on another file.");

Input cobj=new Input();

Output ccobj=new Output();

String str=cobj.enterinput();

ccobj.Disp(str);

}

}

package p15;

import java.util.Scanner;

public class Input

{

protected String enterinput()

{

Scanner inp=new Scanner(System.in);

System.out.println("Enter a line please:");

String str=inp.nextLine();

return str;

}

}

package p15;

public class Output

{

protected void Disp(String str)

{

System.out.println(str);

}

}

**P16.A program to create a multilevel package and also creates a reusable class to generate Fibonacci series, where the function to generate Fibonacci Series is given in a different file belonging to the same package.**

package p16;

import java.util.Scanner;

import Fib.Fibonacci;

public class P16

{

public static void main(String[] args)

{

Fibonacci f=new Fibonacci();

System.out.println("A program to create a multilevel package and also creates a reusable class to generate Fibonacci series, where the function to generate Fibonacci Series is given in a different file belonging to the same package.");

System.out.print("Enter a number to compute Fibonacci series till that number of terms:");

Scanner inp=new Scanner(System.in);

int n=inp.nextInt();

System.out.print("Fibonacci Series till "+n+"th number of terms is:");

if(n==0)

System.out.println("0");

if(n==1)

System.out.println("0 1");

if(n==2)

System.out.println("0 1 1");

if(n>2)

f.generate\_Fibonacci(n);

if(n<=0)

System.out.print("Invalid Entry!");

}

}

package Fib;

public class Fibonacci

{

public void generate\_Fibonacci(int n)

{

int f0=0,f1=1,s;

System.out.print("0 1 ");

for(int i=0;i<=n-2;i++)

{

s=f0+f1;

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

f0=f1;

f1=s;

}

}

}

**P17.Program to show the difference between public and private access specifiers. The program should also show that the primitive data types are passed by value and objects are passed by reference and to learn the use of final keyword.**

package p17;

import DifferentPackageClass.\*;

import differentpackageclass.DifferentPackageClass;

import java.util.Scanner;

public class P17

{

public static void main(String[] args)

{

System.out.println("Program to show the difference between public and private access specifiers. The program should also show that the primitive data types are passed by value and objects are passed by reference and to learn the use of final keyword.");

Scanner inp=new Scanner(System.in);

System.out.println("Enter your choice:\n1. Public function of this class.\n2. Private function of this class.\n3. Protected function of this class.\n4. Default function of this class.\n5. Public function of New class of same package.\n6. Private function of New class of same package.\n7. Protected function of New class of same package.\n8. Default function of New class of same package.\n9.Public function of DifferentPackageClass class of different package.\n10.Private function of DifferentPackageClass class of different package.\n11.Protected function of DifferentPackageClass class of different package.\n12.Default function of DifferentPackageClass class of different package.");

int n=inp.nextInt();

New cobj1=new New();

DifferentPackageClass cobj2=new DifferentPackageClass();

switch(n)

{

case 1:pubfunct();break;

case 2:prifunct();break;

case 3:profunct();break;

case 4:defaultfunct();break;

case 5:cobj1.pubfunct();break;

case 6:System.out.println("Cannot access a private function of another class in the same package.");break;

case 7:cobj1.profunct();break;

case 8:cobj1.defaultfunct();break;

case 9:cobj2.pubfunct();break;

case 10:System.out.println("Cannot access a private function of another class in the different package.");break;

case 11:System.out.println("Cannot access a protected function of another class in the different package.");break;

case 12:System.out.println("Cannot access a default function of another class in the different package.");break;

}

}

public static void pubfunct()

{

System.out.println("Public function of this class.");

}

private static void prifunct()

{

System.out.println("Private function of this class.");

}

protected static void profunct()

{

System.out.println("Protected function of this class.");

}

static void defaultfunct()

{

System.out.println("Default function of this class.");

}

}

class New

{

public static void pubfunct()

{

System.out.println("Public function of New class of same package.");

}

private static void prifunct()

{

System.out.println("Private function of New class of same package.");

}

protected static void profunct()

{

System.out.println("Protected function of New class of same package.");

}

static void defaultfunct()

{

System.out.println("Default function of New class of same package.");

}

}

package differentpackageclass;

public class DifferentPackageClass

{

public static void pubfunct()

{

System.out.println("Public function of DifferentPackageClass class of different package.");

}

private static void prifunct()

{

System.out.println("Private function of DifferentPackageClass class of different package.");

}

protected static void profunct()

{

System.out.println("Protected function of DifferentPackageClass class of different package.");

}

static void defaultfunct()

{

System.out.println("Default function of DifferentPackageClass class of different package.");

}

}

**P18.A Program 'DivideByZero' that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero.**

package p18;

import java.util.Scanner;

public class P18

{

public static void main(String[] args)

{

try

{

System.out.println("A Program 'DivideByZero' that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero.");

int a,b,c;

Scanner inp=new Scanner(System.in);

System.out.println("Enter the two numbers a and b respectively:");

a=inp.nextInt();

b=inp.nextInt();

c=a/b;

System.out.println("a/b="+c);

}

catch(ArithmeticException e)

{

System.out.println("Error:"+e);

}

}

}

P19.A Program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.

package p19;

public class P19

{

public static void main(String[] args)

{

System.out.println("A Program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.");

int ar[]={3,8,9,1,9};

System.out.println("5th element of the array is:"+ar[4]);

try

{

try

{

int result=ar[4]/0;

System.out.println("Resultof division:"+result);

}

catch(ArithmeticException e2)

{

e2.printStackTrace();

}

System.out.println("5th element of the array is:"+ar[5]);

}

catch(ArrayIndexOutOfBoundsException e1)//Exception keyword is used if we don't know which exception to use.

{

e1.printStackTrace();

}

System.out.println("Hello World");

}

}

P20. A Program to create your own exception types to handle situation specific to your application(Hint: Define a subclass of which itself is a subclass of throwable).

package p20;

import java.util.Scanner;

public class P20

{

public static void main(String[] args)

{

System.out.println("A Program to create your own exception types to handle situation specific to your application(Hint: Define a subclass of which itself is a subclass of throwable).");

System.out.print("Enter a postive integer:");

try//block

{

Scanner inp=new Scanner(System.in);

int n=inp.nextInt();

if(n<0) throw new Myexception("Negative Number!");

if(n==0) throw new Myexception1("Zero Entered!");

}

catch(Myexception | Myexception1 e)//function // 1try mcatch, mtry 1catch, mtry mcatch

{

e.printStackTrace();

}

/\* or

catch(Myexception1 e1)

{

e1.printStackTrace();

}\*/

}

}

class Myexception extends Exception//Exception class doesn't includes throw by itself. It extends throwable class to include throw.

{

String str;

Myexception(String str)

{

this.str=str;//this is used when we refer to the global not local.

}

public String toString()//We are overwriting the toString function of the Exception class in our program by defining it again.

{

return("User defined exception:"+str);

}

}

class Myexception1 extends Exception

{

String str1;

Myexception1(String str1)

{

this.str1=str1;

}

public String toString()

{

return("User defined exception 1:"+str1);

}

}