Program 1. Write a java program for Creation and Casting of Variables.

**Code:**

import myPack.Intro;

import static java.lang.System.\* ;

public class Casting1{

public static void main(String args[]){

Intro.print("Type casting") ;

//implicit castings

byte byt = 2 ;

short srt = byt ;

int nt = srt;

float flt = nt ;

double dbl = flt ;

dbl = 25434.545345 ;

// explicit casting

flt = (float)dbl ;

nt = (int)flt ;

srt = (short)nt ;

byt = (byte)srt ;

out.println("flt : "+flt);

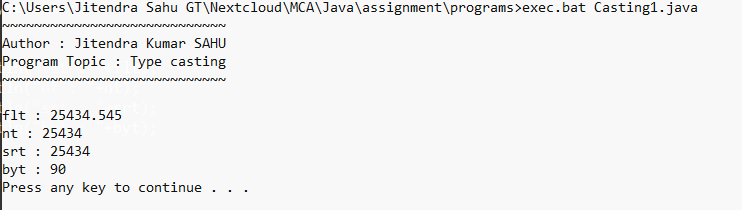
out.println("nt : "+nt);

out.println("srt : "+srt);

out.println("byt : "+byt);

}

}

**Output:**

Program 2. Write a java program to demonstrate the various Operators.

**Code:**

import myPack.Intro;

import static java.lang.System.\* ;

public class Operator{

public static void main(String args[]){

Intro.print("Different kind of operators in JAVA") ;

// Arithmetic Operators

out.println("Arithmetic Operators") ;

int a = 5 , b = 6 ;

out.println(a + " + " + b +" = " + (a+b)) ;

out.println(a + " - " + b +" = " + (a-b)) ;

out.println(a + " \* " + b +" = " + (a\*b)) ;

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

out.println(a + " % " + b +" = " + (a%b)) ;

// Comparison Operators

out.println("\nComparison Operators") ;

out.println(a + " = " + b +" = " + (a=b)) ;

out.println(a + " != " + b +" = " + (a!=b)) ;

out.println(a + "<" + b +" = " + (a<b)) ;

out.println(a + ">" + b +" = " + (a>b)) ;

out.println(a + "<= " + b +" = " + (a<=b)) ;

out.println(a + ">= " + b +" = " + (a>=b)) ;

// Bitwise operators

out.println("\nBitwise Operators") ;

out.println(a + "&" + b +" = " + (a&b)) ;

out.println(a + " | " + b +" = " + (a|b)) ;

out.println(a + " ^ " + b +" = " + (a^b)) ;

out.println(a + "<<" + b +" = " + (a<<b)) ;

out.println(a + ">>" + b +" = " + (a>>b)) ;

out.println(" ~ " + b +" = " + (~b)) ;

// Logical operators

int d = 7 , e = 8 ;

out.println("\nLogical Operators") ;

out.println(a + " == " + b +"&&" + d + " == " + e + " : " +( a==b && d==e)) ;

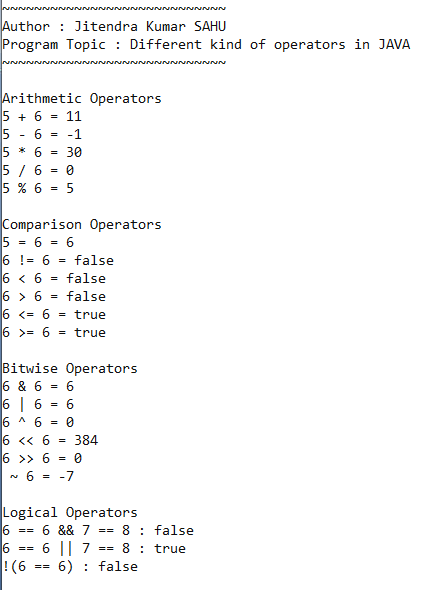
out.println(a + " == " + b +" || " + d + " == " + e + " : " +( a==b || d==e)) ;

out.println("!("+a + " == " + b +") : " + !( a==b)) ;

}

}

**Output:**



Program 3. Write a java program for printing the current date in different formats.

**Code:**

import myPack.Intro;

import static java.lang.System.\* ;

import java.text.SimpleDateFormat ;

import java.util.Date ;

public class CurrentDateInVariousFormat{

public static void main(String args[]){

Intro.print("Printing current date in defferent formates") ;

Date currentDate = new Date() ;

SimpleDateFormat f1 = new SimpleDateFormat("dd/MM/yyyy") ;

SimpleDateFormat f2 = new SimpleDateFormat("dd-MMM-yyyy") ;

SimpleDateFormat f3 = new SimpleDateFormat("MMM dd yyyy") ;

System.out.println("Current date in format1 : "+f3.format(currentDate)) ;

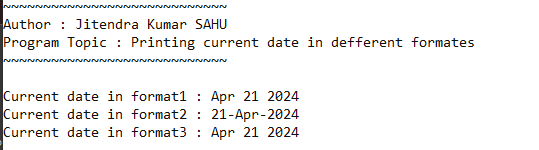
System.out.println("Current date in format2 : "+f2.format(currentDate)) ;

System.out.println("Current date in format3 : "+f3.format(currentDate)) ;

}

}

**Output:**



Program 4. Write a java program for Inputting Data From Keyboard through Scanner Class.

**Code:**

import myPack.Intro;

import static java.lang.System.\* ;

import java.util.Scanner ;

public class InputWithScanner{

public static void main(String args[]){

Intro.print("Input using Scanner") ;

// creating object

Scanner sc = new Scanner(System.in) ;

float a ;

String s ;

// input number

out.print("Enter number: ") ;

a = sc.nextFloat() ;

sc.nextLine() ; // to escap line

out.print("Enter string: ") ;

s = sc.nextLine() ;

sc.close() ;

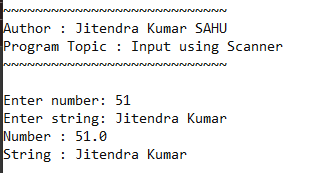
out.println("Number : " + a ) ;

out.println("String : " + s ) ;

}

}

**Output:**



Program 5. Write a java program for Inputting Data From Keyboard through BufferedReader Class.

**Code:**

import myPack.Intro;

import static java.lang.System.\* ;

import java.io.BufferedReader ;

import java.io.InputStreamReader ;

import java.io.IOException;

public class BufferedInputReaderExample{

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

Intro.print("") ;

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

out.print("Enter a number : ") ;

float flt = Float.parseFloat(br.readLine()) ;

out.print("Enter string : ") ;

String s = br.readLine() ;

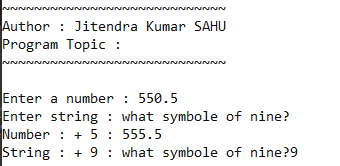
out.println("Number : + 5 : "+(flt+5)) ;

out.println("String : + 9 : "+s+9) ;

}

}

**Output:**



Program 6. Write a java program for Inputting Data From Keyboard through Console Class.

**Code:**

import myPack.Intro;

import static java.lang.System.\* ;

import java.io.\* ;

public class InputWithConsoleClass{

public static void main(String args[]){

Intro.print("Input output using console") ;

Console cl = System.console() ;

if(cl == null) {

out.println("console not found") ;

return ;

}

// taking input from console

cl.printf("Enter a number : ") ;

float number = Float.parseFloat(cl.readLine()) ;

cl.printf("Enter a string : ") ;

String s = cl.readLine() ;

cl.printf("Enter you password : ");

char[] pass = cl.readPassword() ;

cl.printf("\number + 5 : %f",number) ;

s = s+5 ;

cl.printf("\nstring + 5: %s",s) ;

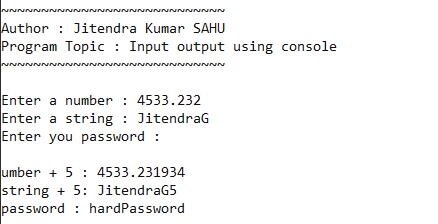
cl.printf("\npassword : ");

out.println(pass) ;

}

}

**Output:**



Program 7. Write a java program to demonstrate the use of for–each loop.

**Code:**

import myPack.Intro;

import static java.lang.System.\* ;

public class ForEachLoop{

public static void main(String args[]){

Intro.print("Demostration of for each loop") ;

int arr[] = {12,34,56,7,87,98,9,45,43,3} ;

out.println("Items of array are : ") ;

//using for each loop to iterate over array elements and printing values

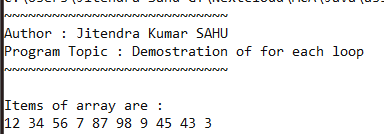
for (int i : arr) out.print(i+"") ;

out.println();

}

}

**Output:**



Program 8. Write a java program to demonstrate ragged arrays.

**Code:**

import myPack.Intro;

import static java.lang.System.\* ;

public class RaggedArrayInJava{

public static void main(String args[]){

Intro.print("Program to demonstrate ragged array") ;

int raggedArray[][] = {

{34,5,5,3,5},

{23,6,87,2},

{23,87,34},

{34,3},

{32,5,769,98,0,067}

};

// printing ragged array

out.println("printing ragged array ") ;

for (int i = 0 ; i< 5 ; i++ ) {

for(int j : raggedArray[i] ) out.print(j+"") ;

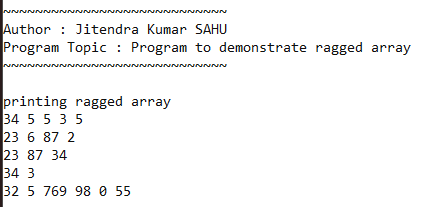
out.println() ;

}

}

}

**Output:**



Program 9. Write a java program to demonstrate anonymous arrays.

**Code:**

import myPack.Intro;

import static java.lang.System.\* ;

public class AnonymousArray{

static void printArray(int arr[]){

for (int i : arr) out.print(i+"") ;

}

static float getAvg(int arr[]){

int sum = 0 ;

for (int i : arr) sum += i ;

return sum / arr.length ;

}

public static void main(String args[]){

Intro.print("Program to Anonymous array") ;

out.println("Array elements : ") ;

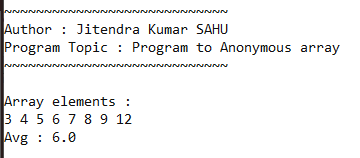
printArray(new int[] {3,4,5,6,7,8,9,12}) ;

out.println("\nAvg : " + getAvg(new int[] {3,4,5,6,7,8,9,12})) ;

}

}

**Output:**



Program 10. Write a java program to demonstrate the methods of Arrays Class.

**Code:**

import myPack.Intro;

import java.util.Arrays ;

public class MethodsOfArraysClass{

public static void main(String args[]){

Intro.print("Methods of Arrays Class");

// Demonstrating some methods of Arrays class

int[] numbers = {5, 3, 8, 2, 9};

Arrays.sort(numbers);

System.out.println("Sorted array: " + Arrays.toString(numbers));

System.out.println("Index of 8: " + Arrays.binarySearch(numbers, 8));

// Additional functions

int[] copiedArray = Arrays.copyOf(numbers, 3);

System.out.println("Copied array: " + Arrays.toString(copiedArray));

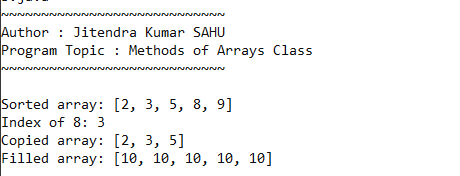
int[] filledArray = new int[5];

Arrays.fill(filledArray, 10);

System.out.println("Filled array: " + Arrays.toString(filledArray));

}

}

**Output:**

Program 11. Write a java program for Application Of Classes And Objects.

**Code:**

import myPack.Intro ;

class Person{

String name ;

int age;

Person(String name, int age){

this.name = name ;

this.age = age ;

}

void show(){

System.out.println("Person name : "+name) ;

System.out.println("Person age : "+age) ;

}

void eat(){

System.out.println(name + " is eating rice and curry");

}

}

public class ClassesAndObject{

public static void main(String args[]){

Intro.print("Demonstration of classes and object");

Person a = new Person("Jitendra Kumar", 22) ;

Person b = new Person("Purusottam", 21) ;

a.show();

a.eat();

System.out.println();

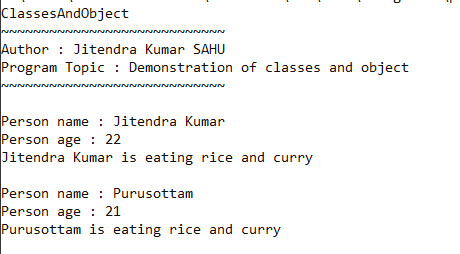
b.show();

b.eat() ;

}

}

**Output:**



Program 12. Write a java program to demonstrate method overloading.

**Code:**

import myPack.Intro;

class Adder {

void printSum(int a, int b){

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

}

void printSum(int a, int b, int c){

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

}

void printSum(float a, float b){

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

}

}

public class MethodOverloading{

public static void main(String args[]){

Intro.print("Demonstration of method overloading") ;

Adder a = new Adder() ;

a.printSum(3, 5);

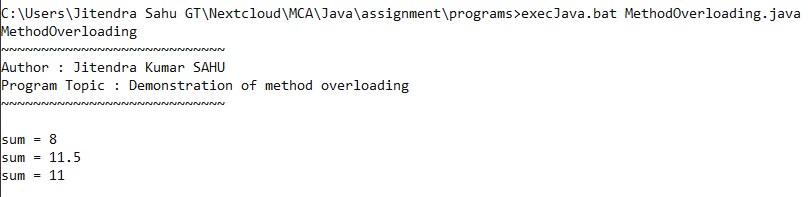
a.printSum(5.2f, 6.3f);

a.printSum(2, 3,6);

}

}

**Output:**



Program 13. Write a java program to demonstrate constructor overloading.

**Code:**

import myPack.Intro;

class DemoConstOverloading {

// Constructor overloading

DemoConstOverloading() {

System.out.println("Default constructor");

}

DemoConstOverloading(int x) {

System.out.println("Parameterized constructor with one parameter: " + x);

}

DemoConstOverloading(int x, String str) {

System.out.println("Parameterized constructor with two parameters: " + x + ", " + str);

}

}

public class ConstructorOverloading {

public static void main(String[] args) {

Intro.print("Constructor Overloading demonstration");

DemoConstOverloading obj1 = new DemoConstOverloading();

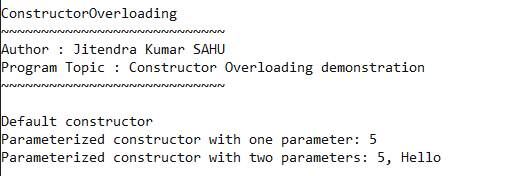
DemoConstOverloading obj2 = new DemoConstOverloading(5);

DemoConstOverloading obj3 = new DemoConstOverloading(5, "Hello");

}

}

**Output:**



Program 14. Write a java program Using Single Inheritance.

**Code:**

import myPack.Intro;

class ParentClass {

void displayParent() {

System.out.println("Parent class method");

}

}

class ChildClass extends ParentClass {

void displayChild() {

System.out.println("Child class method");

}

}

public class SingleInheritance {

public static void main(String[] args) {

Intro.print("Single Inheritance");

// Single inheritance demonstration

ChildClass obj = new ChildClass();

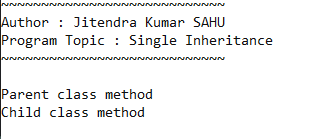
obj.displayParent();

obj.displayChild();

}

}

**Output:**



Program 15. Write a java program Using Super And This Keyword.

**Code:**

import myPack.Intro;

class ParentClass {

void display() {

System.out.println("Parent class method");

}

}

class ChildClass extends ParentClass {

void display() {

super.display(); // Calls the parent class method

System.out.println("Child class method");

}

}

public class SuperAndThisKeyword {

public static void main(String[] args) {

Intro.print("Super and This Keyword demonstration");

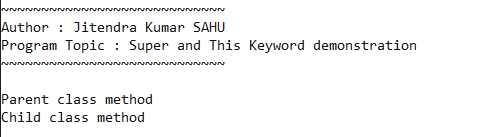
ChildClass obj = new ChildClass();

obj.display();

}

}

**Output:**



Program 16. Write a java program to demonstrate multilevel inheritance.

**Code:**

import myPack.Intro;

class GrandParentClass {

void displayGrandParent() {

System.out.println("GrandParent class method");

}

}

class ParentClass extends GrandParentClass {

void displayParent() {

System.out.println("Parent class method");

}

}

class ChildClass extends ParentClass {

void displayChild() {

System.out.println("Child class method");

}

}

public class MultilevelInheritance {

public static void main(String[] args) {

Intro.print("Multilevel Inheritance");

// Multilevel inheritance demonstration

ChildClass obj = new ChildClass();

obj.displayGrandParent();

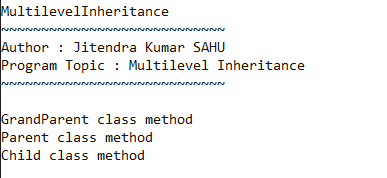
obj.displayParent();

obj.displayChild();

}

}

**Output:**



Program 17. Write a java program to demonstrate method overriding.

**Code:**

import myPack.Intro;

class ParentClass {

void func1(){

System.out.println("func1 one from parent");

}

void display() {

System.out.println("Parent class method");

}

}

class ChildClass extends ParentClass {

@Override

void display() {

System.out.println("Child class method");

}

}

class MethodOverriding {

public static void main(String[] args) {

Intro.print("Method Overriding");

// Method overriding demonstration

ChildClass obj = new ChildClass();

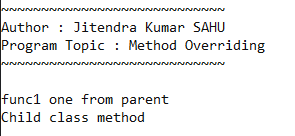
obj.func1();

obj.display();

}

}

**Output:**



Program 18. Write a java program Using Multiple Inheritance Concept through interfaces.

**Code:**

import myPack.Intro;

interface Interface1 {

void method1();

}

interface Interface2 {

void method2();

}

class TestClass implements Interface1, Interface2 {

public void method1() {

System.out.println("Method1 implementation");

}

public void method2() {

System.out.println("Method2 implementation");

}

}

public class MultipleInhritWithInf {

public static void main(String[] args) {

Intro.print("Multiple Inheritance through Interfaces");

TestClass obj = new TestClass();

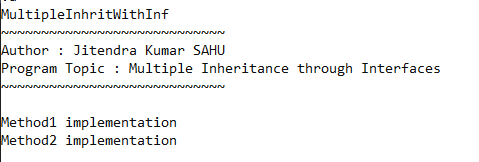
obj.method1();

obj.method2();

}

}

**Output:**



Program 19. Write a java program to demonstrate the concept of inner class.

**Code:**

import myPack.Intro;

public class InnerClassDemo {

public static void main(String[] args) {

Intro.print("Concept of Inner Class");

// Creating object of Outer class

Outer outer = new Outer();

// Accessing inner class method

outer.display();

}

static class Outer {

void display() {

System.out.println("Inside Outer class method");

// Inner class definition

class Inner {

void innerMethod() {

System.out.println("Inside Inner class method");

}

}

// Creating object of Inner class

Inner inner = new Inner();

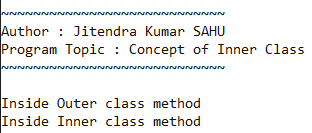
inner.innerMethod();

}

}

}

**Output:**



Program 20. Write a java program to demonstrate the concept of local class.

**Code:**

import myPack.Intro;

public class LocalClassDemo {

public static void main(String[] args) {

Intro.print("Concept of Local Class");

// Calling method with local class

displayMessage();

}

static void displayMessage() {

System.out.println("Inside method");

// Local class definition

class Local {

void localMethod() {

System.out.println("Inside local method");

}

}

// Creating object of Local class

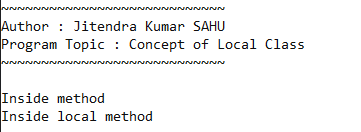
Local local = new Local();

local.localMethod();

}

}

**Output:**



Program 21. Write a java program that creates its own package containing two classes.

**Code:**

***Animal.java***

package AnimalPack ;

public class Animal{

String name , breed ;

public void setName(String name) {

this.name = name ;

}

public void setBreed(String breed) {

this.breed = breed ;

}

}

***Dog.java***

package AnimalPack ;

public class Dog extends Animal{

public Dog(String name , String breed){

setName(name) ;

setBreed(breed) ;

}

public void print(){

System.out.println("Dog properties : ") ;

System.out.println("name : " +name) ;

System.out.println("breed : "+breed) ;

}

}

***Cat.java***

package AnimalPack ;

public class Cat extends Animal{

public Cat(String name , String breed){

setName(name) ;

setBreed(breed) ;

}

public void print(){

System.out.println("Cat properties : ") ;

System.out.println("name : " +name) ;

System.out.println("breed : "+breed) ;

}

}

***TestAnimal.java***

import myPack.Intro;

import AnimalPack.Dog ;

import AnimalPack.Cat ; // imported two classes from AnimalPack

public class TestAnimals{

public static void main(String arg[]){

Intro.print("Package having two Classes") ;

Dog dg = new Dog("Diggu", "Pug") ;

dg.print() ;

System.out.println() ;

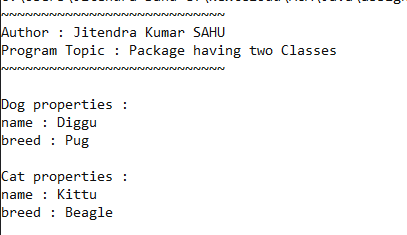
Cat ct = new Cat("Kittu", "Beagle") ;

ct.print() ;

}

}

**Output:**



Program 22. Write a java program Using Try And Catch Statement.

**Code:**

import myPack.Intro;

public class TryAndCatch{

public static void main(String args[]){

Intro.print("Try catch statement") ;

int arr[] = {3,4,5,7,8} ;

try{

for (int i = 0; i <= 5; i++) {

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

}

}catch (IndexOutOfBoundsException e){

System.err.println("\nException caught : ") ;

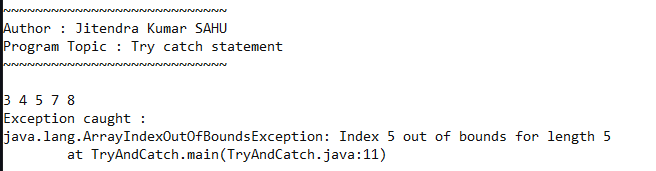
e.printStackTrace() ;

}

}

}

**Output:**

****

Program 23. Write a java program Using Multiple Catch Statements.

**Code:**

import myPack.Intro;

public class MultpleCatch {

public static void main(String args[]) {

Intro.print("Multiple catch block in try catch statement");

int arr[] = { 3, 4, 5, 7, 8 };

float c = 0;

try {

c = arr[1] / (arr.length - 5);

System.err.println("c= " + c);

for (int i = 0; i <= 5; i++) {

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

}

} catch (IndexOutOfBoundsException e) {

System.err.println("\nException caught : ");

e.printStackTrace();

} catch (ArithmeticException ae) {

System.err.println("\nException caught : ");

ae.printStackTrace();

} catch (Exception e) {

System.err.println("\nException caught : ");

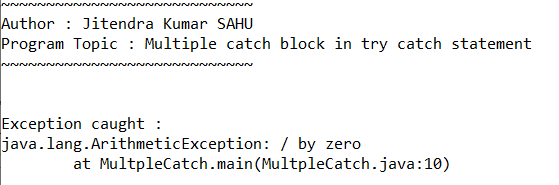
e.printStackTrace();

}

}

}

**Output:**

****

Program 24. Write a java program to demonstrate the MultiCatch feature.

**Code:**

import myPack.Intro;

public class MultipleCatchFeature {

public static void main(String args[]) {

Intro.print("Multiple catch block in try catch statement");

int arr[] = { 3, 4, 5, 7, 8 };

float c = 0;

try {

c = arr[1] / (arr.length - 5);

System.err.println("c= " + c);

for (int i = 0; i <= 5; i++) {

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

}

} catch (IndexOutOfBoundsException | ArithmeticException e) {

e.printStackTrace();

} catch (Exception e) {

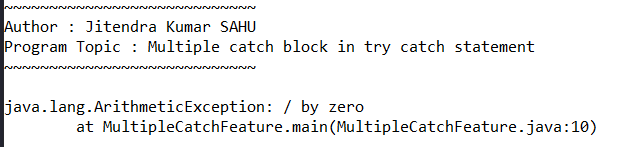
e.printStackTrace();

}

}

}

**Output:**

****

Program 25. Write a java program to demonstrate the use of finally block.

**Code:**

import myPack.Intro;

public class FinallyTryCatch {

public static void main(String args[]) {

Intro.print("Finally block in try catch statement");

int arr[] = { 3, 4, 5, 7, 8 };

float c = 0;

try {

c = arr[1] / (arr.length - 5);

System.err.println("c= " + c);

for (int i = 0; i <= 5; i++) {

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

}

} catch (IndexOutOfBoundsException e) {

System.err.println("\nException caught : ");

e.printStackTrace();

} catch (ArithmeticException ae) {

System.err.println("\nException caught : ");

ae.printStackTrace();

} catch (Exception e) {

System.err.println("\nException caught : ");

e.printStackTrace();

}finally{

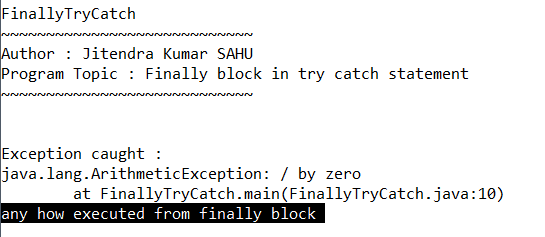
System.out.println("any how executed from finally block");

}

}

}

**Output:**

****

Program 26. Write a java program Using Nested Try Statements.

**Code:**

import myPack.Intro;

public class NestedTryAndCatch {

public static void main(String args[]) {

Intro.print("Nested Try catch statement");

int arr[] = { 3, 4, 5, 7, 8 };

try {

for (int i = 0; i < 5; i++) {

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

}

try {

float div = arr[0] / (9 - 5 - 4);

System.out.println("div : " + div);

} catch (ArithmeticException e) {

e.printStackTrace();

}

} catch (IndexOutOfBoundsException e) {

System.err.println("\nException caught : ");

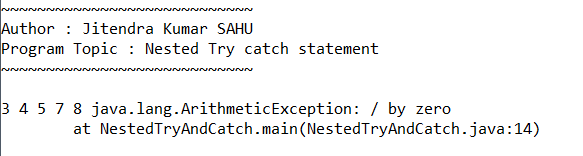
e.printStackTrace();

}

}

}

**Output:**



Program 27. Write a java program To Create Your Own Exception Class And Display Corresponding Error Message.

**Code:**

import myPack.Intro;

class myOwnException extends Exception {

myOwnException(){

super("your own exception occurred >\_<");

}

}

public class MakingOwnException{

public static void main(String[] args) {

Intro.print("Creating my own exception");

try{

throw new myOwnException() ;

}catch(myOwnException e){

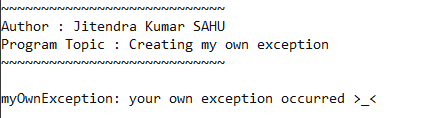
System.err.println(e);

}

}

}

**Output:**



Program 28. Write a java program For Creating And Executing Threads by extending the Thread class.

**Code:**

class ConcurrentFun extends Thread {

char c;

ConcurrentFun(char c) {

this.c = c;

}

public void run() {

for (int i = 0; i < 10; i++) {

System.out.println(c + " = " + i);

}

System.out.println("Finished " + c);

}

}

public class MultiThreading {

public static void main(String args[]) {

ConcurrentFun f1 = new ConcurrentFun('i');

ConcurrentFun f2 = new ConcurrentFun('k');

f1.start();

f2.start();

for (int i = 0; i < 10; i++) {

System.out.println("m = " + i);

}

System.out.println("Finished Main");

}

}

**Output:**



Program 29. Write a java program To run Three Threads by implementing the Runnable Interface.

**Code:**

import myPack.Intro;

class Counter implements Runnable {

char ch;

Counter(char c) {

ch = c;

}

public void run() {

for (int i = 1; i <= 5; i++) {

System.out.printf("%c%d\n",ch,i);

}

System.out.println("Executed thread "+ch);

}

}

public class MultithreadRunnable {

public static void main(String[] args) {

Intro.print("multithreading with runnable");

Thread ct1 = new Thread(new Counter('a'));

Thread ct2 = new Thread(new Counter('b'));

Thread ct3 = new Thread(new Counter('c'));

ct1.start();

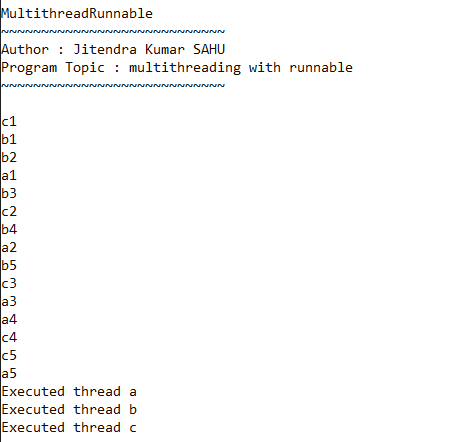
ct2.start();

ct3.start();

}

}

**Output:**



Program 30. Write a java program to demonstrate the use of join() method.

**Code:**

import myPack.Intro;

class Counter implements Runnable {

char ch;

Counter(char c) {

ch = c;

}

public void run() {

for (int i = 1; i <= 5; i++) {

System.out.printf("%c%d\n", ch, i);

}

System.out.println("Executed thread " + ch);

}

}

class AnotherThread extends Thread {

public void run() {

for (int i = 0; i < 5; i++) {

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

System.out.println("i = " + i);

}

System.out.println("Executed AN THREAD");

}

}

public class JoinThread {

public static void main(String[] args) {

Intro.print("Join function in multithreading");

Thread ct1 = new Thread(new Counter('a'));

Thread ct2 = new Thread(new Counter('b'));

AnotherThread AN = new AnotherThread();

AN.start();

try {

AN.join(2100); // thread name an will continue

// it's execution for 2100 ms then only another thread will get chance

} catch (InterruptedException e) {

System.out.println(e);

}

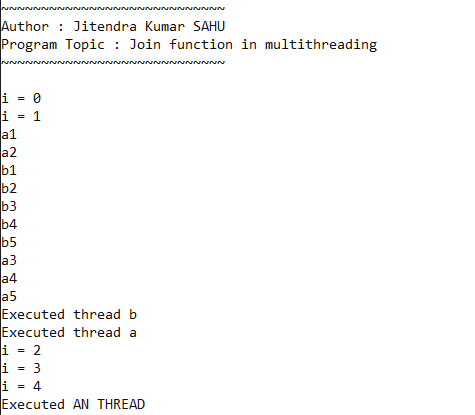
ct1.start();

ct2.start();

}

}

**Output:**



Program 31. Write a java program to demonstrate Multithreading using wait () & notify().

**Code:**

import myPack.\*;

public class WaitAndNotify {

public static void main(String[] args) {

new Intro("Wait and notify method") ;

SharedData shared = new SharedData();

Thread proThread = new Thread(() -> {

try {

shared.produce("how are you");

} catch (InterruptedException e) {

e.printStackTrace();

}

});

Thread conThread = new Thread(() -> {

try {

shared.consume();

} catch (InterruptedException e) {

e.printStackTrace();

}

});

proThread.start();

conThread.start();

}

}

class SharedData {

private String msg;

private boolean isProduced = false;

public synchronized void produce(String msg) throws InterruptedException {

while (isProduced) {

wait();

}

this.msg = msg;

System.out.println("Produced : " + msg);

isProduced = true;

notify();

}

public synchronized void consume() throws InterruptedException {

while (!isProduced) {

wait();

}

System.out.println("consumed : " + msg);

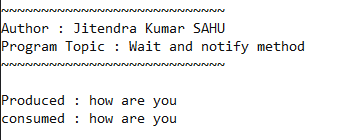
isProduced = !isProduced;

notify();

}

}

**Output:**

****

Program 32. Write a java program to demonstrate The String Class & its methods.

**Code:**

import myPack.Intro;

public class StringClassAndMethod {

public static void main(String[] args) {

Intro.print("The String Class & its methods");

String str = "Hello, World!";

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

// Length of the string

System.out.println("Length: " + str.length());

// Character at a specific index

System.out.println("Character at index 7: " + str.charAt(7));

// Substring

System.out.println("Substring (7, 12): " + str.substring(7, 12));

// Replace

String replacedStr = str.replace("World", "Java");

System.out.println("Replaced String: " + replacedStr);

System.out.println("Uppercase: " + str.toUpperCase());

System.out.println("Lowercase: " + str.toLowerCase());

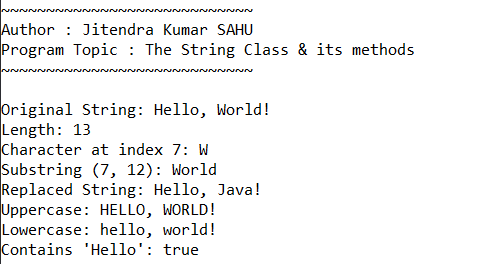
// Check if string contains a sequence

System.out.println("Contains 'Hello': " + str.contains("Hello"));

}

}

**Output:**

****

Program 33. Write a java program to demonstrate StringBuffer Class & its methods.

**Code:**

import myPack.Intro;

import java.util.Scanner;

public class StringBufferDemo {

public static void main(String[] args) {

Intro.print("StringBuffer Class & its methods");

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a string: ");

String initialString = scanner.nextLine();

StringBuffer sb = new StringBuffer(initialString);

System.out.println("Original StringBuffer: " + sb);

// Append

System.out.print("Enter a string to append: ");

String appendString = scanner.nextLine();

sb.append(appendString);

System.out.println("After append: " + sb);

// Insert

System.out.print("Enter the position to insert: ");

int insertPosition = scanner.nextInt();

scanner.nextLine(); // Consume newline

System.out.print("Enter a string to insert: ");

String insertString = scanner.nextLine();

if (insertPosition >= 0 && insertPosition <= sb.length()) {

sb.insert(insertPosition, insertString);

System.out.println("After insert: " + sb);

} else {

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

}

// Replace

System.out.print("Enter start index for replace: ");

int replaceStart = scanner.nextInt();

System.out.print("Enter end index for replace: ");

int replaceEnd = scanner.nextInt();

scanner.nextLine(); // Consume newline

System.out.print("Enter a string for replacement: ");

String replaceString = scanner.nextLine();

if (replaceStart >= 0 && replaceEnd <= sb.length() && replaceStart < replaceEnd) {

sb.replace(replaceStart, replaceEnd, replaceString);

System.out.println("After replace: " + sb);

} else {

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

}

// Delete

System.out.print("Enter start index for delete: ");

int deleteStart = scanner.nextInt();

System.out.print("Enter end index for delete: ");

int deleteEnd = scanner.nextInt();

scanner.nextLine(); // Consume newline

if (deleteStart >= 0 && deleteEnd <= sb.length() && deleteStart < deleteEnd) {

sb.delete(deleteStart, deleteEnd);

System.out.println("After delete: " + sb);

} else {

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

}

// Reverse

sb.reverse();

System.out.println("After reverse: " + sb);

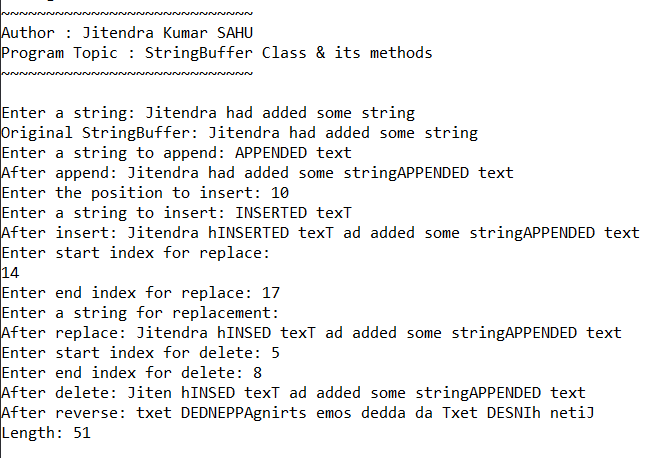
// Length

System.out.println("Length: " + sb.length());

}

}

**Output:**

****

Program 34. Write a java program to demonstrate various Wrapper Classes.

**Code:**

import myPack.Intro;

import java.util.Scanner;

public class WrapperClassesDemo {

public static void main(String[] args) {

Intro.print("Various Wrapper Classes");

Scanner scanner = new Scanner(System.in);

// Integer

System.out.print("Enter an integer: ");

int intInput = scanner.nextInt();

Integer intObj = Integer.valueOf(intInput);

System.out.println("Integer value: " + intObj);

// Double

System.out.print("Enter a double: ");

double doubleInput = scanner.nextDouble();

Double doubleObj = Double.valueOf(doubleInput);

System.out.println("Double value: " + doubleObj);

// Boolean

System.out.print("Enter a boolean (true/false): ");

boolean boolInput = scanner.nextBoolean();

Boolean boolObj = Boolean.valueOf(boolInput);

System.out.println("Boolean value: " + boolObj);

// Character

System.out.print("Enter a character: ");

char charInput = scanner.next().charAt(0);

Character charObj = Character.valueOf(charInput);

System.out.println("Character value: " + charObj);

// Auto-boxing

int primitiveInt = intObj;

double primitiveDouble = doubleObj;

boolean primitiveBool = boolObj;

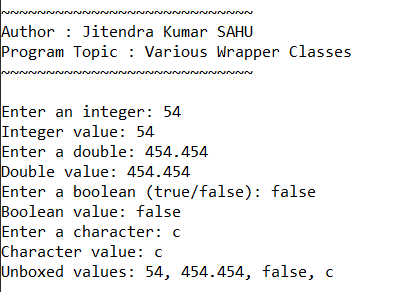
char primitiveChar = charObj;

System.out.println("Unboxed values: " + primitiveInt + ", " + primitiveDouble + ", " + primitiveBool + ", " + primitiveChar);

}

}

**Output:**

****

Program 35. Write a java program to demonstrate HashSet Class & its methods.

**Code:**

import myPack.Intro;

import java.util.HashSet;

import java.util.Scanner;

public class HashSetDemo {

public static void main(String[] args) {

Intro.print("HashSet Class & its methods");

HashSet<String> set = new HashSet<>();

Scanner scanner = new Scanner(System.in);

// Adding elements

System.out.println("Enter elements for the HashSet (type 'exit' to stop):");

while (true) {

String input = scanner.nextLine();

if (input.equalsIgnoreCase("exit")) {

break;

}

set.add(input);

}

System.out.println("HashSet: " + set);

// Check if set contains an element

System.out.print("Enter an element to check: ");

String elementToCheck = scanner.nextLine();

System.out.println("Contains '" + elementToCheck + "': " + set.contains(elementToCheck));

// Remove an element

System.out.print("Enter an element to remove: ");

String elementToRemove = scanner.nextLine();

set.remove(elementToRemove);

System.out.println("After removing '" + elementToRemove + "': " + set);

// Size of the set

System.out.println("Size of HashSet: " + set.size());

// Iterating over the elements

System.out.println("Iterating over HashSet:");

for (String item : set) {

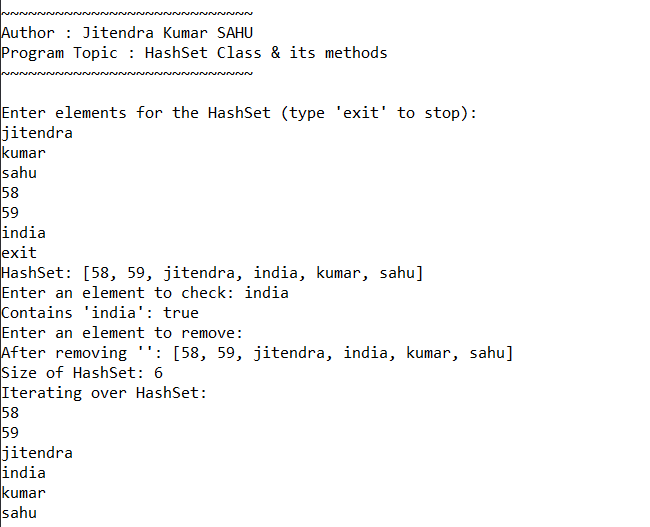
System.out.println(item);

}

}

}

**Output:**

****

Program 36. Write a java program to demonstrate ArrayList Class & its methods.

**Code:**

import myPack.Intro;

import java.util.ArrayList;

import java.util.Scanner;

public class ArrayListDemo {

public static void main(String[] args) {

Intro.print("ArrayList Class & its methods");

ArrayList<String> list = new ArrayList<>();

Scanner scanner = new Scanner(System.in);

// Adding elements

System.out.println("Enter elements for the ArrayList (type 'exit' to stop):");

while (true) {

String input = scanner.nextLine();

if (input.equalsIgnoreCase("exit")) {

break;

}

list.add(input);

}

System.out.println("ArrayList: " + list);

// Accessing elements

System.out.print("Enter an index to access: ");

int index = scanner.nextInt();

scanner.nextLine(); // Consume newline

if (index >= 0 && index < list.size()) {

System.out.println("Element at index " + index + ": " + list.get(index));

} else {

System.out.println("Index out of bounds");

}

// Removing an element

System.out.print("Enter an index to remove: ");

int removeIndex = scanner.nextInt();

scanner.nextLine(); // Consume newline

if (removeIndex >= 0 && removeIndex < list.size()) {

list.remove(removeIndex);

System.out.println("After removing element at index " + removeIndex + ": " + list);

} else {

System.out.println("Index out of bounds");

}

// Size of the list

System.out.println("Size of ArrayList: " + list.size());

// Iterating over the elements

System.out.println("Iterating over ArrayList:");

for (String item : list) {

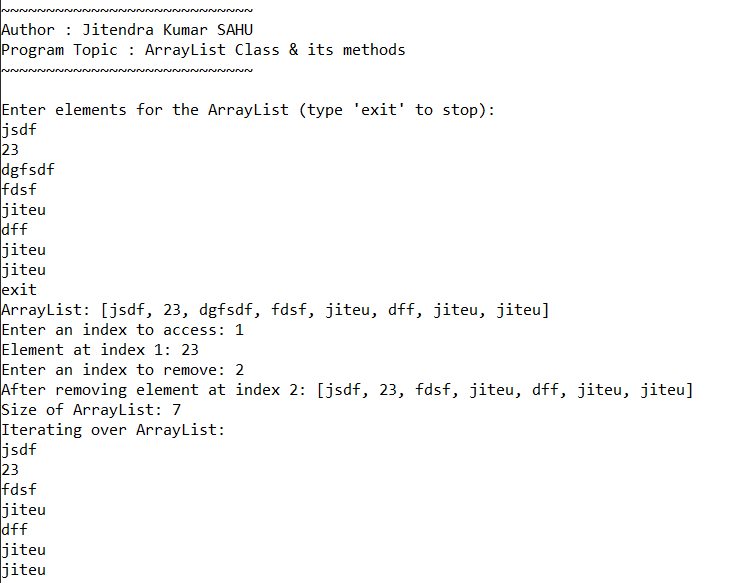
System.out.println(item);

}

}

}

**Output:**

****

Program 37. Write a java program to copy a File.

**Code:**

import myPack.Intro;

import java.io.\*;

import java.util.Scanner;

public class FileCopyDemo {

public static void main(String[] args) {

Intro.print("Copy a File");

String sourcePath = "myFile.txt" ;

String destinationPath = "myNewFile.txt" ;

File inputFile = new File(sourcePath);

File outputFile = new File(destinationPath);

try (FileInputStream fis = new FileInputStream(inputFile);

FileOutputStream fos = new FileOutputStream(outputFile)) {

byte[] buffer = new byte[1024];

int length;

while ((length = fis.read(buffer)) > 0) {

fos.write(buffer, 0, length);

}

System.out.println("File copied successfully.");

} catch (IOException e) {

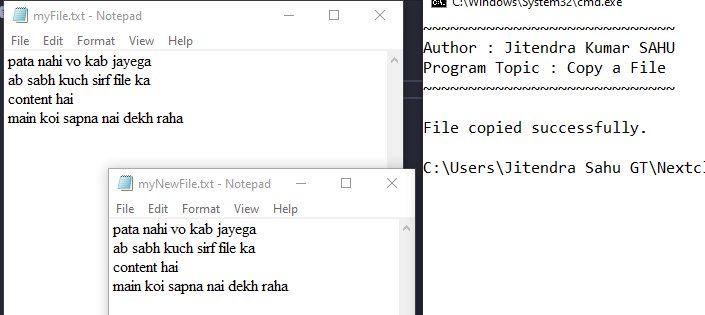
System.out.println("An error occurred: " + e.getMessage());

}

}

}

**Output:**

****

Program 38. Write a java program to Count the numbers of Characters in a File.

**Code:**

import myPack.Intro;

import java.io.\*;

public class CharacterCountFileDemo {

public static void main(String[] args) {

Intro.print("Count the numbers of Characters in a File");

String filePath ="myFile.txt";

File file = new File(filePath);

int charCount = 0;

try (FileReader fr = new FileReader(file)) {

int character;

while ((character = fr.read()) != -1) {

charCount++;

}

System.out.println("Number of characters in the file: " + charCount);

} catch (IOException e) {

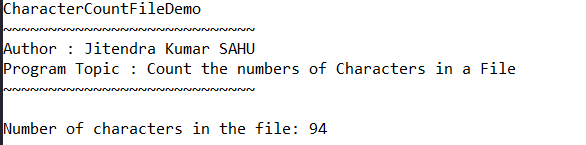
System.out.println("An error occurred: " + e.getMessage());

}

}

}

**Output:**

****

Program 39. Write a java program to demonstrate Object Serialization.

**Code:**

import myPack.Intro;

import java.io.\*;

class Person1 implements Serializable {

private static final long serialVersionUID = 1L;

String name;

int age;

Person1(String name, int age) {

this.name = name;

this.age = age;

}

}

public class ObjectSerializationDemo {

public static void main(String[] args) {

Intro.print("Object Serialization");

Person1 person = new Person1("John Doe", 30);

try (FileOutputStream fos = new FileOutputStream("person.ser");

ObjectOutputStream oos = new ObjectOutputStream(fos)) {

oos.writeObject(person);

System.out.println("Object has been serialized");

} catch (IOException e) {

System.out.println("An error occurred: " + e.getMessage());

}

try (FileInputStream fis = new FileInputStream("person.ser");

ObjectInputStream ois = new ObjectInputStream(fis)) {

Person1 deserializedPerson = (Person1) ois.readObject();

System.out.println("Object has been deserialized");

System.out.println("Name: " + deserializedPerson.name);

System.out.println("Age: " + deserializedPerson.age);

} catch (IOException | ClassNotFoundException e) {

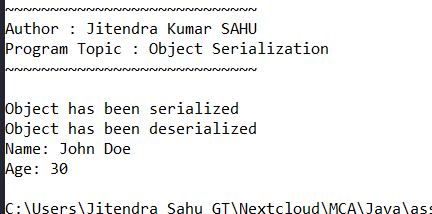
System.out.println("An error occurred: " + e.getMessage());

}

}

}

**Output:**

****

Program 40. Write a java program to demonstrate Keyboard Event.

**Code:**

import myPack.Intro;

import javax.swing.\*;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

public class KeyboardEventDemo extends JFrame implements KeyListener {

private JTextArea textArea;

public KeyboardEventDemo() {

Intro.print("Keyboard Event");

textArea = new JTextArea();

textArea.addKeyListener(this);

add(new JScrollPane(textArea));

setTitle("Keyboard Event Demo");

setSize(400, 300);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setVisible(true);

}

@Override

public void keyTyped(KeyEvent e) {

textArea.append("Key Typed: " + e.getKeyChar() + "\n");

}

@Override

public void keyPressed(KeyEvent e) {

textArea.append("Key Pressed: " + e.getKeyChar() + "\n");

}

@Override

public void keyReleased(KeyEvent e) {

textArea.append("Key Released: " + e.getKeyChar() + "\n");

}

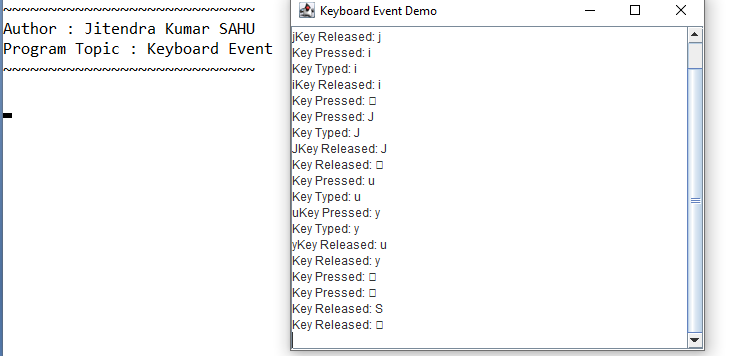
public static void main(String[] args) {

new KeyboardEventDemo();

}

}

**Output:**

****

Program 41. Write a java program to demonstrate Mouse Event.

**Code:**

import myPack.Intro;

import javax.swing.\*;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

public class MouseEventDemo extends JFrame implements MouseListener {

private JTextArea textArea;

public MouseEventDemo() {

Intro.print("Mouse Event");

textArea = new JTextArea();

textArea.addMouseListener(this);

add(new JScrollPane(textArea));

setTitle("Mouse Event Demo");

setSize(400, 300);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setVisible(true);

}

@Override

public void mouseClicked(MouseEvent e) {

textArea.append("Mouse Clicked: " + e.getPoint() + "\n");

}

@Override

public void mousePressed(MouseEvent e) {

textArea.append("Mouse Pressed: " + e.getPoint() + "\n");

}

@Override

public void mouseReleased(MouseEvent e) {

textArea.append("Mouse Released: " + e.getPoint() + "\n");

}

@Override

public void mouseEntered(MouseEvent e) {

textArea.append("Mouse Entered: " + e.getPoint() + "\n");

}

@Override

public void mouseExited(MouseEvent e) {

textArea.append("Mouse Exited: " + e.getPoint() + "\n");

}

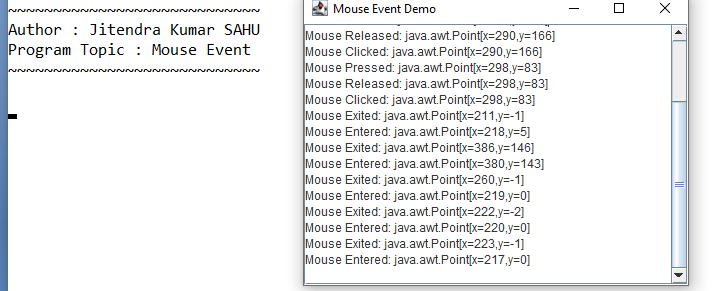
public static void main(String[] args) {

new MouseEventDemo();

}

}

**Output:**

****

Program 42. Write a java program to establish connection to the database.

**Code:**

import myPack.Intro;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class DatabaseConnectionDemo {

public static void main(String[] args) {

Intro.print("Establish connection to the database");

final String dbString = "jdbc:mysql://localhost:3306/test";

Connection con = null;

Statement stmt = null;

ResultSet rset = null;

try {

con = DriverManager.getConnection(dbString, "root", "");

System.out.println("Connection stablized");

stmt = con.createStatement();

String query = "select id , sname from student ";

rset = stmt.executeQuery(query);

while (rset.next()) {

System.out.println("id : " + rset.getInt("id") + " name : " + rset.getString("sname"));

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

try {

if (rset != null) rset.close();

if (stmt != null) stmt.close();

if (con != null) con.close();

} catch (SQLException e) {

e.printStackTrace();

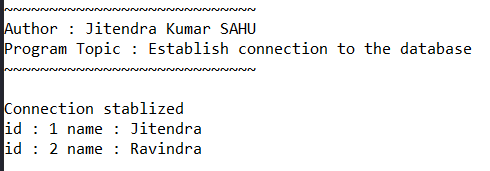
}

}

}

}

**Output:**



Program 43. Write a java program to create a table named employee with fields as emp\_id, emp\_name, age, dept.

**Code:**

import myPack.Intro ;

import java.sql.Statement ;

import java.sql.Connection ;

import java.sql.SQLException ;

import java.sql.DriverManager ;

public class CreateEmployeeTable {

public static void main(String[] args) {

new Intro("create employee table in db") ;

final String dbUrl = "jdbc:mysql://localhost/test" ;

final String username = "root" ;

final String password = "" ;

try(Connection con = DriverManager.getConnection(dbUrl,username,password) ){

Statement stmt = con.createStatement() ;

String sql = "create TABLE employee(emp\_id int(2), emp\_name varchar(15),age int(2), dept varchar(20))" ;

stmt.execute(sql) ;

System.out.println("Table created!\n") ;

}catch(SQLException e){

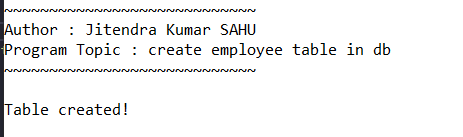
e.printStackTrace() ;

}

}

}

**Output:**

****

Program 44. Write a java program to create a table and drop it.

**Code:**

import myPack.Intro;

import java.sql.Statement;

import java.sql.Connection;

import java.sql.SQLException;

import java.sql.DriverManager;

import java.util.Scanner;

public class CreateAndDropTable {

public static void main(String[] args) {

new Intro("create and delete a table in db");

final String dbUrl = "jdbc:mysql://192.168.1.43/test";

final String username = "root";

final String password = "root";

Scanner sc = new Scanner(System.in);

try (Connection con = DriverManager.getConnection(dbUrl, username, password)) {

Statement stmt = con.createStatement();

String sql\_createTable = "create TABLE temp(id int(2), tname varchar(15))";

String sql\_dropTable = "drop TABLE temp";

stmt.execute(sql\_createTable);

System.out.println("Table created!\n");

System.out.println("Enter to delete table!\n");

sc.nextLine();

stmt.execute(sql\_dropTable);

System.out.println("Table Dropped!\n");

} catch (SQLException e) {

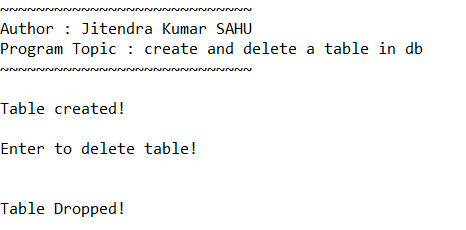
e.printStackTrace();

}

}

}

**Output:**

****

***Below DBconfig Class is used in following several classes that deals with database connection.***

***DBconfig.java***

package jkdatabase;

public class DBconfig {

static String db\_name = "test" ;

public static final String dbString = "jdbc:mysql://192.168.30.128:3306/"+db\_name;

public static final String username = "root";

public static final String password = "root";

}

Program 45. Write a java program to insert multiple rows in a table using prepared statement.

**Code:**

import myPack.Intro;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.sql.PreparedStatement;

import jkdatabase.DBconfig;

public class InsertMultipleWithPrepare {

public static void main(String[] args) {

Intro.print("Multiple Insert Using Prepare Statement");

Statement stmt = null;

ResultSet rset = null;

try (Connection con = DriverManager.getConnection(DBconfig.dbString, DBconfig.username, DBconfig.password)) {

String insertQuery = "insert into employee (emp\_id, emp\_name, age , dept) values (?,?,?,?)";

PreparedStatement prepStmt = con.prepareStatement(insertQuery);

System.out.println("connection established!");

con.setAutoCommit(false);

prepStmt.setString(1, "1");

prepStmt.setString(2, "Jitendra Sahu");

prepStmt.setString(3, "21");

prepStmt.setString(4, "CS&IT");

prepStmt.addBatch();

prepStmt.setString(1, "2");

prepStmt.setString(2, "Mohan Markam");

prepStmt.setString(3, "26");

prepStmt.setString(4, "Social");

prepStmt.addBatch();

prepStmt.setString(1, "3");

prepStmt.setString(2, "Kanhaiya");

prepStmt.setString(3, "41");

prepStmt.setString(4, "LibraryScience");

prepStmt.addBatch();

int[] affectedRecords = prepStmt.executeBatch();

con.commit();

System.out.println("Rows inserted : " + affectedRecords.length);

} catch (SQLException e) {

e.printStackTrace();

} finally {

try {

if (rset != null) rset.close();

if (stmt != null) stmt.close();

} catch (SQLException e) {

e.printStackTrace();

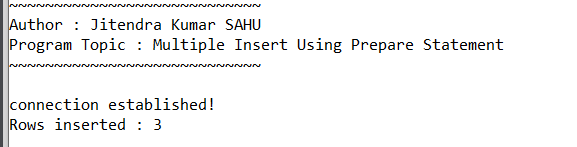
}

}

}

}

**Output:**

****

Program 46. Write a java program to display contents of a table on the console.

**Code:**

DisplayTableContentToConsole.java

import java.sql.SQLException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import jkdatabase.DBconfig;

// import java.sql.

import myPack.Intro;

public class DisplayTableContentToConsole {

public static void main(String[] args) {

Intro.print("Display table content to console");

Statement stmt = null;

ResultSet rset = null;

try (Connection con = DriverManager.getConnection(DBconfig.dbString, DBconfig.username, DBconfig.password)) {

System.out.println("Connection stabilized");

stmt = con.createStatement();

String sql = "select \* from employee";

rset = stmt.executeQuery(sql);

int i = 1;

System.out.println("Employee table data : ");

while (rset.next()) {

int emp\_id = rset.getInt(1);

String name = rset.getString(2);

int age = rset.getInt(3);

String dept = rset.getString(4);

System.out.printf("row %d : %d\t%s\t%d\t%s\n", i++, emp\_id, name, age, dept);

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

try {

if (rset != null)rset.close();

if (stmt != null)stmt.close();

} catch (SQLException e) {

e.printStackTrace();

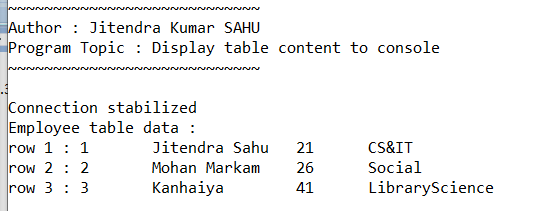
}

}

}

}

**Output:**

****

Program 47. Write a java program to update rows using result set.

**Code:**

import java.sql.SQLException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.Scanner;

import jkdatabase.DBconfig;

// import java.sql.

import myPack.Intro;

public class UpdateTableUsingResultSet {

public static void main(String[] args) {

Intro.print("Update row using ResultSet");

Statement stmt = null;

ResultSet rset = null;

try (Connection con = DriverManager.getConnection(DBconfig.dbString, DBconfig.username, DBconfig.password)) {

System.out.println("Connection stabilized");

stmt = con.createStatement(

ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_UPDATABLE);

String sql = "select \* from employee";

rset = stmt.executeQuery(sql);

int i = 1;

System.out.println("Employee table data before update : ");

while (rset.next()) {

int emp\_id = rset.getInt(1);

String name = rset.getString(2);

int age = rset.getInt(3);

String dept = rset.getString(4);

System.out.printf("row %d : %d\t%s\t%d\t%s\n", i++, emp\_id, name, age, dept);

}

// update starts from here

Scanner sc = new Scanner(System.in);

System.out.println("|\nEnter name for id = 2");

String newName = sc.nextLine();

// re-initialize rset so it again point to starting record

rset = stmt.executeQuery(sql) ;

System.out.println("Employee table data after update : ");

i = 1 ;

while (rset.next()) {

int emp\_id = rset.getInt(1);

if (emp\_id == 2){

rset.updateString("emp\_name",newName) ;

rset.updateRow();

}

String name = rset.getString(2);

int age = rset.getInt(3);

String dept = rset.getString(4);

System.out.printf("row %d : %d\t%s\t%d\t%s\n", i++, emp\_id, name, age, dept);

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

try {

if (rset != null)

rset.close();

if (stmt != null)

stmt.close();

} catch (SQLException e) {

e.printStackTrace();

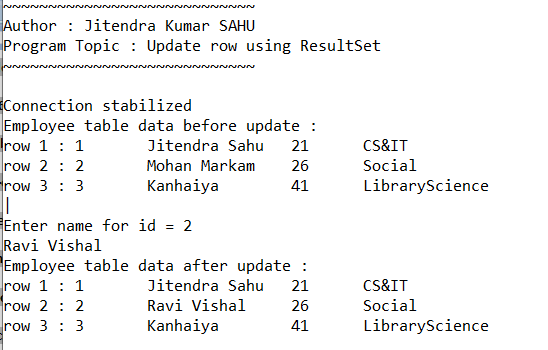
}

}

}

}

**Output:**

****

Program 48. Write a java program to describe the functions of metadata objects**. (resultset & database)**

**Code:**

import java.sql.Connection;

import java.sql.DatabaseMetaData;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.SQLException;

import java.sql.Statement;

import jkdatabase.DBconfig ;

import myPack.Intro;

public class DBMetadataExample {

// JDBC URL, username, and password of MySQL server

public static void main(String[] args) {

new Intro("printing database metadata");

Connection connection = null;

Statement statement = null;

ResultSet resultSet = null;

try {

// Establish the connection

connection = DriverManager.getConnection(DBconfig.dbString, DBconfig.username, DBconfig.password);

// Get DatabaseMetaData

DatabaseMetaData dbMetaData = connection.getMetaData();

// Print general database information

System.out.println("Database Product Name: " + dbMetaData.getDatabaseProductName());

System.out.println("Database Product Version: " + dbMetaData.getDatabaseProductVersion());

System.out.println("Database URL: " + dbMetaData.getURL());

System.out.println("Database User: " + dbMetaData.getUserName());

// Print tables in the database

ResultSet tables = dbMetaData.getTables(null, null, "%", new String[]{"TABLE"});

System.out.println("\nTables in the database:");

// while (tables.next()) {

// System.out.println(tables.getString("emp\_id"));

// }

// Create a statement to execute a query

statement = connection.createStatement();

resultSet = statement.executeQuery("SELECT \* FROM employee");

// Get ResultSetMetaData

ResultSetMetaData rsMetaData = resultSet.getMetaData();

// Print column information

int columnCount = rsMetaData.getColumnCount();

System.out.println("\nColumns in the Employee table:");

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

System.out.println("Column " + i + ": " + rsMetaData.getColumnName(i) + " - " + rsMetaData.getColumnTypeName(i));

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

// Close resources in reverse order of their creation

try {

if (resultSet != null) resultSet.close();

if (statement != null) statement.close();

if (connection != null) connection.close();

} catch (SQLException e) {

e.printStackTrace();

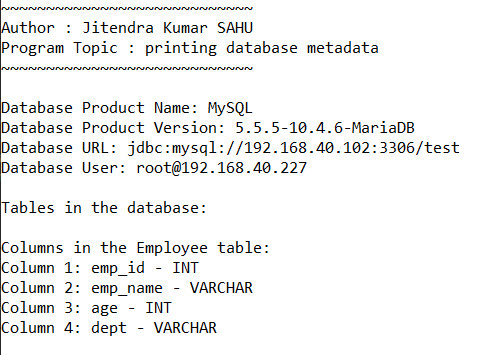
}

}

}

}

**Output:**

****

Program 49. Write a java program to demonstrate the ArrayList class.

**Code:**

import myPack.Intro;

import java.util.ArrayList;

import java.util.Scanner;

public class ArrayListDemo {

public static void main(String[] args) {

Intro.print("ArrayList Class & its methods");

ArrayList<String> list = new ArrayList<>();

Scanner scanner = new Scanner(System.in);

// Adding elements

System.out.println("Enter elements for the ArrayList (type 'exit' to stop):");

while (true) {

String input = scanner.nextLine();

if (input.equalsIgnoreCase("exit")) {

break;

}

list.add(input);

}

System.out.println("ArrayList: " + list);

// Accessing elements

System.out.print("Enter an index to access: ");

int index = scanner.nextInt();

scanner.nextLine(); // Consume newline

if (index >= 0 && index < list.size()) {

System.out.println("Element at index " + index + ": " + list.get(index));

} else {

System.out.println("Index out of bounds");

}

// Removing an element

System.out.print("Enter an index to remove: ");

int removeIndex = scanner.nextInt();

scanner.nextLine(); // Consume newline

if (removeIndex >= 0 && removeIndex < list.size()) {

list.remove(removeIndex);

System.out.println("After removing element at index " + removeIndex + ": " + list);

} else {

System.out.println("Index out of bounds");

}

// Size of the list

System.out.println("Size of ArrayList: " + list.size());

// Iterating over the elements

System.out.println("Iterating over ArrayList:");

for (String item : list) {

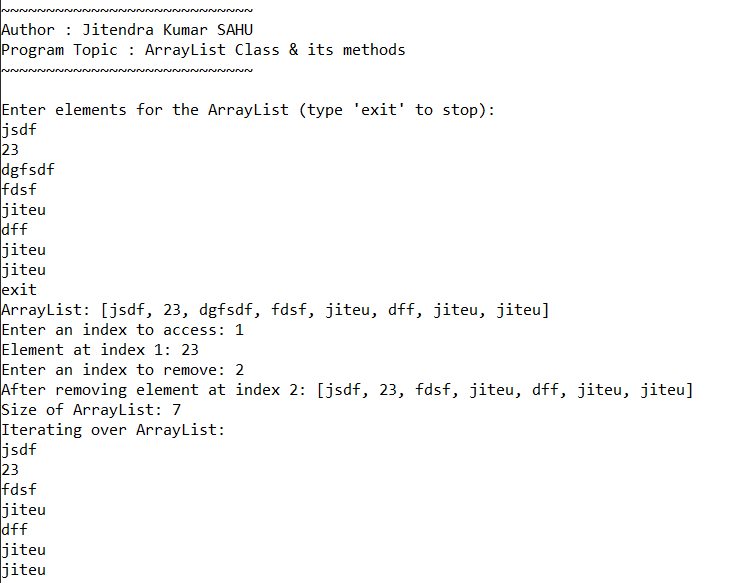
System.out.println(item);

}

}

}

**Output:**

****

Program 50. Write a java program to demonstrate the HashSet class.

**Code:**

import myPack.Intro;

import java.util.HashSet;

import java.util.Scanner;

public class HashSetDemo2 {

public static void main(String[] args) {

Intro.print("HashSet Class & its methods");

HashSet<String> set = new HashSet<>();

Scanner scanner = new Scanner(System.in);

// Adding elements

System.out.println("Enter elements for the HashSet (type 'exit' to stop):");

while (true) {

String input = scanner.nextLine();

if (input.equalsIgnoreCase("exit")) {

break;

}

set.add(input);

}

System.out.println("HashSet: " + set);

// Check if set contains an element

System.out.print("Enter an element to check: ");

String elementToCheck = scanner.nextLine();

System.out.println("Contains '" + elementToCheck + "': " + set.contains(elementToCheck));

// Remove an element

System.out.print("Enter an element to remove: ");

String elementToRemove = scanner.nextLine();

set.remove(elementToRemove);

System.out.println("After removing '" + elementToRemove + "': " + set);

// Size of the set

System.out.println("Size of HashSet: " + set.size());

// Iterating over the elements

System.out.println("Iterating over HashSet:");

for (String item : set) {

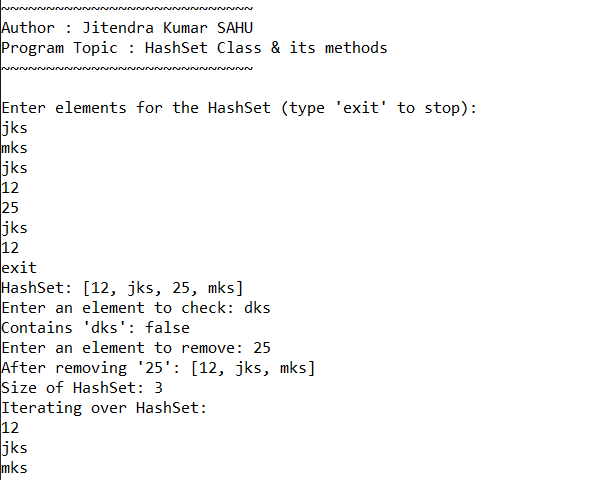
System.out.println(item);

}

}

}

**Output:**

****

Program 51. Write a java program to demonstrate the HashMap class.

**Code:**

import myPack.Intro;

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class HashMapDemo {

public static void main(String[] args) {

Intro.print("HashMap Class & its methods");

HashMap<String, Integer> map = new HashMap<>();

Scanner scanner = new Scanner(System.in);

// Adding elements

System.out.println("Enter elements for the HashMap (key-value pairs, type 'exit' to stop):");

while (true) {

System.out.print("Enter key: ");

String key = scanner.nextLine();

if (key.equalsIgnoreCase("exit")) {

break;

}

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

int value = scanner.nextInt();

scanner.nextLine(); // Consume newline

map.put(key, value);

}

System.out.println("HashMap: " + map);

// Accessing a value

System.out.print("Enter a key to get its value: ");

String keyToGet = scanner.nextLine();

if (map.containsKey(keyToGet)) {

System.out.println("Value for key '" + keyToGet + "': " + map.get(keyToGet));

} else {

System.out.println("Key not found");

}

// Removing an element

System.out.print("Enter a key to remove: ");

String keyToRemove = scanner.nextLine();

map.remove(keyToRemove);

System.out.println("After removing key '" + keyToRemove + "': " + map);

// Size of the map

System.out.println("Size of HashMap: " + map.size());

// Iterating over the elements

System.out.println("Iterating over HashMap:");

for (Map.Entry<String, Integer> entry : map.entrySet()) {

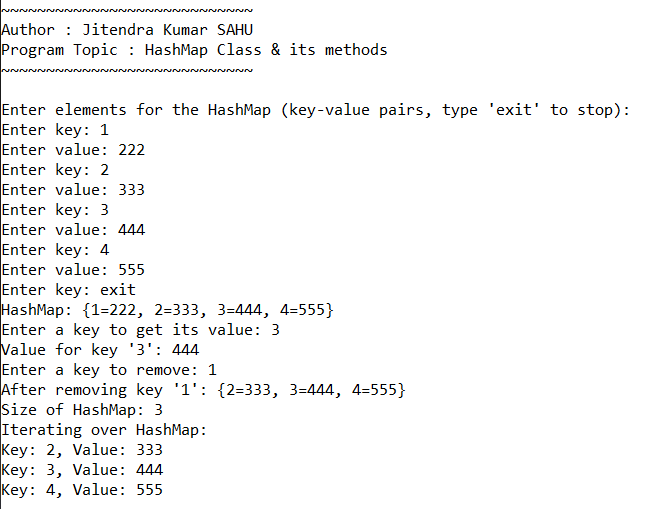
System.out.println("Key: " + entry.getKey() + ", Value: " + entry.getValue());

}

}

}

**Output:**

****

Program 52. Write a java program to demonstrate the Vector class.

**Code:**

import myPack.Intro;

import java.util.Vector;

import java.util.Scanner;

public class VectorDemo {

public static void main(String[] args) {

Intro.print("Vector class demonstration");

Vector<String> vector = new Vector<>();

Scanner scanner = new Scanner(System.in);

System.out.println("Menu:");

System.out.println("1. Add element");

System.out.println("2. Display elements");

System.out.println("3. Exit");

while (true) {

System.out.print(">> ");

int choice = scanner.nextInt();

scanner.nextLine(); // Consume newline

switch (choice) {

case 1:

System.out.print("Enter element to add: ");

String element = scanner.nextLine();

vector.add(element);

break;

case 2:

System.out.println("Vector Elements:");

for (String elem : vector) {

System.out.println(elem);

}

break;

case 3:

System.out.println("Exiting...");

scanner.close();

return;

default:

System.out.println("Invalid choice. Please try again.");

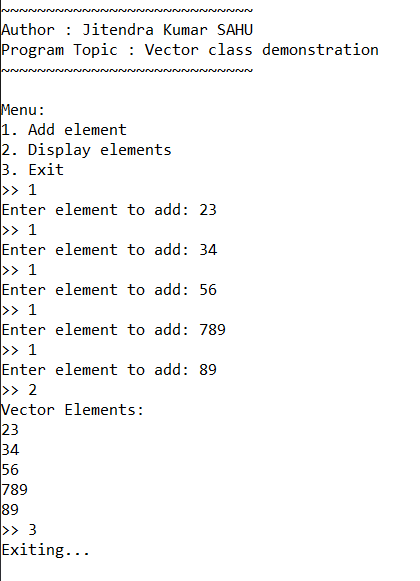
}

}

}

}

**Output:**

****

Program 53. Write a java program to demonstrate the LinkedList class.

**Code:**

import myPack.Intro;

import java.util.LinkedList;

import java.util.Scanner;

public class LinkedListDemo {

public static void main(String[] args) {

Intro.print("LinkedList Class & its methods");

LinkedList<String> linkedList = new LinkedList<>();

Scanner scanner = new Scanner(System.in);

// Adding elements

System.out.println("Enter elements for the LinkedList (type 'exit' to stop):");

while (true) {

String input = scanner.nextLine();

if (input.equalsIgnoreCase("exit")) {

break;

}

linkedList.add(input);

}

System.out.println("LinkedList: " + linkedList);

// Adding element at the first position

System.out.print("Enter an element to add at the first position: ");

String firstElement = scanner.nextLine();

linkedList.addFirst(firstElement);

System.out.println("After adding at the first position: " + linkedList);

// Adding element at the last position

System.out.print("Enter an element to add at the last position: ");

String lastElement = scanner.nextLine();

linkedList.addLast(lastElement);

System.out.println("After adding at the last position: " + linkedList);

// Accessing elements

System.out.print("Enter an index to access: ");

int index = scanner.nextInt();

scanner.nextLine(); // Consume newline

if (index >= 0 && index < linkedList.size()) {

System.out.println("Element at index " + index + ": " + linkedList.get(index));

} else {

System.out.println("Index out of bounds");

}

// Removing an element

System.out.print("Enter an index to remove: ");

int removeIndex = scanner.nextInt();

scanner.nextLine(); // Consume newline

if (removeIndex >= 0 && removeIndex < linkedList.size()) {

linkedList.remove(removeIndex);

System.out.println("After removing element at index " + removeIndex + ": " + linkedList);

} else {

System.out.println("Index out of bounds");

}

// Size of the LinkedList

System.out.println("Size of LinkedList: " + linkedList.size());

// Iterating over the elements

System.out.println("Iterating over LinkedList:");

for (String item : linkedList) {

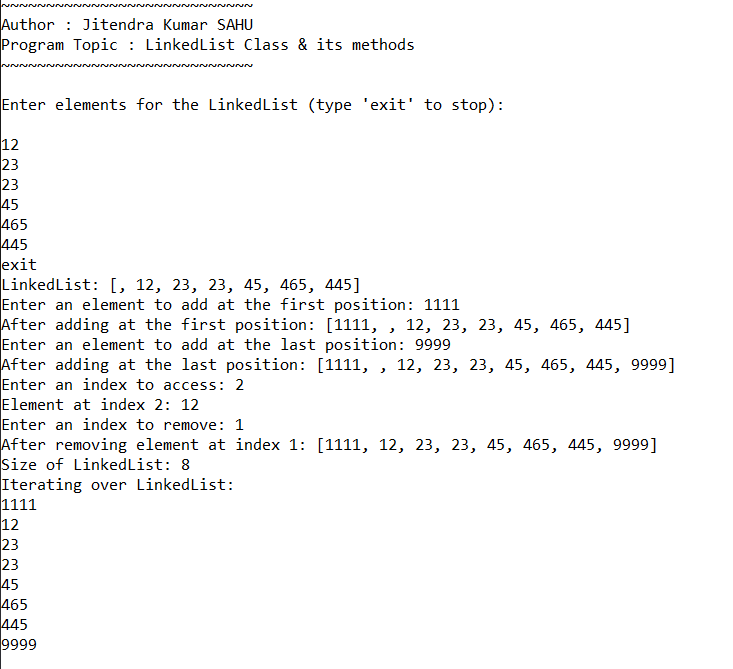
System.out.println(item);

}

}

}

**Output:**

****

Program 54. Write a java program to demonstrate the JTextField class.

**Code:**

import myPack.Intro;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class JTextFieldDemo extends JFrame {

private JTextField inputField;

private JTextArea displayArea;

public JTextFieldDemo() {

Intro.print("JTextField demonstration");

setTitle("JTextField Demo");

setSize(400, 300);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new BorderLayout());

JPanel inputPanel = new JPanel();

inputPanel.setLayout(new FlowLayout());

JLabel inputLabel = new JLabel("Enter text:");

inputPanel.add(inputLabel);

inputField = new JTextField(15);

inputPanel.add(inputField);

JButton addButton = new JButton("Add");

inputPanel.add(addButton);

add(inputPanel, BorderLayout.NORTH);

displayArea = new JTextArea();

displayArea.setEditable(false);

add(new JScrollPane(displayArea), BorderLayout.CENTER);

addButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String text = inputField.getText();

if (!text.isEmpty()) {

displayArea.append(text + "\n");

inputField.setText("");

}

}

});

setVisible(true);

}

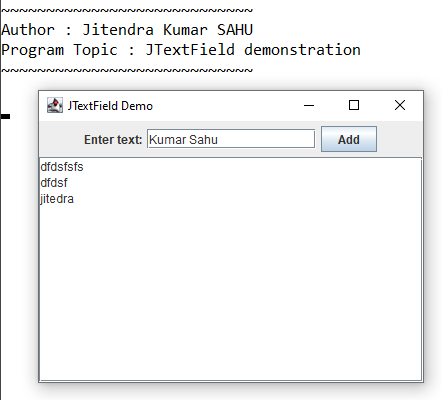
public static void main(String[] args) {

new JTextFieldDemo();

}

}

**Output:**

****

Program 55. Write a java program to demonstrate the JButton class.

**Code:**

import myPack.Intro;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class JButtonDemo extends JFrame {

private JTextField inputField;

private JTextArea displayArea;

public JButtonDemo() {

Intro.print("JButton demonstration");

setTitle("JTextField Demo");

setSize(400, 300);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new BorderLayout());

JPanel inputPanel = new JPanel();

inputPanel.setLayout(new FlowLayout());

JLabel inputLabel = new JLabel("Enter text:");

inputPanel.add(inputLabel);

inputField = new JTextField(15);

inputPanel.add(inputField);

JButton addButton = new JButton("Add");

inputPanel.add(addButton);

add(inputPanel, BorderLayout.NORTH);

displayArea = new JTextArea();

displayArea.setEditable(false);

add(new JScrollPane(displayArea), BorderLayout.CENTER);

addButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String text = inputField.getText();

if (!text.isEmpty()) {

displayArea.append(text + "\n");

inputField.setText("");

}

}

});

setVisible(true);

}

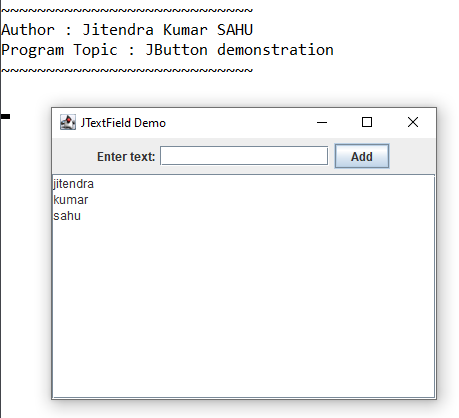
public static void main(String[] args) {

new JButtonDemo();

}

}

**Output:**

****

Program 56. Write a java program to demonstrate the JToggleButton class.

**Code:**

import myPack.Intro;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class JToggleButtonDemo extends JFrame {

private JTextArea displayArea;

private JToggleButton toggleButton;

public JToggleButtonDemo() {

Intro.print("JToggleButton");

setTitle("JToggleButton Demo");

setSize(400, 300);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new BorderLayout());

JPanel inputPanel = new JPanel();

inputPanel.setLayout(new FlowLayout());

toggleButton = new JToggleButton("Toggle");

inputPanel.add(toggleButton);

add(inputPanel, BorderLayout.NORTH);

displayArea = new JTextArea();

displayArea.setEditable(false);

add(new JScrollPane(displayArea), BorderLayout.CENTER);

toggleButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (toggleButton.isSelected())

displayArea.append("Toggle : on\n");

else

displayArea.append("Toggle : off\n") ;

}

});

setVisible(true);

}

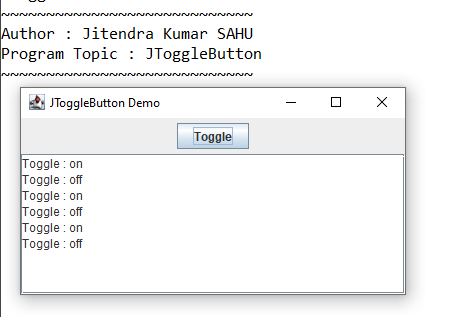
public static void main(String[] args) {

new JToggleButtonDemo();

}

}

**Output:**

****

Program 57. Write a java program to demonstrate the JCheckbox class.

**Code:**

import myPack.Intro;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class JCheckBoxDemo extends JFrame {

private JTextArea displayArea;

private JCheckBox checkBox;

public JCheckBoxDemo() {

Intro.print("JCheckBox");

setTitle("JCheckBox Demo");

setSize(400, 300);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new BorderLayout());

JPanel inputPanel = new JPanel();

inputPanel.setLayout(new FlowLayout());

checkBox = new JCheckBox("Check me");

inputPanel.add(checkBox);

add(inputPanel, BorderLayout.NORTH);

displayArea = new JTextArea();

displayArea.setEditable(false);

add(new JScrollPane(displayArea), BorderLayout.CENTER);

checkBox.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent e){

if (checkBox.isSelected()) displayArea.append("Checkbox : checked\n") ;

else displayArea.append("checkbox : unchecked\n") ;

}

}) ;

setVisible(true);

}

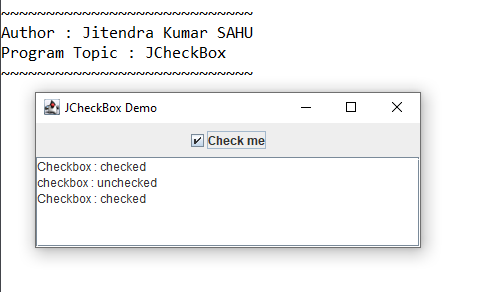
public static void main(String[] args) {

new JCheckBoxDemo();

}

}

**Output:**

****

Program 58. Write a java program to demonstrate the JRadioButton class.

**Code:**

import myPack.Intro;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class JRadioButtonDemo extends JFrame {

private JTextArea displayArea;

private JRadioButton radioButton;

public JRadioButtonDemo() {

Intro.print("JRadioButton");

setTitle("JRadioButton Demo");

setSize(400, 300);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new BorderLayout());

JPanel inputPanel = new JPanel();

inputPanel.setLayout(new FlowLayout());

radioButton = new JRadioButton("click to select");

inputPanel.add(radioButton);

add(inputPanel, BorderLayout.NORTH);

displayArea = new JTextArea();

displayArea.setEditable(false);

add(new JScrollPane(displayArea), BorderLayout.CENTER);

radioButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (radioButton.isSelected()) {

displayArea.append("RadioButton: Selected\n");

} else {

displayArea.append("RadioButton: Deselected\n");

}

}

});

setVisible(true);

}

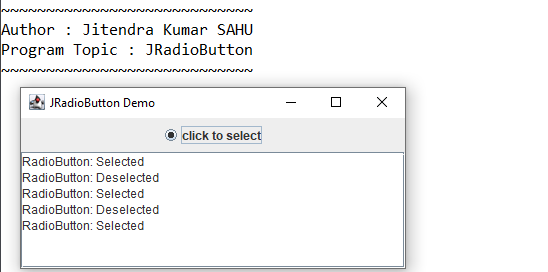
public static void main(String[] args) {

new JRadioButtonDemo();

}

}

**Output:**

****

Program 59. Write a java program to demonstrate the JComboBox class.

**Code:**

import myPack.Intro;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class JComboBoxDemo extends JFrame {

private JTextArea displayArea;

private JComboBox<String> comboBox ;

public JComboBoxDemo() {

Intro.print("JComboBox");

setTitle("JComboBox Demo");

setSize(400, 300);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new BorderLayout());

JPanel inputPanel = new JPanel();

inputPanel.setLayout(new FlowLayout());

String[] comboBoxItems ={"Item 1", "Item 2", "Item 3"};

comboBox = new JComboBox<> (comboBoxItems);

inputPanel.add(comboBox);

add(inputPanel, BorderLayout.NORTH);

displayArea = new JTextArea();

displayArea.setEditable(false);

add(new JScrollPane(displayArea), BorderLayout.CENTER);

comboBox.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

displayArea.append("ComboBox: " + comboBox.getSelectedItem() + "\n");

}

});

setVisible(true);

}

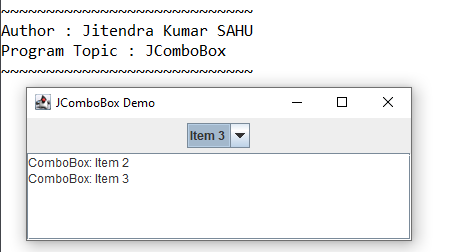
public static void main(String[] args) {

new JComboBoxDemo();

}

}

**Output:**

****

Program 60. Write a java program to demonstrate the JList class.

**Code:**

import myPack.Intro;

import javax.swing.\*;

import java.awt.\*;

public class JListDemo extends JFrame {

private JTextArea displayArea;

private JList<String> list;

public JListDemo() {

Intro.print("JList");

setTitle("JList Demo");

setSize(400, 300);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new BorderLayout());

JPanel inputPanel = new JPanel();

inputPanel.setLayout(new FlowLayout());

String[] listItems = {"Item A", "Item B", "Item C"};

list = new JList<>(listItems);

list.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

JScrollPane listScrollPane = new JScrollPane(list);

listScrollPane.setPreferredSize(new Dimension(100, 60));

inputPanel.add(listScrollPane);

add(inputPanel, BorderLayout.NORTH);

displayArea = new JTextArea();

displayArea.setEditable(false);

add(new JScrollPane(displayArea), BorderLayout.CENTER);

list.addListSelectionListener(e -> {

if (!e.getValueIsAdjusting()) {

displayArea.append("List: " + list.getSelectedValue() + "\n");

}

});

setVisible(true);

}

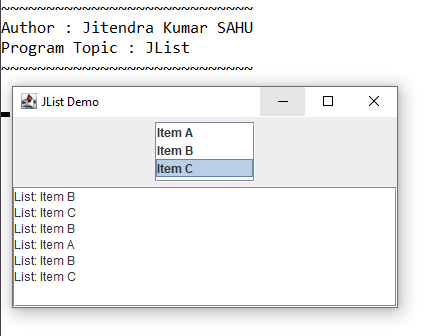
public static void main(String[] args) {

new JListDemo();

}

}

**Output:**

****