**String Program**

**package** strfunpack;

**publicclass** strmun {

**publicstaticvoid** main(String[] args) {

// **TODO** Auto-generated method stub

/\*System.out.println("sarathi");

int a;

a=10;

System.out.println(a);

int b=20;

char c='g';\*/

String comment=("this is true");

String result="Truth triumph";

String result1="Truth triumph";

System.***out***.println(comment.concat(result));

System.***out***.println(result.equals(result1));

System.***out***.println(comment.length());

System.***out***.println(comment.charAt(3));

System.***out***.println(comment.indexOf("true"));

String revresult = **new** Strin-gBuffer(result).reverse().toString();

System.***out***.println(revresult);

}

}

**Array with loop**

**package** ary2;

**publicclass** forloop {

**publicstaticvoid** main(String[] args) {

// **TODO** Auto-generated method stub

//int i;

**int**a[]= {34,57,766,3542,365,763};

**char**d[]= {'r','k','r'};

String c[]= {"the","most","important"};

**int**b[]=**newint**[3];

{

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

{

System.***out***.println("the value is "+a[i]);

}

**int**j=6;

**while**(j<=5)

{

System.***out***.println("j value is:"+j);

j++;

}

-------------------------------------------------------------

**int**k=6;

**do** {

System.***out***.println("K value is:"+k);

k++;

}

**while**(k<=5);

-------------------------------------------------------------

}}}

**Array List:**

**publicclass** arraylist {

**publicstaticvoid** main(String[] args) {

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

list.add("Mango|orange");

list.add("1");

list.add("d");

list.add(" ");

list.add("@#$");

System.***out***.println(list);

list.set(2,"selenium");

list.set(3,"java");

System.***out***.println(list);

// list.remove("Mango");

list.remove(0);

System.***out***.println(list);

}

}

If and switch

**package** ifconp;

**publicclass** ifcondi {

**publicstaticvoid** main(String[] args) {

// **TODO** Auto-generated method stub

**int**m=70;

**if**(m>=50)

{

System.***out***.println("Pass Mark");

}

**else**

{

System.***out***.println("Fail Mark");

}

**Int** mark=70;

**if**(mark>=90)

{

System.***out***.println("outstanding mark");

}

**elseif**(mark>=80)

{

System.***out***.println("distinction mark");

}

**elseif**(mark>=60)

{

System.***out***.println("fistclass mark");

}

**else**

{

System.***out***.println("not elegible to join ");

}

**int**days = 9;

**switch** (days) {

**case** 1:

System.***out***.println("Monday");

**break**;

**case** 2:

System.***out***.println("Tuesday");

**break**;

**case** 3:

System.***out***.println("Wednesday");

**break**;

**case** 4:

System.***out***.println("Thursday");

**break**;

**case** 5:

System.***out***.println("Friday");

**break**;

**case** 6:

System.***out***.println("Saturday");

**break**;

**case** 7:

System.***out***.println("Sunday");

**break**;

**default**:

System.***out***.println("Enter the value between 1 to 7");

**break**;

}

}

}

Methods

**package** methodp;

**publicclass** methodclass {

// method definition

**void** logicMethod3(){

**int**z=10;

System.***out***.println("execution of sub one " +z);

}

**publicvoid** logicMethod4() {

**int**A, P=10000, r=3, t=5;

A = P\*(1 + r\*t) ;

System.***out***.println("Interest value is " +A);

}

**publicvoid** logicMethod5(){

**int**h=80;

h=h\*h;

System.***out***.println("execution of sub one " +h);

}

**publicstaticvoid** main(String[] args) {

// **TODO** Auto-generated method stub

methodclass a=**new** methodclass();

a.logicMethod4();

a.*logicMethod5*();

a.logicMethod3();

}

}

Method overloading

**package** methoverloadp;

**class** DisplayOverloading

{

**Public void** display(**int** a)

{

System.***out***.println( "rijin " +a);

}

**Public static void** display(**char**a, **int**num)

{

**int**c=10, b=30;

num=c+b;

System.***out***.println(a+" "+num);

}

**Public void** display(String c)

{

System.***out***.println(c);

}

**publicvoid**display(**int**v)

{

System.***out***.println(v);

}

**publicvoid** display(**char**v)

{

System.***out***.println(v);

}

}

**publicclass** moverload {

**publicstaticvoid** main(String[] args) {

// **TODO** Auto-generated method stub

DisplayOverloading w = **new** DisplayOverloading();

w.display(60);

w.display("rate");

w.display(88);

w.display('r');

DisplayOverloading.*display*('z',10);

}

}

Method overriding

**package** methodoverridingp;

**class** ABC{

//Overridden method

**publicvoid** disp()

{

System.***out***.println("disp() method of parent1 class");

}

**publicvoid** disp1()

{

System.***out***.println("disp() method of parent2 class");

}

}

**class** methodclass **extends** ABC{

//Overriding method

**publicvoid** disp(){

System.***out***.println("disp() method of Child1 class");

}

**publicvoid** disp1()

{

System.***out***.println("disp() method of Child2 class");

}

**publicvoid**newMethod(){

System.***out***.println("new method of main class");

}

**publicstaticvoid** main( String args[]) {

/\* When Parent class reference refers to the parent class object

\* then in this case overridden method (the method of parent class)

\* is called.\*/

ABC x = **new** ABC();

x.disp();

x.disp1();

/\* When parent class reference refers to the child class object

\* then the overriding method (method of child class) is called.

\* This is called dynamic method dispatch and runtime polymorphism

\*/

ABC y = **new** methodclass();

y.disp();

y.disp1();

methodclass z = **new** methodclass();

z.newMethod();

} }

**Inheritance**

**publicclass**inc**extends** tech{

**publicstaticvoid** main(String args[]){

//tech v=new tech();//While create object for SubClass which cannot call main class method

incv = **new**inc();

v.add();

v.sub();

v.multi(); }

**privatevoid** multi() {

System.***out***.println("this is private method of main");

}}

**class** tech {

String Subject1 = "Physics";

String Subject2 = "Chemistry";

String IIT = "Enterance Score";

**int**totalmark=1089;

**void** add(){

**int**Phy=187, che=194;

**int**total;

total = Phy + che;

System.***out***.println("Marks");

System.***out***.println("Total Marks of Physics and Chemistry = " +total);

}

**void** sub() {

**int**a=1, b=2;

**int**c=b-a;

System.***out***.println("sub value="+c);

}

Implements for multiple Inheritance

**package** interfacep;

**interface** Print{

**void** print();

//void display();

}

**interface** Showable{

**void** show();

}

**class** interclas **implements** Print, Showable{

**publicvoid** print(){

**int**a=20,b=10,c;

c=a+b;

System.***out***.println(c);}

**publicvoid** show(){

**int**a=10,b=20,c;

c=a-b;

System.***out***.println(c);}

**publicvoid** display(){

**int**a=10,b=20,c;

c=a+b;

System.***out***.println(c);}

**Public static void** main(String args[]){

interclas obj = **new** interclas();

//Print obj = new Print(); //Object creation not possible in interface name

obj.print();

obj.show();

obj.display();

}

}

Poly morphism

**package** polyobjectp;

**publicclass** objmulti {

**publicstaticvoid** main(String[] args) {

// **TODO** Auto-generated method stub

Animal a;

a=**new** Dog();

a.eat();

a=**new** Cat();

a.eat();

a=**new** Lion();

a.eat();

}

}

**class** Animal{

**void** eat(){

System.***out***.println("eating...");

}

}

**class** Dog **extends** Animal{

**void** eat(){

System.***out***.println("eating bread...");

}

}

**class** Cat **extends** Animal{

**void** eat(){

System.***out***.println("eating rat...");

}

}

**class** Lion **extends** Animal{

**void** eat(){System.***out***.println("eating meat...");}

}

Encapsulation

**package** methodp;

**publicclass** EncapTest {

**private**Stringname;

**private**StringidNum;

**privateint**age;

**publicint** getAge() {

**return**age;

}

**public**String getName() {

**return**name;

}

**public**String getIdNum() {

**return**idNum;

}

**publicvoid** setAge( **int**newAge) {

age = newAge;

}

**publicvoid** setName(StringnewName) {

name = newName;

}

**publicvoid** setIdNum( StringnewId) {

idNum = newId;

}

}

Encap main

**package** methodp;

**publicclass** RunEncap {

**publicstaticvoid** main(String args[]) {

EncapTest encap = **new** EncapTest();

encap.setName("James");

encap.setAge(20);

encap.setIdNum("12343ms");

System.***out***.print("Name : " + encap.getName() + " Age : " + encap.getAge() + " IDNUM :" + encap.getIdNum());

}

}

Constructor

**package** construtorp;

//Java Program to create and call a default constructor

**publicclass** conclass {

//creating a default constructor

conclass(){

System.***out***.println("Bike is created");

}

//main method

**publicstaticvoid** main(String args[]){

//calling a default constructor

conclass b=**new** conclass();

//conclass c=new conclass();

}

}

Default Constructor

**package** construtorp;

//Let us see another example of default constructor

//which displays the default values

**publicclass** condefalult {

**int**id=100;

String name="Gopi";

condefalult(){

System.***out***.println(id+" "+name);

}

//method to display the value of id and name

**void** display(){

System.***out***.println(id+" "+name);

}

**publicstaticvoid** main(String args[]){

//creating objects

condefalult s1=**new** condefalult();

condefalult s2=**new** condefalult();

//displaying values of the object

s1.display();

s2.display();

}

}

Parameterized constructor

**package** construtorp;

//Java Program to demonstrate the use of parameterized constructor

**publicclass** parameterconstructor {

**int**id;

String name;

//creating a parameterized constructor

parameterconstructor(**int**i,String n){

id = i;

name = n;

System.***out***.println(id+" "+name);

}

//method to display the values

**void** display(){

System.***out***.println(id+" "+name);

}

**publicstaticvoid** main(String args[]){

//creating objects and passing values

parameterconstructor s1 = **new** parameterconstructor(111,"Karan");

parameterconstructor s2 = **new** parameterconstructor(222,"Aryan");

//calling method to display the values of object

s1.display();

s2.display();

}

}

Abstract:

Normal Abstract:

**package** packabstract;

**abstractclass**Base {

**abstractvoid** fun();

}

**class** Derived **extends**Base {

**void** fun() { System.***out***.println("Derived fun() called"); }

}

**class** classtoinherit {

**publicstaticvoid** main(String args[]) {

// Uncommenting the following line will cause compiler error as the

// line tries to create an instance of abstract class.

// Base b = new Base();

// We can have references of Base type.

Derived b = **new** Derived();

b.fun();

}

}

Abstract with constructor

**package** packabstract;

**abstractclass** Bas {

Bas() { System.***out***.println("Base Constructor Called"); }

**abstractvoid** fun();

}

**class** Derive **extends** Bas {

Derive() { System.***out***.println("Derived Constructor Called"); }

**void** fun() { System.***out***.println("Derived fun() called"); }

}

**class** classsub1 **extends** Derive{

**publicstaticvoid** main(String args[]) {

classsub1 d = **new** classsub1();

d.fun();

}

}

**Exception Handling:**

**publicclass** arthmeticexp {

**public static void** main(String[] args)

{

**try** {

**int** a = 10, b = 5, c = 5, result;

result = 10 / (0);

System.***out***.println("result" + result);

}

**catch** (ArithmeticException e) {

System.***out***.println("Exception caught:Division by zero");

}

**finally** {

System.***out***.println("I am in final block");

}

System.***out***.println("no exe");

result = a / (b - c);

System.***out***.println("result" + result);

}

}

Association

**package** Assopack;

**class** CarClass{

String carName;

**int**carId;

CarClass(String name, **int**id)

{

**this**.carName = name;

**this**.carId = id;

}

}

**class** Driver **extends** CarClass{

String driverName;

Driver(String name, String cname, **int**cid){

**super**(cname, cid);

**this**.driverName=name;

}

}

**class** assoclass{

**publicstaticvoid** main(String args[])

{

Driver obj = **new** Driver("Andy", "Ford", 9988);

System.***out***.println(obj.driverName+" is a driver of car Id: "+obj.carId);

}

}

**Palindrome**

**package** palindrome;

**publicclass** pali {

**publicstaticvoid** main(String args[]){

**int**r,sum=0,temp;

**int**n=727;//It is the number variable to be checked for palindrome

temp=n;

**while**(n>0){

r=n%10; //getting remainder

sum=(sum\*10)+r;

n=n/10;

}

**if**(temp==sum)

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

**else**

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

}

}

Access Modifier

Pack1

Methodclass

**package** Insideclass;

**publicclass** methodclass {

**privatevoid** pri() {

**int**a=10, b=20;

**int**c=a+b;

System.***out***.println(c);

}

**publicvoid** pub() {

**int**a=10, b=20;

**int**c=a-b;

System.***out***.println(c);

}

**protectedvoid**pro() {

**int**a=10, b=20;

**int**c=a\*b;

System.***out***.println(c);

}

**void** def()

{

**int**a=10, b=20;

**int**c=a\*b;

System.***out***.println(c);

}

**publicstaticvoid** main(String[] args) {

// **TODO** Auto-generated method stub

methodclass m=**new** methodclass();

m.pri();

m.pub();

m.pro();

m.def();

}

}

Outsideclass

**package** Insideclass;

**publicclass** outsideclass {

**publicstaticvoid** main(String[] args) {

// **TODO** Auto-generated method stub

methodclass n=**new** methodclass();

//n.pri();

n.pub();

n.pro();

n.def();

}

}

Outsidepackage

OutPackclass

**package** OutsidePackage;

**import** Insideclass.methodclass;

**publicclass** OutPackClass {

**publicstaticvoid** main(String[] args) {

// **TODO** Auto-generated method stub

methodclass mc=**new** methodclass();

mc.pub();

//mc.pro();

//mc.def();

}

}

Aggregation

**package** aggregationp;

//Java program to illustrate

//the concept of Aggregation.

**import**java.io.\*;

**import** java.util.\*;

//student class

**class** Student

{

String name;

**int**id ;

String dept;

Student(String name, **int**id, String dept)

{

**this**.name = name;

**this**.id = id;

**this**.dept = dept;

}

}

/\* Department class contains list of student

Objects. It is associated with student

class through its Object(s). \*/

**class** Department

{

String name;

**private** List<Student>students;

Department(String dname, List<Student>students)

{

**this**.name = dname;

**this**.students = students;

}

**public** List<Student>getStudents()

{

**return**students;

}

}

/\* Institute class contains list of Department

Objects. It is asoociated with Department

class through its Object(s).\*/

**class** Institute

{

String instituteName;

**private** List<Department>departments;

Institute(String instituteName, List<Department>departments)

{

**this**.instituteName = instituteName;

**this**.departments = departments;

}

// count total students of all departments

// in a given institute

**publicint**getTotalStudentsInInstitute()

{

**int**noOfStudents = 0;

List<Student>students;

**for**(Department dept : departments)

{

students = dept.getStudents();

**for**(Student s : students)

{

noOfStudents++;

}

}

**return**noOfStudents;

}

}

//main method

**class** aggre

{

**publicstaticvoid** main (String[] args)

{

Student s1 = **new**Student("Mia", 1, "CSE");

Student s2 = **new**Student("Priya", 2, "CSE");

Student s3 = **new**Student("John", 1, "EE");

Student s4 = **new**Student("Rahul", 2, "EE");

// making a List of

// CSE Students.

List <Student>cse\_students = **new** ArrayList<Student>();

cse\_students.add(s1);

cse\_students.add(s2);

// making a List of

// EE Students

List <Student>ee\_students = **new** ArrayList<Student>();

ee\_students.add(s3);

ee\_students.add(s4);

Department CSE = **new**Department("CSE", cse\_students);

Department EE = **new**Department("EE", ee\_students);

List <Department>departments = **new** ArrayList<Department>();

departments.add(CSE);

departments.add(EE);

// creating an instance of Institute.

Institute institute = **new**Institute("KGISL", departments);

System.***out***.print("Total students in institute: ");

System.***out***.print(institute.getTotalStudentsInInstitute());

}

}

Association

**package** association;

//class bank

**class** Bank

{

**private** String name;

// bank name

Bank(String name)

{

**this**.name = name;

}

**public** String getBankName()

{

**returnthis**.name;

}

}

//employee class

**class** Employee

{

**private** String name;

// employee name

Employee(String name)

{

**this**.name = name;

}

**public** String getEmployeeName()

{

**returnthis**.name;

}

}

//Association between both the

//classes in main method

**class** Association

{

**publicstaticvoid** main (String[] args)

{

Bank bank = **new** Bank("HSBC");

Employee emp = **new** Employee("Geetha");

System.***out***.println(emp.getEmployeeName() +

" is employee of " + bank.getBankName());

}

}

Composition

package Compositionpack;

//Java program to illustrate

//the concept of Composition

import java.io.\*;

import java.util.\*;

//class book

class Book

{

public String title;

public String author;

Book(String title, String author)

{

this.title = title;

this.author = author;

}

}

//Libary class contains

//list of books.

class Library

{

// reference to refer to list of books.

private final List<Book> books;

Library (List<Book> books)

{

this.books = books;

}

public List<Book>getTotalBooksInLibrary(){

return books;

}

}

//main method

class compoclass

{

public static void main (String[] args)

{

// Creating the Objects of Book class.

Book b1 = new Book("Effective Java", "Jos");

Book b2 = new Book("Thinking in Java", "Bruce");

Book b3 = new Book("Java: The Complete Reference", "Herbert");

// Creating the list which contains the

// no. of books.

List<Book> books = new ArrayList<Book>();

books.add(b1);

books.add(b2);

books.add(b3);

Library library = new Library(books);

List<Book> TNB = library.getTotalBooksInLibrary();

for(Book bn : TNB){

System.out.println("Title : " + bn.title + " and "

+" Author : " + bn.author);

}

}

}

**Selenium Programs**

package mouseoverp;

import java.io.File;

import java.io.IOException;

import java.util.ArrayList;

import java.util.concurrent.TimeUnit;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.interactions.Actions;

public class mouseover {

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

// TODO Auto-generated method stub

System.setProperty("webdriver.chrome.driver","C:\\SeleniumTraining\\chromedriver.exe");

WebDriver d=new ChromeDriver();

d.get("http://executeautomation.com/demosite/index.html?UserName=Rajkumar&Password=coolbuddy9&Login=Login");

d.manage().window().maximize();

//d.manage().timeouts().implicitlyWait(15, TimeUnit.SECONDS);

d.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

Actions a=new Actions(d);

WebElement moveonmenu = d.findElement(By.xpath(".//\*[@id='Automation Tools']"));

a.moveToElement(moveonmenu).build().perform();

Thread.sleep(3000);

WebElement elem2 = d.findElement(By.xpath("//\*[@id=\"cssmenu\"]/ul/li[2]/ul/li[2]/a"));

a.moveToElement(elem2).build().perform();

Thread.sleep(3000);

WebElement elem3 = d.findElement(By.xpath("//\*[@id=\"cssmenu\"]/ul/li[2]/ul/li[2]/ul/li[2]/a/span"));

a.moveToElement(elem3).click().build().perform();

Thread.sleep(73000);

d.get("https://www.amazon.in");

Thread.sleep(8000);

JavascriptExecutor jse = (JavascriptExecutor)d;

jse.executeScript("window.scrollBy(0,3250)", "");

Thread.sleep(4000);

jse.executeScript("window.scrollBy(0,-1250)", "");

d.get("http://www.facebook.com");

File src= ((TakesScreenshot)d).getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(src, new File("C:/newaccinvalid.png"));

d.get("http://www.india.com");

File src1= ((TakesScreenshot)d).getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(src1, new File("C:/jhgjhg.png"));

d.navigate().back();

d.navigate().forward();

d.get("http://executeautomation.com/demosite/index.html?UserName=Rajkumar&Password=coolbuddy9&Login=Login");

d.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

//Code to open a new link from current page to new page

((JavascriptExecutor)d).executeScript("window.open()");

ArrayList<String> tabs = new ArrayList<String>(d.getWindowHandles());

d.switchTo().window(tabs.get(1));

d.get("http://google.com");

//To stay and move to the current second window

((JavascriptExecutor)d).executeScript("window.open()");

ArrayList<String> tabs1 = new ArrayList<String>(d.getWindowHandles());

d.switchTo().window(tabs1.get(2));

d.get("http://google.com");

//To move back to the parent window

ArrayList<String> tab = new ArrayList<String>(d.getWindowHandles());

d.switchTo().window(tab.get(0));

d.get("https://www.amazon.in");

// ArrayList<String> tab2 = new ArrayList<String>(d.getWindowHandles());

// d.switchTo().window(tab2.get(2));

// d.get("https://www.amazon.in/gp/product/B07DWP6GWB/ref=s9\_acss\_bw\_cg\_Sneakpee\_2b1\_w?pf\_rd\_m=A1K21FY43GMZF8&pf\_rd\_s=merchandised-search-5&pf\_rd\_r=J62K17T4SV2WKCT4C7VC&pf\_rd\_t=101&pf\_rd\_p=89ca6c5f-3ec5-4fb9-ba0f-268b8c8b0ffc&pf\_rd\_i=1389396031");

//d.navigate().back();

}

}

**Assertion**

package testpack;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

public class assertion {

public static void main(String[] args) {

// TODO Auto-generated method stub

/\*String expected []= {"Arun","Kavi"};

String actual []= {"Arun","Kavi"};

Assert.assertEquals(expected,actual);

System.out.println("Both expected and Actual are same");\*/

WebDriver d;

System.setProperty("webdriver.chrome.driver", "C:\\SeleniumTraining\\chromedriver.exe");

d=new ChromeDriver();

d.get("https://www.softwaretestingmaterial.com");

//Actual title is "Software Testing Material - A site for Software Testers"

//We took title as "Software Testing Material" to make the test fail

String Title = "software Testing Material";

String GetTitle = d.getTitle();

System.out.println("Assertion starts here...");

Assert.assertEquals(Title, GetTitle);

System.out.println("As Per Expected");

}

}

**Double click and right click**

package SeleChromep;

import org.openqa.selenium.Alert;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

public class doubleclick {

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

// TODO Auto-generated method stub

WebDriver driver;

System.setProperty("webdriver.chrome.driver","C:\\Seleniumtemp\\chromedriver.exe");

driver= new ChromeDriver();

//Launch the Application Under Test (AUT)

//driver.get("http://demo.guru99.com/test/simple\_context\_menu.html");

////driver.manage().window().maximize();

/\*

driver.get("http://demo.guru99.com/test/simple\_context\_menu.html");

driver.manage().window().maximize();

//Double click the button to launch an alertbox

Actions action = new Actions(driver);

WebElement link =driver.findElement(By.xpath("//button[text()='Double-Click Me To See Alert']"));

action.doubleClick(link).perform();

Thread.sleep(5000);

//driver.switchTo().alert().accept();

//Switch to the alert box and click on OK button

Alert alert = driver.switchTo().alert();

System.out.println("Alert Text\n" +alert.getText());

alert.accept();

//Closing the driver instance

driver.quit();

\*/

driver.get("http://swisnl.github.io/jQuery-contextMenu/demo.html");

//To maximize the browser

driver.manage().window().maximize();

//Create an object 'action' of an Actions class

Thread.sleep(1000);

Actions action1 = new Actions(driver);

By locator = By.cssSelector(".context-menu-one");

//By locator = By.xpath("/html/body/div/section/div/div/div/p/span");

//Wait for the element. Used Explicit wait

WebDriverWait wait = new WebDriverWait(driver, 5);

wait.until(ExpectedConditions.presenceOfElementLocated(locator));

WebElement rightClickElement=driver.findElement(locator);

//contextClick() method to do right click on the element

action1.contextClick(rightClickElement).build().perform();

Thread.sleep(3000);

WebElement rce=driver.findElement(By.cssSelector(".context-menu-icon-quit"));

Thread.sleep(4000);

rce.click();

driver.switchTo().alert().accept();

// Alert alert = driver.switchTo().alert();

// alert.accept();

//driver.findElement(By.cssSelector(".context-menu-icon-edit")).click();

// WebElement getCopyText =driver.findElement(By.cssSelector(".context-menu-icon-edit"));

WebElement getCopyText1 =driver.findElement(By.name("Edit"));

//getText() method to get the text value

String GetText =getCopyText1.getText();

//To print the value

System.out.println(GetText);

//To close the browser

driver.close();

}

}

**Drag And Drop**

**package** mouseoverp;

**import**org.openqa.selenium.By;

**import**org.openqa.selenium.JavascriptExecutor;

**import**org.openqa.selenium.WebDriver;

**import**org.openqa.selenium.WebElement;

**import**org.openqa.selenium.chrome.ChromeDriver;

**import**org.openqa.selenium.interactions.Actions;

**publicclass** draganddrop {

**publicstaticvoid** main(String[] args) {

// **TODO** Auto-generated method stub

**DropDown Select:**

System.*setProperty*("webdriver.chrome.driver","C:\\SeleniumTraining\\chromedriver.exe");

WebDriver driver=**new**ChromeDriver();

driver.get("http://demo.guru99.com/test/drag\_drop.html");

JavascriptExecutor jse = (JavascriptExecutor)driver;

jse.executeScript("window.scrollBy(0,1250)", "");

//jse.executeScript("window.scrollBy(0,-1250)", "");

//Element which needs to drag.

WebElement From=driver.findElement(By.*xpath*("//\*[@id='credit2']/a"));

//Element on which need to drop.

WebElement To=driver.findElement(By.*xpath*("//\*[@id='bank']/li"));

//Using Action class for drag and drop.

Actions act=**new** Actions(driver);

//Dragged and dropped.

act.dragAndDrop(From, To).build().perform();

}

}

**File Upload**

package mouseoverp;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class fileupload {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.setProperty("webdriver.chrome.driver","C:\\SeleniumTraining\\chromedriver.exe");

WebDriver driver=new ChromeDriver();

String baseUrl = "http://demo.guru99.com/test/upload/";

driver.get(baseUrl);

WebElement uploadElement = driver.findElement(By.id("uploadfile\_0"));

// enter the file path onto the file-selection input field

uploadElement.sendKeys("C:\\Users\\JSP SARATHI\\Desktop\\Java syllabus.doc");

// check the "I accept the terms of service" check box

driver.findElement(By.id("terms")).click();

// click the "UploadFile" button

driver.findElement(By.name("send")).click();

}

}

**Link Text**

**package** SeleChromep;

**import**org.openqa.selenium.By;

**import**org.openqa.selenium.WebDriver;

**import**org.openqa.selenium.chrome.ChromeDriver;

**publicclass** Linktext {

**publicstaticvoid** main(String[] args) {

// **TODO** Auto-generated method stub

String baseUrl = "http://demo.guru99.com/test/link.html";

System.*setProperty*("webdriver.chrome.driver", "C:\\SeleniumTraining\\chromedriver.exe");

WebDriver driver =**new** ChromeDriver();

driver.get(baseUrl);

driver.findElement(By.*linkText*("click here")).click();

System.***out***.println("title of page is: " + driver.getTitle());

//driver.quit();

}

}

Selenium Programs

package mouseoverp;

import java.io.File;

import java.io.IOException;

import java.util.ArrayList;

import java.util.concurrent.TimeUnit;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.ui.Select;

public class mouseover {

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

// TODO Auto-generated method stub

**Select**: System.setProperty("webdriver.chrome.driver","C:\\SeleniumTraining\\chromedriver.exe");

WebDriver d=new ChromeDriver();

d.get("http://brm.tremplintech.in/web\_pages/ord\_reg.aspx");

d.findElement(By.xpath("//\*[@id='txt\_unam']")).sendKeys("sylix");

d.findElement(By.xpath("//\*[@id='txt\_pass']")).sendKeys("admin");

d.findElement(By.xpath("//\*[@id=\'Button3\']")).click();

Thread.sleep(6000);

//d.switchTo().alert().accept();

d.findElement(By.xpath("//\*[@id=\"hmenu\"]/ul/li[2]/a")).click();

Thread.sleep(6000);

Select state=new Select(d.findElement(By.id("ContentPlaceHolder1\_ddl\_state")));

//state.selectByIndex(7);

state.selectByVisibleText("Assam");

Thread.sleep(65000);

**MouseOver:**

d.get("http://executeautomation.com/demosite/index.html?UserName=Rajkumar&Password=colbuddy9&Login=Login");

//d.get("http://executeautomation.com/demosite/index.html?UserName=Rajkumar&Password=coolbuddy9&Login=Login");

d.manage().window().maximize();

//Thread.sleep(30000);

//d.manage().timeouts().implicitlyWait(15, TimeUnit.SECONDS);

d.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

Actions a=new Actions(d);

WebElement moveonmenu = d.findElement(By.xpath(".//\*[@id='Automation Tools']"));

a.moveToElement(moveonmenu).build().perform();

Thread.sleep(3000);

WebElement elem2 = d.findElement(By.xpath("//\*[@id=\"cssmenu\"]/ul/li[2]/ul/li[2]/a"));

a.moveToElement(elem2).build().perform();

Thread.sleep(3000);

WebElement elem3 = d.findElement(By.xpath("//\*[@id=\"cssmenu\"]/ul/li[2]/ul/li[2]/ul/li[2]/a/span"));

a.moveToElement(elem3).click().build().perform();

d.get("https://www.amazon.in");

Thread.sleep(8000);

JavascriptExecutor jse = (JavascriptExecutor)d;

jse.executeScript("window.scrollBy(0,3250)", "");

Thread.sleep(4000);

jse.executeScript("window.scrollBy(0,-1250)", "");

d.get("http://www.facebook.com");

File src= ((TakesScreenshot)d).getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(src, new File("C:/newinvalid.png"));

d.navigate().refresh();

d.get("http://www.india.com");

File src1= ((TakesScreenshot)d).getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(src1, new File("C:/invalid.png"));

d.navigate().back();

d.navigate().refresh();

d.navigate().forward();

\*/

**Move to Tab**d.get("http://executeautomation.com/demosite/index.html?UserName=Rajkumar&Password=coolbuddy9&Login=Login");

d.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

Thread.sleep(3000);

//Code to open a new link from current page to new page

((JavascriptExecutor)d).executeScript("window.open()");

ArrayList<String> tabs = new ArrayList<String>(d.getWindowHandles());

d.switchTo().window(tabs.get(1));

d.get("http://google.com");

Thread.sleep(3000);

d.navigate().refresh();

//To stay and move to the current second window

((JavascriptExecutor)d).executeScript("window.open()");

ArrayList<String> tabs1 = new ArrayList<String>(d.getWindowHandles());

d.switchTo().window(tabs1.get(2));

d.get("http://google.com");

Thread.sleep(3000);

//To move back to the parent window

ArrayList<String> tab = new ArrayList<String>(d.getWindowHandles());

d.switchTo().window(tab.get(0));

d.get("https://www.amazon.in");

d.navigate().refresh();

Thread.sleep(3000);

ArrayList<String> tab2 = new ArrayList<String>(d.getWindowHandles());

d.switchTo().window(tab2.get(2));

Thread.sleep(3000);

**Navigate Function:** d.get("https://www.amazon.in/gp/product/B07DWP6GWB/ref=s9\_acss\_bw\_cg\_Sneakpee\_2b1\_w?pf\_rd\_m=A1K21FY43GMZF8&pf\_rd\_s=merchandised-search-5&pf\_rd\_r=J62K17T4SV2WKCT4C7VC&pf\_rd\_t=101&pf\_rd\_p=89ca6c5f-3ec5-4fb9-ba0f-268b8c8b0ffc&pf\_rd\_i=1389396031");

d.navigate().back();

d.navigate().refresh();

}

}

**DD Process through JXL**

**package** ddp;

**import**java.io.File;

**import**java.io.FileInputStream;

**import**java.io.FileNotFoundException;

**import**java.io.IOException;

**import**org.apache.commons.io.FileUtils;

**import**org.openqa.selenium.By;

**import**org.openqa.selenium.OutputType;

**import**org.openqa.selenium.TakesScreenshot;

**import**org.openqa.selenium.WebDriver;

**import**org.openqa.selenium.chrome.ChromeDriver;

**import**jxl.Sheet;

**import**jxl.Workbook;

**import**jxl.read.biff.BiffException;

**publicclass** ddc {

**publicstaticvoid** main(String[] args) **throws** BiffException, IOException {

// **TODO** Auto-generated method stub

System.*setProperty*("webdriver.chrome.driver", "C:\\Chrome.exe\\chromedriver.exe");

WebDriver d=**new**ChromeDriver();

d.get("http://brm.tremplintech.in/web\_pages/ord\_reg.aspx");

FileInputStream f= **new** FileInputStream("C:\\SeleniumTraining\\ddjxl1.xls");

Workbook b=Workbook.*getWorkbook*(f);

Sheet s = b.getSheet(0);

**int**rowcount =s.getRows();

**for**(**int**i=1; i<rowcount; i++){

String username= s.getCell(0, i).getContents();

String password=s.getCell(1, i).getContents();

String cid=s.getCell(2, i).getContents();

String pid=s.getCell(3,i).getContents();

//String cname=s.getCell(4, i).getContents();

d.findElement(By.*xpath*("//\*[@id=\"txt\_unam\"]")).sendKeys(username);

d.findElement(By.*xpath*("//\*[@id='txt\_pass']")).sendKeys(password);

d.findElement(By.*xpath*("//\*[@id=\'Button3\']")).click();

d.findElement(By.*xpath*("//\*[@id=\"LinkButton1\"]")).click();

//d.findElement(By.xpath("//\*[@id=\"LinkButton1\"]")).click();

/\*d.findElement(By.xpath("//\*[@id=\"ContentPlaceHolder1\_txt\_customerid\"]")).sendKeys(cid);

d.findElement(By.xpath("//\*[@id=\"ContentPlaceHolder1\_txt\_phoneno\"]")).sendKeys(pid);

d.findElement(By.xpath("//\*[@id=\"ContentPlaceHolder1\_but\_insert\"]")).click();\*/

}

}

}

Window Handling

**package** seleniumpack;

**import**java.util.Set;

**import**org.openqa.selenium.By;

**import**org.openqa.selenium.WebDriver;

**import**org.openqa.selenium.chrome.ChromeDriver;

**publicclass** getwindowhandle {

**publicstaticvoid** main(String[] args) **throws** InterruptedException {

System.*setProperty*("webdriver.chrome.driver","C:\\selenium\\chromedriver.exe");

WebDriver d=**new**ChromeDriver();

d.navigate().to("http://www.leafground.com/pages/Window.html");

Thread.*sleep*(4000);

String oldwindow=d.getWindowHandle();

d.findElement(By.*id*("home")).click();

Set <String>handles=d.getWindowHandles();

**for** (String newwindow :handles)

{

d.switchTo().window(newwindow);

}

d.findElement(By.*xpath*("//\*[@id=\"post-153\"]/div[2]/div/ul/li[1]/a/img")).click();

Thread.*sleep*(3000);//EDIT

d.close();

d.switchTo().window(oldwindow);

d.findElement(By.*xpath*("//\*[@id=\"contentblock\"]/section/div[2]/div/div/button")).click();

**int**number= d.getWindowHandles().size() ;

System.***out***.println("NO OF OPENED WINDOWS="+number);//COUNT

d.findElement(By.*id*("color")).click();

Set<String>newwindows=d.getWindowHandles();

**for** (String child :newwindows)

{

**if**(!child.equals(oldwindow))

{

d.switchTo().window(child);

d.close();

}

}

}

}

**Link Text**

package seleniumpack;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class linktext {

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

// TODO Auto-generated method stub

String baseUrl = "http://demo.guru99.com/test/link.html";

System.setProperty("webdriver.chrome.driver", "C:\\selenium\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.get(baseUrl);

Thread.sleep(12000);

driver.findElement(By.linkText("click here")).click();

System.out.println("title of page is: " + driver.getTitle());

//driver.quit();

}

**Slider**

package seleniumpack;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.interactions.Actions;

public class slider {

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

// TODO Auto-generated method stub

System.setProperty("webdriver.chrome.driver","C:\\selenium\\chromedriver.exe");

WebDriver driver=new ChromeDriver();

driver.get("http://only-testing-blog.blogspot.in/2014/09/selectable.html");

//WebElement dragElementFrom = driver.findElement(By.xpath("//span[contains(@class, 'ui-slider-handle')]"));

WebElement dragElementFrom = driver.findElement(By.xpath("//\*[@id=\"slider\"]/span"));

Thread.sleep(5000);

//To Move jQuery slider by 100 pixel offset using dragAndDropBy method of Actions class.

// new Actions(driver).dragAndDropBy(dragElementFrom, 200, 0).build().perform();

//Thread.sleep(10000);

//After 5 seconds, This will Move jQuery slider by 100 pixel offset using the combination of clickAndHold, moveByOffset and release methods of Actions class.

new Actions(driver).clickAndHold(dragElementFrom).moveByOffset(500,0).release().perform();

}

}

**Tag Element:**

package seleniumpack;

import java.util.List;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class tagelement {

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

System.setProperty("webdriver.chrome.driver","C:\\selenium\\chromedriver.exe");

WebDriver a = new ChromeDriver();

a.get("http://www.leafground.com/pages/autoComplete.html");

Thread.sleep(4000);

a.manage().window().maximize();

a.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

WebElement in=a.findElement(By.id("tags"));

in.sendKeys("R");

List<WebElement> li=a.findElements(By.xpath("//\*[@id=\"ui-id-7\"]/li"));

//List<WebElement> li=a.findElements(By.xpath("//\*[@id=\"ui-id-1\"]/li"));

System.out.println(li.size());

Thread.sleep(4000);

for (WebElement e :li)

{

if(e.getText().equals("Selenium"))

{

e.click();

System.out.println("yes");

break;

}

else

{

System.out.println("NOT");

}

}

}

}

Create a project and create two packages in that one for Pages and another one for Main class

Project Structure like:

POMProject(project name)

Pages(package1)

Homepage(page 1)

Loginpage(page 2)

Test(Package 2)

TestLogin(Main Class)

**1.Pages Package -> Homepage**

**package** Pages;

**import**org.openqa.selenium.By;

**import**org.openqa.selenium.WebDriver;

**publicclass** Homepage {

WebDriver driver;

By homePageUserName = By.*xpath*("//table//tr[@class='heading3']");

**public**Homepage(WebDriver driver){

**this**.driver = driver;

}

//Get the User name from Home Page

**public**StringgetHomePageDashboardUserName(){

**return**driver.findElement(homePageUserName).getText();

}

}

**2.Pages Package->Loginpage**

**package** Pages;

**import**org.openqa.selenium.By;

**import**org.openqa.selenium.WebDriver;

**publicclass** Loginpage {

WebDriver driver;

By user99GuruName = By.*name*("uid");

By password99Guru = By.*name*("password");

By titleText =By.*className*("barone");

By login = By.*name*("btnLogin");

**public**Loginpage(WebDriver driver){

**this**.driver = driver;

}

//Set user name in textbox

**publicvoid**setUserName(String strUserName){

driver.findElement(user99GuruName).sendKeys(strUserName);

}

//Set password in password textbox

**publicvoid**setPassword(String strPassword){

driver.findElement(password99Guru).sendKeys(strPassword);

}

//Click on login button

**publicvoid**clickLogin(){

driver.findElement(login).click();

}

//Get the title of Login Page

**public** String getLoginTitle(){

**return**driver.findElement(titleText).getText();

}

/\*\*

\* This POM method will be exposed in test case to login in the application

\* **@param** strUserName

\* **@param** strPasword

\* **@return**

\*/

**publicvoid** loginToGuru99(String strUserName,StringstrPasword){

//Fill user name

**this**.setUserName(strUserName);

//Fill password

**this**.setPassword(strPasword);

//Click Login button

**this**.clickLogin();

}

}

**3.Test Package->Test Login(Main Class)**

package Test;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

import Pages.Homepage;

import Pages.Loginpage;

public class TestLogin {

String driverPath = "C:\\geckodriver.exe";

WebDriver driver;

Loginpage objLogin;

Homepage objHomePage;

@BeforeTest

public void setup(){

System.setProperty("webdriver.chrome.driver","C:\\selenium\\chromedriver.exe");

WebDriver driver=new ChromeDriver();

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

driver.get("http://demo.guru99.com/V4/");

}

/\*\*

\* This test case will login in http://demo.guru99.com/V4/

\* Verify login page title as guru99 bank

\* Login to application

\* Verify the home page using Dashboard message

\*/

@Test(priority=0)

public void test\_Home\_Page\_Appear\_Correct(){

//Create Login Page object

objLogin = new Loginpage(driver);

//Verify login page title

String Expectedresult="Guru99 Bank Home Page";

String loginPageTitle = objLogin.getLoginTitle();

Assert.assertEquals(loginPageTitle,Expectedresult );

//Assert.assertTrue(loginPageTitle.toLowerCase().contains("guru99 bank home page"));

//login to application

objLogin.loginToGuru99("mgr123", "mgr!23");

// go the next page

objHomePage = new Homepage(driver);

//Verify home page

Assert.assertTrue(objHomePage.getHomePageDashboardUserName().toLowerCase().contains("manger id : mgr123"));

}

}

POI Program:

Create Project DDPOI and Create four class to create Excel File, Open Excel File, Write Excel file, Read Excel File.

Create Package-> PoiOUT

First Class: CreateWorkBook

Second Class: OpenWorkBook

Third Class: Writesheet

Forth Class: Readsheet

Class to Create Workbook Lin.xlsx

**package** PoiOUT;

**import**java.io.File;

**import**java.io.FileOutputStream;

**import**org.apache.poi.xssf.usermodel.XSSFWorkbook;

**publicclass** CreateWorkBook {

**publicstaticvoid** main(String[] args)**throws** Exception {

//Create Blank workbook

XSSFWorkbook workbook = **new**XSSFWorkbook();

//Create file system using specific name

FileOutputStream out = **new**FileOutputStream(**new** File("Lin.xlsx"));

//write operation workbook using file out object

workbook.write(out);

out.close();

System.***out***.println("Lin.xlsx Create successfully");

}

}

Second Class To open Lin.xlsx:

**package** PoiOUT;

**import**java.io.File;

**import**java.io.FileInputStream;

**import**org.apache.poi.xssf.usermodel.XSSFWorkbook;

**publicclass** OpenWorkBook {

**publicstaticvoid**main(String args[])**throws** Exception {

File file = **new** File("Lin.xlsx");

FileInputStream fIP = **new** FileInputStream(file);

//Get the workbook instance for XLSX file

XSSFWorkbook workbook = **new** XSSFWorkbook(fIP);

**if**(file.isFile() &&file.exists()) {

System.***out***.println("Lin.xlsx file open successfully.");

} **else** {

System.***out***.println("Error to open openworkbook.xlsx file.");

}

}

}

Third Class to write excel file here employee profile data write by Set Collection through TreeSet.

package PoiOUT;

import java.io.File;

import java.io.FileOutputStream;

import java.util.Map;

import java.util.Set;

import java.util.TreeMap;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.xssf.usermodel.XSSFRow;

import org.apache.poi.xssf.usermodel.XSSFSheet;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

public class Writesheet {

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

//Create blank workbook

XSSFWorkbook workbook = new XSSFWorkbook();

//Create a blank sheet

XSSFSheet spreadsheet = workbook.createSheet(" Employee Info ");

//Create row object

XSSFRow row;

//This data needs to be written (Object[])

Map < String, Object[] > empinfo =

new TreeMap < String, Object[] >();

empinfo.put( "1", new Object[] { "EMP ID", "EMP NAME", "DESIGNATION" });

empinfo.put( "2", new Object[] { "tp01", "jiji", "Technical Manager" });

empinfo.put( "3", new Object[] { "tp02", "Malathi", "Proof Reader" });

empinfo.put( "4", new Object[] { "tp03", "Prdeep", "Technical Writer" });

empinfo.put( "5", new Object[] { "tp04", "Sarathi", "Technical Writer" });

empinfo.put( "6", new Object[] { "tp05", "Kriti", "Technical Writer" });

//Iterate over data and write to sheet

Set < String > keyid = empinfo.keySet();

int rowid = 0;

for (String key : keyid) {

row = spreadsheet.createRow(rowid++);

Object [] objectArr = empinfo.get(key);

int cellid = 0;

for (Object obj : objectArr) {

Cell cell = row.createCell(cellid++);

cell.setCellValue((String)obj);

}

}

//Write the workbook in file system

FileOutputStream out = new FileOutputStream(new File("Lin.xlsx"));

workbook.write(out);

out.close();

System.out.println("Lin.xlsx written successfully");

}

}

Forth class for Read data by using POI and Display excel data in console:

package PoiOUT;

import java.io.File;

import java.io.FileInputStream;

import java.util.Iterator;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.ss.usermodel.Row;

import org.apache.poi.xssf.usermodel.XSSFRow;

import org.apache.poi.xssf.usermodel.XSSFSheet;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

public class Readsheet {

static XSSFRow row;

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

FileInputStream fis = new FileInputStream(new File("Lin.xlsx"));

XSSFWorkbook workbook = new XSSFWorkbook(fis);

XSSFSheet spreadsheet = workbook.getSheetAt(0);

Iterator < Row > rowIterator = spreadsheet.iterator();

while (rowIterator.hasNext()) {

row = (XSSFRow) rowIterator.next();

Iterator < Cell > cellIterator = row.cellIterator();

while ( cellIterator.hasNext()) {

Cell Cell = (Cell)cellIterator.next();

switch (Cell.getCellType()) {

// case Cell.CELL\_TYPE\_NUMERIC:

case STRING:

System.out.print(Cell.getStringCellValue() + " \t\t ");

break;

case NUMERIC:

// case Cell.CELL\_TYPE\_STRING:

System.out.print(Cell.getNumericCellValue() + " \t\t ");

break;

default:

break;

}

}

System.out.println();

}

fis.close();

}

}