Important Examples (Run and Explain)–

Create table query –

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) `java\_db`.`emp` (`id` INT(10) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL ,

`first` VARCHAR(20) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL , `last` VARCHAR(20) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL ,

`age` INT(3) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL ) ENGINE = InnoDB;

jdbc\_test1

import java.sql.\*;

public class jdbc\_test1 {

static final String DB\_URL = "jdbc:mysql://localhost/java\_db";

static final String USER = "root";

static final String PASS = "";

static final String QUERY = "SELECT id, first, last, age FROM Employees";

public static void main(String[] args) {

// Open a connection

try(Connection conn = DriverManager.getConnection(DB\_URL, USER, PASS);

Statement stmt = conn.createStatement();) {

ResultSet rs = stmt.executeQuery(QUERY);

int sum = 0;

while (rs.next()) {

// Retrieve by column name

int a=rs.getInt("id"),b=rs.getInt("age");

System.out.print("ID: " + rs.getInt("id"));

System.out.print(", Age: " + rs.getInt("age"));

System.out.print(", First: " + rs.getString("first"));

System.out.println(", Last: " + rs.getString("last"));

}

} catch (SQLException e) {

e.printStackTrace();

}

}

}

jdbc\_test2

import java.sql.\*;

public class jdbc\_test2 {

static final String DB\_URL = "jdbc:mysql://localhost/java\_db";

static final String USER = "root";

static final String PASS = "";

static final String QUERY = "INSERT INTO Employees (Id,first,last,age) values ('6','chandler','bing','55')";

static final String QUERY2 = "DELETE FROM `Employees` WHERE `employees`.`ID` = 6";

static final String QUERY3 = "UPDATE Employees SET first='friends' where id=6";

public static void main(String[] args) {

// Open a connection

try(Connection conn = DriverManager.getConnection(DB\_URL, USER, PASS);

Statement stmt = conn.createStatement();) {

int i = stmt.executeUpdate(QUERY);

System.out.println("insert happened");

} catch (SQLException e) {

System.out.println("insert not happening");

try(Connection conn = DriverManager.getConnection(DB\_URL, USER, PASS);

Statement stmt = conn.createStatement();){

int i = stmt.executeUpdate(QUERY2);

System.out.println("delete happened now again inserting");

int j = stmt.executeUpdate(QUERY);

System.out.println("insert finally happened");

}

catch (SQLException e1){

System.out.println("even delete did not happen");

e1.printStackTrace();

}

e.printStackTrace();

}

try(Connection conn = DriverManager.getConnection(DB\_URL, USER, PASS);

Statement stmt = conn.createStatement();){

int i = stmt.executeUpdate(QUERY3);

System.out.println("update happened");

}

catch (SQLException e1){

e1.printStackTrace();

}

}

}

jdbc\_test3

import java.sql.\*;

public class jdbc\_test3 {

static final String DB\_URL = "jdbc:mysql://localhost/java\_db";

static final String USER = "root";

static final String PASS = "";

static final String QUERY = "CREATE TABLE REGISTRATION (id INTEGER not NULL, first VARCHAR(255), last VARCHAR(255), age INTEGER, PRIMARY KEY ( id ))"; ;

public static void main(String[] args) {

// Open a connection

try(Connection conn = DriverManager.getConnection(DB\_URL, USER, PASS);

Statement stmt = conn.createStatement();) {

boolean bool = stmt.execute(QUERY);

if (!bool) System.out.println("table created");

else System.out.println("table error");

} catch (SQLException e) {

e.printStackTrace();

}

}

}

prob1\_swing

import java.awt.event.\*;

import java.awt.\*;

import javax.swing.\*;

class prob1\_swing extends JFrame {

static JFrame f;

static JLabel l;

prob1() {

}

public static void main(String[] args) {

f = new JFrame("label chk");

l = new JLabel();

l.setText("label text");

JPanel p = new JPanel();

p.add(l);

/\*JLabel l1 = new JLabel();

l1.setText("label text1");

JLabel l2 = new JLabel();

l2.setText("label text2");

p.add(l1);

p.add(l2);\*/

p.setLayout(new BorderLayout());

JLabel l1 = new JLabel();

l1.setText("label text1");

JLabel l2 = new JLabel();

l2.setText("label text2");

p.add(l1,BorderLayout.SOUTH);

p.add(l2,BorderLayout.CENTER);

p.add(l,BorderLayout.EAST);

f.add(p);

f.setSize(500, 500);

f.show();

f.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

}

prob2\_swing

import java.awt.event.\*;

import java.awt.\*;

import javax.swing.\*;

class prob2\_swing extends JFrame {

JFrame f;

JLabel l,l2;

JButton b;

int count;

prob2\_swing() {

f = new JFrame("button chk");

l = new JLabel();

l.setText("Button Count");

JPanel p = new JPanel();

p.add(l);

l2 = new JLabel();

l2.setText("Press Count -"+count);

p.add(l2);

b = new JButton();

b.setText("Increment");

b.addActionListener(new ButtonListener());

p.add(b);

f.add(p);

f.setSize(500, 500);

f.show();

f.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

// main class

public static void main(String[] args) {

new prob2\_swing();

}

private class ButtonListener implements ActionListener {

public void actionPerformed(ActionEvent event) {

count++;

l2.setText("Press Count -"+count);

}

}

}

prob3\_swing

import java.awt.event.\*;

import java.awt.\*;

import javax.swing.\*;

class prob3\_swing extends JFrame {

JFrame f;

JLabel l,l2,l3;

JButton b;

JTextField x,y;

prob3\_swing() {

f = new JFrame("text input chk");

l = new JLabel();

l.setText("x");

JPanel p = new JPanel();

p.setLayout(new GridLayout(6,1));

p.add(l);

x = new JTextField();

p.add(x);

l2 = new JLabel();

l2.setText("y");

p.add(l2);

y = new JTextField();

p.add(y);

b = new JButton();

b.setText("add");

b.addActionListener(new ButtonListener());

p.add(b);

l3 = new JLabel();

l3.setText("SUM-None");

p.add(l3);

l.setHorizontalAlignment(JLabel.CENTER);

l.setVerticalAlignment(JLabel.CENTER);

l2.setHorizontalAlignment(JLabel.CENTER);

l2.setVerticalAlignment(JLabel.CENTER);

l3.setHorizontalAlignment(JLabel.CENTER);

l3.setVerticalAlignment(JLabel.CENTER);

b.setHorizontalAlignment(JButton.CENTER);

b.setVerticalAlignment(JButton.CENTER);

x.setHorizontalAlignment(JTextField.CENTER);

y.setHorizontalAlignment(JTextField.CENTER);

f.add(p);

f.setSize(500, 500);

f.pack();

f.show();

f.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

// main class

public static void main(String[] args) {

new prob3\_swing();

}

private class ButtonListener implements ActionListener {

public void actionPerformed(ActionEvent event) {

int x\_new = Integer.parseInt(x.getText());

int y\_new = Integer.parseInt(y.getText());

l3.setText("SUM-"+(x\_new+y\_new));

}

}

}

prob4\_swing

import java.awt.event.\*;

import java.awt.\*;

import javax.swing.\*;

import javax.swing.JFrame;

class prob4\_swing extends JFrame implements ItemListener{

static JFrame f;

static JLabel l, l1, l3, l4;

static JComboBox c1, c2;

public static void main(String[] args) {

f = new JFrame("frame");

prob4 s = new prob4();

String s1[] = { "Jalpaiguri", "Mumbai", "Noida", "Kolkata", "New Delhi" };

String s2[] = { "male", "female", "others" };

c1 = new JComboBox(s1);

c2 = new JComboBox(s2);

c1.setSelectedIndex(3);

c2.setSelectedIndex(0);

c1.addItemListener(s);

c2.addItemListener(s);

c1.setEditable(true);

l = new JLabel("select your city ");

l1 = new JLabel("Jalpaiguri selected");

l3 = new JLabel("select your gender ");

l4 = new JLabel("Male selected");

l.setForeground(Color.red);

l1.setForeground(Color.blue);

l3.setForeground(Color.red);

l4.setForeground(Color.blue);

JPanel p = new JPanel();

p.add(l);

p.add(c1);

p.add(l1);

p.add(l3);

p.add(c2);

p.add(l4);

p.setLayout(new FlowLayout());

f.add(p);

f.setSize(400, 400);

f.show();

f.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

//c1.addItem("currc1");

}

public void itemStateChanged(ItemEvent e) {

// if the state combobox 1 is changed

if (e.getSource() == c1) {

l1.setText(c1.getSelectedItem() + " selected");

c1.addItem("currc1");

}

// if state of combobox 2 is changed

else {

l4.setText(c2.getSelectedItem() + " selected");

}

}

}

prob5\_swing

import java.awt.event.\*;

import java.awt.\*;

import javax.swing.\*;

class prob5\_swing extends JFrame implements ItemListener {

static JFrame f;

static JLabel l, l1;

static JCheckBox c1, c2;

public static void main(String[] args) {

f = new JFrame("frame");

prob5 s = new prob5();

f.setLayout(new FlowLayout());

c1 = new JCheckBox("checkbox 1", false);

c2 = new JCheckBox("checkbox 2", false);

c1.addItemListener(s);

c2.addItemListener(s);

l = new JLabel("checkbox1 not selected");

l1 = new JLabel("checkbox2 not selected");

l.setForeground(Color.red);

l1.setForeground(Color.blue);

JPanel p = new JPanel();

p.add(c1);

p.add(c2);

p.add(l);

p.add(l1);

f.add(p);

f.setSize(600, 300);

f.show();

f.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

public void itemStateChanged(ItemEvent e) {

// if the state of checkbox1 is changed

if (e.getSource() == c1) {

if (e.getStateChange() == 1)

l.setText("checkbox 1 selected");

else

l.setText("checkbox 1 not selected");

}

// if the state of checkbox2 is changed

else {

if (e.getStateChange() == 1)

l1.setText("checkbox 2 selected");

else

l1.setText("checkbox 2 not selected");

}

}

}

prob6\_swing

import java.awt.event.\*;

import java.awt.\*;

import javax.swing.\*;

class prob6 extends JFrame implements ItemListener,ActionListener {

static JFrame f;

static JLabel l, l1;

static JCheckBox c1, c2;

static JButton b1;

public static void main(String[] args) {

f = new JFrame("frame");

prob6 s = new prob6();

f.setLayout(new FlowLayout());

c1 = new JCheckBox("checkbox 1", false);

c2 = new JCheckBox("checkbox 2", false);

c1.addItemListener(s);

c2.addItemListener(s);

l = new JLabel("checkbox1 not selected");

l1 = new JLabel("checkbox2 not selected");

l.setForeground(Color.red);

l1.setForeground(Color.blue);

b1 = new JButton();

b1.setText("final state");

b1.addActionListener(s);

JPanel p = new JPanel();

p.add(c1);

p.add(c2);

p.add(l);

p.add(l1);

p.add(b1);

f.add(p);

f.setSize(600, 300);

f.show();

f.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

public void itemStateChanged(ItemEvent e) {

// if the state of checkbox1 is changed

if (e.getSource() == c1) {

if (e.getStateChange() == 1)

l.setText("checkbox 1 selected");

else

l.setText("checkbox 1 not selected");

}

// if the state of checkbox2 is changed

else {

if (e.getStateChange() == 1)

l1.setText("checkbox 2 selected");

else

l1.setText("checkbox 2 not selected");

}

}

public void actionPerformed(ActionEvent event) {

l.setText(c1.getLabel()+"-"+c1.isSelected());

l1.setText(c2.getLabel()+"-"+c2.isSelected());

}

}

Basic1

import java.io.\*;

class basic1 implements Serializable {

int id;

String name;

static int count;

basic1(int id, String name){

this.id = id;

this.name = name;

count++;

}

void display(){

System.out.println("Created object name-"+this.name+

" id-"+this.id);

System.out.println("Objects no.-"+count);

}

static int area(int n){

System.out.println("square-"+n\*n);

return n\*n;

}

static int area(int n, int o){

System.out.println("rectangle-"+n\*o);

return n\*o;

}

static double area(double n, double o){

System.out.println("triangle-"+0.5\*n\*o);

return 0.5\*n\*o;

}

static void area(double n){

System.out.println("circle-"+3.14\*n\*n);

}

public static void main(String[] args) {

basic1 obj = new basic1(123,"mine");

obj.display();

/\*System.out.println(area(obj.id));

System.out.println(area(obj.id,obj.id));

System.out.println(area((float)obj.id,(float)obj.id));

area((long)obj.id);\*/

try {

obj.file\_io(obj);

}

catch(IOException e) {

e.printStackTrace();

}

}

void file\_io(basic1 obj) throws IOException {

File file = new File("curr.dat");

FileOutputStream fileStream1 = new FileOutputStream(file);

ObjectOutputStream objStream1 = new ObjectOutputStream(fileStream1);

obj.id++;

objStream1.writeObject(obj);

objStream1.close();

FileInputStream fileStream2 = new FileInputStream(file);

ObjectInputStream objStream2 = new ObjectInputStream(fileStream2);

//Object new\_obj = objStream2.readObject();

try{

Object new\_obj = objStream2.readObject();

//new\_obj.display();

basic1 obj2 = (basic1) new\_obj;

obj2.display();

}

catch(Exception e){

e.printStackTrace();

}

objStream2.close();

}

}

Basic2

class Parent {

Parent(){

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

this.m1();

}

private void m1() {

System.out.println("From parent m1()");

}

protected void m2() {

System.out.println("From parent m2()");

}

}

class Child extends Parent {

public void m1() {

System.out.println("From child m1()");

}

@Override

public void m2() {

System.out.println("From child m2()");

}

public void m3() {

System.out.println("From child m3()");

}

}

class basic2 {

public static void main(String[] args) {

Parent obj1 = new Parent();

obj1.m2();

//Child obj3 = new Child();

//obj3.m3();

//Parent obj2 = new Child();

//obj2.m2();

//obj2.m3();

// obj2.m3();

// obj1.m1();

// obj3.m1();

}

}

test\_abs

public class test\_abs extends Product {

test\_abs(int pid, String n) {

super(pid, n);

}

public void display() {

System.out.println("Product Id = "+pro\_Id + " " + " Product Name = "+pro\_name);

}

public void display1() {

System.out.println("Child");

//super.display1();

}

public static void main(String args[]) {

test\_abs[] obj = new test\_abs[5];

obj[0] = new test\_abs(23907,"Dell Laptop");;

System.out.println("Product Object:");

obj[0].display();

obj[0].display1();

Product x = (Product)new test\_abs(23907,"Dell Laptop");

x.display1();

}

}

abstract class Product {

int pro\_Id;

String pro\_name;

Product(int pid, String n) {

pro\_Id = pid; pro\_name = n;

}

abstract void display();

public void display1() {

System.out.print("Parent");

}

}

Test1\_abs

abstract class Sunstar {

//abstract int chk1 = 23;

int chk2 = 46;

abstract void printInfo();

Sunstar(){

System.out.println("abstract class constructor");

}

}

class Employee extends Sunstar {

int chk2 = 23;

Employee(int x){

//super();

System.out.println(x);

}

void printInfo() {

String name = "avinash";

int age = 21;

float salary = 222.2F;

System.out.println(name);

System.out.println(age);

System.out.println(salary);

}

void abstract\_var(){

System.out.println(chk2);

System.out.println(super.chk2);

}

}

class test1\_abs {

public static void main(String args[]) {

//Sunstar s = new Employee();

//s.printInfo();

Employee x = new Employee(4);

//x.abstract\_var();

//System.out.println(Sunstar.chk2);

}

}

/\*class Employee2 extends Sunstar {

int printInfo() {

String name = "new";

int age = 23;

float salary = 222.2F;

System.out.println(name);

System.out.println(age);

System.out.println(salary);

return 1;

}

void printInfo(int n) {

String name = "new";

int age = 23;

float salary = 222.2F;

System.out.println(name);

System.out.println(age);

System.out.println(salary);

return n;

}

}\*/

Test\_inter

//Interface declaration: by first user

interface Drawable{

int a = 10;

void draw();

}

//Implementation: by second user

class Rectangle implements Drawable{

public void draw(){

System.out.println("drawing rectangle");

System.out.println(a++);

}

}

class Circle implements Drawable{

public void draw(){

System.out.println("drawing circle");

System.out.println(a);

}

}

//Using interface: by third user

public class test\_inter{

public static void main(String args[]){

Drawable d=new Circle();//In real scenario, object is provided by method e.g. getDrawable()

d.draw();

Drawable d1=new Rectangle();//In real scenario, object is provided by method e.g. getDrawable()

d1.draw();

}

}

Test\_inter2

// Java Program to Demonstrate Working of

// Comparator Interface

// Importing required classes

import java.io.\*;

import java.lang.\*;

import java.util.\*;

// Class 1

// A class to represent a Student

class Student {

// Attributes of a student

int rollno;

String name, address;

// Constructor

public Student(int rollno, String name, String address) {

// This keyword refers to current instance itself

this.rollno = rollno;

this.name = name;

this.address = address;

}

// Method of Student class

// To print student details in main()

public String toString() {

// Returning attributes of Student

return this.rollno + " " + this.name + " " + this.address;

}

}

// Class 2

// Helper class implementing Comparator interface

class Sortbyroll implements Comparator<Student> {

// Method

// Sorting in ascending order of roll number

public int compare(Student a, Student b) {

return a.rollno - b.rollno;

}

}

// Class 3

// Helper class implementing Comparator interface

class Sortbyname implements Comparator<Student> {

// Method

// Sorting in ascending order of name

public int compare(Student a, Student b) {

return a.name.compareTo(b.name);

}

}

// Class 4

// Main class

class test\_inter2 {

// Main driver method

public static void main(String[] args) {

// Creating an empty ArrayList of Student type

ArrayList<Student> ar = new ArrayList<Student>();

// Adding entries in above List

// using add() method

ar.add(new Student(111, "Mayank", "london"));

ar.add(new Student(131, "Anshul", "nyc"));

ar.add(new Student(121, "Solanki", "jaipur"));

ar.add(new Student(101, "Aggarwal", "Hongkong"));

// Display message on console for better readability

System.out.println("Unsorted");

// Iterating over entries to print them

for (int i = 0; i < ar.size(); i++)

System.out.println(ar.get(i));

// Sorting student entries by roll number

Collections.sort(ar, new Sortbyroll());

// Display message on console for better readability

System.out.println("\nSorted by rollno");

// Again iterating over entries to print them

for (int i = 0; i < ar.size(); i++)

System.out.println(ar.get(i));

// Sorting student entries by name

Collections.sort(ar, new Sortbyname());

// Display message on console for better readability

System.out.println("\nSorted by name");

// // Again iterating over entries to print them

for (int i = 0; i < ar.size(); i++)

System.out.println(ar.get(i));

}

}

Output –

Unsorted

111 Mayank london

131 Anshul nyc

121 Solanki jaipur

101 Aggarwal Hongkong

Sorted by rollno

101 Aggarwal Hongkong

111 Mayank london

121 Solanki jaipur

131 Anshul nyc

Sorted by name

101 Aggarwal Hongkong

131 Anshul nyc

111 Mayank london

121 Solanki Jaipur

Java File IO

Reading from file BufferedReader –

import java.io.\*;  
import java.util.\*;  
  
public class fileio\_test {  
 public static void main(String[] args) throws IOException {  
 // File path is passed as parameter  
 File file = new File(".\\test.txt");  
 // Creating an object of BufferedReader class  
 BufferedReader br = new BufferedReader(new FileReader(file));  
 // Declaring a string variable  
 String st;  
 // Condition holds true till  
 // there is character in a string  
 while ((st = br.readLine()) != null)  
 // Print the string  
 System.*out*.println(st);  
 }  
}

Reading from file Scanner –

import java.io.\*;  
import java.util.\*;  
  
public class fileio\_test {  
 public static void main(String[] args) throws IOException {  
 // pass the path to the file as a parameter  
 File file = new File(".\\test.txt");  
 Scanner sc = new Scanner(file);  
 while (sc.hasNextLine())  
 System.*out*.println(sc.next());  
 }  
}

Writing to file FileWriter –

import java.io.\*;  
import java.util.\*;  
  
public class fileio\_test {  
 public static void main(String[] args) throws IOException {  
 // Content to be assigned to a file  
 String text =  
 "this is just a check file\n" +  
 "this is line 2\n" +  
 "at line 3 we will end";  
  
 // Try block to check if exception occurs  
 try {  
 // Create a FileWriter object  
 // to write in the file  
 FileWriter fWriter = new FileWriter(".\\check.txt");  
  
 // Writing into file  
 fWriter.write(text);  
  
 // Closing the file writing connection  
 fWriter.close();  
  
 // Display message for successful execution of  
 // program on the console  
 System.*out*.println("File is created successfully with the content.");  
 }  
 // Catch block to handle if exception occurs  
 catch (IOException e) {  
 System.*out*.print(e.getMessage());  
 }  
 //now reading  
 File file = new File(".\\check.txt");  
 Scanner sc = new Scanner(file);  
 while (sc.hasNextLine())  
 System.*out*.println(sc.nextLine());  
  
 }  
}

Writing to file BufferedWriter –

import java.io.\*;  
import java.util.\*;  
  
public class fileio\_test {  
 public static void main(String[] args) throws IOException {  
 // Content to be assigned to a file  
 String text =  
 "this is just a check file\n" +  
 "this is line 2\n" +  
 "at line 3 we will end";  
  
 // Step 1: Create BufferedWriter object  
 BufferedWriter f\_writer = new BufferedWriter(new FileWriter(".\\check.txt"));  
 // Step 2: Write text(content) to file  
 f\_writer.write(text);  
 // Step 3: Display message showcasing  
 // successful execution of the program  
 System.*out*.println("File is created successfully with the content.");  
 // Step 4: Close the BufferedWriter object  
 f\_writer.close();  
  
 //now reading from file that was created previously  
 File file = new File(".\\check.txt");  
 Scanner sc = new Scanner(file);  
 while (sc.hasNextLine())  
 System.*out*.println(sc.nextLine());  
 }  
}