1. Multithreading :  
   import java.util.\*;

class Thread4 extends Thread {

public void run() {

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

if(i %2==0) {

System.***out***.println("Even: "+i);

}

try {

Thread.*sleep*(500);

}

catch(Exception e){

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

}

}

}

}

class Thread5 implements Runnable{

public void run() {

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

if((i %2)!= 0) {

System.***out***.println("Even: "+i);

}

try {

Thread.*sleep*(500);

}

catch(Exception e){

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

}

}

}

}

public class Example {

public static void main(String[] args) {

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

Thread4 t = new Thread4();

t.start();

Thread5 t2 =new Thread5();

Thread t3 = new Thread(t2);

t3.start();

}

}

2)AWT:

import java.awt.\*;

import java.awt.event.\*;

public class Awt extends Frame {

// Constructor to set up the frame

public Awt() {

// Set the title of the frame

setTitle("Geometric Shapes using AWT");

// Set the size of the frame

setSize(600, 400);

setLocationRelativeTo(null); //places figure in center;

// Close the application when the user clicks the close button

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent we) {

dispose();

}

});

// Make the frame visible

setVisible(true);

}

// Overriding the paint() method to draw shapes

public void paint(Graphics g) {

// Set font for the text

Font customFont = new Font("Arial", Font.BOLD | Font.ITALIC, 18); // Create a new font (Arial, bold and italic, size 18)

g.setFont(customFont); // Set the font for drawing text

// Drawing a Rectangle (Outline)

g.setColor(Color.RED); // Set color to red

g.drawRect(50, 50, 200, 100); // Draw a rectangle at (50, 50) with width 200 and height 100

// Drawing a Filled Rectangle

g.setColor(Color.YELLOW); // Set color to yellow

g.fillRect(300, 50, 200, 100); // Draw a filled rectangle at (300, 50) with width 200 and height 100

// Drawing an Oval (Outline)

g.setColor(Color.BLUE); // Set color to blue

g.drawOval(50, 200, 200, 100); // Draw an oval at (50, 200) with width 200 and height 100

// Drawing a Filled Oval

g.setColor(Color.GREEN); // Set color to green

g.fillOval(300, 200, 200, 100); // Draw a filled oval at (300, 200) with width 200 and height 100

// Drawing a Line

g.setColor(Color.BLACK); // Set color to black

g.drawLine(50, 350, 550, 350); // Draw a line from (50, 350) to (550, 350)

// Drawing Text

g.setColor(Color.MAGENTA); // Set color to magenta

g.drawString("Hello, AWT Graphics!", 200, 380); // Draw text at (200, 380)

}

public static void main(String[] args) {

// Create an instance of the GeometricShapesAWT class

new Awt();

}

}

3)File H:

import java.io.\*;

public class fileh {

public static void main(String args[]) {

String sources = "sources.rtf";

String destination = "destination.rtf";

try {

BufferedReader reader = new BufferedReader(new FileReader(sources));

BufferedWriter writer = new BufferedWriter(new FileWriter(destination));

String line;

while ((line = reader.readLine()) != null) {

writer.write(line);

writer.newLine();

}

reader.close();

writer.close();

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

} catch (IOException e) {

System.***out***.println("error occurred");

e.printStackTrace();

}

}

}

4)Matrix Addition:

import java.util.\*;

public class Matrixadd {

public static void main(String[] args) {

// Initialize Scanner to take user input

Scanner scanner = new Scanner(System.***in***);

// Taking matrix dimensions input

System.***out***.print("Enter the number of rows: ");

int rows = scanner.nextInt();

System.***out***.print("Enter the number of columns: ");

int cols = scanner.nextInt();

// Create two matrices of the given size

int[][] matrix1 = new int[rows][cols];

int[][] matrix2 = new int[rows][cols];

int[][] resultMatrix = new int[rows][cols]; // Resultant matrix for storing the sum

// Input matrix 1

System.***out***.println("Enter elements of matrix 1:");

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

for (int j = 0; j < cols; j++) {

System.***out***.print("Element at [" + i + "][" + j + "]: ");

matrix1[i][j] = scanner.nextInt();

}

}

// Input matrix 2

System.***out***.println("Enter elements of matrix 2:");

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

for (int j = 0; j < cols; j++) {

System.***out***.print("Element at [" + i + "][" + j + "]: ");

matrix2[i][j] = scanner.nextInt();

}

}

// Perform matrix addition

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

for (int j = 0; j < cols; j++) {

resultMatrix[i][j] = matrix1[i][j] + matrix2[i][j];

}

}

// Displaying the result matrix

System.***out***.println("\nResult of Matrix Addition:");

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

for (int j = 0; j < cols; j++) {

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

}

System.***out***.println(); // Move to the next line for each row

}

// Close the scanner

scanner.close();

}

}

5)Interface:

package pkg3;

import java.util.\*;

interface Sports{

void playsport();

}

interface Music {

void playmusic();

}

class Person implements Sports , Music {

String name,name2;

public Person(String name,String name2) {

this.name = name;

this.name2=name2;

}

public void playsport() {

System.***out***.println(name +" Plays cricket");

}

public void playmusic() {

System.***out***.println(name2 +" Plays Guitar");

}

}

public class Multiple {

public static void main (String [] args) {

Person obj = new Person("Tina","Rina");

obj.playsport();

obj.playmusic();

}

}

6) Exception :

package pkg5;

import java.util.\*;

public class Try {

public static void main(String[] args) {

try {

int a=10;

int b=0;

int c=a/b;

System.***out***.println("Result is :"+c);

}

catch(ArithmeticException e){

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

}

try {

String str =null;

System.***out***.println("This is error :"+str.toUpperCase());

}

catch(NullPointerException e){

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

}

try {

String str="Hello";

System.***out***.println("String:"+str.charAt(15));

}

catch(StringIndexOutOfBoundsException e){

System.***out***.println("String IOOB:");

}

try {

int arr []= new int[3];

arr[5]=10;

System.***out***.println("There is some error :");

}

catch(ArrayIndexOutOfBoundsException e) {

System.***out***.println("AIOOBE :");

}

finally {

System.***out***.println("This is finnaly block");

}

System.***out***.println("This is execution flow");

}

}

&

7) Matching ractangles:

import java.util.\*;

class Rect {

double width ;

double length;

double area;

String colour;

Scanner sc = new Scanner (System.***in***);

public void get\_length() {

System.***out***.println("Enter the lenth: ");

length = sc.nextDouble();

}

public void get\_width() {

System.***out***.println("Enter the witdh: ");

width = sc.nextDouble();

}

public double get\_area() {

area=length \* width;

System.***out***.println("your area is : "+(area));

return area;

}

public String get\_colour() {

System.***out***.println("Enter the colour: ");

colour = sc.next();

return colour;

}

}

public class Rectangle\_2 {

public static void main(String[] args) {

Rect r1 , r2;

r1 = new Rect();

r2 = new Rect();

r1.get\_length();

r1.get\_width();

r1.get\_area();

r1.get\_colour();

r2.get\_length();

r2.get\_width();

r2.get\_area();

r2.get\_colour();

if((r1.area == r2.area ) && r1.colour.equalsIgnoreCase(r2.colour) )

{//Mistake in this don't equate functions , equate only variables;

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

}

else {

System.***out***.println("Non matching");

}

}

}

8) Package:

package pkg1;

public class ClassA {

public int a = 1;

protected int b = 2;

int c = 3; // package-private, only accessible within pkg1

private int d = 4; // private, only accessible within ClassA

public void publicMethod() {

System.***out***.println("In public method");

}

protected void protectedMethod() { // Changed to protected for subclass access

System.***out***.println("In protected method");

}

void defaultMethod() { // package-private, accessible only within pkg1

System.***out***.println("In default method");

}

private void privateMethod() { // private, only accessible within ClassA

System.***out***.println("In private method");

}

public void show() { // Public method to display all variables and methods in ClassA

System.***out***.println("public var a: " + a);

System.***out***.println("protected var b: " + b);

System.***out***.println("Default var c: " + c);

System.***out***.println("private var d: " + d);

publicMethod();

protectedMethod();

defaultMethod();

privateMethod();

}

}

package pkg2;

import pkg1.ClassA;

public class ClassB extends ClassA {

public void display() {

// Accessing inherited public and protected members from ClassA

System.***out***.println("Public var a: " + a); // Accessible (public)

System.***out***.println("Protected var b: " + b); // Accessible (protected)

publicMethod(); // Accessible (public)

protectedMethod(); // Accessible (protected)

// Note: `c` (default) and `d` (private) are not accessible in ClassB

// `defaultMethod()` and `privateMethod()` are also not accessible

}

}

package pkg3;

import pkg1.ClassA;

import pkg2.ClassB;

public class ClassC {

public static void main(String[] args) {

// Create an instance of ClassA

ClassA temp1 = new ClassA();

System.out.println("public var a: " + temp1.a); // Access public variable 'a'

temp1.publicMethod(); // Call public method in ClassA

// temp1.show();

// Create an instance of ClassB

ClassB temp2 = new ClassB();

temp2.display(); // Call display method in ClassB

}

}