

Develop a java program that prints all real solutions to Quadratic equation $ax^2+bx+c=0$. Read in a, b, c and use the Quadratic formula .If the discriminant b^2-4ac is negative ,display a message stating that there are no real solutions.

```
import java.util.Scanner;
class First
{
    public static void main(String args[])
    {
        int a,b,c;
        double r1,r2,d;
        System.out.println("enter the coefficients of quadratic equation");
        Scanner s1= new Scanner(System.in);
        a=s1.nextInt();
        b=s1.nextInt();
        c=s1.nextInt();
        if(a==0)
        {
            System.out.println("coefficients are invalid");
        }
        else
        {
            d=b*b-(4*a*c);
            if(d>0)
            {
                System.out.println("it has real and distinct roots");
                r1=(-b+Math.sqrt(d))/(2*a);
                r2=(-b-Math.sqrt(d))/(2*a);
                System.out.println("the roots are "+r1+" and "+r2);
            }
            else if(d==0)
            {
                System.out.println("it has real and equal roots");
                r1=(-b)/(2*a);
                System.out.println("the roots are "+r1+" and "+r1);
            }
        }
    }
}
```

```

        }
        else
        {
            System.out.println("it has no real roots");
        }
    }
}

C:\Users\Naidu Renusree\Pictures\1BM22CS168>java First
enter the coefficients of quadratic equation
1
2
1
it has real and equal roots
the roots are -1.0 and -1.0

C:\Users\Naidu Renusree\Pictures\1BM22CS168>java First
enter the coefficients of quadratic equation
1
4
1
it has real and distinct roots
the roots are -0.2679491924311228 and -3.732050807568877

C:\Users\Naidu Renusree\Pictures\1BM22CS168>java First
enter the coefficients of quadratic equation
1
1
4
it has no real roots

C:\Users\Naidu Renusree\Pictures\1BM22CS168>

```

Develop a java program to create a class Student with members usn , name, an array credits and an array marks .Include methods to accept and display details and a method to calculate SGPA of a student.

```

import java.util.Scanner;

class Student
{
    int usn;
    String name;
    int credits[] = new int[5];
    float marks[] = new float[5];
    void studentdetails(int usn, String name, int credits[], float marks[])
}

```

```

{

    this.usn=usn;

    this.name=name;

    this.credits=credits;

    this.marks=marks;

}

void printdetails()

{

    System.out.println("the usn of student is"+usn);

    System.out.println("the name of student is"+name);

    System.out.println("the credits of subjects are");

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

    {

        Sustem.out.println(credits[i]);

    }

    System.out.println("the marks of the students are");

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

    {

        System.oout.println(marks[i]);

    }

}

}

Float studentsgpa()

{

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

    {

        If(marks[i]>=90)

        {

            sum=sum+(credits[i]*10);

        }

        else if(marks[i]>=80)

```

```

{
    sum=sum+(credits[i]*9);
}

else if(marks[i]>=70)

{
    sum=sum+(credits[i]*8);
}

else if(marks[i]>=60)

{
    sum=sum+(credits[i]*7);
}

else if(marks[i]>=50)

{
    sum=sum+(credits[i]*6);
}

else if(marks[i]>=35)

{
    sum=sum+(credits[i]*5);
}

else

{
    sum=sum+(credits[i]*0);
}

}

int creditssum=0;

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

{
    Creditssum=credits+credits[i];
}

Float sgpa=sum/(float)creditssum;

return sgpa;

```

```
}

class Student1

{

    public static void main(string xx[])

    {

        int usn;

        String name;

        int credits[] = new int[5];

        float marks[] = new float[5];

        float sgpa; Student s = new Student();

        Scanner s1 = new Scanner(System.in);

        System.out.println("enter the usn of student");

        usn = s1.nextInt();

        System.out.println("enter the name of student");

        name = s1.next();

        System.out.println("enter the credits of subject");

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

        {

            credits[i] = s1.nextInt();

        }

        System.out.println("enter the marks of student");

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

        {

            marks[i] = s1.nextFloat();

        }

        S.students(usn, name, credits, marks);

        S.printdetails();

        sgpa = S.studentsgpa();

        System.out.println("the sgpa of student is "+sgpa);

    }

}
```

```
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Naidu Renusree\Downloads\1BM22CS168>javac Student1.java

C:\Users\Naidu Renusree\Downloads\1BM22CS168>java Student1
enter the usn of student
1
enter the name of student
arun
enter the credits of subject
4
4
3
2
2
enter the marks of student
90
88
97
98
80
the usn of student is 1
the name of student is arun
the credits of subjects are
4
4
3
2
2
```

```
enter the marks of student
90
88
97
98
80
the usn of student is 1
the name of student is arun
the credits of subjects are
4
4
3
2
2
the marks of the student are
90.0
88.0
97.0
98.0
80.0
the sgpa of student is 9.6

C:\Users\Naidu Renusree\Downloads\1BM22CS168>
```

Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

```
import java.util.Scanner;
```

```
class Books {  
    String name, author;  
    int price, numPages;  
  
    Books(String name, String author, int price, int numPages) {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.numPages = numPages;  
    }  
  
    public String toString() {  
        String name, author, price, numPages;  
        name = "Book name s: " + this.name + "\n";  
        author = "Author name : " + this.author + "\n";  
        price = "Price : " + this.price + "\n";  
        numPages = "Number of pages : " + this.numPages + "\n";  
        return name + author + price + numPages;  
    }  
}  
  
class BookMain {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int n;  
        String name, author;  
        int price, numPages;  
        System.out.println("Enter the number of books:");  
        n = sc.nextInt();  
        Books b[] = new Books[n];
```

```

System.out.println("Enter Name,author,price and number of pages:");

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

    name = sc.next();

    author = sc.next();

    price = sc.nextInt();

    numPages = sc.nextInt();

    b[i] = new Books(name, author, price, numPages);

}

System.out.println("Book details:");

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

    System.out.println(b[i].toString());

}

}

```

```

C:\Users\Naidu Renusree\Pictures\1BM22CS168>javac BookMain.java

C:\Users\Naidu Renusree\Pictures\1BM22CS168>java BookMain
Enter the number of books:
3
Enter Name,author,price and number of pages:
thegirlwithnoname
jillgregory
385
785
bookname
rishi
485
500
nameofthebook
sushma
500
600
Book details:
Book name: thegirlwithnoname
Author name : jillgregory
Price : 385
Number of pages : 785

Book name: bookname
Author name : rishi
Price : 485
Number of pages : 500

Book name: nameofthebook
Author name : sushma
Price : 500
Number of pages : 600

```

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

```
abstract class shape{
protected int dimension1;
protected int dimension2;
public shape(int dimension1,int dimension2){
this.dimension1=dimension1;
this.dimension2=dimension2;
}
public abstract void printarea();
}

class rectangle extends shape{
public rectangle(int length,int width){
super(length,width);
}
public void printarea(){
double area=dimension1*dimension2;
System.out.println("area of rectangle:"+area);
}
}

class triangle extends shape{
public triangle(int base,int height){
super(base,height);
}
public void printarea(){
double area=0.5*dimension1*dimension2;
System.out.println("area of triangle:"+area);
}
}

class circle extends shape{
public circle(int radius){
super(radius,0);
}
}
```

```

public void printarea(){
    double area=Math.PI*dimension1*dimension1;
    System.out.println("area of circle:"+area);
}
}

public class rect{
    public static void main(String xx[]){
        rectangle r=new rectangle(2,10);
        triangle t=new triangle(3,7);
        circle c=new circle(5);
        r.printarea();
        t.printarea();
        c.printarea();
    }
}

```

```

C:\Users\EXAM\Desktop\1BM22CS168>javac rect.java

C:\Users\EXAM\Desktop\1BM22CS168>java rect
area of rectangle:20.0
area of triangle:10.5
area of circle:78.53981633974483

```

```
C:\Users\EXAM\Desktop\1BM22CS168>
```

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks: a) Accept deposit from customer and update the balance. b) Display the balance. c) Compute and deposit interest d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

```

import java.util.Scanner;
class account
{
    String name;

```

```
int accno;
String type;
double balance;
account(String name,int accno,String type,double balance)
{
    this.name=name;
    this.accno=accno;
    this.type=type;
    this.balance=balance;
}
void deposit(double amount)
{
    balance+=amount;
}
void withdraw(double amount)
{
    if((balance-amount)>=0)
    {
        balance-=amount;
    }
    else
    {
        System.out.println("insufficient balance,cant withdraw");
    }
}
void display()
{
    System.out.println("name:"+name+"accno:"+accno+"type:"+type+"balance:"+balance);
}
}
class savAcct extends account
{
private static double rate=5;
savAcct(String name,int accno,double balance)
{
    super(name,accno,"savings",balance);
}
void interest()
{
    balance+=balance*(rate)/100;
    System.out.println("balance:"+balance);
}
}
class curAcct extends account
{

private double minBal=500;
private double serviceCharges=50;
curAcct(String name,int accno,double balance)
{
    super(name,accno,"current",balance);
```

```

}

void checkmin()
{
if(balance<minBal)
{
System.out.println("balance is less than min balance,service charges imposed:"+serviceCharges);
balance-=serviceCharges;
System.out.println("balance is:"+balance);
}

}

class accountMain
{
public static void main(String a[])
{
Scanner s=new Scanner(System.in);
System.out.println("enter the name :");
String name=s.next();
System.out.println("enter the type(current/savings):");
String type=s.next();
System.out.println("enter the account number:");
int accno=s.nextInt();
System.out.println("enter the intial balance:");
double balance=s.nextDouble();
int ch;
double amount1,amount2;
account acc=new account(name,accno,type,balance);
savAcct sa=new savAcct(name,accno,balance);
curAcct ca=new curAcct(name,accno,balance);
while(true)
{
if(acc.type.equals("savings"))
{
System.out.println("\nMenu\n1.deposit 2.withdraw 3.compute interest 4.display");
System.out.println("enter the choice:");
ch=s.nextInt();
switch(ch)
{
case 1:System.out.println("enter the amount:");
amount1=s.nextInt();
sa.deposit(amount1);
break;
case 2:System.out.println("enter the amount:");
amount2=s.nextInt();
sa.withdraw(amount2);
break;
case 3:sa.interest();
break;
case 4:sa.display();
break;
}
}
}

```

```
case 5:System.exit(0);
default:System.out.println("invalid input");
break;
}
}
else
{
System.out.println("\nMenu\n1.deposit 2.withdraw 3.display");
System.out.println("enter the choice:");
ch=s.nextInt();
switch(ch)
{
case 1:System.out.println("enter the amount:");
amount1=s.nextInt();
ca.deposit(amount1);
break;
case 2:System.out.println("enter the amount:");
amount2=s.nextInt();
ca.withdraw(amount2);
ca.checkmin();
break;
case 3:ca.display();
break;
case 4:System.exit(0);
default:System.out.println("invalid input");
break;
}
}
}
}
```

```
C:\Users\EXAM\Desktop\1BM22CS168>javac accountMain.java

C:\Users\EXAM\Desktop\1BM22CS168>java accountMain
enter the name :
renusree
enter the type(current/savings):
savings
enter the account number:
168
enter the intial balance:
1000

Menu
1.deposit 2.withdraw 3.compute interest 4.display
enter the choice:
2
enter the amount:
200

Menu
1.deposit 2.withdraw 3.compute interest 4.display
enter the choice:
3
balance:840.0

Menu
1.deposit 2.withdraw 3.compute interest 4.display
enter the choice:
4
name:renusreeaccno:168type:savingsbalance:840.0

Menu
1.deposit 2.withdraw 3.compute interest 4.display
enter the choice:
5

C:\Users\EXAM\Desktop\1BM22CS168>
```

Create a package CIE which has two classes- Student and Internals. The class Student has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

package CIE;

```
public class Internals {
    private int[] internalMarks = new int[5];

    public Internals() {

    }
```

```
public void setInternalMarks(int[] internalMarks) {
    this.internalMarks = internalMarks;
}

public int[] getInternalMarks() {
    return internalMarks;
}
}

package CIE;

public class Student {
    public String usn;
    public String name;
    public int sem;

    public Student() {
        this("", "", 0);
    }

    public Student(String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }

    public void setUsn(String usn) {
        this.usn = usn;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setSem(int sem) {
        this.sem = sem;
    }

    public String getUsn() {
        return usn;
    }

    public String getName() {
        return name;
    }

    public int getSem() {
```

```

        return sem;
    }
}
package SEE;
import CIE.Student;

public class External extends Student {
    public int[] seeMarks = new int[5];

    // Default constructor
    public External() {
        this("", "", 0, new int[5]);
    }

    // Constructor with parameters
    public External(String usn, String name, int sem, int[] seeMarks) {
        super(usn, name, sem);
        this.seeMarks = seeMarks;
    }

    // Setter method for seeMarks
    public void setSeeMarks(int[] seeMarks) {
        this.seeMarks = seeMarks;
    }

    // Getter method for seeMarks
    public int[] getSeeMarks() {
        return seeMarks;
    }
}

import CIE.Student;
import CIE.Internals;
import SEE.External;
import java.util.Scanner;

public class FinalMarks {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Allow the user to enter the number of students
        System.out.print("Enter the number of students: ");
        int n = scanner.nextInt();

        Student[] students = new Student[n];
        Internals[] internals = new Internals[n];
        External[] externals = new External[n];

        // Initialize students, internals, and externals
        for (int i = 0; i < n; i++) {
            students[i] = new Student();
            System.out.print("Enter USN for student " + (i + 1) + ": ");
        }
    }
}

```

```

        students[i].setUsn(scanner.nextInt());

        System.out.print("Enter name for student " + (i + 1) + ": ");
        students[i].setName(scanner.next());

        System.out.print("Enter semester for student " + (i + 1) + ": ");
        students[i].setSem(scanner.nextInt());

        internals[i] = new Internals();
        // Assuming a simple method to input internal marks with validation
        internals[i].setInternalMarks(inputMarksWithValidation("internal", i, scanner, 0, 50));

        externals[i] = new External(students[i].getUsn(), students[i].getName(), students[i].getSem(),
new int[5]);
        // Assuming a simple method to input external marks with validation
        externals[i].setSeeMarks(inputMarksWithValidation("external", i, scanner, 0, 100));

        // Calculate final marks for the ith student and display
        int[] finalMarks = new int[5];
        for (int j = 0; j < 5; j++) {
            finalMarks[j] = internals[i].getInternalMarks()[j] + externals[i].getSeeMarks()[j] / 2;
        }

        System.out.println("Student " + (i + 1) + " Final Marks: " +
finalMarks[0] + ", " + finalMarks[1] + ", " + finalMarks[2] + ", " +
finalMarks[3] + ", " + finalMarks[4]);
    }

    scanner.close();
}

private static int[] inputMarksWithValidation(String type, int studentIndex, Scanner scanner, int min, int max) {
    int[] marks = new int[5];
    System.out.println("Enter " + type + " marks for student " + (studentIndex + 1) + ": ");
    for (int i = 0; i < 5; i++) {
        int mark;
        do {
            System.out.print("Subject " + (i + 1) + ": ");
            mark = scanner.nextInt();
            if (mark < 0 || mark > max) {
                System.out.println("Invalid input. " + type + " marks should be between 0 and " + max + ".
Please try again.");
            }
        } while (mark < 0 || mark > max);
        marks[i] = mark;
    }
    return marks;
}

```

```

}

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Naidu Renusree\Downloads\1BM22CS168\project>javac FinalMarks.java

C:\Users\Naidu Renusree\Downloads\1BM22CS168\project>java FinalMarks
Enter the number of students: 2
Enter USN for student 1: 100
Enter name for student 1: arun
Enter semester for student 1: 1
Enter internal marks for student 1:
Subject 1: 48
Subject 2: 50
Subject 3: 47
Subject 4: 46
Subject 5: 44
Enter external marks for student 1:
Subject 1: 98
Subject 2: 99
Subject 3: 96
Subject 4: 98
Subject 5: 99
Student 1 Final Marks: 97, 99, 95, 95, 93
Enter USN for student 2: 101
Enter name for student 2: arjun
Enter semester for student 2: 1
Enter internal marks for student 2:
Subject 1: 48
Subject 2: 46
Subject 3: 47
Subject 4: 48
Subject 5: 49
Enter external marks for student 2:
Subject 1: 98
Subject 2: 96
Subject 3: 97
Subject 4: 92
Subject 5: 99
Student 2 Final Marks: 97, 94, 95, 94, 98

```

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age=father’s age.

```

class wrongage extends Exception
{
    public wrongage(String Message)
    {
        super(Message);
    }
}
class father
{
    private int age;
    public father(int age) throws wrongage
    {
        if(age=fatherage)
        {
            throw new wrongage("son's age cannot be greater
than or equal to father's age");
        }
    }
}

```

```

        this.sonage=sonage;
    }
    public int getsonage()
    {
        return sonage;
    }
}
public class exceptioninheritance
{
    public static void main(String xx[])
    {
        try{ son son1=new son(40,20);
        System.out.println("father's age:"+son1.getage());
        System.out.println("son's age:"+son1.getsonage());
        son invalidson=new son(30,35);
        System.out.println("this line will not be reached");
    }
    catch(wrongage e)
    {
        System.out.println("exception:"+e.getMessage());
    }
}
}

```

```

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Naidu Renusree\Downloads\1BM22CS168>javac exceptioninheritance.java

C:\Users\Naidu Renusree\Downloads\1BM22CS168>java exceptioninheritance
father's age:40
son's age:20
exception:son's age cannot be greater than or equal to father's age

C:\Users\Naidu Renusree\Downloads\1BM22CS168>

```

Write a program that creates two threads 1)thread displays “BMS College of Engineering” Once every ten seconds and another displays “CSE” once every two seconds.

```

class DisplayMessageThread extends Thread {
    private final String message;
    private final long interval; // in milliseconds

    DisplayMessageThread(String message, long interval) {
        this.message = message;
        this.interval = interval;
    }

    public void run() {
        try {
            while (true) {
                System.out.println(message);
                Thread.sleep(interval);
            }
        }
    }
}

```

```
        }
    } catch (InterruptedException e) {
        System.out.println(Thread.currentThread().getName() + " interrupted.");
    }
}

public class TwoThreadDemo {
    public static void main(String[] args) {
        DisplayMessageThread thread1 = new DisplayMessageThread("BMS College of Engineering",
10000); // 10 seconds
        DisplayMessageThread thread2 = new DisplayMessageThread("CSE", 2000); // 2 seconds

        thread1.setName("Thread 1");
        thread2.setName("Thread 2");

        thread1.start();
        thread2.start();

        try {
            // Let the threads run for a while
            Thread.sleep(30000); // Let the program run for 30 seconds
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted.");
        }

        // Interrupt both threads to stop them
        thread1.interrupt();
        thread2.interrupt();

        System.out.println("Main thread exiting.");
    }
}
```

```
C:\Users\EXAM\Desktop\1BM22CS168>javac TwoThreadDemo.java

C:\Users\EXAM\Desktop\1BM22CS168>java TwoThreadDemo
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
Main thread exiting.
Thread 1 interrupted.
Thread 2 interrupted.

C:\Users\EXAM\Desktop\1BM22CS168>
```

Write a program that creates a user interface to perform division. The user enters two numbers in the text fields Num1 and Num2 where not an integer, the program would throw an arithmetic exception. Display the exception in a message dialog box.

```
import java.awt.*;
import java.awt.event.*;

public class DivisionMain extends Frame implements ActionListener
{
    TextField num1,num2;
    Button dResult;
    Label outResult;
    String out="";
    double resultNum;
    int flag=0;

    public DivisionMain()
    {
        setLayout(new FlowLayout());

        dResult = new Button("RESULT");
        Label number1 = new Label("Number 1:",Label.RIGHT);
        Label number2 = new Label("Number 2:",Label.RIGHT);
        num1=new TextField(5);
        num2=new TextField(5);
        outResult = new Label("Result:",Label.RIGHT);

        add(number1);
        add(num1);
        add(number2);
```

```

        add(num2);
        add(dResult);
        add(outResult);

        num1.addActionListener(this);
        num2.addActionListener(this);
        dResult.addActionListener(this);
        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent we)
            {
                System.exit(0);
            }
        });
    }

    public void actionPerformed(ActionEvent ae)
    {
        double n1,n2;
        try
        {
            if (ae.getSource() == dResult)
            {
                n1=Double.parseDouble(num1.getText());
                n2=Double.parseDouble(num2.getText());

                /*if(n2==0)
                    throw new ArithmeticException();*/
                out=n1+" "+n2;
                resultNum=n1/n2;
                out+=String.valueOf(resultNum);
                repaint();
            }
        }
        catch(ArithmetcException e2)
        {
            flag=1;
            out="Divide by 0 Exception! "+e2;
            repaint();
        }
        catch(NumberFormatException e1)
        {
            flag=1;
            out="Number Format Exception! "+e1;
            repaint();
        }
    }

    public void paint(Graphics g)
    {
        if(flag==0)

            g.drawString(out,outResult.getX()+outResult.getWidth(),outResult
.getY()+outResult.getHeight()-8);
        else
            g.drawString(out,100,200);
    }
}

```

```
    flag=0;  
}  
  
public static void main(String[] args)  
{  
    DivisionMain dm=new DivisionMain();  
    dm.setSize(new Dimension(800,400));  
    dm.setTitle("DivionOfIntegers");  
    dm.setVisible(true);  
}
```

Number 1: Number 2: **RESULT** Result: 4.0 2.02.0

Page No.	Teacher's Sign / Remarks
18/12/23	

Date _____
Page _____

1) Quadratic Equation

develop a java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a,b,c and use the quadratic formula. If the discriminant b^2-4ac is negative, display a message stating that there are no real solutions.

```
Import java.util.Scanner;
```

```
class Quadratic
```

```
{
```

```
    int a,b,c;
```

```
    double r1,r2,d;
```

```
    void getd()
```

```
{
```

```
    Scanner s = new Scanner(System.in);
```

```
    System.out.println("Enter the coefficients");
```

```
import java.util.Scanner;
```

```
class first
```

```
{
```

```
    public static void main (String xx[])
```

```
{
```

```
    int a,b,c,d;
```

```
    float r1,r2;
```

```
    System.out.println("Enter the coefficients of  
quadratic equation ");
```

```
Scanner s1 = new Scanner (System.in);
```

```
a=s1.nextInt();
```

```
b=s1.nextInt();
```

```
c=s1.nextInt();
```

```
if (a<=0)
```

```
{
```

```

System.out.println("coefficients are invalid");
else{
    d = b*b - 4*a*c;
    if (d > 0){
        System.out.println("it has real and distinct roots");
        r1 = (-b + Math.sqrt(d)) / (2*a);
        r2 = (-b - Math.sqrt(d)) / (2*a);
        System.out.println("the roots are " + r1 + " and " + r2);
    } else if (d == 0)
    {
        System.out.println("it has real and equal roots");
        r1 = (-b) / (2*a);
        System.out.println("the roots are " + r1 + " and " + r1);
    }
    else
    {
        System.out.println("it has no real roots");
    }
}

```

Output:
enter the coefficient of quadratic equation.
1
2
1

it has real and equal roots
the roots are -1 and -1
enter the coefficients of quadratic equation.
1
1
1
it has real and distinct roots.
the roots are -0.2579491924311228 and -3.73205030756
enter the coefficients of quadratic equations
1
1
1
it has no real roots

18/10/24
develop a java program to create a class student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```

import util.java.scanner
class student
{
    String usn;
    String name;
    int credits[];
    int marks[];
    void acceptDetails()
    {
        System.out.println("Enter USN: ");
        usn = scanner.nextLine();
        System.out.println("Enter Name: ");
        name = scanner.nextLine();
        System.out.println("Enter number of subjects: ");
        int n = scanner.nextInt();
        credits = new int[n];
        marks = new int[n];
        for (int i = 0; i < n; i++)
        {
            System.out.println("Enter credit for subject " + (i + 1));
            credits[i] = scanner.nextInt();
            System.out.println("Enter marks for subject " + (i + 1));
            marks[i] = scanner.nextInt();
        }
    }
    float calculateSGPA()
    {
        float sgpa = 0;
        for (int i = 0; i < credits.length; i++)
        {
            sgpa += marks[i] * credits[i];
        }
        sgpa /= credits.length;
        return sgpa;
    }
}

```

```

    void studentdetails(int usn, String name, int credits[], float marks[])
    {
        this.usn = usn;
        this.name = name;
        this.credits = credits;
        this.marks = marks;
    }

    void printdetails()
    {
        System.out.println("the usn of student is " + usn);
        System.out.println("the name of student is " + name);
        System.out.println("the credits of subjects are");
        for (int i = 0; i < 5; i++)
        {
            System.out.println(credits[i]);
        }
        System.out.println("the marks of the students are");
        for (int i = 0; i < 5; i++)
        {
            System.out.println(marks[i]);
        }
    }

    float average()
    {
        float sum = 0;
        for (int i = 0; i < 5; i++)
        {
            sum = sum + (credits[i] * marks[i]);
        }
        float avg = sum / 5;
        return avg;
    }
}

class student
{
    public static void main(String[] args)
    {
        int usn; String name; int credits[] = new int[5];
        float marks[] = new float[5];
        float avg;
        Scanner s = new Scanner(System.in);
        System.out.println("enter the usn of student");
        usn = s.nextInt();
        System.out.println("enter the name of student");
        name = s.nextLine();
        System.out.println("enter the credits of subjects");
        for (int i = 0; i < 5; i++)
    }
}

```

```
2
    credits[i] = s.nextInt();
}
System.out.print("Enter the marks of student");
for(int i=0; i<5; i++)
{
    marks[i] = s.nextInt();
}
s.println("Student ID, name, credits, marks");
s.printdetails();
sgpa = s.getsgpa();
System.out.println("The sgpa of student is " + sgpa);
}
3.
```

Output:

Enter the usn of student.

Enter the name of student.

Arun

Enter the credits of subjects

4.

5

3

Enter the marks of students

90

88

97

98

80

The usn of student is 1

Then name of student is Arun.

The credits of subjects are

4

3

2

The marks of the student are

90.0

88.0

97.0

98.0

80.0

The sgpa of student is 9.6

9.6

Q9
Create a class Book which contains four members: name, author, price, numPages. Include a constructor to set the values for these members. Include methods to set and get the details of the objects. Create a toString() method that can display the complete details of the book. Develop a Java program to create a book object.

```

import java.util.Scanner;
class Books
{
    String name;
    String author;
    int price, numPages;
    Books(String name, String author, int price, int numPages)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    public String toString()
    {
        String name, author, price, numPages;
        name = "Book name: " + this.name + "\n";
        author = "Author name: " + this.author + "\n";
        price = "Price: " + this.price + "\n";
        numPages = "Number of pages: " + this.numPages + "\n";
        return name + author + price + numPages;
    }
}
class BookMain
{
    public static void main(String[] args)
    {
    }
}

```

S.
Scanner sc = new Scanner(System.in);
int n;
String name, author;
int price, numPages;
System.out.println("Enter the number of books:");
n = sc.nextInt();
Books b[] = new Books[n];
System.out.println("Enter name, author, price and number of pages:");
for(int i=0; i<n; i++)
{
 name = sc.nextLine();
 author = sc.nextLine();
 price = sc.nextInt();
 numPages = sc.nextInt();
 b[i] = new Books(name, author, price, numPages);
}
System.out.println("Book details:");
for(int i=0; i<n; i++)
{
 System.out.println(b[i].toString());
}

Output:
Enter the number of books:
3

Enter Name, author, price and number of pages;

Book details:

Name

Author

The Girl with no name

Jill Gregory

385

785.

The boy with no name

Bookname

Rishi

685

500

Name of the book

Sushma

500

600.

Book details:

Book name : The girl with no name

Author name : Jill Gregory

price : 385

number of pages : 785

Book names : Book name

Author name : Rishi

price : 685

Name of pages : 500

Book names : Name of the Book

Author name : Sushma

price : 600

number of pages : 600

do?

Packages

Create a package ac which has two classes Student and Internals. The class student has members like usn, name and sem. The class Internals has an array that stores the Internals marks of all 5 courses of the current semester of the student. Create another package see which has the class External which is derived class of the student. This class has an array that stored the marks scored in the courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

import java.util.Scanner;

package ac;

public class Student

{

String name;

String usn;

int sem;

public Student (String name, String usn, int sem)

{

this.name = name;

this.usn = usn;

this.sem = sem;

}

class Internals extends Student {

int [] Internal Marks;

public Internals (String name, String usn, int sem,

int [] internalMarks)

{

```

    Super(name, usn, sem);
    <this>.InternalMarks = InternalMarks;
}

3
{
    package see;
    import java.util.*;
    class External extends Internal {
        {
            int[] SeeMarks;
            public External(String name, String usn, int sem, int[] SeeMarks) {
                Super(name, usn, sem);
                this.SeeMarks = SeeMarks;
            }
        }
        public class FinalMarks {
            public static void main(String[] args) {
                Scanner s1 = new Scanner(System.in);
                System.out.println("Enter the no. of students");
                int n = s1.nextInt();
                String[] names = new String[n];
                String[] usn = new String[n];
                int[] Sem = new int[n];
                int[][] InternalMarks = new int[n][5];
                int[][] SeeMarks = new int[n][5];
                for (int i = 0; i < n; i++) {
                    System.out.println("Enter details for student " + (i + 1));
                    s1.nextLine();
                    String str = s1.nextLine();
                    String[] strArr = str.split(" ");
                    names[i] = strArr[0];
                    usn[i] = strArr[1];
                    Sem[i] = Integer.parseInt(strArr[2]);
                    InternalMarks[i] = new int[5];
                    SeeMarks[i] = new int[5];
                    for (int j = 0; j < 5; j++) {
                        InternalMarks[i][j] = Integer.parseInt(strArr[j + 3]);
                    }
                }
            }
        }
    }
}

```

Date	
Page	

```

    System.out.println("Name");
    marks[i] = s1.nextInt();
    System.out.println("USN");
    usn[i] = s1.nextLine();
    System.out.println("Semester");
    sem[i] = s1.nextInt();
    System.out.println("Internal Marks for 5 courses");
    for (int j = 0; j < 5; j++) {
        System.out.println("Course " + (j + 1));
        SeeMarks[i][j] = s1.nextInt();
    }
    int[] finalMarks = new int[5];
    for (int i = 0; i < 5; i++) {
        InternalMarks[i] = new InternalMarks(names[i], usn[i],
                                              Sem[i], InternalMarks[i]);
        External[i] = new External(names[i], usn[i],
                                   Sem[i], SeeMarks[i]);
        for (int j = 0; j < 5; j++) {
            finalMarks[i][j] = InternalMarks[i][j] + External[i].SeeMarks[i][j];
        }
    }
    System.out.println("Final Marks for " + n + " students in 5 courses");
}

```

<pre> for (int i = 0; i < n; i++) { System.out.println("Names[" + i + "]"); for (int j = 0; j < 5; j++) { System.out.println("Marks[" + i + "][" + j + "]"); } } System.out.println("Total marks for " + name); System.out.println("Total marks for " + name); </pre> <p>Output</p> <pre> Enter USN: 143 Enter name: Penu Enter Semester: 3 Enter CIE marks Enter Internal marks for Penu Subject 1 marks: 32 Subject 2 marks: 35 Subject 3 marks: 33 Subject 4 marks: 40 Enter SEE marks for Penu Subject 1 marks: 80 Subject 2 marks: 88 Subject 3 marks: 43 Subject 4 marks: 89 Subject 5 marks: 90 Enter USN: 144 </pre>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; border-right: 1px solid black; padding: 2px;">Date _____</td> <td style="width: 50%; padding: 2px;">Page _____</td> </tr> </table> <p>Enter name: Naidu Enter Sem: 3 Enter CIE marks: Enter Internal marks for Naidu Subject 1 marks: 93 Subject 2 marks: 82 Subject 3 marks: 87 Subject 4 marks: 40 Subject 5 mark: 31 Enter SEE marks for Naidu Subject 1 marks: 90 Subject 2 marks: 88 Subject 3 mark: 90 Subject 4 marks: 82 Subject 5 marks: 98</p> <p>Displaying data!</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; border-right: 1px solid black; padding: 2px; vertical-align: top;"> USN: 143 name: Penu semester: 3 subject 1: 81 Subject 2: 81 Subject 3: 73 Subject 4: 81 Subject 5: 80 </td> <td style="width: 50%; padding: 2px; vertical-align: top;"> OSN: 144 Name: Naidu semester: 3 Subject 1: 90 Subject 2: 95 Subject 3: 81 Subject 4: 45 Subject 5: 90 </td> </tr> </table>	Date _____	Page _____	USN: 143 name: Penu semester: 3 subject 1: 81 Subject 2: 81 Subject 3: 73 Subject 4: 81 Subject 5: 80	OSN: 144 Name: Naidu semester: 3 Subject 1: 90 Subject 2: 95 Subject 3: 81 Subject 4: 45 Subject 5: 90
Date _____	Page _____				
USN: 143 name: Penu semester: 3 subject 1: 81 Subject 2: 81 Subject 3: 73 Subject 4: 81 Subject 5: 80	OSN: 144 Name: Naidu semester: 3 Subject 1: 90 Subject 2: 95 Subject 3: 81 Subject 4: 45 Subject 5: 90				

Date _____
Page _____

Exception Handling

Write a program that demonstrates:

- i) Create a user defined named Reorder. Create a class called Item with members - Id, name, price, quantity, reorder-level. Include methods - i) purchase which adds item s to the existing quantity and ii) sell which reduces the same. Create two objects of item and set the values given by the user. Raise the exception reorder when the quantity of any item goes below reorder-level (int value level below which ordering is to be initiated).

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called Father and derived class Son which extends the base class. In Father class implement a method which takes the age and throws the exception Wrongage() when the input age >= 10. In Son class implement a constructor that checks birth father and son's age and throws an exception if son's age is >= father's age.

Exception

```

class Wrongage extends Exception {
    void wrongage (String message) {
        super(message);
    }
}

```

Class Father

```

class Father {
    private int age;
    public Father (int age) throws Wrongage {
        public int getAge () {
            return age;
        }
    }
}

```

Class Son extends Father

```

class Son extends Father {
    public Son (int fatherage, int sonage) throws Wrongage {
        super(fatherage);
        if (sonage >= fatherage) {
            throw new Wrongage ("Son's age cannot be greater than or equal to father's age");
        }
        this.sonage = sonage;
    }
    public int getSonage () {
        return sonage;
    }
}

```

```

public class exception inheritance demo
public static void main (String xx[])
{
    try {
        Son son = new Son (20, 20);
        System.out.println ("Father's age :" + son.getAge());
        System.out.println ("Son's age :" + son.getSonAge());
        Son invalidSon = new Son (20, 35);
        System.out.println ("This line will not be reached.");
    } catch (Exception e) {
        System.out.println ("exception: " + e.getMessage());
    }
}

Output:
Father's age: 20
Son's age : 20
exception: Son's age cannot be greater than or
equal to Father's age.
if (.getMessage() is not used:
exception: wrong: Son's age cannot be greater than
or equal to Father's age.

```

29/11/17

Date	Page
------	------

```

Generics:
class Generics
{
    Generics (G1, G2)
    {
        ob = 0;
    }
    void showType()
    {
        System.out.println ("Type of ob: " + ob.getClass().getName());
    }
    G1 retObj()
    {
        return ob;
    }
}

public class GenMain
{
    public static void main (String [] args)
    {
        Generics < Integer > g1 = new Generics < Integer > (200);
        g1.showType();
        int x = g1.retObj();
        System.out.println ("Value: " + x);
        Generics < Double > g2 = new Generics < Double > (34.5);
        g2.showType();
        double d = g2.retObj();
        System.out.println ("Value: " + d);
        Generics < Character > g3 = new Generics < Character > ('A');
        g3.showType();
        char c = g3.retObj();
    }
}

```

```

s.op ("value "+1);
Gener<String> g1=new Gener<String>("Hii");
g1.showType();
String s1=g1.value();
System.out.println(s1);
Gener<boolean> g2=new Gener<boolean>(true);
g2.showType();
boolean b2=g2.value();
System.out.println(b2);
}
}


```

Output

```

Type of ob: java.lang.Integer value: 200.
Type of ob: java.lang.Double value: 34.8
Type of ob: java.lang.Character value: 71
Type of ob: java.lang.String value: Hii
Type of ob: java.lang.Boolean value: true.


```

Theano
class Thread1
{
public static void main (String ss []) {
Thread t = Thread.currentThread();
System.out.println ("CT: " + t);
t.setName ("current main Thread");
System.out.println ("CT: " + t);
try {
for (int n = s; n > 0; n--)
s =

```
System.out.println("n");
Thread.sleep(500);
}
}
Catch (InterruptedException ie)
{
    System.out.println("The sleeping thread is
woken up.");
}
```

output:
3
CT : Thread [main, 5, main]
C : Thread [current main Thread] is main

MultiThread

Class NewThread implements Runnable
f

String name;
Thread t;

New Thread (String threadName);
3

name = threadName;
t = new Thread (this, name);
System.out.println ("New Thread : ");
t.start();
}

<pre> public void run() { try { for (int i = 5; i > 0; i--) { System.out.println(name + ":" + i); Thread.sleep(5000); } } catch (InterruptedException e) { System.out.println(name + " interrupted"); } } class MultiThreadDemo { public static void main(String args[]) { new NewThread("One"); new NewThread("Two"); new NewThread("Three"); try { Thread.sleep(10000); System.out.println("Main thread is awake"); } catch (InterruptedException e) { } } } </pre>	<p>Date _____ Page _____</p> <p> System.out.println("Main thread interrupted"); y System.out.println("Main thread exiting"); y output: New thread: Thread [One,5,main] New thread: Thread [Two,5,main] one:5 New thread: Thread [Three,5,main] Two: 5 Three: 5 Two: 4 One: 4 Three: 3 One: 3 Two: 3 Two: 2 One: 2 Three: 2 One: 1 Two: 1 Three: 1 two exiting. one exiting. three exiting. Main Thread is awake Main thread exiting. </p>
---	---

Multi Threading

- write a program that creates two threads, one thread displaying "BMS college of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

class DisplayMessageThread extends Thread

{

 private final String message;

 private long interval;

 DisplayMessageThread(string message, long interval)

{

 this.message = message;

 this.interval = interval;

}

 public void run()

 {try

 while(true)

 {

 System.out.println(message);

 Thread.sleep(interval);

 }

 }catch(InterruptedException e)

 {

 System.out.println(Thread.currentThread().getname() + "Interrupted.");

 }

public class TwoThreadDemo

{

 public static void main(string[] args)

{

 DisplayMessageThread thread1 = new DisplayMessageThread

 ("BMS college of Engg.inching",10000);

 DisplayMessageThread thread2 = new DisplayMessageThread

 ("CSE",2000);

 thread1.setName("Thread1");

 thread2.setName("Thread2");

 thread1.start();

 Thread2.start();

 }try

 Thread.sleep(3000);

 }catch(InterruptedException)

 {

 System.out.println("Main thread interrupted");

 }

 thread1.interrupt();

 thread2.interrupt();

 System.out.println("Main thread exiting.");

 }

Output:

BMS College of Engineering

CSE

CSE

CSE

CSE

<p>BMS college of Engineering. CSE CSE CSE CSE CSE BMS college of Engineering. CSE CSE CSE CSC CSE</p> <p>Main thread exiting Thread 1 interrupted. Thread 2 interrupted.</p> <p>Shape:</p> <pre> abstract class Shape{ protected int dimension1; protected int dimension2; public Shape (dimension1, dimension2) { this.dimension1 = dimension1; this.dimension2 = dimension2; } public abstract void printArea(); public void printArea() { int area = dimension1 * dimension2; System.out.println("Area of rectangle: " + area); } </pre>	<p>Date _____ Page _____</p> <p>class Triangle extends Shape public Triangle (int base, int height) { super(base, height); public void printArea() { double area = 0.5 * dimension1 * dimension2; System.out.println("Area of Triangle: " + area); } }</p> <p>class Circle extends Shape public Circle (int radius) { super(radius, 0); public void printArea() { double area = Math.PI * dimension1 * dimension2; System.out.println("Area of circle: " + area); } }</p> <p>public class Rectangle public void main (String [] args) { Rectangle r = new Rectangle (2, 10); Triangle t = new Triangle (5, 7); Circle c = new Circle (5); r.printArea(); t.printArea(); c.printArea(); }</p>
---	--

Output:
Area of Rectangle: 20
Area of Triangle: 10.5
Area of Circle: 78.53918

Develop a Java program to create a class Bank that maintains two kinds of accounts for HS customers, one called Saving account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this, derive the classes Current and Savings to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

- a) Accept deposit from customer and update the balance.
- b) Display the balance.
- c) Compute and deposit interest.
- d) Permit withdrawal and update the balance.

```
import java.util.Scanner;  
class Account {  
    String name;  
    int accno;
```

String type;
double balance;
Account (String name, int accno, String type, double balance)
{
 this.name = name;
 this.accno = accno;
 this.type = type;
 this.balance = balance;
}
void deposit (double amount)
{
 balance += amount;
}
void withdraw (double amount)
{
 if ((balance - amount) >= 0)
 {
 balance -= amount;
 }
 else
 {
 System.out.println ("insufficient balance, can't withdraw");
 }
}
System.out.println ("name: " + name + " accno: " +
 accno + " type: " + type + " balance: " + balance);
}

class savings extends account

```
private static double rate=5;  
savAcct (String name, int accno, double balance,  
        Super(name, accno, "Savings", balance),
```

void interest()

```
balance+=balance*(rate)/100;  
System.out.println("balance: "+balance);
```

class current extends account

```
private double minBal = 500;  
private double serviceCharges = 50;  
currAcct (String name, int accno, double balance,  
          Super(name, accno, "Current", balance),
```

super();

void checkMin()

```
i){balance < minBal)  
System.out.println("balance is less than min balance,  
service charges imposed: "+serviceCharges);  
balance-=serviceCharges;
```

System.out.println("balance is: "+balance);

}

class account Main

```
public static void main(String args)
```

```
{Scanner s = new Scanner (System.in);  
System.out.println("Enter the name:");  
String name = s.nextLine();  
System.out.println("Enter the type (current/savings):");  
String type = s.nextLine();  
System.out.println("Enter the account number:");  
int accno = s.nextInt();  
System.out.println("Enter the initial balance:");  
double balance = s.nextDouble();  
int ch;
```

double amount1, amount2;

account acc = new account (name, accno, type,
balance);

sav Acct sa = new SavAcct (name, accno, balance);
currAcct ca = new currAcct (name, accno, balance,
balance);

```
{if (acc.type.equals("savings"))
```

```
System.out.println("deposit 2.withdraw");
```

3.computer interest 4.display";

System.out.println("Enter the choice");

ch = s.nextInt();

switch(ch)

```
{case 1: System.out.println("Enter the amount:");
```



```

import java.awt.*;
import java.awt.event.*;
public class divisionMain extends Frame implements
ActionListener
{
    JTextField num1, num2;
    JButton dresult;
    Label outResult;
    String out = "";
    double resultNum;
    int flag;
    public divisionMain()
    {
        setLayout(new FlowLayout());
        dresult = new JButton("result");
        label number1 = new label("Number 1:", label.RIGHT);
        label number2 = new label("Number 2:", label.RIGHT);
        num1 = new TextField("5");
        num2 = new TextField("5");
        outResult = new label("result:", label.RIGHT);
        add(number1);
        add(num1);
        add(number2);
        add(num2);
        add(dresult);
        add(outResult);
        num1.addActionListener(this);
        num2.addActionListener(this);
        dresult.addActionListener(this);
        addWindowListener(new WindowAdapter())
    }
}

```

Date	
Page	

```

public void windowClosing(WindowEvent e)
{
    System.exit(0);
}

public void actionPerformed(ActionEvent ae)
{
    double n1, n2;
    try
    {
        if(ae.getSource() == dresult)
        {
            n1 = Double.parseDouble(num1.getText());
            n2 = Double.parseDouble(num2.getText());
            throw new ArithmeticException();
        }
        else
        {
            out = n1 + " + " + n2;
            resultNum = n1 / n2;
            out += String.valueOf(resultNum);
            repaint();
        }
    }
    catch(ArithmeticException e1)
    {
        flag = 1;
        out = "divide by 0 exception!" + e1;
    }
    catch(NumberFormatException e2)
    {
        flag = 1;
        out = "Number format exception! " + e2;
        repaint();
    }
}

```

```

public void paint(Graphics g)
{
    if(flag >= 0)
        g.drawString("out", result.getX() + 10, result.get
        get width() + result.getY() - result.getheight()
        else
            g.drawString("out", 100, 200);
        flag = 0;
}
public static void main(String[] args)
{
    DivisionMain dm = new DivisionMain();
    dm.setSize(new Dimension(600, 400));
    dm.setTitle("Division of Integers");
    dm.setVisible(true);
}

```

Output

number 1: 4 number 2: 2 [result] result: 4 2 2.0
 number 1: 4 number 2: 0 [result] result:
 divide by 0 Exception! Java.lang.ArithemticException: / by zero
 number 1: 4 number 2: a [result] result:
 number format exception! Java.lang.NumberFormatException: for
 input string: "a"

Report
} out[
AWT "Abstract Window Toolkit"

The AWT is a package in Java which provides classes to create and manage graphical user interfaces (GUIs).

AWT includes various components, event handling, layout managers and other utilities.

Components:-

① Frame:-

A frame is a window with title & border, used as a main container in which other components are added.

Syntax: class IntegerDivision extends Frame.

Dialog:

It takes some form of input from the user used to display modal dialog to interact with user used to perform action, prompt the user, display messages.

Text Field:

It is a single line text displayed used for guiding the user to accept input.

Label:

display area for short text.

display static text.

Prompts the user to enter text input.

e.g. Num 1:

Button:

It triggers the actions when clicked.

Used for performing specific actions.

Used for submitting forms, confirming, etc.

Check Box:-

Component which can be checked or unchecked.

Used to enable or disable.

Used in the form with multiple choices.

Event Handling.

Event:

Events are generated by user actions.

Used for handling user input, such as mouse,

clicks, keypress or window events.

Events are processed by event listeners attached to the relevant components.

Event Listener:
Object that receives and handles events.
This is for the implementation of event driven behaviour in GUI's.
Event listener are registered with specific components to listen for particular types of events.

Action Event

Event that indicates that a component defined action occurred.
Used for handling user actions.
Action generated when user clicks on component like button, checkbox, etc.

Action Listener:

It receives action events.
It responds to the user action.

Window Event:

Event indicating change to the state of a window.
Handler window related events such as window opening, closing, resizing etc.

Window Listener:

It receives window events.
Used to implement methods to handle window related events.

Layout Management.

Flow Layout:

Arranges components in a vertical flow one after another.

Used for arranging rows or column with equal spacing between them.

Used for creating forms etc.

Border Layout:

Layout manager that divides the container into 5 regions: north, south, east, west and center.

Used for arranging components at the edges, center of the container.

Layout Manager:

Responsible for arranging components in container, central size & position of components.

8/21/2