



B.M.S COLLEGE OF ENGINEERING

Basavanagudi-560019

Subject:-

Object Oriented Java

LAB-Observation

Submitted by:

**Shree Sudhanva K
3E -Batch 3
1BM22CS262**

observation Lab

Develop a Java program that prints all the solution

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

```
class Quadratic
```

```
{
```

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

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

```
    void getd()
```

```
{
```

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

```
        System.out.print("Enter coefficient");
```

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

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

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

```
,
```

```
    void compute()
```

```
{
```

```
    while(a == 0)
```

```
{
```

```
        System.out.println("Not a Quadratc");
```

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

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

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

```
,
```

```
        d = b * b - 4 * a * c;
```

```
        if(d == 0)
```

```
{
```

```
            r1 = (-b / 2 * a);
```

```
            System.out.println("Roots are real & distinct");
```

```
        } else
```

else if ($d > 0$)

{

$$r_1 = \frac{(-b) + (\text{math.sqrt}(d))}{\text{double}(2^{\circ}a)}$$

$$r_2 = \frac{(-b) - (\text{math.sqrt}(d))}{\text{double}(2^{\circ}a)}$$

System.out.println("Roots are real and
distinct");

System.out.println("Root1 = " + r₁ + " Root2 = " + r₂)

{

else if ($d < 0$)

{

System.out.println("Roots are imaginary");

$$r_1 = (-b) / 2^{\circ}a;$$

$$r_2 = \text{Math.sqrt}(-d / 2^{\circ}a);$$

$$r_2 = \text{Math.sqrt}(-d)$$

System.out.println("Root1 = " + r₁ + " + " + r₂ + "i");

System.out.println("Root2 = " + r₂ + " - " + r₁ + "i");

{

{

class quadratic Main

{

public static void main (String args[])

Quadratic q = new Quadratic();

q.getd();

q.compute();

System.out.println("Shree Sudhanva K")

IBM Z22CS26Z")

{

output

Enter the coefficients

1

3

2

Roots are real and distinct

root 1 = -1.0 Root 2 = -2.0

Shree Sudhanva LC 1BM22CS262

Enter the coefficients

1

2

1

Roots are real and equal

Root 1 = Root 2 = -10

Shree Sudhanva LC 1BM22CS262

Enter the coefficients

2

1

3

Roots are Imaginary

Root 1 = $0.0 + i 1.1989578805$

Root 2 = $0.0 + i 1.1989578805$

Shree Sudhanva LC 1BM22CS262

80
1212)23

19/12/23

Lab 2

SURYA Gold

Date

Page

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

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

```
class Subject {
```

```
    int subjectMarks;
```

```
    int credits;
```

```
    int grade;
```

}

```
class Student
```

}

```
Subject subject[3];
```

```
String name;
```

```
String USN;
```

```
double SGPA = 0;
```

```
Scanner s;
```

```
Student()
```

}

```
int i;
```

```
Subject t = new Subject(8);
```

```
for (i=0; i<8; i++)
```

```
    subject[i] = new Subject();
```

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

}

```
void getStudentDetails()
```

}

```
System.out.println("Enter your name");
```

```
name = s.nextLine();
```

```
System.out.println("Enter your USN");
```

```
USN = s.nextLine();
```

void

{

3

void getMarks()

{

for (int i = 0; i < 8; i++)

{

System.out.print("Enter marks of subject " + i + ": ");

System.out.

subject[i] = subjectMarks = s.nextInt();

System.out.print("Enter credits for

subject " + i + " : ");

subject[i].credits = s.nextInt();

subject[i].grade = (subject[i].

subjectMarks / 10) + 1;

if (subject[i].grade > 11)

subject[i].grade = 10;

if (subject[i].grade < 1)

subject[i].grade = 0;

}

}

void computeSGPA()

{

int totalCredits = 0

for (int i = 0; i < 8; i++)

{

sgpa += (subject[i].credits * subject[i].grade)

totalCredits += subject[i].credits;

}

sgpa = sgpa / totalCredits;

}

}

class main

5

public static void main (String args[])

Student s1 = new Student();

s1.getStudentDetails();

s1.getMarks();

s1.computeSGPA();

System.out.println ("Name:" + s1.name);

System.out.println ("USN:" + s1.USN);

System.out.println ("SGPA:" + s1.SGPA);

3

Output

Enter your Name

Shreya Sudhanya

Enter your USN

13BM22CS262

Enter marks of Subject 1

89

Enter marks of Subject 2 83

Enter marks of Subject 3

Enter credits of Subject 1

4

Enter marks of subject 2

69

Enter credits of subject 2

4

Enter marks of subject 3

80

Enter credits of subject 3

3

Enter marks of subject 4

90

Enter credits of subject 4

3

Enter marks of subject 5

90

Enter credits of subject 5

3

Enter marks of subject 6

95

Enter credits of subject 6

1

Enter marks of subject 7

91

Enter credits of subject 7

1

Enter marks of subject 8

98

Enter credits of subject 8

1

Name : Shree Sudhanva

UAN : 1BM22CS262

SGPA : 9.05

26/12/23

Lab 3

create Java program to develop book object

26/11/23

SURYA Gold
Date _____
Page _____

import java.util.Scanner;

class Book {

 String name;
 String author;
 float Price;
 int numPage;

 public Book {

 this.name = name;
 this.author = author;
 this.Price = Price;
 this.numPage = numPage;

}

 public String toString () {

 String bookDetails = "Book name: " + this.name +
 "\n" + "Author name: "
 + this.author + "\n" + "Price: "
 + this.Price + "\n" + "Number of"
 "Pages: " + this.numPage +
 "\n";

 return bookDetails;

}

 public class XYZ {

 public static void main (String [] args) {

 Scanner s = new Scanner (System.in);

 System.out.println ("Enter number of books");

 int n = s.nextInt ();

 Book [] books = new Book [n];

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

 System.out.println ("Enter details of book");

 System.out.print ("Name: ");

 String name = System.in.readLine();

 System.out.print ("Author: ");

 String author = System.in.readLine();

 System.out.print ("Price: ");

 float price = System.in.readFloat();

 System.out.print ("No. of pages: ");

 int numPage = System.in.readInt();

 books[i] = new Book (name, author, price, numPage);

 }

 }

 }

}

string

System

String

System

Print

System

Print

bo

3

Super

for

3

super

for

3

3

3

3

outpu

Enter

Enter

Name

Author

Price

No.

Enter

name

Author

Price

Page

Book

```

String name = s.next();
System.out.print("Author: ");
String author = s.nextLine();
System.out.print("Price: ");
float price = s.nextDouble();
System.out.print("No of Pages: ");
int numPages = s.nextInt();
book[0] = new Book(name, author, price, numPages);
    
```

3

```

System.out.println("In Book Details:");
for (i = 0; i < n; i++) {
    System.out.println("Book " + (i + 1) + ": " + book[i]);
}
    
```

3 System.out.println("Shree Sandhwan") - 3

Output

Enter number of Books 2
 Enter details for Book 1:
 Name: JavaProgram
 Author: Shreyas
 Price: 200
 No of Pages: 100

Enter details for Book 2:

Name: abcd
 Author: xyz
 Price: 200
 Page: 50

Book details

BOOK 1 : java program

Book name : Java program

Author name : Shreyas

Price : 200

No of pages : 100

Book 2

Book name : ABCD

Author name : XYZ

Price : 200

No of pages : 50

Shree Sudhanwa

16/12/23
26/12/23

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named `Display()`. Provide 3 classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the class contain only the method `Display()` that prints the area of the given shape.

class InputScanner {

Scanner sc;

InputScanner() {

sc = new Scanner(System.in);

}

Abstract class Shape Extends InputScanner {

double a;

double b;

~~abstract void getInput();~~

~~abstract void display();~~

3

Class rectangle extends Shape {

rectangle (int a, int b) {

shape (double x)

?

a = x;

3

shape (double x, double y)

? a = x; b = y; ?

~~class Rec extends shape {~~

~~void getInout() { System.out.println("Enter sides"); }~~

~~a = sc.nextInt(); nextDouble();~~

~~b = sc.nextDouble();~~

~~}~~

~~void display() {~~

~~System.out.print("Area of rectangle is ");~~

~~a * b);~~

~~}~~

~~class Triangle extends shape {~~

~~void getInout() {~~

~~System.out.println("Enter sides");~~

~~a = sc.nextInt(); nextDouble();~~

~~b = sc.nextDouble();~~

~~}~~

~~void display() {~~

~~System.out.println("Area of triangle is " + (a * b) / 2);~~

~~}~~

~~class Circle extends shape {~~

~~void getInout() {~~

~~System.out.println("Enter radius");~~

~~a = sc.nextInt(); nextDouble();~~

~~void display() {~~

~~System.out.print("Area of circle is " + (Math.PI * a * a));~~

~~* set side * side * a * a)~~

~~3~~

3rd class Chase Math 2

rectangle shape void main(string args[]){}

Rec p = new Rec();

Triangle t = new Triangle();

Circle c = new Circle();

rectangle

R. get input();

R. display();

t. get input();

~~t. display();~~

C. get input();

C. display();

3

output

Enter sides

10

5

Area of rectangle is 50

Enter sides

10

5

Area of triangle is 25

Enter radius

5

~~area of circle = 78.53981635~~

Sudhanna

IBN22CSZ62

Develop Java Program to develop a class bank
with current account & savings account

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

Class Input {

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

}

Class account extends Input {

```
String name; int accNo; double balance;
```

Public void getDetails()

{

```
System.out.print("Enter the name: ");
```

```
name = sc.nextLine();
```

```
System.out.print("Enter the accNo: ");
```

```
accNo = sc.nextInt();
```

}

void deposit()

```
System.out.print("Enter the amount to deposit");
```

```
double amt = sc.nextDouble();
```

```
balance += amt;
```

```
System.out.println("Amount deposited");
```

}

Public void withdraw()

```
System.out.print("Enter amount to withdraw");
```

```
double amt = sc.nextDouble();
```

```
if (balance >= amt) {
```

```
balance -= amt;
```

```
System.out.println("withdraw successful");
```

}

```
else { System.out.println("Insufficient balance"); }
```

}

void display() {

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

System.out.println("Account No: " + accNo);

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

3

class Savings extends Account {

final double interestRate = 0.04; //

void computeInterest() {

double interest = balance * interestRate;

balance += interest;

System.out.println("Interest credited: " + interest);

3

~~class Current extends Account {~~

~~final double minBalance = 500;~~

~~final double penalty = .00;~~

@ override

public void withdraw() {

super.withdraw();

checkMinBalance();

3

~~private void checkMinBalance() {~~

~~if (balance < minBalance) {~~

~~balance -= penalty;~~

~~System.out.println("Penalty applied");~~

3

5

3

class bank extends Account

9

public static void main(String[] args)

```
Scanner sc = new Scanner (System. in);
Saving ob1 = new Saving();
Current ob2 = new Current();
ob1. get Details();
ob2. get details();
```

int choice;

String ac;

~~System.out.println("1. Deposit")
2. withdraw
3. display
4. compute Interest
5. exit")~~

do {

System.out.print("Enter choice: ")

choice = sc.nextInt();

System.out.print("Enter acc type: ")

ac = sc.next();

switch(choice){

case 1:

```
if(ac.equals("saving")) ob1.deposit();
else ob2.deposit(); break;
```

case 2:

```
if(ac.equals("saving")) ob1.withdraw();
else ob2.withdraw(); break;
```

case 3:

```
if(ac.equals("saving")) ob1.display();
else ob2.display(); break;
```

case 4:

```
ob1. computeInterest();
break;
```

Case 5

System.out.println("caught");
break;

default

System.out.println("invalid choice");

g

switch(choice) {

3

3

3

~~output~~

Enter name : someone

Enter age no: 123

Menu:

1. Deposit

2. withdraw

3. Display Balance

4. Compute Interest

5. exit

Enter your choice : 1

Enter the acc type : savings.

Enter amount to be deposited: 1000

Amount deposited successfully.

Enter your choice : 1

Enter account type : current

Enter the amount to deposit : 5000

Amount deposited successfully.

Enter choice : 2

Enter the acc type savings

Enter amount to withdraw : 800

Sudhansu

Amad withdrawal successfully : (BNR)CS6 >

Enter choice : 5

16/01/22

SURYA Gold

Date _____

Page _____

import java.util.Scanner
public class GenericStack<T> {
 Object[] stackArray;

- 1) write a Java program to create a generic class stack which holds 5 integers and 5 double value

import java.util.Scanner;
import java.util.Arrays;

class Stack<T> {
 static final int N = 5;
 private int top = -1;
 private Object[] stackArray;

public GenericStack()

 stackArray = new Object[N];

 public void push(T value)

 if (top < N - 1)

 stackArray[++top] = value;

 else

 System.out.println("Stack overflow");

 public T pop()

 if (top > -1)

 return (T) stackArray[--top];

 else

 System.out.println("Stack empty");

 return null;

multiple boolean Precondition
2 return true if all 16

multiple boolean Postcondition
2 return true if all 16

class Name

3 public static void main (String args[])

Convert single Char to Integer

= new ConvertChar();

Convert Char (double) doubleVal

= new ConvertChar();

for (int i = 0; i < 6; i++)

 integerVal = arr[i];

for (double d = 0; d < 6; d++)

 doubleVal = arr[d];

System.out.println("Inputed integer fields ");

while (integerVal != 0 && integerVal <= 16)

System.out.print("Inputed Char fields ");

3

System.out.println("Inputed double fields ");

while (doubleVal != 0 && doubleVal <= 16)

System.out.println("Inputed double fields ");

3

3

3

Output

Popped integers from the stack

5

4

3

2

1

Popped double from the stack:

5.0

4.0

3.0

2.0

1.0

String

Public class string1 {

 Public static void main (String Arg[]) {

Output:

demosticate string length:

5

str concat:

He is 9 years old

9/1/24

white after a program an abstract class
shape with abstract methods to calculate area

import java.lang.Math;

abstract class shape {

 double a;

 double b;

 double c;

 abstract void calculateArea();

 abstract void calculatePerimeter();

2

class Triangle extends shape

 Triangle(double x, double y, double z)

 a = x

 b = y

 c = z

3

 void calculateArea()

2

 double s = (a+b+c)/2;

 System.out.println("Area = " + Math.sqrt(s*(s-a)) * (s-b) * (s-c));

3

 void calculatePerimeter()

2

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

3

calculate circle extends shape

Circle (doubler)

{

$\pi = 3.14$

3

void calculateArea()

{

System.out.println ("Area = " + (Math.PI * r * r))

void calculatePerimeter()

{

System.out.println ("Perimeter = " + (2 * Math.PI * r))

}

3

class Shape { }

public static void main (String args) { }

triangle t = new triangle (2.0, 3.0);

t.calculateArea();

t.calculatePerimeter();

circle c = new circle (5);

c.calculateArea();

c.calculatePerimeter();

2 3

Output

Area = 4.145

Perimeter = 11.0

Area = 78.5398

Perimeter = 31.4159

Shruti Sathawa

P Lab - 6

create package CIE which has 2 classes student and internal. The class student has members vrn, name, sem. The class derived from student has an array that stores internal marks spread in 3 subjects

student.java

Package CIE

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

```
public class student {
```

```
protected String vrn = new String();
```

```
protected String name = new String();
```

```
protected int sum;
```

```
public void inputStudentDetails() {
```

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

```
this.vrn = s.nextLine();
```

```
this.name = s.nextLine();
```

```
this.sem = s.nextInt();
```

3

```
public void displayStudentDetails() {
```

```
System.out.println(this.vrn + " " +
```

```
this.name);
```

7

3

23-1-26

Package (I.E.);

import java.util.*;

Public class Internals extends Student {
 Protected int marks[] = new int[5];
 Public void Input (I.E. marks[]){
 Scanner s = new Scanner (System.in);
 for (int i=0; i<s.length(); i++)
 marks[i] = s.nextInt();
 }
 ?

External.java

Package SEE

import (I.E. Internals);

Import java.util.Scanner;

Public class External extends Internals {
 int marks[];

Protected int finalMarks[];

Public External () {

marks = new int [5];

finalMarks = new int [5]

3

Public void Output SEE marks[] {

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

marks[i] = s.nextInt();

3

Public void calculateFinalMarks () {

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

finalMarks[i] = marks[i]/2

+ Super. marks[i]

3

```

public void displayFinalMarks() {
    displayStudentDetails();
    for (i = 0; i < s; i++) {
        System.out.print("subject: " + (i + 1) F +
            finalMarks[i]);
    }
}

```

3

Math. Java

```
import SEE.Externals;
```

```
class Main {
```

```
    public static void main (String args[]) {
```

```
        int numofStudents = 2;
```

```
        External finalMarks[] = new External[2];
```

```
        for (int i = 0; i < numofStudents; i++) {
```

```
            finalMarks[i] = new External();
```

```
            finalMarks[i] = inputStudentDetails();
```

```
            System.out.println("Enter CIE marks")
```

```
            finalMarks[i] = inputCIEmarks();
```

```
            System.out.println("Enter SEE");
```

```
            finalMarks[i] = inputSEEmarks();
```

```
        for (int i = 0; i < numofStudents; i++) {
```

```
            final[i] = calculateFinalMarks();
```

```
            final[i].displayFinalMarks();
```

3

3

7

void display()

System.out.println("Father age")

3

3

(class Son extends Father

int sonAge

son() {

super();

try {

check();

3

catch (WrongAge e) {

System.out.println(e);

3

3

void check() throws WrongAge {

if (sonAge < 0)

throw new WrongAge("cannot be negtive");

else if (sonAge > fatherAge)

throw new WrongAge("son Age cannot be greater than father age");

else if (sonAge == fatherAge)

throw new WrongAge("Son Age can't be eq to father age");

3

void display()

System.out.println(e);

3

3

class main {

public static void main (String args[]) {

 Son s = new Son();
 s.display();

}

3
Output

50

30

Father age : 50

Son age : 30

- 12

Age cannot be negative

20

40

~~Son's age cannot be greater than Father's age.~~

06/02/10

Labs - 08

SURYA Gold

Date

Page

- * write a program which creates two threads, one thread displaying BMSC College of Engineering once every ten & another displaying "CSE" every two sec.

class displayMessage implements Runnable {

String message;
int interval;

displayMessage(String message, int interval);

this.message = message;

this.interval = interval;

}

public void run() {

while (true) {

System.out.println(message);

try {

Thread.sleep(interval * 1000);

}

catch (InterruptedException e) {

System.out.println(e);

}

}

3

public class multo {

public static void main(String args[]) {

Thread T1 = new Thread(new display
message("BMSCIE", 10));

Thread T2 = new Thread(new display
message("CSE", 2));

T1 - start();

T2 - start();

3

3

Output

CSE

CSE

CSE

BMSCS

CSE

CSIE

CSE

CSF

CSE

BMSCIE

CSE

CSE

CSE

CSE

CSE

BMSCS

Sudhanya

Lab-9

Demonstrate intercommunication and deadlock

class QS

int n;

boolean valueset = False;

Synchronized void put (int n)

while (!valueset) {

try {

System.out.println("In. continuous
wait()");

}

catch (InterruptedException e)

e

System.out.println("Interrupted exception
handled");

System.out.println("Best: " + n);

valueset = True;

System.out.println("Interrupted by");

notify();

return n;

}

Synchronized void put (int n)

while (!valueset)

try {

System.out.println("

Produced waiting");

wait();

}

catch (InterruptedException e)

e

y

public void run() {

 int i = 0;

 while (i < 15) {

 int r = q.get();

 }

System.out.println ("consumed " + s);

 i++;

}

}

3

class PCFinal {

 public static void main (String args) {

 Q q = new Q();

 new Producer (q);

 new Consumer (q);

}

3

output

Put : 0

Integrate consumer

Producer waiting

got : 0

Integrate producer

Consumed waiting

~~Put : 1~~

Integrate consumer

Producer waiting

got : 1

Integrate producer

consumed 1

Put : 2

Integrate consumer

Producer waiting

Got : 2

Integrate consumer

consumed : 2

Put 3

Integrate consumer

Producer waiting

got : 3

Integrate consumer

consumed : 3

put : 4

Deadlocking

Class A?

Synchronized void fo(B b)?
 String name = thread.currentThread().getName();
 System.out.println(name + " entered A::fo");
 try {

 thread.sleep(1000);
 }

catch (Exception e) {

 System.out.println("A interrupted");
 }

 System.out.println("B interrupted") trying to call
 B.last();

 b.last();

}

void last {

 System.out.println("inside A (last)");

}

Class B?

Synchronized void bar(A a)?

String name = thread.currentThread().getName();
 System.out.println(name + " entered B::bar");
 try {

 thread.sleep(1000);
 }

catch (Exception E) {

 System.out.println("B interrupted")

 System.out.println(name + " trying to call
 A.last());

 a.last();

}

void last() {

System.out.println("Inside A last")

}

class Deadlock implements runnable {

A a = new A();

B b = new B();

Deadlock t {

thread - currentThread().getName()
("Main Thread");

thread.set()

thread t = new Thread(this,
"Rushing Thread");

t.start();

a.foo(b);

System.out.println("Back to main thread");

}

public void run() {

b.bar(a);

System.out.println("Back to other thread");

}

public static void main (String args) {

new Deadlock();

}

outcomes

Main thread entered A first

Rushing thread entered B first

Rushing thread to call A's code

Inside A last

Back to main thread

Main thread trying to call B::last()
Inside A::last
~~But~~ in other thread.

4

13.08.20

Shree Siddhava 1

Lab 10

20/2/24
write a program that creates a GUI application to perform integer division. The user enters two numbers in the text. The division of num1 / num2 is displayed in result field when the divide button is clicked. If num1 or num2 were not an integer, the program would throw an Arithmetic exception in a message.

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

class swingDemo {

 swing demo()

```
}  
JFrame frm = new JFrame("Divide App");  
frm.setSize(275, 150);  
frm.setLayout(new FlowLayout());  
frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

JLabel jlab = new JLabel("Enter the
divisor and
dividend: ");

```
JButton button = new JButton("calculate");  
JLabel err = new JLabel(" ");  
JLabel alab = new JLabel();  
JLabel blab = new JLabel();  
JLabel cmlab = new JLabel();
```

```

jfrm.add (err);
jfrm.add (jlab);
jfrm.add (afile);
jfrm.add (bjfile);
jfrm.add (button);
jfrm.add (alab);
jfrm.add (blab);
jfrm.add (anlab);

```

ActionListener l = new ActionListener

{

public void actionPerformed (ActionEvent
e)

System.out.println ("Action event for
a Text field (l);")

g

agtf.add ActionListener (l);
bjtf.add ActionListener (l);

button.ActionListener (new ActionListener)

public void actionPerformed (ActionEvent e)

{

try {

int a = Integer.parseInt (ajtf.getText());

int b = Integer.parseInt (bjtf.getText());

int ans = a/b;

alab.setText ("\nA = " + a);

blab.setText ("\nB = " + b);

{

catch (number from at Exception e)

alab.setText(" ");

blab.setText(" ");

ans lab = set Text(" ");

3 text ("Enter only Integer!");

catch (ArithmaticException e) {

alab.setText(" ");

blab.setText(" ");

anslab.setText(" ");

err.setText("B should be non zero");

3

g3)

3 from setVisible(true));

3

public static void main(String args[]){

SwingUtilities.invokeLater(new Runnable(){

g

Public void run() {

new SwingDemo();

g

3;

3

3

output

Enter the dividend and divisor

10

2

~~A=10~~

B=2

Ans=5

Calculate

Shree Sudhamik

Function

JFrame :- The Java X Swing JFrame class is a type of container which inherits the Java.awt.Frame class. JFrame works like the main window.

setSize (int width, int height) - used to size a frame using width and height parameters

setLayout () :- methods allow you to set the layout of the container. The layout manager helps to lay out the components by this container.

setDefaultCloseOperation() - methods used to specify one of several options for the close button of frame. Exit on - close - Exit option

JTextField - the object of a JTextField class is a Text Component that allows the edition of a single line text. It inherits JTextField Component class.

add (frame) - adds new frame to the existing frame

setText () - This method substitutes new text in all or part of the text field. This works only with the first line of multi-line text field.

setVisible () :- Is a method that has return type boolean.

~~This
22.02.24~~

LAB: 1

Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read a, b, c and use the quadratic formula. If the discriminant b^2-4ac is negative, display a 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 of a,b,c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }
    void compute()
    {
        while(a==0)
        {
            System.out.println("Not a quadratic equation");
            System.out.println("Enter a non zero value for a:");
            Scanner s = new Scanner(System.in);
            a = s.nextInt();
        }
        d = b*b-4*a*c;
        if(d==0)
        {
            r1 = (-b)/(2*a);
            System.out.println("Roots are real and equal");
            System.out.println("Roo1 = Root2 = " + r1);
        }
        else if(d>0)
        {
            r1 = ((-b)+(Math.sqrt(d)))/(double)(2*a);
            r2 = ((-b)-(Math.sqrt(d)))/(double)(2*a);
            System.out.println("Roots are real and distinct");
            System.out.println("Roo1 = " + r1 + " Root2 = " + r2);
        }
        else if(d<0)
        {
            System.out.println("Roots are imaginary");
            r1 = (-b)/(2*a);
            r2 = Math.sqrt(-d)/(2*a);
            System.out.println("Root1 = " + r1 + " + i" + r2);
            System.out.println("Root1 = " + r1 + " - i" + r2);
        }
    }
}
```

```
class QuadraticMain
{
    public static void main(String args[])
    {
        Quadratic q = new Quadratic();
        q.getd();
        q.compute();
    }
}
```

LAB: 2

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

//Develop a java program to create a class Student with members usn,name, an array creadits 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{
```

```
    String name;
```

```
    String usn;
```

```
    double SGPA;
```

```
    Subject subject[];
```

```
    Scanner s;
```

```
    Student(){
```

```
        int i;
```

```
        subject=new Subject[9];
```

```
        for(i=0;i<9;i++){
```

```
            subject[i]=new Subject();
```

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

```
        }
```

```
}
```

```
    void getStudentDetails(){
```

```
        System.out.println("Enter your Name:");
```

```
        name=s.next();
```

```
        System.out.println("Enter your USN:");
```

```
        usn=s.next();
```

```
}
```

```
    void getMarks(){
```

```
        for(int i=0;i<9;i++){
```

```
            System.out.println("Enter marks for subject "+(i+1)+":");
```

```
            subject[i].subjectMarks=s.nextInt();
```

```
            System.out.println("Enter credits for subject "+(i+1)+":");
```

```
            subject[i].credits=s.nextInt();
```

```
            subject[i].grade=(subject[i].subjectMarks/10)+1;
```

```
            if (subject[i].grade==11){
```

```
                subject[i].grade=10;
```

```
}
```

```
            if (subject[i].grade<=4){
```

```
                subject[i].grade=0;
```

```
}
```

```
}
```

```
}
```

```
    void computeSGPA(){
```

```
        int effectiveScore=0;
```

```
        int totalCreadits=0;
```

```
for(int i=0;i<9;i++){
    effectiveScore += (subject[i].grade*subject[i].credits);
    totalCreadits += subject[i].credits;
}

SGPA=(double)effectiveScore/(double)totalCreadits;
}

class Subject{
    int subjectMarks;
    int credits;
    int grade;
}

class Main{
    public static void main(String args[]){
        Student s1=new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();

        System.out.println("Name:"+s1.name);
        System.out.println("USN:"+s1.usn);
        System.out.println("SGPA :" +s1.SGPA);
    }
}
```

LAB: 3

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 Book{
    String name, author;
    int price, page_num;

    Book(String name, String author, int price, int page_num){
        this.name=name;
        this.author=author;
        this.price=price;
        this.page_num=page_num;
    }

    String toStrings(){
        String name, author, price, page_num;
        name="Book name: "+this.name+"\n";
        author="Author name: "+this.author+"\n";
        price="Price: "+this.price+"\n";
        page_num="Number of pages: "+this.page_num+"\n";

        return (name+author+price+page_num);
    }

    public static void main(String args[]){
        int n;
        String name, author;
        int price, page_num;

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter the no. of books:");
        n=sc.nextInt();

        Book b[]=new Book[n];

        for (int i=0;i<n;i++){
            System.out.println("Enter name of book:");
            sc.nextLine();
            name=sc.nextLine();

            System.out.println("Enter author of a book:");
            author=sc.nextLine();

            System.out.println("Enter the price of book:");
            price=sc.nextInt();

            System.out.println("Enter the no. of pages of book:");
            page_num=sc.nextInt();
        }
    }
}
```

```
b[i]=new Book(name,author,price,page_num);  
}  
  
System.out.println("Book Details:");  
for(int i=0;i<n;i++){  
    System.out.println("Book "+(i+1)+" :\n"+b[i].toString());  
}  
}  
}
```

LAB: 4

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.

```
import java.util.Scanner;
public class InputScanner {
    Scanner s;
    InputScanner(){
        s=new Scanner(System.in);
    }
}
abstract public class Shape extends InputScanner{
    double a,b;
    abstract void getInput();
    abstract void displayArea();
}
public class Rectangle extends Shape {
    void getInput() {
        System.out.println("Enter the dimension of rectangle:");
        a=s.nextDouble();
        b=s.nextDouble();
    }

    void displayArea() {
        System.out.println("Area of reactangle:"+ (a*b));
    }
}
public class Circle extends Shape{
    void getInput() {
        System.out.println("Enter the dimension of circle:");
        a=s.nextDouble();
    }

    void displayArea() {
        System.out.println("Area of circle:"+(3.14*a*a));
    }
}
public class Triangle extends Shape {
    void getInput() {
        System.out.println("Enter the dimension of triangle:");
        a=s.nextDouble();
        b=s.nextDouble();
    }

    void displayArea() {
        System.out.println("Area of triangle:"+((a*b)/2));
    }
}
public class MainMethod {

    public static void main(String[] args) {
```

```
Rectangle R=new Rectangle();
R.getInput();
R.displayArea();

Triangle T=new Triangle();
T.getInput();
T.displayArea();

Circle C=new Circle();
C.getInput();
C.displayArea();
}

}
```

LAB: 5

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.

```
import java.util.Scanner;
public class Account {
    String customer_name;
    int account_no;
    String type_acc;
    double balance=0;
    Scanner sc;

    Account(String customer_name,int account_no,String type_acc){
        this.customer_name=customer_name;
        this.account_no=account_no;
        this.type_acc=type_acc;
        sc=new Scanner(System.in);
    }

    void deposit() {
        System.out.println("Enter the deposit amount:");
        int dipo=sc.nextInt();
        balance+=dipo;
    }

    void withdrawal() {
        System.out.println("Enter the withdrawal amount:");
        int with=sc.nextInt();
        if(with>balance) {
            System.out.println("Insufficient Balance");
        }
        else{
            balance-=with;
        }
    }

    void display() {
        System.out.println("Customer name:"+customer_name);
        System.out.println("Account number:"+account_no);
        System.out.println("Type of Account:"+type_acc);
        System.out.println("Balance:"+balance);
    }

    void applyinterest() {}

}

public class Cur_acct extends Account {

    Cur_acct(String customer_name,int account_no,String type_acc){
        super(customer_name,account_no,type_acc);}
}
```

```

void withdrawal() {
    System.out.println("Enter the withdrawal amount:");
    int with=sc.nextInt();
    if (balance<=2000) {
        double pen=balance/(0.06);
        System.out.println("Insufficient balance penalty to be paid:"+pen);
        balance+=pen;
    }
    else{
        balance-=with;
    }
}

public class Sav_acct extends Account{
Sav_acct(String customer_name,int account_no,String type_acc){
super(customer_name,account_no,type_acc);
}

void applyinterest() {
System.out.println("Enter the interest rate:");
int rate=sc.nextInt();
double interest=balance*rate;
balance+=interest;
System.out.println("Balance after interest:"+balance);
}
}

import java.util.Scanner;
public class Bank {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter customer name:");
        String cus_name=sc.nextLine();
        System.out.println("Enter account number:");
        int acc_no=sc.nextInt();

        Account ca=new Cur_acct(cus_name,acc_no,"Current Account");
        Account sa=new Sav_acct(cus_name,acc_no,"Saving Account");

        int choice;
        while(true){
            System.out.println("----MENU----");
            System.out.println("1.Deposite\n2.Withdrawl\n3.Compute interest for Saving account\n4.Display account details\n5.Exit");
            choice=sc.nextInt();

            switch(choice) {
            case 1:
                System.out.println("Enter the type of account:\n1.Saving Account \n2.Current Account");
                int acc=sc.nextInt();
                if(acc==1) {

```

```
        sa.diposite();
    }
    else {
        ca.diposite();
    }
    break;
    case 2:
        System.out.println("Enter the type of account:\n1.Saving Account \n2.Current Account");
        int acc1=sc.nextInt();
        if(acc1==1) {
            sa.withdrawal();
        }
        else {
            ca.withdrawal();
        }
        break;
    case 3:
        sa.applyinterest();
        break;
    case 4:
        System.out.println("Enter the type of account:\n1.Saving Account \n2.Current Account");
        int acc2=sc.nextInt();
        if(acc2==1) {
            sa.display();
        }
        else {
            ca.display();
        }
        break;
    case 5:
        break;
    default:
        System.out.println("Invalid Choice");
        break;
    }

    if(choice==5) {
        break;
    }
}
}
```

LAB: 6

Create a package CIE which has two classes- Student and Internals. The class Student has members like usn, name, sem. The class Internals derived from Student 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;
import java.util.Scanner;
public class Student{
    public String usn,name;
    public int sem;
    public void inputStudentDetails(){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the student usn:");
        usn=sc.nextLine();
        System.out.println("Enter the student name:");
        name=sc.nextLine();
        System.out.println("Enter student semester:");
        sem=sc.nextInt();
    }
}

public void dispylevel(){
    System.out.println("Student USN:"+usn);
    System.out.println("Student Name:"+name);
    System.out.println("Student Sem:"+sem);
}
}

package CIE;
import java.util.Scanner;
public class Internals extends Student{
    public int marks[] = new int[5];
    public void inputCIEmarks(){
        Scanner sc=new Scanner(System.in);
        for(int i=0;i<5;i++){
            System.out.println("Enter the marks for subject "+(i+1)+":");
            marks[i]=sc.nextInt();
        }
    }
}

package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends CIE.Internals{
    public int marks[];
    public int finalMarks[];

    public Externals(){
        marks=new int[5];
        finalMarks=new int[5];
    }
}
```

```

public void inputSEEMarks(){
    Scanner sc=new Scanner(System.in);
    for(int i=0;i<5;i++){
        System.out.println("Enter subject "+(i+1)+" marks:");
        marks[i]=sc.nextInt();
    }
}

public void calculateFinalMarks(){
    for(int i=0;i<5;i++){
        finalMarks[i]=marks[i]/2+super.marks[i];
    }
}

public void displayFinalMarks(){
    inputStudentDetails();
    for(int i=0;i<5;i++){
        System.out.println("Subject "+(i+1)+" final marks:"+finalMarks[i]);
    }
}

import SEE.Externals;
class PackageMain{
    public static void main(String args[]){
        int numOfStudent=2;
        Externals finalMarks[]=new Externals[numOfStudent];
        for(int i=0;i<numOfStudent;i++){
            finalMarks[i]=new Externals();
            finalMarks[i].inputStudentDetails();
            System.out.println("Enter CIE Marks:");
            finalMarks[i].inputCIEmarks();
            System.out.println("Enter SEE Marks:");
            finalMarks[i].inputSEEMarks();
        }
        System.out.println("Displaying data:\n");
        for(int i=0;i<numOfStudent;i++){
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].display();
            finalMarks[i].displayFinalMarks();
        }
    }
}

```

LAB: 7

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<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.

```
public class WrongAge extends Exception{
    WrongAge(String msg){
        super(msg);
    }
}
import java.util.Scanner;
class InputScanner{
    Scanner sc;
    InputScanner(){
        sc=new Scanner(System.in);
    }
}
class Father extends InputScanner{
    int fatherAge;
    Father() throws WrongAge{
        System.out.println("Enter father's age:");
        fatherAge=sc.nextInt();
        if(fatherAge<0){
            throw new WrongAge("Age cannot be negative");
        }
    }
    void display(){
        System.out.println("Father's age:"+fatherAge);
    }
}
class Son extends Father{
    int sonAge;
    Son() throws WrongAge{
        System.out.println("Enter Son's age:");
        sonAge=sc.nextInt();
        if(sonAge>= fatherAge){
            throw new WrongAge("Son's age cannot be greater than father's age");
        }
        else if(sonAge<0){
            throw new WrongAge("Age cannot be negative");
        }
    }
    void display(){
        super.display();
        System.out.println("Son's age:"+sonAge);
    }
}
public static void main(String args[]){
```

```
try{
    Son son=new Son();
    son.display();
}
catch(WrongAge e){
    System.out.println(e);
}
```

LAB: 8

Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

```
class BMS_College_of_Engineering implements Runnable {  
    public void run() {  
        while (true) {  
            try {  
                System.out.println("BMS College of Engineering");  
                Thread.sleep(10000); // Sleep for 10000 milliseconds (10 seconds)  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
  
    class CSE implements Runnable {  
        public void run() {  
            while (true) {  
                try {  
                    System.out.println("CSE");  
                    Thread.sleep(2000); // Sleep for 2000 milliseconds (2 seconds)  
                } catch (InterruptedException e) {  
                    e.printStackTrace();  
                }  
            }  
        }  
    }  
  
    public class ThreadMain {  
        public static void main(String[] args) {  
            Thread t1 = new Thread(new BMS_College_of_Engineering());  
            Thread t2 = new Thread(new CSE());  
            t1.start();  
            t2.start();  
        }  
    }  
}
```

LAB: 9

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

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

class SwingDemo{
    SwingDemo(){
        // create jframe container
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        // to terminate on close
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // text label
        JLabel jlab = new JLabel("Enter the divider and divident:");

        // add text field for both numbers
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);

        // calc button
        JButton button = new JButton("Calculate");

        // labels
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();

        // add in order :
        jfrm.add(err); // to display error bois
        jfrm.add(jlab);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
        jfrm.add(button);
        jfrm.add(alab);
        jfrm.add(blab);
        jfrm.add(anslab);

        ActionListener l = new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                System.out.println("Action event from a text field");
            }
        };
        ajtf.addActionListener(l);
        bjtf.addActionListener(l);
    }
}
```

```

button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try{
            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());
            int ans = a/b;

            alab.setText("\nA = " + a);
            blab.setText("\nB = " + b);
            anslab.setText("\nAns = " + ans);
        }
        catch(NumberFormatException e){
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("Enter Only Integers!");
        }
        catch(ArithmaticException e){
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("B should be NON zero!");
        }
    }
});

// display frame
jfrm.setVisible(true);
}

public static void main(String args[]){
    // create frame on event dispatching thread
    SwingUtilities.invokeLater(new Runnable(){
        public void run(){
            new SwingDemo();
        }
    });
}
}

```

LAB: 10

Demonstrate Inter process Communication and deadlock.

```
class Q {  
    int n;  
    boolean valueSet = false;  
  
    synchronized int get() {  
        while(!valueSet)  
            try {  
                System.out.println("\nConsumer waiting\n");  
                wait();  
            } catch(InterruptedException e) {  
                System.out.println("InterruptedException caught");  
            }  
        System.out.println("Got: " + n);  
        valueSet = false;  
        System.out.println("\nIntimate Producer\n");  
        notify();  
        return n;  
    }  
  
    synchronized void put(int n) {  
        while(valueSet)  
            try {  
                System.out.println("\nProducer waiting\n");  
                wait();  
            } catch(InterruptedException e) {  
                System.out.println("InterruptedException caught");  
            }  
        this.n = n;  
        valueSet = true;  
        System.out.println("Put: " + n);  
        System.out.println("\nIntimate Consumer\n");  
        notify();  
    }  
}  
  
class Producer implements Runnable {  
    Q q;  
    Producer(Q q) {  
        this.q = q;  
        new Thread(this, "Producer").start();  
    }  
    public void run() {  
        int i = 0;  
        while(i<5) {  
            q.put(i++);  
        }  
    }  
}
```

```

class Consumer implements Runnable {
    Q q;
    Consumer(Q q) {
        this.q = q;
        new Thread(this, "Consumer").start();
    }

    public void run() {
        int i=0;
        while(i<5) {
            int r=q.get();
            System.out.println("consumed:"+r);
            i++;
        }
    }
}

class PCFixed {
    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
    }
}

```

Deadlock:

```

class A {
    synchronized void foo(B b) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered A.foo");
        try {
            Thread.sleep(1000);
        } catch(Exception e) {
            System.out.println("A Interrupted");
        }
        System.out.println(name + " trying to call B.last()");
        b.last();
    }

    void last() {
        System.out.println("Inside A.last");
    }
}

class B {
    synchronized void bar(A a) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered B.bar");
        try {

```

```
        Thread.sleep(1000);
    } catch(Exception e) {
        System.out.println("B Interrupted");
    }
    System.out.println(name + " trying to call A.last()");
    a.last();
}
void last() {
    System.out.println("Inside A.last");
}
}

class Deadlock implements Runnable
{
    A a = new A();
    B b = new B();
    Deadlock() {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this,"RacingThread");
        t.start();
        a.foo(b); // get lock on a in this thread.
        System.out.println("Back in main thread");

    }
    public void run() {
        b.bar(a); // get lock on b in other thread.
        System.out.println("Back in other thread");
    }
}

public static void main(String args[]) {
    new Deadlock();
}
}
```