VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming (23CS3PCOOJ)

Submitted by

Bhoomi Suresh Kota (1BM23CS065)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Sep-2024 to Jan-2025

B.M.S. College of Engineering,

Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled “Object Oriented Java Programming (23CS3PCOOJ)” carried out by **Bhoomi Suresh Kota (1BM23CS065),** who is a bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Mrs. Swathi Sridharan Assistant Professor Department of CSE, BMSCE	**Dr. Jyothi S Nayak** Professor & HOD Department of CSE, BMSCE

Index

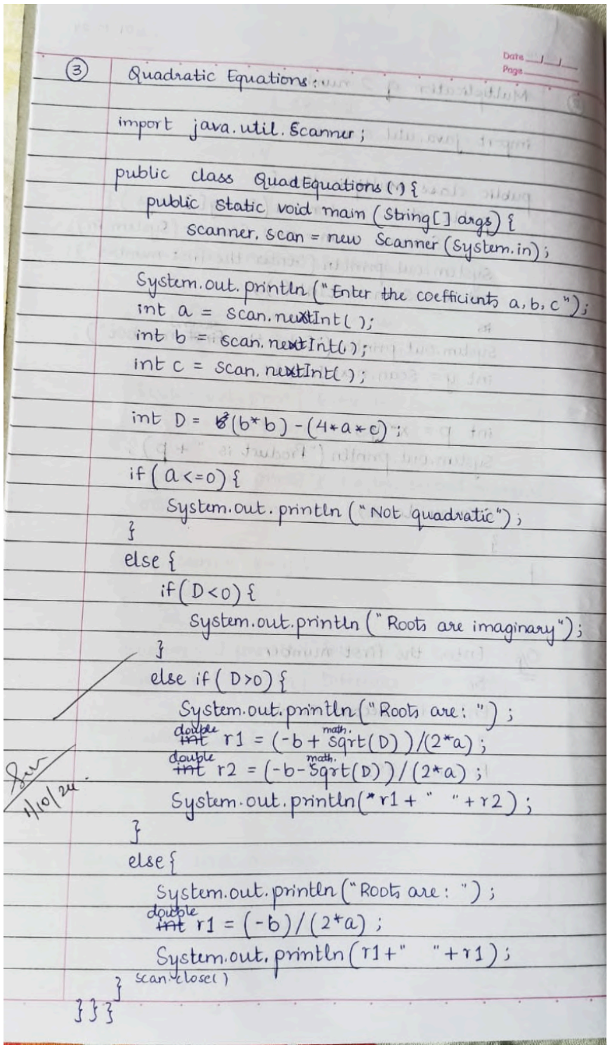
Sl. No.	Date	Experiment Title	Page No.
1	01-10-2024	Quadratic Equations	4
2	08-10-2024	SGPA Calculator	6
3	15-10-2024	Getter setter methods	9
4	22-10-2024	Abstract classes	12
5	29-10-2024	Bank Account	16
6	12-11-2024	Packages	20
7	19-11-2024	Interfaces	23
8	26-11-2024	Exceptions	26
9	3-12-2024	Threads	28

Github Link:
https://github.com/BhoomiSuresh/OOJ.git

Program 1

Implement Quadratic Equation

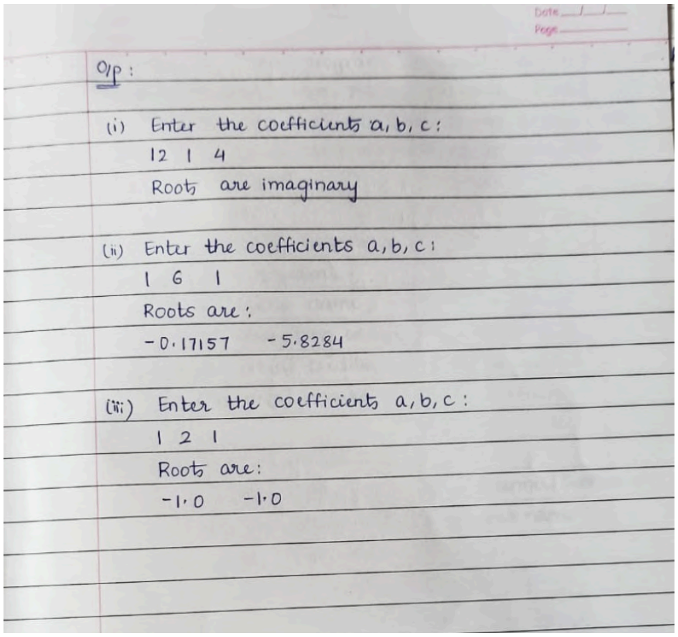
Algorithm:



③ Quadratic Equations :

```
import java.util.Scanner;

public class QuadEquations () {
    public static void main (String[] args) {
        Scanner, scan = new Scanner (System.in);

      System.out.println ("Enter the coefficients a, b, c");
      int a = scan.nextInt( );
      int b = scan.nextInt( );
      int c = scan.nextInt( );

      int D = (b*b) - (4*a*c);

      if (a<=0) {
          System.out.println ("Not quadratic");
      }
      else {
          if (D<0) {
              System.out.println ("Roots are imaginary");
          }
          else if (D>0) {
              System.out.println ("Roots are: ");
              double r1 = (-b + math.sqrt(D))/(2*a);
              double r2 = (-b-math.sqrt(D))/(2*a);
              System.out.println("r1 + "  "+r2);
      }
      else {
          System.out.println ("Roots are: ");
          double r1 = (-b)/(2*a);
          System.out.println(r1+"  "+r1);
      }  scan.close()
}}}
```

1/10/24



O/p :

```
(i)   Enter the coefficients a, b, c :
      12  1  4
      Roots are imaginary

(ii)  Enter the coefficients a, b, c :
      1  6  1
      Roots are :
      -0.17157    -5.8284

(iii) Enter the coefficients a, b, c :
      1  2  1
      Roots are:
      -1.0    -1.0
```

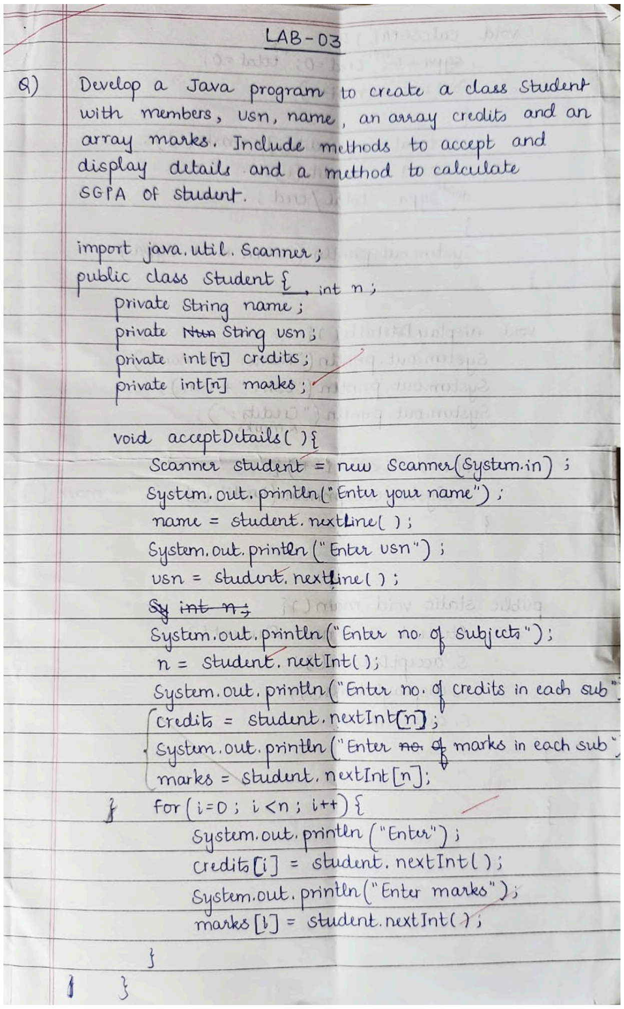
Code:

```
import java.util.*;
import java.lang.Math;
public class Main{
public static void main(String[] args) {
Scanner scan = new Scanner(System.in);
System.out.println("Enter the coefficients a, b, c");
int a = scan.nextInt();
int b= scan.nextInt();
int c = scan.nextInt();
int D = (b*b) - (4*a*c);
if(a <= 0)
System.out.println("Not quadratic");
else {
if(D < 0){
System.out.println("Roots are imaginary");
}
else if (D > 0) {
System.out.println("Roots are: ");
double r1= (-b+Math.sqrt(D)) /(2*a);
double r2=(-b-Math.sqrt(D))/(2*a);
System.out.println(r1+" "+r2);
}
else {
System.out.println("Roots are: ");
double r1 = (- b) / (2 * a);
System.out.println ( r1+" "+r1);
}
}
}
}
```

Program 2

Implement SGPA Calculator

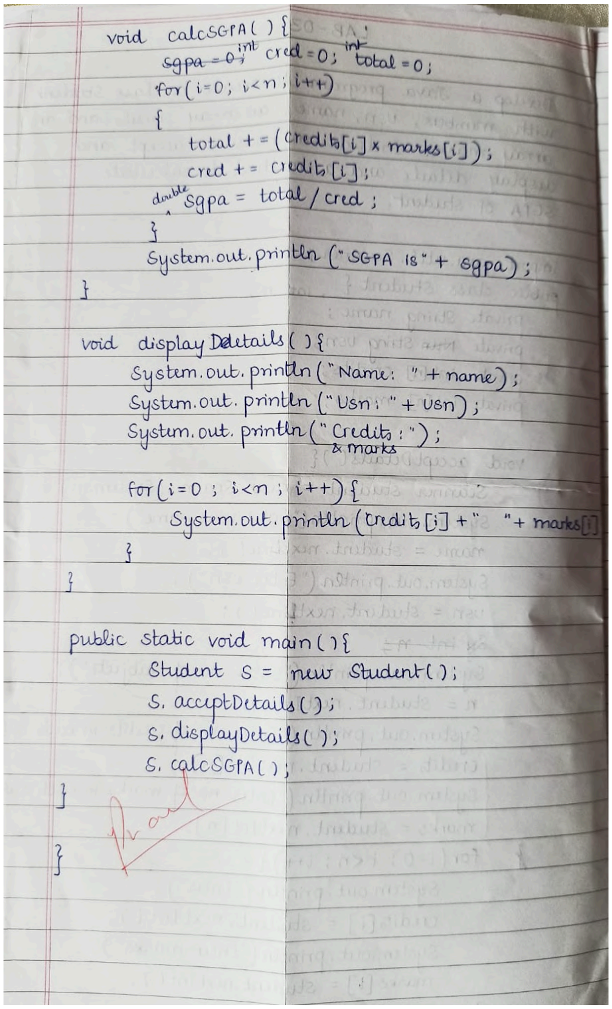
Algorithm:



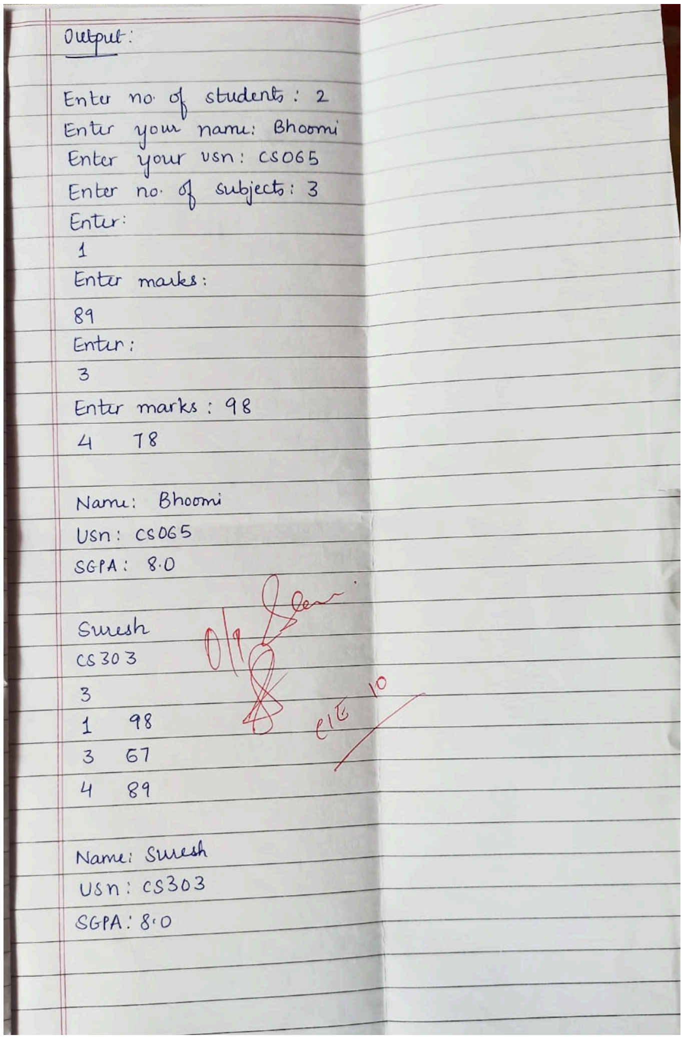
LAB-03

Q) 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 student.

```
import java.util.Scanner;
public class Student {, int n;
    private String name;
    private ~~Nun~~ String usn;
    private int[n] credits;
    private int[n] marks;

    void acceptDetails( ){
        Scanner student = new Scanner(System.in) ;
        System.out.println("Enter your name") ;
        name = student.nextLine( ) ;
        System.out.println ("Enter usn") ;
        usn = student.nextLine( ) ;
        ~~Sy int n;~~
        System.out.println("Enter no. of subjects");
        n = Student.nextInt( );
        System.out.println("Enter no. of credits in each sub"
        credits = student.nextInt[n];
        System.out.println("Enter ~~no. of~~ marks in each sub"
        marks = student.nextInt[n];
    }   for ( i=0 ; i<n ; i++) {
            System.out.println ("Enter") ;
            credits[i] = student.nextInt( );
            System.out.println("Enter marks");
            marks[i] = student.nextInt( );

        }
}   }
```



```
void calcSGPA( ) {
    ~~sgpa = 0;~~ int cred = 0; int total = 0;
    for(i=0; i<n; i++)
    {
        total += (credits[i] x marks[i]);
        cred += credits[i];
    double sgpa = total / cred ;
    }
    System.out.println ("SGPA is" + sgpa);
}

void display ~~D~~details ( ) {
    System.out.println ("Name:  " + name);
    System.out.println ("Usn: " + usn);
    System.out.println ("Credits & marks : ");

    for(i=0 ; i<n ; i++) {
        System.out.println (credits[i] + "   "+ marks[i]
    }
}

public static void main ( ){
        Student S = new Student( );
        S. acceptDetails( );
        S. displayDetails( );
        S. calcSGPA( );
}

}
```



Output:

```
Enter no. of students : 2
Enter your name: Bhoomi
Enter your usn: CS065
Enter no. of subjects: 3
Enter:
1
Enter marks:
89
Enter:
3
Enter marks : 98
4    78

Name: Bhoomi
Usn: CS065
SGPA: 8.0

Suresh
CS303
3
1    98
3    67
4    89

Name: Suresh
Usn: CS303
SGPA: 8.0
```

CIE 10

Code:

```
import java.util.Scanner;

public class Student {
  private String name;
  private String usn;
  private int[] credits;
  private int[] marks;
  private int n;

  public void acceptDetails() {
    Scanner student = new Scanner(System.in);

    System.out.println("Enter your name:");
    name = student.nextLine();
```

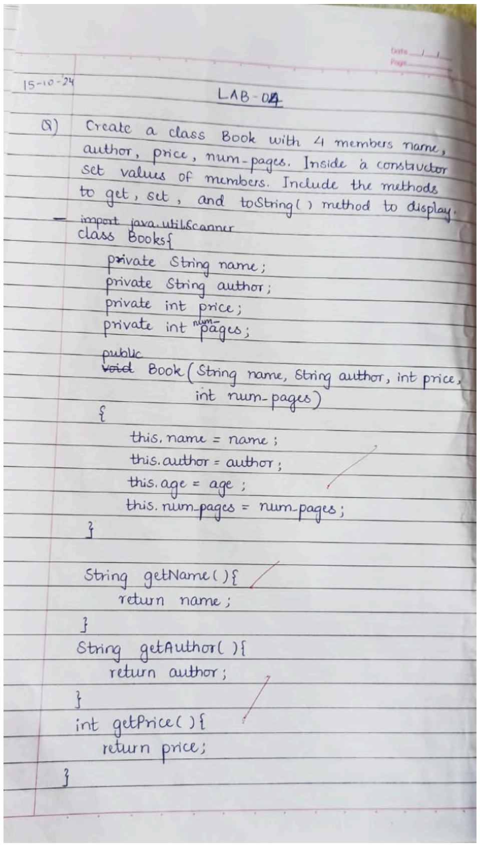
```
        System.out.println("Enter USN:");
        usn = student.nextLine();

        System.out.println("Enter no of subjects:");
        n = student.nextInt();

        credits = new int[n];
        marks = new int[n];

        System.out.println("Enter no of credits in each sub:");
        for (int i = 0; i < n; i++) {
            credits[i] = student.nextInt();
        }

        System.out.println("Enter no of marks in each sub:");
        for (int i = 0; i < n; i++) {
            marks[i] = student.nextInt();
        }

        student.close();
    }

    public void displayDetails() {
        System.out.println("Name: " + name);
        System.out.println("USN: " + usn);
        System.out.println("Credits and Marks:");

        for (int i = 0; i < n; i++) {
            System.out.println(credits[i] + " " + marks[i]);
        }
    }

    public void calcSGPA() {
        int total = 0;
        int cred = 0;

        for (int i = 0; i < n; i++) {
            total += credits[i] * marks[i];
            cred += credits[i];
        }

        double sgpa = (double) total / cred;
        System.out.println("SGPA: " + sgpa);
    }
```

```
public static void main(String[] args) {
   Student student = new Student();
   student.acceptDetails();
   student.displayDetails();
   student.calcSGPA();
}}
```

Program 3

Implement getter setter methods

Algorithm:



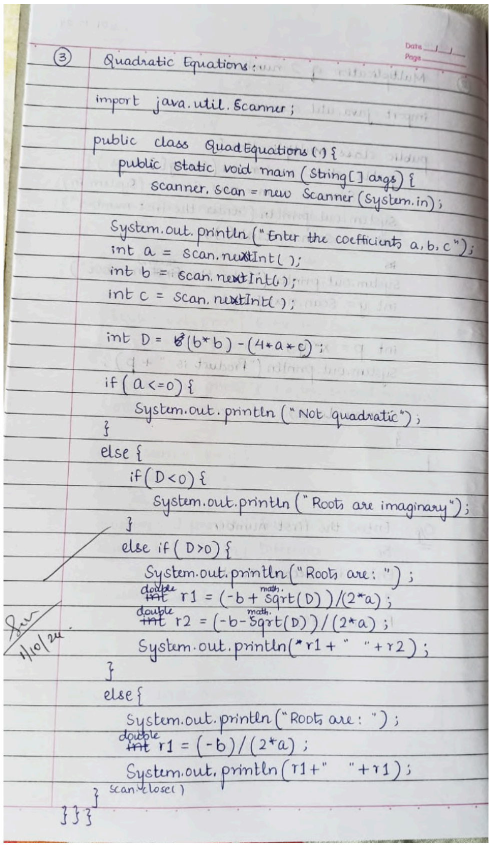
Date / /
Page

15-10-24

LAB-04

Q) Create a class Book with 4 members name, author, price, num-pages. Inside a constructor set values of members. Include the methods to get, set, and toString( ) method to display.

```
~~import java.util.Scanner~~
class Books{
    private String name;
    private String author;
    private int price;
    private int num-pages;

    ~~void~~ public Book(String name, String author, int price,
                    int num-pages)
    {
        this.name = name;
        this.author = author;
        this.age = age;
        this.num_pages = num-pages;
    }

    String getName(){
        return name;
    }
    String getAuthor(){
        return author;
    }
    int getPrice(){
        return price;
}
```

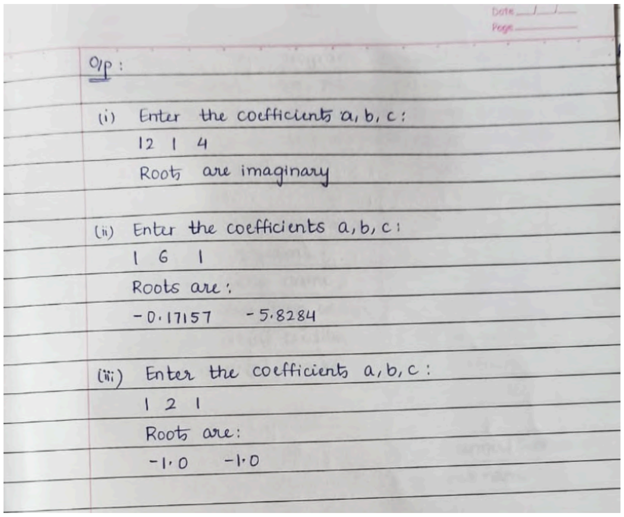


Date / /
Page

③ Quadratic Equations:

```
import java.util.Scanner;

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

     System.out.println("Enter the coefficients a,b,c");
     int a = scan.nextInt();
     int b = scan.nextInt();
     int c = scan.nextInt();

     int D = (b*b)-(4*a*c);

     if(a<=0){
         System.out.println("Not quadratic");
     }
     else{
         if(D<0){
             System.out.println("Roots are imaginary");
         }
         else if(D>0){
             System.out.println("Roots are: ");
             ~~int~~ double r1 = (-b+math.sqrt(D))/(2*a);
             ~~int~~ double r2 = (-b-math.sqrt(D))/(2*a);
             System.out.println(r1+"  "+r2);
         }
         else{
             System.out.println("Roots are: ");
             ~~int~~ double r1 = (-b)/(2*a);
             System.out.println(r1+"  "+r1);
         } scan.close()
}}}
```

1/10/24



```
O/p :

(i)   Enter the coefficients a, b, c :
      12  1  4
      Roots are imaginary

(ii)  Enter the coefficients a, b, c :
      1  6  1
      Roots are :
      -0.17157    -5.8284

(iii) Enter the coefficients a, b, c :
      1  2  1
      Roots are:
      -1.0    -1.0
```

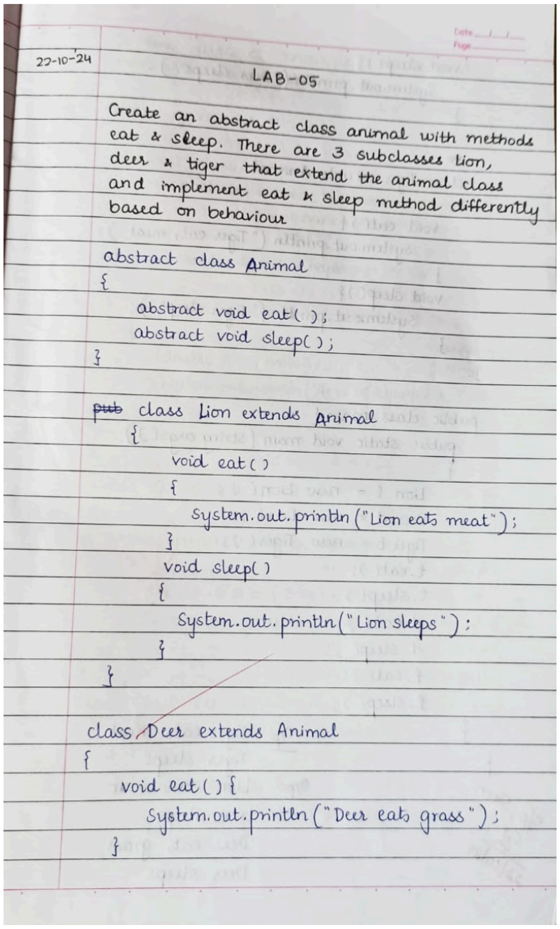
Code:

```
import java.util.Scanner;

class Book {
    private String name;
    private String author;
    private double price;
    private int numPages;

    public Book(String name, String author, double price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

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

    public String getName() {
        return name;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public String getAuthor() {
        return author;
    }
    public void setPrice(double price) {
        this.price = price;
    }

    public double getPrice() {
        return price;
    }

    public void setNumPages(int numPages) {
        this.numPages = numPages;
    }

    public int getNumPages() {
        return numPages;
    }

    @Override
    public String toString() {
        return "Book{" +
                "name='" + name + '\'' +
                ", author='" + author + '\'' +
                ", price=" + price +
                ", numPages=" + numPages +
                '}';
    }
}

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

        System.out.print("Enter the number of books: ");
        int n = scanner.nextInt();

        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for book " + (i + 1));
            System.out.print("Name: ");
            String name = scanner.next();
            System.out.print("Author: ");
            String author = scanner.next();
            System.out.print("Price: ");
            double price = scanner.nextDouble();
            System.out.print("Number of pages: ");
            int numPages = scanner.nextInt();

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

        System.out.println("\nBook Details:");
        for (Book book : books) {
            System.out.println(book);
        }
    }
}
```

Program 4

Implement Abstract classes

Algorithm:



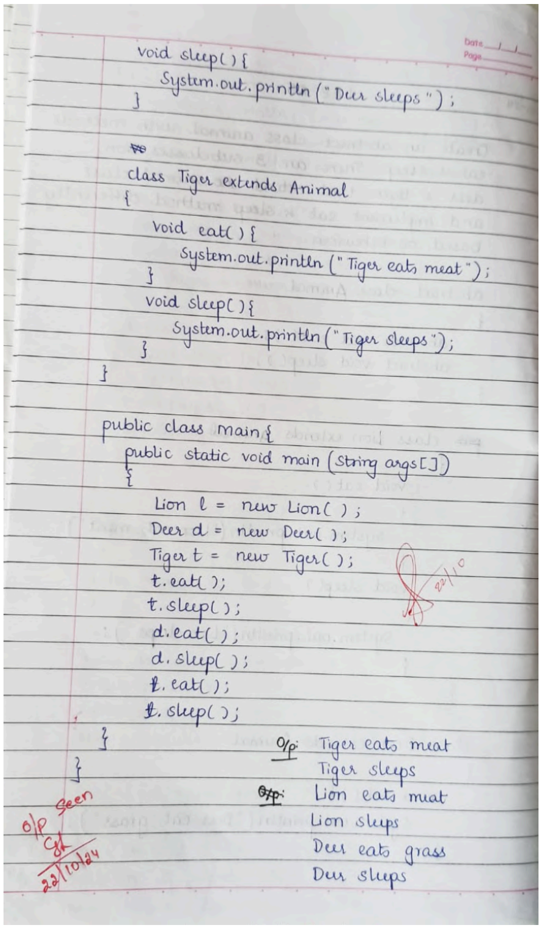
Date / /
Page

22-10-24

LAB-05

Create an abstract class animal with methods eat & sleep. There are 3 subclasses lion, deer & tiger that extend the animal class and implement eat & sleep method differently based on behaviour

```
abstract class Animal
{
    abstract void eat( );
    abstract void sleep( );
}

~~pub~~ class Lion extends Animal
  {
      void eat ( )
      {
         System.out.println ("Lion eats meat");
      }
      void sleep( )
      {
         System.out.println("Lion sleeps");
      }
  }

class Deer extends Animal
{
   void eat ( ) {
      System.out.println ("Deer eats grass");
   }
```



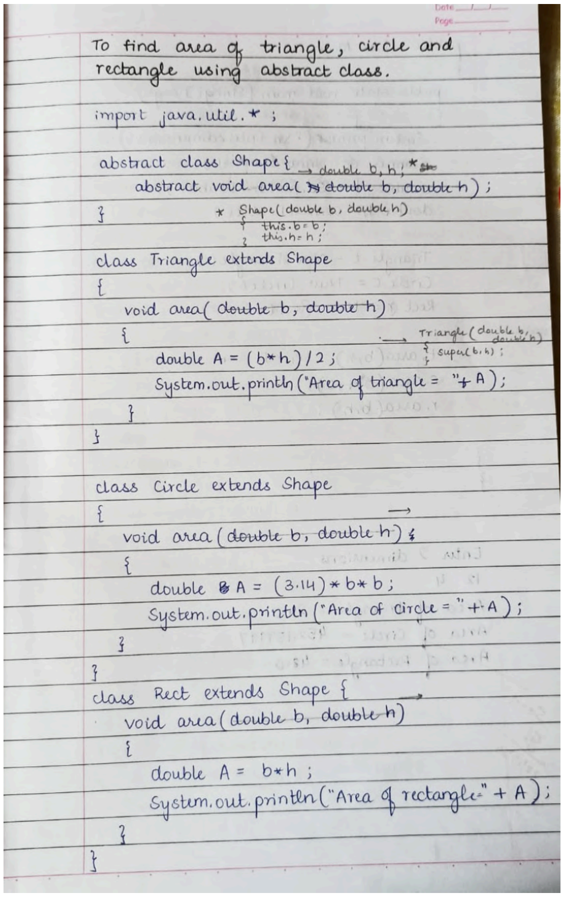
Date / /
Page

```
    void sleep( ) {
       System.out.println ("Deer sleeps");
    }

  class Tiger extends Animal
  {
     void eat( ) {
        System.out.println ("Tiger eats meat");
     }
     void sleep( ){
        System.out.println("Tiger sleeps");
     }
  }

public class main{
   public static void main (String args[])
   {
      Lion l = new Lion( );
      Deer d = new Deer( );
      Tiger t = new Tiger( );
      t.eat( );
      t.sleep( );
      d.eat( );
      d.sleep( );
      l.eat( );
      l.sleep( );
   }
}
```

22/10

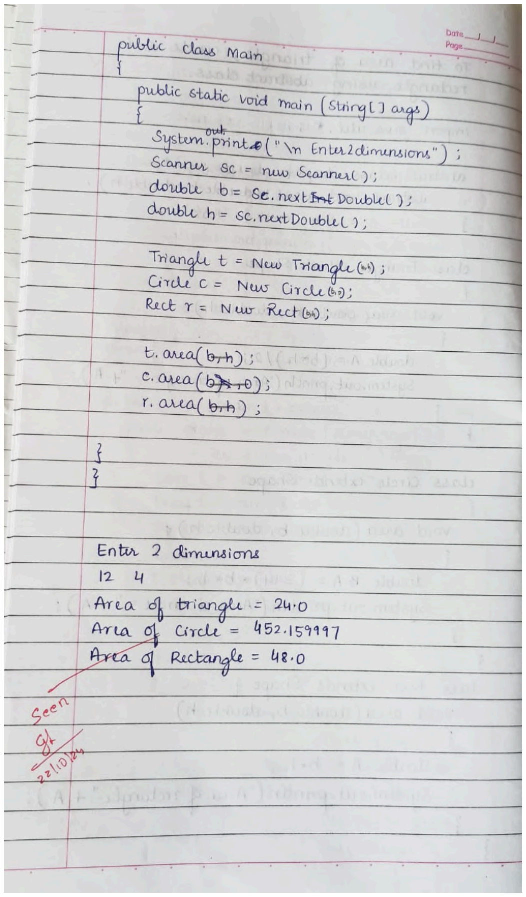
O/p:
```
Tiger eats meat
Tiger sleeps
Lion eats meat
Lion sleeps
Deer eats grass
Deer sleeps
```

o/p seen
22/10/24



To find area of triangle, circle and rectangle using abstract class.

```
import java.util.* ;

abstract class Shape{ → double b, h; *
    abstract void area( ~~double b, double h~~ ) ;
}              * Shape( double b, double h)
                 {  this.b = b;
                 }  this.h = h;
class Triangle extends Shape
{
   void area( ~~double b, double h~~ )
   {                                   → Triangle ( double b, double h)
                                         { super(b,h) ;
       double A = (b*h)/2 ;              }
       System.out.printlh ("Area of triangle =  "+ A );
   }
}

class Circle extends Shape
{                                      →
   void area ( ~~double b, double h~~ ) {
   {
      double A = (3.14) * b* b ;
      System.out.println ("Area of circle = "+ A ) ;
   }
}
class Rect extends Shape {           →
   void area( ~~double b, double h~~ )
   {
      double A =  b*h ;
      System.out.println("Area of rectangle=" + A );
   }
}
```



```
public class Main
{
  public static void main (String[] args)
  {
     System.out.print("\n Enter2dimensions") ;
     Scanner sc = new Scanner();
     double  b = sc.next~~Int~~Double();
     double h = sc.nextDouble();

     Triangle t = New Triangle(b,h) ;
     Circle c =  New  Circle(b,0);
     Rect r = New  Rect(b,h);

     t. area( ~~b,h~~ ) ;
     c. area( ~~b, 0~~ ) ;
     r. area( ~~b,h~~ ) ;

  }
}
```

```
Enter 2 dimensions
12  4
Area of triangle = 24.0
Area of Circle =  452.159997
Area of Rectangle = 48.0
```

Seen
22/10/24

Code:

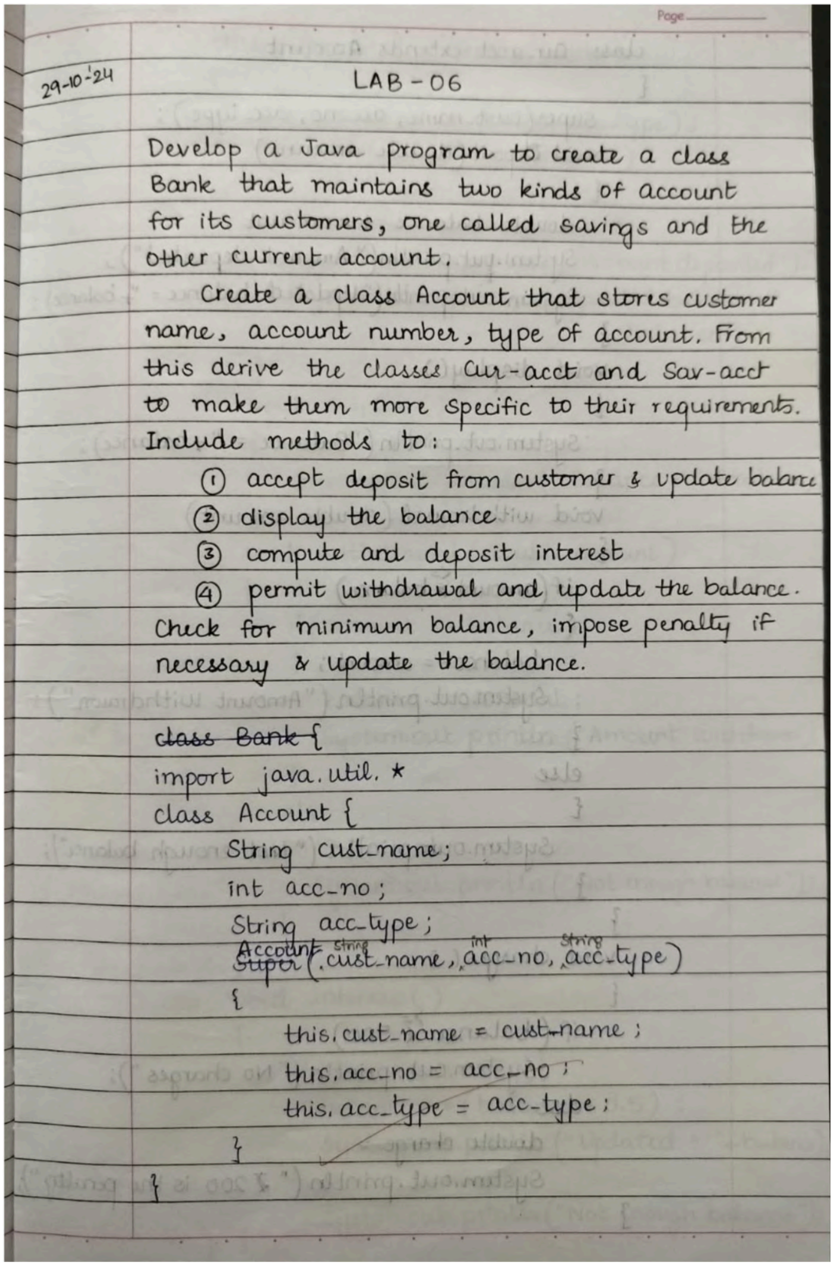
/****** P1 ****/

```
abstract class Animal {
    abstract void eatAndSleep();
}

class Lion extends Animal {
    void eatAndSleep() {
        System.out.println("Lion: Eats meat, and sleeps in a den.");
    }
}

class Deer extends Animal {
    void eatAndSleep() {
        System.out.println("Deer: Grazes on grass, and sleeps under trees.");
    }
}

class Tiger extends Animal {
    void eatAndSleep() {
        System.out.println("Tiger: Eats meat, and rests in dense forests.");
    }
}

public class Main {
    public static void main(String[] args) {
        Animal lion = new Lion();
        Animal deer = new Deer();
        Animal tiger = new Tiger();
        System.out.println("Animal Behaviors:");
        lion.eatAndSleep();
        deer.eatAndSleep();
        tiger.eatAndSleep();
    }
}


/****** P2 ****/

abstract class Shape {
    int x, y;

    abstract void printArea();
}

class Rectangle extends Shape {
    Rectangle(int x, int y) {
        this.x = x;
        this.y = y;
    }

    @Override
    void printArea() {
        System.out.println("Area of Rectangle: " + (x * y));
    }
}

class Triangle extends Shape {
    Triangle(int x, int y) {
        this.x = x;
        this.y = y;
    }

    @Override
    void printArea() {
        System.out.println("Area of Triangle: " + (0.5 * x * y));
    }
}

class Circle extends Shape {
    Circle(int x) {
        this.x = x;
    }

    @Override
    void printArea() {
        System.out.println("Area of Circle: " + (3.14 * x * x));
    }
}

public class Main {
    public static void main(String[] args) {
        Shape rectangle = new Rectangle(5, 4);
        Shape triangle = new Triangle(6, 8);
        Shape circle = new Circle(5);

        rectangle.printArea();
        triangle.printArea();
        circle.printArea();
    }}
```

Program 5

Implement Bank Account

Algorithm:



29-10-24

LAB - 06

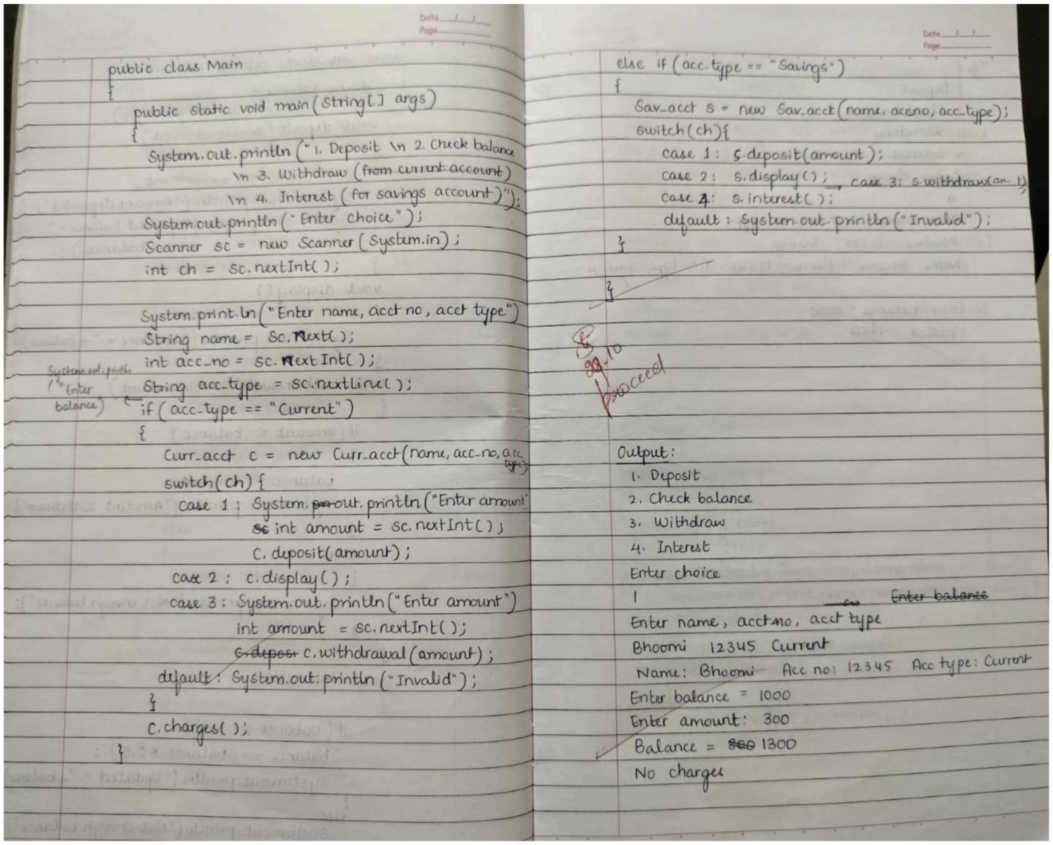
Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings and the other current account.

Create a class Account that stores customer name, account number, type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include methods to:

1. accept deposit from customer & update balance
2. display the balance
3. compute and deposit interest
4. permit withdrawal and update the balance.

Check for minimum balance, impose penalty if necessary & update the balance.

```
~~class Bank {~~
import java.util.*
class Account {
        String cust_name;
        int acc_no;
        String acc_type;
        ~~Super~~ Account(String cust_name, int acc_no, String acc_type)
        {
            this.cust_name = cust_name;
            this.acc_no = acc_no;
            this.acc_type = acc_type;
        }
}
```



```
public class Main
{
   public static void main(String[] args)
   {
      System.out.println("1. Deposit \n 2. Check balance
                \n 3. Withdraw (from current account)
                \n 4. Interest (for savings account)");
      System.out.println("Enter choice");
      Scanner sc = new Scanner(System.in);
      int ch = sc.nextInt();

      System.print.ln("Enter name, acct no, acct type")
      String name = Sc.Next();
      int acc_no = sc.NextInt();
      String acc_type = sc.nextLine();
      if(acc-type == "Current")
      {
         Curr_acct c = new Curr_acct(name, acc_no, acc type);
         switch(ch) {
           Case 1 : System.~~pr~~out.println("Enter amount"
                    ~~sc~~ int amount = sc.nextInt();
                    c.deposit(amount);
          case 2 : c.display();
          case 3 : System.out.println("Enter amount")
                   int amount = sc.nextInt();
                   ~~c.depos~~ c.withdrawal(amount);
        default : System.out.println("Invalid");
      }
      c.charges();
    }
```

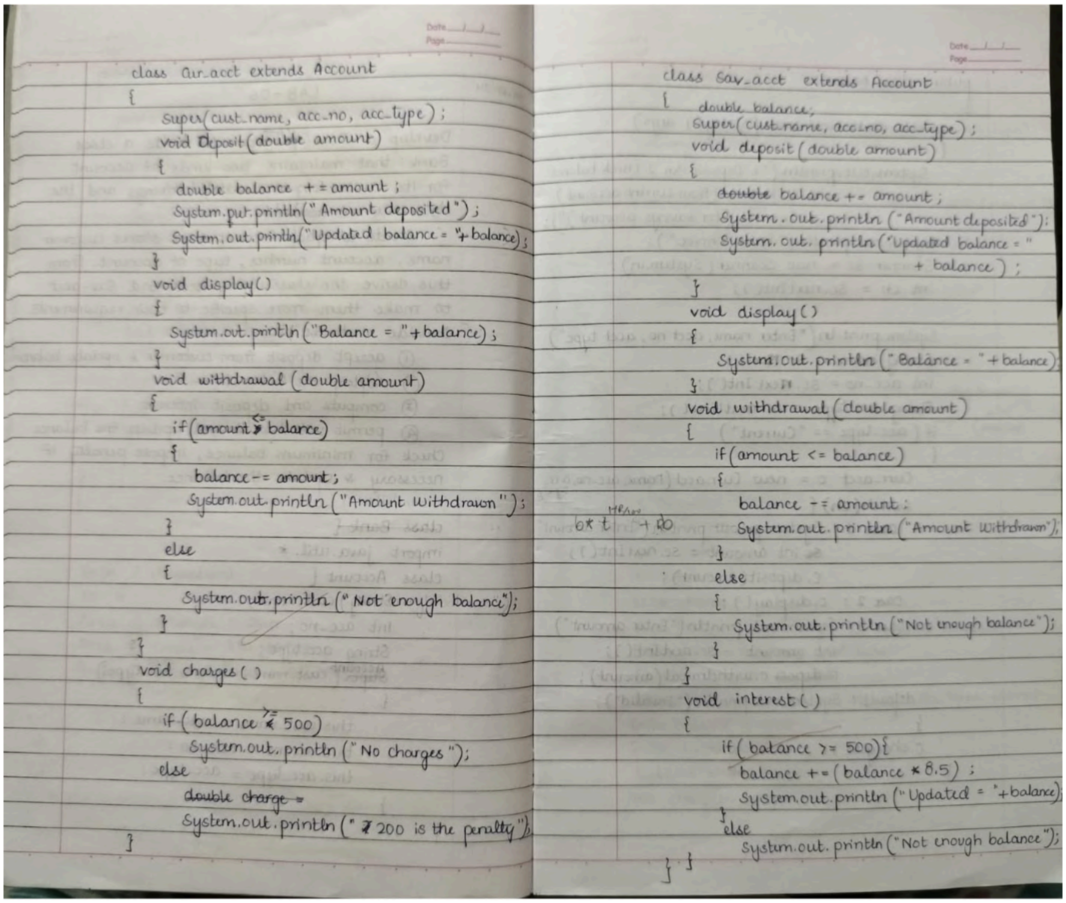
System.out.println ("Enter balance)

```
else if (acc-type == "Savings")
{
   Sav_acct s = new Sav_acct(name, accno, acc_type);
   switch(ch){
      case 1: s.deposit(amount);
      case 2: s.display(); → case 3: s.withdraw(amt);
      case 4: s.interest();
      default : System.out.println("Invalid");
   }
}
```

8 ~~9~~.10
proceed

Output:

```
1. Deposit
2. Check balance
3. Withdraw
4. Interest
Enter choice
1                          ~~Enter balance~~
Enter name, acctno, acct type
Bhoomi  12345  Current
Name: Bhoomi  Acc no: 12345  Acc type: Current
Enter balance = 1000
Enter amount: 300
Balance = ~~800~~ 1300
No charges
```



```
class Cur_acct extends Account
{
    super(cust_name, acc_no, acc_type);
    void Deposit(double amount)
    {
      double balance += amount;
      System.put.println("Amount deposited");
      System.out.println("Updated balance = "+balance);
    }
    void display()
    {
      System.out.println("Balance = "+balance);
    }
    void withdrawal (double amount)
    {
      if(amount <= balance)
      {
        balance -= amount;
        System.out.println ("Amount Withdrawn");
      }
      else
      {
        System.out.println ("Not enough balanci");
      }
   }
   void charges ( )
   {
     if (balance >= 500)
       System.out.println ("No charges");
     else
       ~~double charge =~~
       System.out.println ("₹200 is the penalty");
}
```

```
class Sav_acct extends Account
{  double balance;
   super(cust_name, acc_no, acc_type);
   void deposit (double amount)
   {
     ~~double~~ balance += amount;
     System.out.println ("Amount deposited"):
     System.out.println ("Updated balance = "
                              + balance);
   }
   void display()
   {
     System.out.println ("Balance = "+balance);
   }
   void withdrawal (double amount)
   {
     if(amount <= balance)
     {
       balance -= amount;
       System.out.println ("Amount Withdrawn");
     }
     else
     {
       System.out.println ("Not enough balance");
     }
   }
   void interest ( )
   {
     if( balance >= 500){
      balance += (balance * 8.5) ;
      System.out.println ("Updated = "+balance);
     }
     else
      System.out.println ("Not enough balance");
} }
```

b*t 1+R/100 +R0

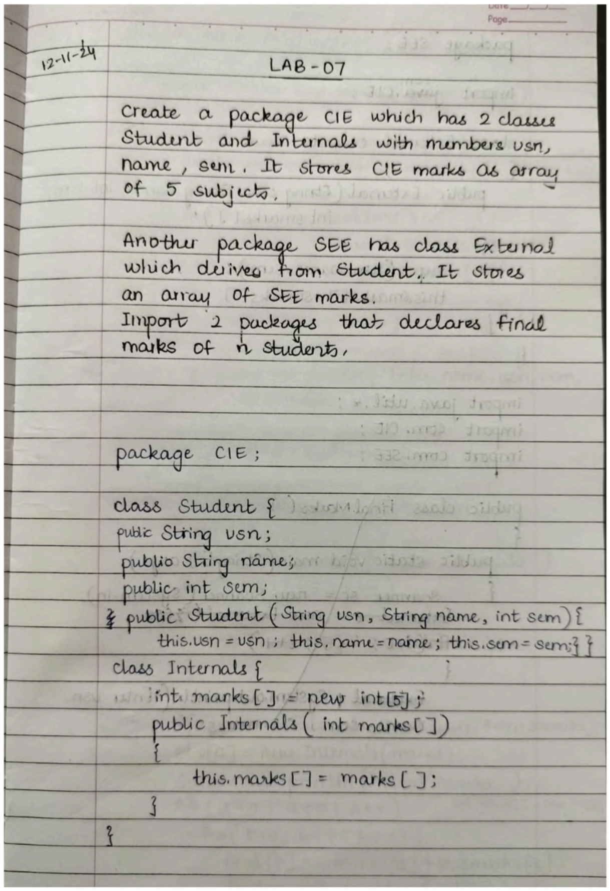
Code:

```
import java.util.Scanner;

class Account {
    String name;
    int accountNumber;
    String typeOfAccount;

    Account(String name, int accountNumber, String typeOfAccount) {
        this.name = name;
        this.accountNumber = accountNumber;
        this.typeOfAccount = typeOfAccount;
    }

    void deposit(int amount) {
    }

    void withdraw(int amount) {
    }

    void displayBalance() {
    }

    void calculateInterest() {
    }
}

class SavingsAccount extends Account {
    double interestRate;

    SavingsAccount(String name, int accountNumber, String typeOfAccount, double interestRate) {
        super(name, accountNumber, typeOfAccount);
        this.interestRate = interestRate;
    }

    @Override
    void deposit(int amount) {
        super.deposit(amount);
    }

    @Override
    void withdraw(int amount) {
        super.withdraw(amount);
    }
    @Override
    void calculateInterest() {
    }
}

class CurrentAccount extends Account {
    int minimumBalance;

    CurrentAccount(String name, int accountNumber, String typeOfAccount, int minimumBalance) {
        super(name, accountNumber, typeOfAccount);
        this.minimumBalance = minimumBalance;
    }

    @Override
    void withdraw(int amount) {
        super.withdraw(amount);
    }
}

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

        String name = scanner.nextLine();
        int accountNumber = scanner.nextInt();
        String typeOfAccount = scanner.next();

        if (typeOfAccount.equalsIgnoreCase("savings")) {
            double interestRate = scanner.nextDouble();
            SavingsAccount savingsAccount = new SavingsAccount(name, accountNumber,
typeOfAccount, interestRate);
        } else if (typeOfAccount.equalsIgnoreCase("current")) {
            int minimumBalance = scanner.nextInt();
            CurrentAccount currentAccount = new CurrentAccount(name, accountNumber,
typeOfAccount, minimumBalance);
        } else {
            System.out.println("Invalid account type.");
        }

        scanner.close();
    }
}
```

Program 6

Implement Packages

Algorithm:



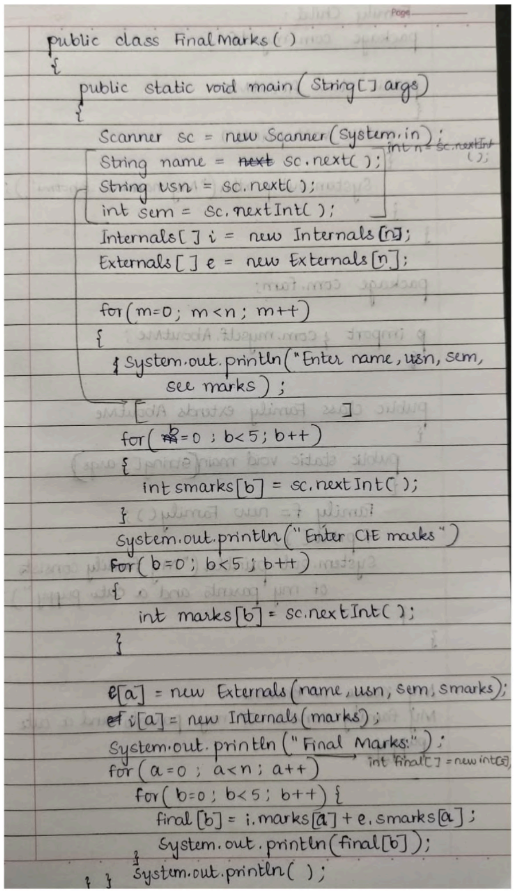
12-11-24

LAB-07

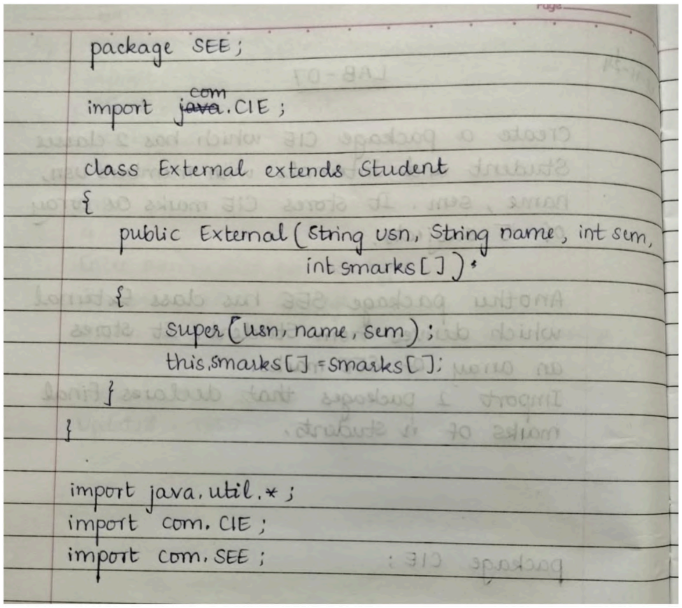
Create a package CIE which has 2 classes Student and Internals with members usn, name, sem. It stores CIE marks as array of 5 subjects.

Another package SEE has class External which derives from Student. It stores an array of SEE marks.
Import 2 packages that declares final marks of n students.

```
package CIE;

class Student {
public String usn;
publio String name;
public int sem;
} public Student (String usn, String name, int sem){
     this.usn = usn ; this.name=name ; this.sem = sem;} }
class Internals {
    int marks[] = new int[5];
    public Internals( int marks[])
    {
        this.marks[] = marks[];
    }
}
```



```
public class FinalMarks ( )
{
   public static void main(String[] args)
   {
      Scanner sc = new Scanner(System.in);   int n= sc.nextInt();
      String name = ~~next~~ sc.next( );
      String usn = sc.next();
      int sem = sc.nextInt( );
      Internals[ ] i = new Internals[n];
      Externals[ ] e = new Externals[n];

      for(m=0; m<n; m++)
      {
       { System.out.println("Enter name, usn, sem,
              see marks);
       [                    ]
        for( b=0 ; b<5; b++)
        {
          int smarks[b] = sc.nextInt( );
        }
        System.out.println("Enter CIE marks")
       For(b=0; b<5 ; b++)
       {
         int marks[b] = sc.nextInt( );
       }

       e[a] = new Externals(name, usn, sem, smarks);
       ~~ef~~ i[a] = new Internals(marks);
       System.out.println("Final Marks:");
       for(a=0 ; a<n; a++)          int final[] = new int[5];
         for(b=0; b<5; b++) {
           final[b] = i.marks[a] + e.smarks[a];
           System.out.println(final[b]);
        }
   } }  System.out.println( );
```



```
package SEE;

import ~~java~~ com.CIE ;

class External extends Student
{
   public External(String usn, String name, int sem,
                          int smarks[])
   {
       super(usn, name, sem);
       this.smarks[] = smarks[];
   }
}

import java.util.*;
import com.CIE ;
import com.SEE ;
```

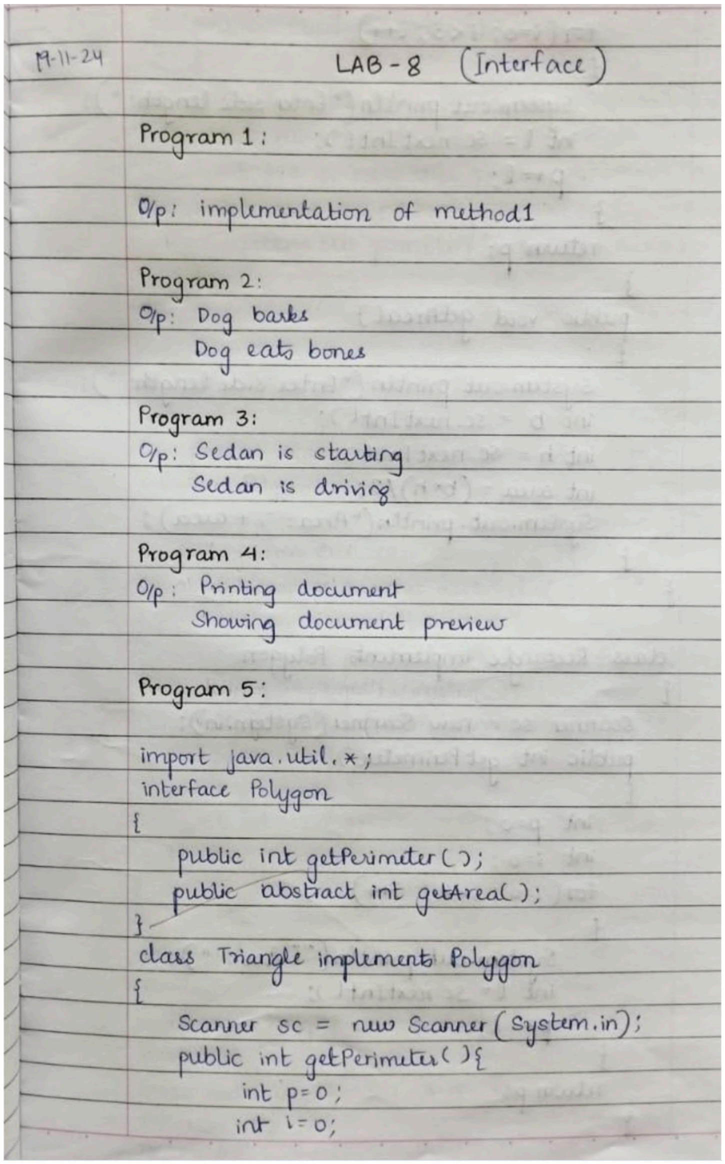
Code:

```
package CIE;

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

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

class Internals {
   int[] marks = new int[5];

   public void setMarks(int[] marks) {
      this.marks = marks;
   }
}

package SEE;

import CIE.Student;

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

   public External(String usn, String name, int sem) {
      super(usn, name, sem);
   }

   public void setMarks(int[] marks) {
      this.marks = marks;
   }
}

import CIE.*;
import SEE.*;

public class Main {
   public static void main(String[] args) {
      int n = 5;
      Student[] students = new Student[n];
      External[] externals = new External[n];

      for (int i = 0; i < n; i++) {
        students[i] = new Student("USN" + i, "Name" + i, 5);
        externals[i] = new External("USN" + i, "Name" + i, 5);

        int[] internalMarks = {80, 75, 90, 85, 95};
        int[] externalMarks = {85, 70, 95, 80, 90};

        students[i].internals.setMarks(internalMarks);
        externals[i].setMarks(externalMarks);
      }

      for (int i = 0; i < n; i++) {
        System.out.println("Student " + (i + 1));
        System.out.println("USN: " + students[i].usn);
        System.out.println("Name: " + students[i].name);
        System.out.println("Semester: " + students[i].sem);

        int[] internalMarks = students[i].internals.marks;
        int[] externalMarks = externals[i].marks;

        for (int j = 0; j < 5; j++) {
          int finalMarks = (internalMarks[j] + externalMarks[j]) / 2;
          System.out.println("Course " + (j + 1) + ": " + finalMarks);
        }

        System.out.println();
      }
   }
}
```

Program 7

Implement Interfaces

Algorithm:



19-11-24

LAB - 8 (Interface)

Program 1 :

O/p : implementation of method1

Program 2 :

```
O/p : Dog barks
      Dog eats bones
```

Program 3 :

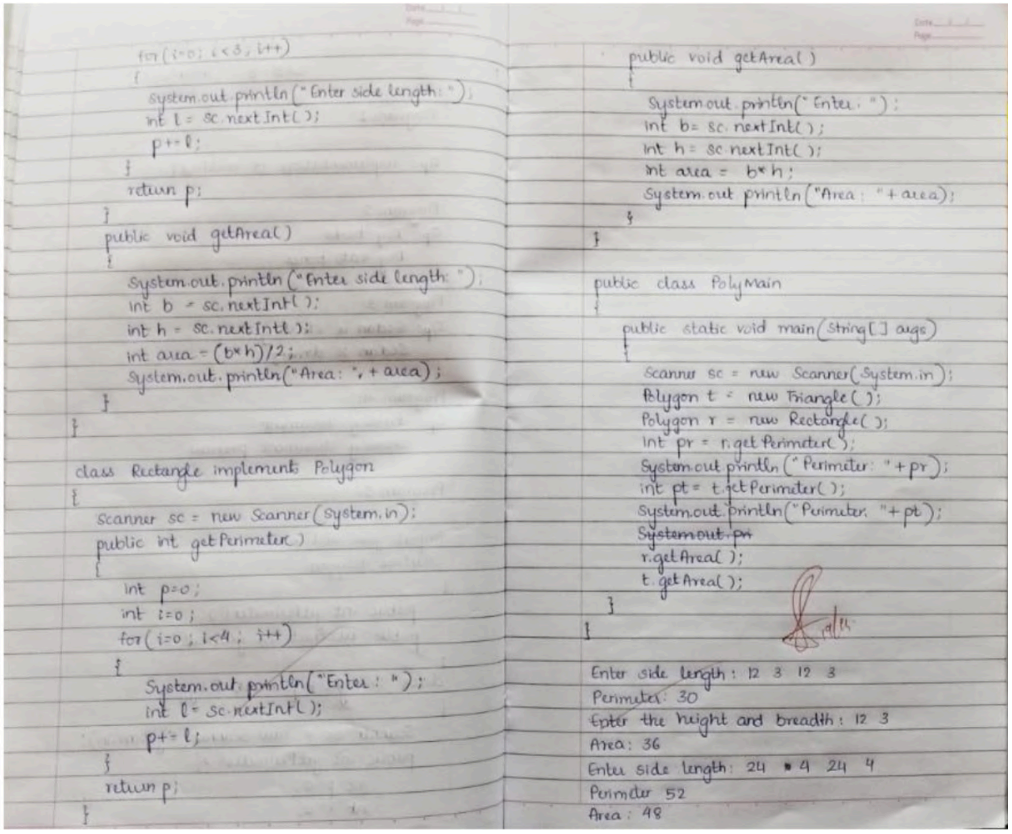
```
O/p : Sedan is starting
      Sedan is driving
```

Program 4 :

```
O/p : Printing document
      Showing document preview
```

Program 5 :

```
import java.util.*;
interface Polygon
{
    public int getPerimeter();
    public abstract int getArea();
}
class Triangle implement Polygon
{
    Scanner sc = new Scanner(System.in);
    public int getPerimeter(){
         int p=0;
         int i=0;
```



```
        for (i=0; i<3, i++)
        {
          system.out.println(" Enter side length: ");
          int l = sc.nextInt( );
          p+=l;
        }
        return p;
    }
    public void getArea( )
    {
        System.out.println("Enter side length: ");
        int b = sc.nextInt( );
        int h = sc.nextInt( );
        int area = (b*h)/2;
        System.out.println("Area: ", + area);
    }
}

class Rectangle implements Polygon
{
   Scanner sc = new Scanner(system.in);
   public int getPerimeter( )
   {
       int p=0;
       int i=0;
       for( i=0 ; i<4 ; i++)
       {
          System.out.println("Enter : ");
          int l = sc.nextInt( );
          p+= l;
       }
       return p;
   }
      public void getArea( )
      {
        System.out.println(" Enter: ");
        int b= sc.nextInt( );
        Int h = sc.nextInt( );
        int area = b*h;
        System.out.println("Area: "+ area);
      }
}

public class PolyMain
{
    public static void main(String[] args)
    {
       Scanner sc = new Scanner(System.in);
       Polygon t = new Triangle( );
       Polygon r = new Rectangle( );
       Int pr = r.getPerimeter( );
       System.out.println(" Perimeter: " + pr);
       int pt = t.getPerimeter( );
       System.out.println(" Perimeter: "+ pt);
       ~~System.out.pr~~
       r.getArea( );
       t.getArea( );
    }
}
```

```
Enter side length: 12 3 12 3
Perimeter: 30
Enter the height and breadth: 12 3
Area: 36
Enter side length: 24 • 4 24 4
Perimeter 52
Area: 48
```

Code:

```
interface Polygon {
   default double getPerimeter() {
      double perimeter = 0.0;
      for (double side : getSides()) {
         perimeter += side;
      }
      return perimeter;
   }

   abstract double getArea();

   double[] getSides();
}

class Rectangle implements Polygon {
   private double length;
   private double width;
```

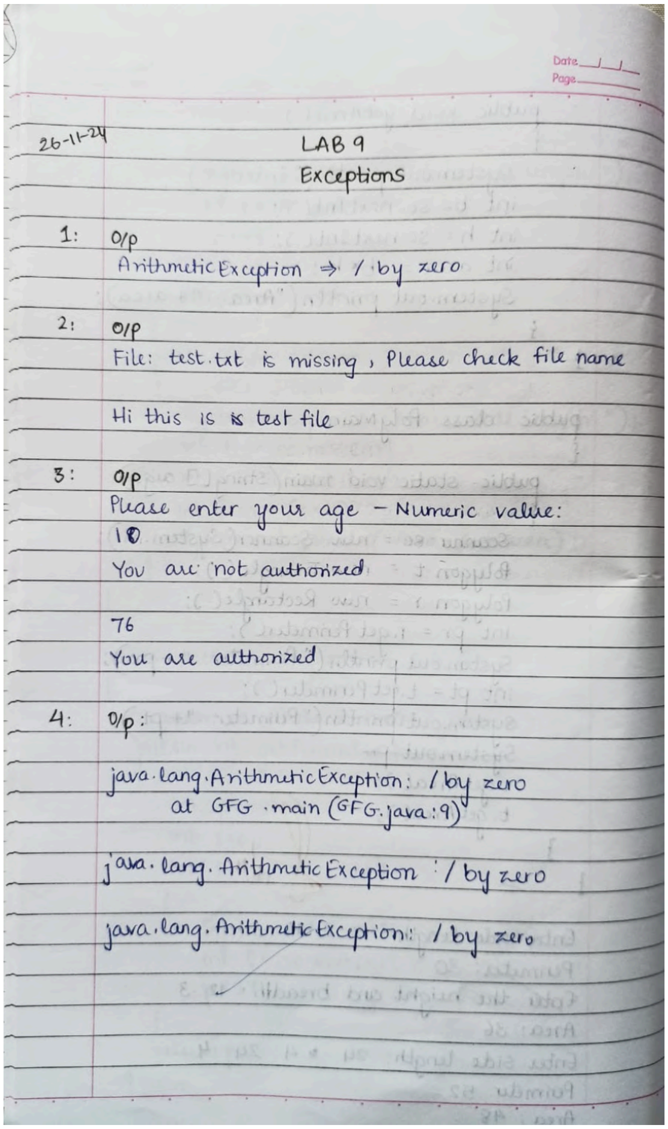
```
  public Rectangle(double length, double width) {
    this.length = length;
    this.width = width;
  }

  @Override
  public double getArea() {
    return length * width;
  }

  @Override
  public double[] getSides() {
    return new double[]{length, width, length, width};
  }
}

class Circle implements Polygon {
  private double radius;

  public Circle(double radius) {
    this.radius = radius;
  }

  @Override
  public double getArea() {
    return Math.PI * radius * radius;
  }

  @Override
  public double[] getSides() {
    return new double[]{2 * Math.PI * radius};
  }
}

public class Main {
  public static void main(String[] args) {
    Polygon rectangle = new Rectangle(5, 4);
    Polygon circle = new Circle(3);

    System.out.println("Rectangle Perimeter: " + rectangle.getPerimeter());
    System.out.println("Rectangle Area: " + rectangle.getArea());

    System.out.println("Circle Perimeter: " + circle.getPerimeter());
    System.out.println("Circle Area: " + circle.getArea());
```

```
 }
}
```

Program 8

Implement Exception

Algorithm:



Date __/__/__
Page ____

26-11-24

LAB 9

Exceptions

1: O/P

```
ArithmeticException ⇒ / by zero
```

2: O/P

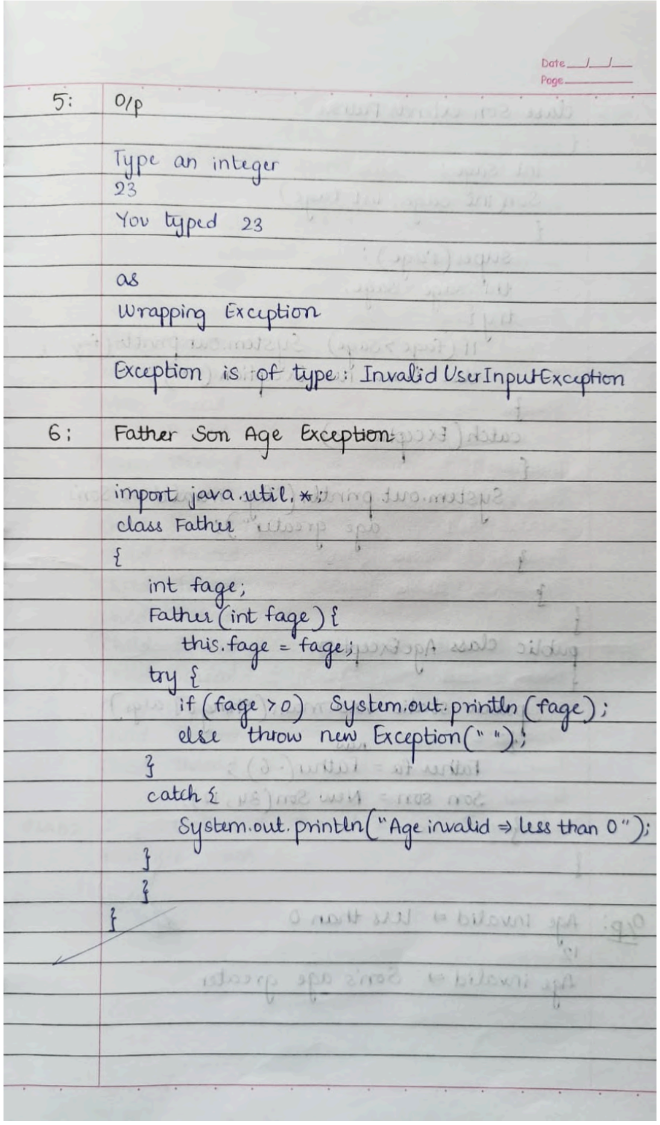
```
File: test.txt is missing , Please check file name

Hi this is ~~is~~ test file
```

3: O/P

```
Please enter your age - Numeric value:
18
You are not authorized

76
You are authorized
```

4: O/P :

```
java.lang.ArithmeticException: / by zero
        at GFG .main (GFG.java:9)

java. lang. ArithmeticException : / by zero

java.lang. ArithmeticException: / by zero
```



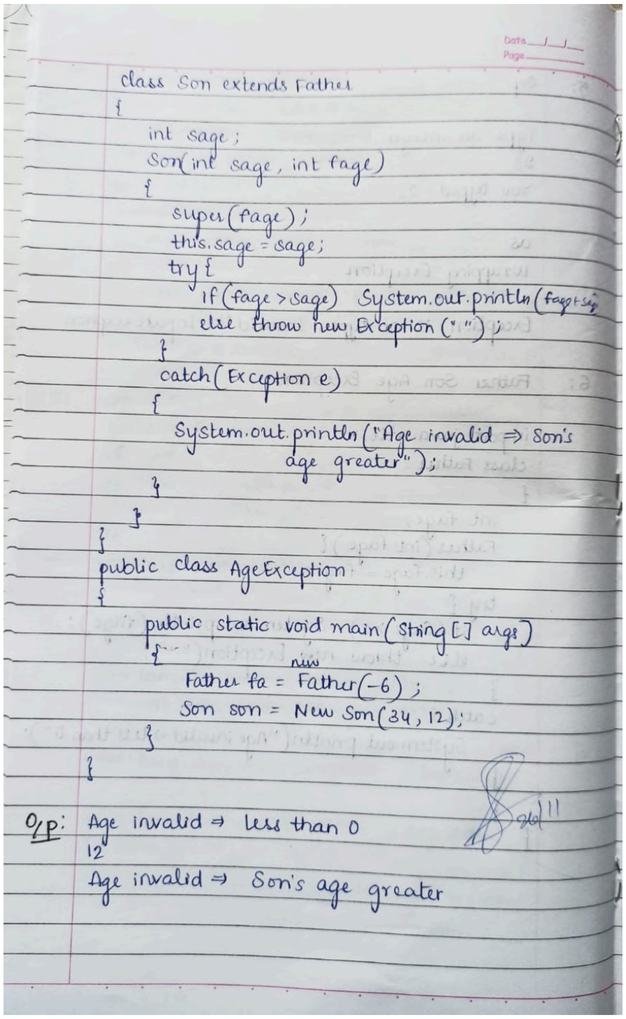
Date __/__/__
Page ____

5: O/P

```
Type an integer
23
You typed 23

as
Wrapping Exception

Exception is of type : Invalid UserInputException
```

6: Father Son Age Exception:

```
import java.util.*;
class Father
{
   int fage;
   Father(int fage){
      this.fage = fage;
   try {
     if (fage >0)  System.out.println (fage);
     else  throw new Exception(" ");
   }
   catch {
      System.out.println("Age invalid ⇒ less than 0");
   }
   }
}
```



```
class Son extends Father
{
    int sage;
    Son(int sage, int fage)
    {
      super(fage);
      this.sage = sage;
      try {
        if (fage > sage)  System.out.println(fage+sage
        else throw new Exception (" ");
     }
     catch( Exception e)
     {
       System.out.println("Age invalid ⇒ Son's
                 age greater");
     }
   }
}
public class AgeException
{
    public static void main(String [] args)
    {
       Father fa = new Father(-6) ;
       Son son = New Son(34, 12);
    }
}
```

O/P:

```
Age invalid ⇒ less than 0
12
Age invalid ⇒ Son's age greater
```

Code:

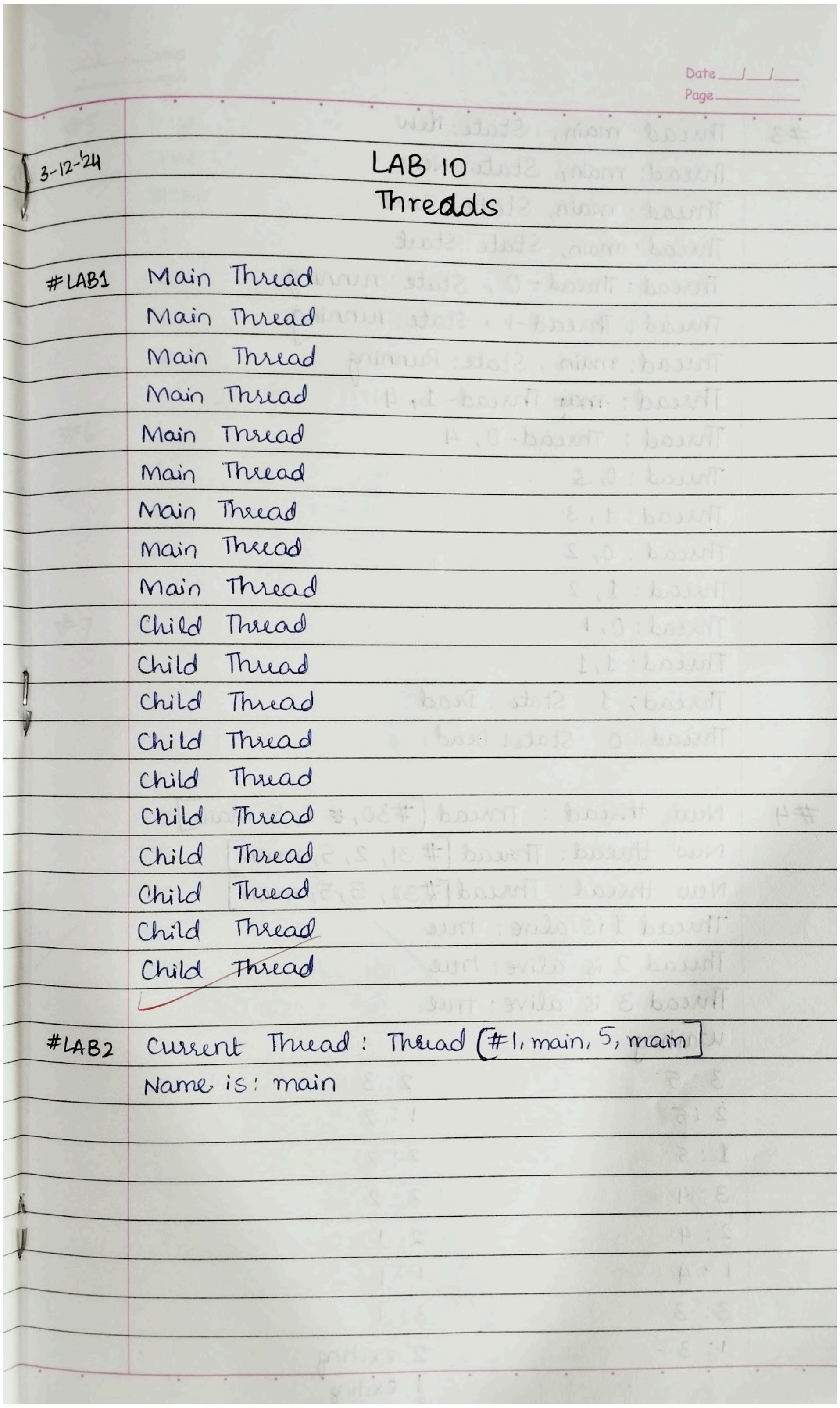
```
import java.util.Scanner;

class WrongAgeException extends Exception {
  public WrongAgeException(String message) {
    super(message);
  }
}

class Father {
  int age;

  public Father(int age) throws WrongAgeException {
    if (age < 0) {
      throw new WrongAgeException("Father's age cannot be negative.");
    }
```

```
        this.age = age;
    }
}

class Son extends Father {
    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age cannot be greater than or equal to father's
age.");
        }
    }
}

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

        try {
            System.out.print("Enter father's age: ");
            int fatherAge = scanner.nextInt();
            System.out.print("Enter son's age: ");
            int sonAge = scanner.nextInt();

            Son son = new Son(fatherAge, sonAge);
            System.out.println("Father's age: " + son.age);
            System.out.println("Son's age: " + sonAge);
        } catch (WrongAgeException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Program 9

Implement Threads

Algorithm:



3-12-'24

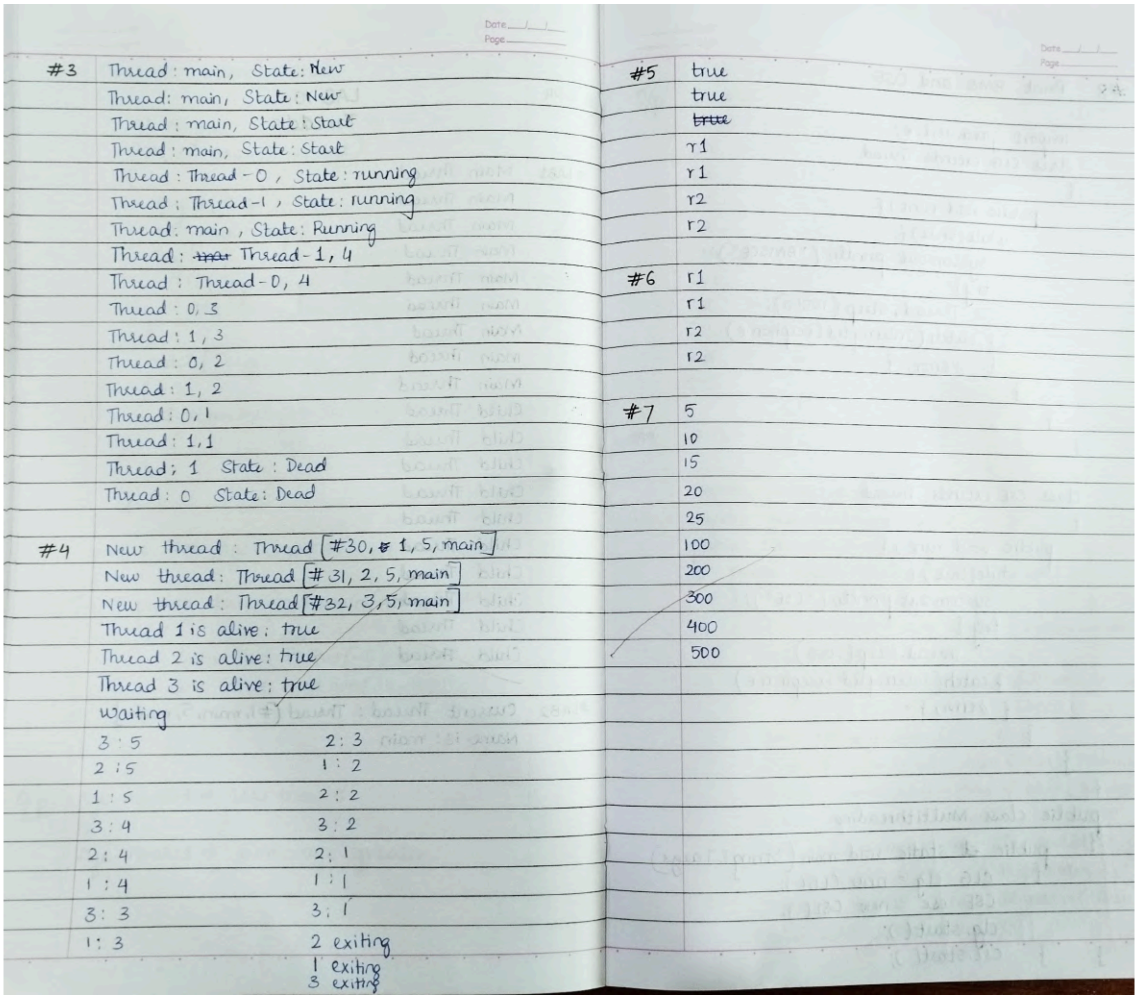
LAB 10
Threads

#LAB1

```
Main Thread
Main Thread
Main Thread
Main Thread
Main Thread
Main Thread
Main Thread
Main Thread
Main Thread
Child Thread
Child Thread
Child Thread
Child Thread
Child Thread
Child Thread
Child Thread
Child Thread
Child Thread
Child Thread
```

#LAB2

```
Current Thread : Thread[#1, main, 5, main]
Name is: main
```



#3

```
Thread : main,  State : New
Thread: main,  State : New
Thread : main, State : Start
Thread : main, State : Start
Thread : Thread - 0 , State : running
Thread : Thread-1 , State : running
Thread: main , State: Running
Thread : ~~mai~~ Thread-1, 4
Thread : Thread-0, 4
Thread : 0, 3
Thread : 1, 3
Thread : 0, 2
Thread : 1, 2
Thread : 0, 1
Thread : 1, 1
Thread; 1  State : Dead
Thread : 0  State : Dead
```

#4

```
New thread : Thread[#30, 1, 5, main]
New thread : Thread[#31, 2, 5, main]
New thread : Thread[#32, 3, 5, main]
Thread 1 is alive : true
Thread 2 is alive : true
Thread 3 is alive : true
Waiting
3 : 5                      2 : 3
2 : 5                      1 : 2
1 : 5                      2 : 2
3 : 4                      3 : 2
2 : 4                      2 : 1
1 : 4                      1 : 1
3 : 3                      3 : 1
1 : 3                      2 exiting
                           1 exiting
                           3 exiting
```

#5

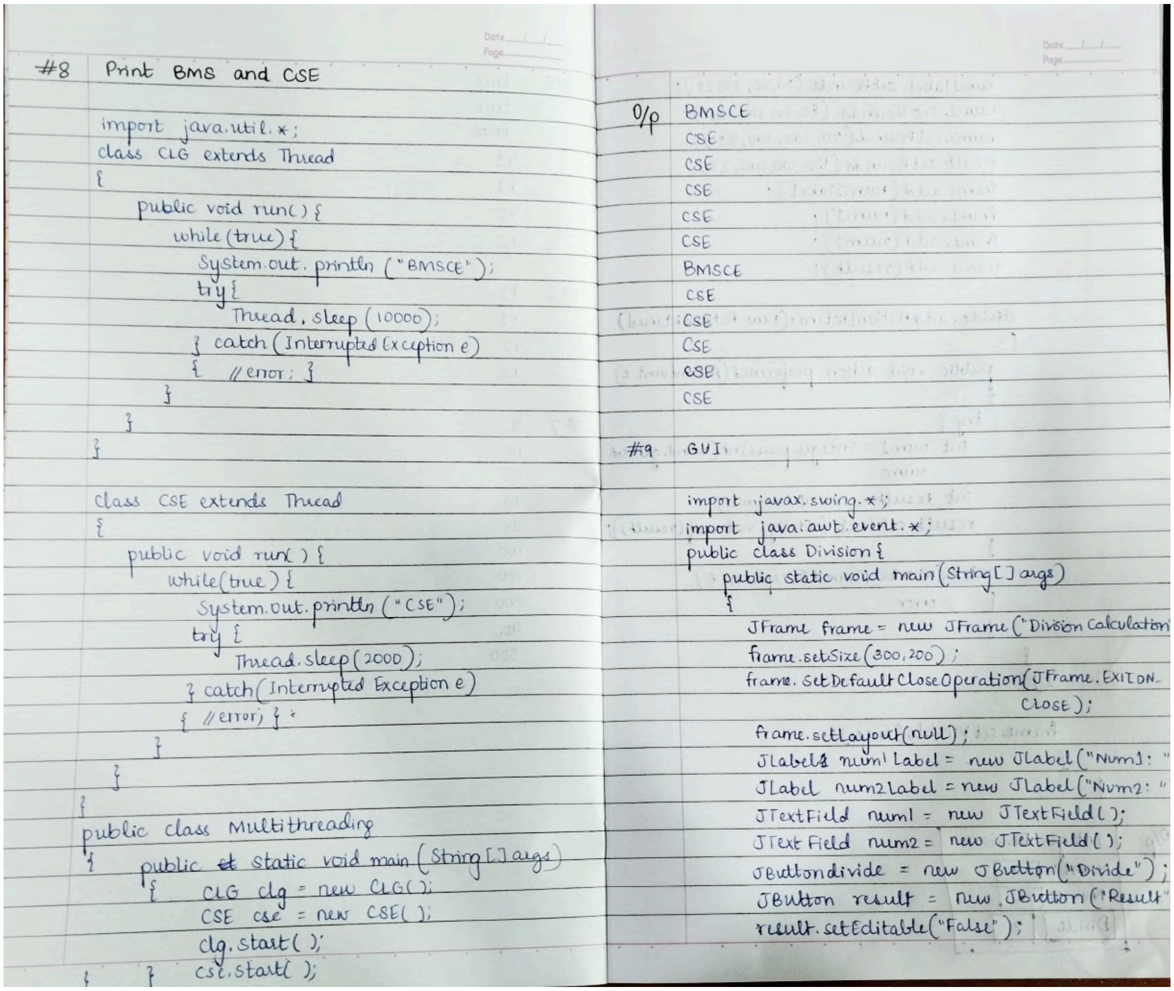
```
true
true
~~true~~
r1
r1
r2
r2
```

#6

```
r1
r1
r2
r2
```

#7

```
5
10
15
20
25
100
200
300
400
500
```



#8 Print BMS and CSE

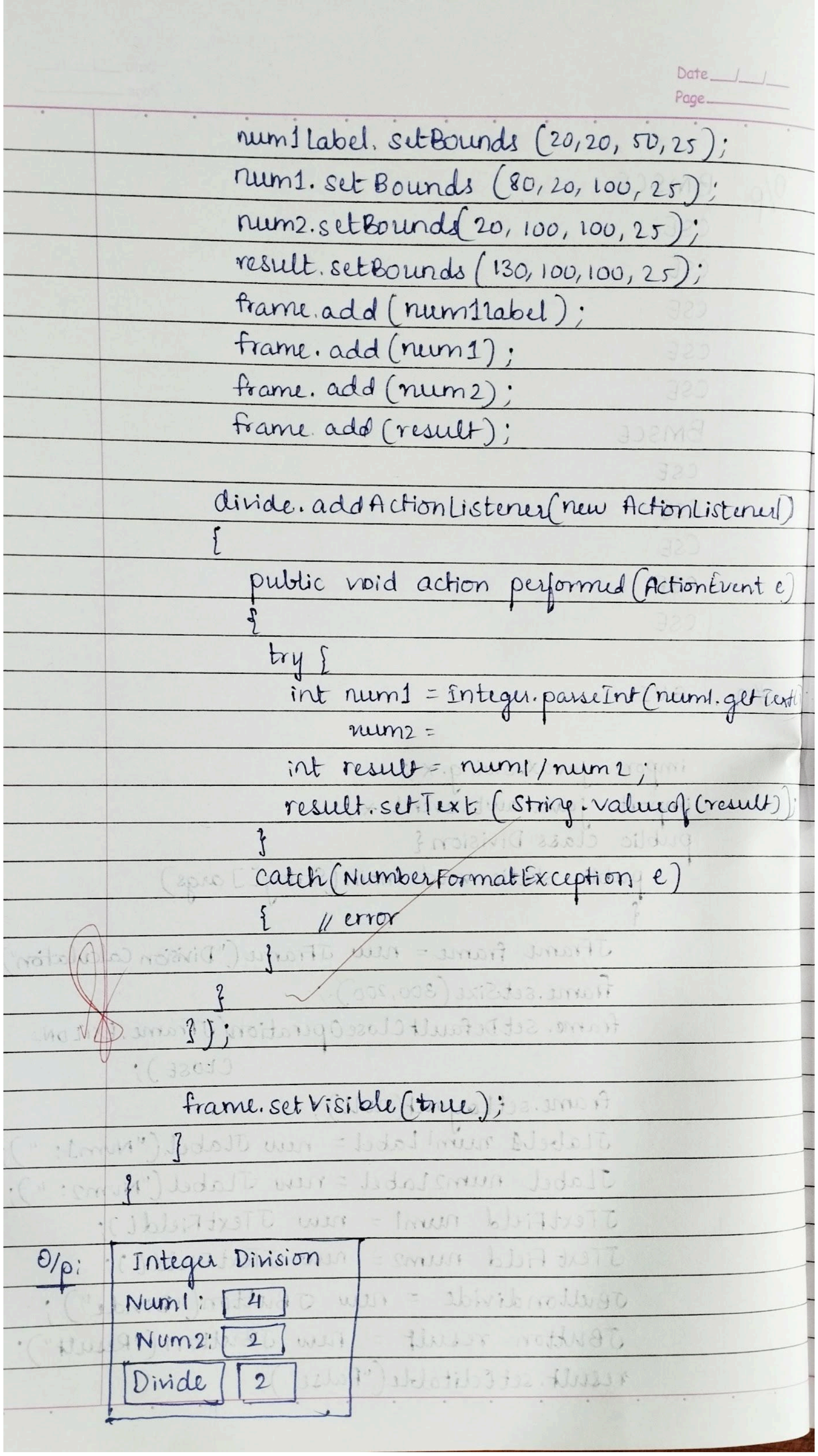
```
import java.util.*;
class CLG extends Thread
{
    public void run(){
        while(true){
            System.out.println("BMSCE");
            try{
                Thread.sleep(10000);
            } catch(InterruptedException e)
            {  //error; }
        }
    }
}

Class CSE extends Thread
{
    public void run(){
        while(true){
            System.out.println("CSE");
            try {
                Thread.sleep(2000);
            } catch(Interrupted Exception e)
            { //error; }
        }
    }
}
public class Multithreading
{   public ~~st~~ static void main(String[] args)
    {   CLG clg = new CLG();
        CSE cse = new CSE();
        clg.start();
{   }   cse.start();
```

O/p

```
BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
CSE
```

#9 GUI

```
import javax.swing.*;
import java.awt.event.*;
public class Division{
    public static void main(String[] args)
    {
        JFrame frame = new JFrame("Division Calculation
        frame.setSize(300,200);
        frame.SetDefaultCloseOperation(JFrame.EXITON_
                                          CLOSE);
        frame.setLayout(null);
        JLabel~~1~~ num1Label = new JLabel("Num1: "
        JLabel num2Label = new JLabel("Num2: "
        JTextField num1 = new JTextField();
        JText Field num2 = new JTextField();
        JButtondivide = new JButton("Divide");
        JButton result = new JButton("Result"
        result.setEditable("False");
```



Date / /
Page

```
num1Label. setBounds (20,20, 50,25);
num1. set Bounds (80, 20, 100, 25);
num2.setBounds(20, 100, 100, 25);
result. setBounds (130, 100,100, 25);
frame.add (num1label);
frame. add (num1);
frame. add (num2);
frame add (result);

divide. addActionListener(new ActionListener()
{
  public void action performed(ActionEvent e)
  {
   try {
    int num1 = Integer.parseInt(num1.getText
        num2 =
    int result = num1/num2;
    result.setText (String.valueof(result));
   }
   catch(NumberFormatException e)
   {   // error
   }
  }
 });

 frame. set Visible(true);
 }
}
```

O/p:

```
Integer Division
Num1 : [ 4 ]
Num2: [ 2 ]
[Divide] [ 2 ]
```

Code:

import java.util.*;
class CLG extends Thread
{

```
public void run()
{
while(true)
{
System.out.println("BMS College of Engineering");
try{
Thread.sleep(10000);
}catch(InterruptedException e)
{
//error;
}
}
}
}

class CSE extends Thread
{
public void run()
{
while(true)
{
System.out.println("CSE");
try{
Thread.sleep(2000);
}catch(InterruptedException e)
{
//error;
}
}
}
}

public class Multithreading
{
public static void main(String[] args)
{
CLG clg = new CLG();
CSE cse = new CSE();
clg.start();
cse.start();
}
}

import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class DivisionCalculator {
 public static void main(String[] args) {
 JFrame frame = new JFrame("Integer Division");
 frame.setSize(300, 200);
 frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 frame.setLayout(null);
 JLabel num1Label = new JLabel("Num1:");
 JLabel num2Label = new JLabel("Num2:");
 JTextField num1Field = new JTextField();
 JTextField num2Field = new JTextField();
 JButton divideButton = new JButton("Divide");
 JTextField resultField = new JTextField();
 resultField.setEditable(false);
 num1Label.setBounds(20, 20, 50, 25);
 num1Field.setBounds(80, 20, 100, 25);
num2Label.setBounds(20, 60, 50, 25);
 num2Field.setBounds(80, 60, 100, 25);
 divideButton.setBounds(20, 100, 100, 25);
 resultField.setBounds(130, 100, 100, 25);
 frame.add(num1Label);
 frame.add(num1Field);
 frame.add(num2Label);
 frame.add(num2Field);
 frame.add(divideButton);
 frame.add(resultField);
 divideButton.addActionListener(new ActionListener() {
 public void actionPerformed(ActionEvent e) {
 try {
 int num1 = Integer.parseInt(num1Field.getText());
 int num2 = Integer.parseInt(num2Field.getText());
 int result = num1 / num2;
 resultField.setText(String.valueOf(result));
 } catch (NumberFormatException ex) {
 JOptionPane.showMessageDialog(frame, "Please enter valid integers!", "Number Format Error",
JOptionPane.ERROR_MESSAGE);
 } catch (ArithmeticException ex) {
 JOptionPane.showMessageDialog(frame, "Cannot divide by zero!", "Arithmetic Error",
JOptionPane.ERROR_MESSAGE); }}
 });
 frame.setVisible(true);
 }
}
```