B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU

LAB REPORT

23CS3PCOOJ

Submitted in partial fulfilment of the requirements for Lab

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by:

PAARTH SANYAL (1BM22CS188)

Department of Computer Science and Engineering, B.M.S

College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560 019 2023-2024.

INDEX

```
import java.util.*;
class quadratic{
 double a,b,c,d;
 Quadratic(double a,doubleb, double c){
   this.a=a;
   this.b=b;
   this.c=c;
}
 void get-root c){
   if(a==0){
      system.out.println("a can't be zero");
 if(d==0){
   System.Out.println("Equal Roots");
   System.Out.println("The roots are" + (-b/2*a)));
   }
  else if (d>0){
  double r1=(-b+ math.sqrt(d))/2*a);
  double r2=(-b-math.sqrt(d))/(2*a);
  System.out.println("The root 1 and root 2 are," +r1,r2);
  }
  else{
     double r1=(-b)/(2*a);
     double r2=(-d)/(2*a);
    math.sqrt(-d)/(2*a);
 System.out.println(r1+" +i" +r2);
System.out.println(r2+"-i" +r2);
```

3
4

Roots are imaginary
Root 1 = 0.0 + i1.19895
root 2 = 0.0 - i1.19895

2 WEEK - 19/12/23

Develop Java prg to develop class with member USN, name, an array, credits and array marks. Include methods to accept and display details and a method to calculate SGPA.

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

class SGPA {
    String usn, name;
    int [] credit, marks;
}
public double calculate () {
    double tot-credits = 0.0;
    double tot-sum = 0.0;
    for (int i=0; i<8; i++) {
        tot-credits += credits [i];
        tot-sum += grade (marks [i]) * credits [i];
    }
    return tot-sum / tot-credits;
}
private double grade (int marks) {
    if (marks >= 90) return 10;
    else if (marks >= 80) return 9;
    else if (marks > 70) return 8;
    else if (marks > 60) return 7;
    else if (marks > 50) return 6;
    else if (marks > 40) return 5;
    else return 0.0;
}
```

```
public void accept (Scanner in) {
  for (int i=0; i<8; i++) {
    credits [i] = in.nextInt();
  }
  sout ("marks");
  for (int i=0; i<8; i++) {
    marks [i] = in.nextInt();
  }
}
public static void main (String args []) {
  Scanner in = new Scanner (System.in);
  SGPA nav = new SGPA
  ("CS188", Paarth Sanyal", credits,
   marks);
  nav.accept (in);
  System.out.println (Arrays.toString (nav.credits));
  System.out.println (nav.carculate ());
```

Output:

credits :	4	4	3	3	3	1	1	1
marks :	100	100	90	90	90	70	70	70

[4,4, 3,3,3, 1,1,1]

9.7

3 WEEK - 26-12-23

```
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 (){
```

```
public int getNumPages(){
    return numPages;
}
public String toString(){
String n = "\n" + "Name of Book: " + name + "\n";
String a = " Author of Book: " + author + "\n";
String p = " Price of Book: Rs" + price + "\n";
String N = " Numb of Pages : " + numPages + "\n";
return  n+a+p+N;
}
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. ner ();
   System . out. print ( " Author :");
   String . author = scanne. next();
   System. out. print  (Price : Rs");
double price = Scanner. nextDouble();
System . out. print (" Number of Pages :");
int NumPages = Scanner. nextInt();

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

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

Output:

Enter the number of books : 2
Enter details for Book 1:
Name : phy
Author : hc-verma
Price : Rs 1234
Number of pages : 345

_____ : 2
_____ rs
Author : rd-sharma
Price : Rs 3454
Number of Pages : 543

Book 1 Details:

Name of book: phy
Author of book : hc- verma
Price of Book : Rs 1234.0
Number of pages : 345

Book 2 Details:

Name of Books: ~~rdd~~
Author of Books : rd-sharma
~~Price~~ of Books : rs 3434 6.0
Number of pages : 543

```
import java.util.*;
abstract class AbsArea{
    int a,b;
  AbsArea(int x){
        a=x;
   }
  AbsArea(int x, int y){
        a=x;
        b=y;
    }
   abstract void area();
}
class rec extends AbsArea{
      rec(int a, int b){
         super(a,b);
    }
    void area(){
        System.out.println("The area of rectangle is": +a*b);
    }
}
class tri extends AbsArea{
     tri(int a, int b){
         super(a,b);
    }
    void area(){
        System.out.println("The area of triangle is:" + a(*b)/2);
    }
}
class cir extends AbsArea{
     cir(int a){
```

```
class Main {
public static void main (String args[]){
    System.out.println (" This is work of Paarthi");
    int x,y;
    Scanner n= new Scanner (System.in);

    System.out.println ("Give input for rectangle");
    x= n.nextInt();
    y= n.nextInt()
    if (x<0||y<0){
        System.out.println (" Invalid input for rectangle. Please enter positive values");
        System.exit(1);
    }

    AbsArea r= new rec(x,y);
    r.area();

    System.out.println ("Give input for triangle");
    x= n.nextInt();
    y= n.nextInt();
    if (x<0||y<0){
        System.out.println ("Invalid input. please enter positive values")
        System.exit(9);
    }
AbsArea t= new tri (x,y);
t.Area()

System.out.println ("Give input for circle");
    x= n.nextInt();
    if (x<0){
        System.out.println( " Invalid input. enter positive value.");
        System.exit(1);
}
```

Output:

```
This is work of Paarth
Give input for rectangle
1 2
The area of rectangle is: 2
Give input for triangle
5 6
The area of triangle is: 15
Give input for circle
-6
Invalid input for circle. Please enter positive value.
```

5 WEEK / 9-01-24

- Class Bank
{
public static void main (String args [])
{
Scanner sc = new scanner (System.in)
int choice;
Savings savings Acc = new Savings Account (1001, "Ram", 5000)
Current current Acc = new Current Account (1004, "krishna", 8000)
do
{
System.out.println ("1. Deposit to Savings Account");
System.out.println ("2 Deposit to Current Account");
System.out.println("3. Withdraw from Savings");
System.out.println("4. Withdraw from Curret");
System.out.println("5. Display Savings account balance");
System.out.println("6. Display Current Accout balance");
System.out.println("7. Exit");
System.out.println("Enter your choice")
choice = sc.nextInt();

Switch (choice){

case1:
System.out.println("Enter amout to deposit")
double depositAmountSavings = sc.nextDouble();
SavingsAcc.deposit(depositAmountSavings);
break;

case2.
System.out.println("Enter amout to deposit")
double depositAmountCurrent = sc.nextDouble()
currentAcc.deposit(depositAmountCurrent);
break;

```
cla
P       case 3:
            System.out.println("Enter the amount to widraw")
            double withdraw Amount savings = sc.nextDouble();
            break;

        case 4:
            System.out.println("Enter amount to withdraw")
            double withdraw Amount current = sc.next double;
            current Acc withdraw ( withdraw Amount current);
            break;

        case 5;
            Savings Acc. display()
            break;

        case 6:
            Current Acc. display()
            break;

        case 7;
            Savings Acc. calculate Interest()
            break;

        case 8;
            System.out. println("Exiting...");
            break;

    default;
        System.out. println("Invalid choice. Please enter an option");
        }
    } while(choice != 8)
```

```
class Account
  {
   int Acc no;
   string accholder;
   double balance;
  Account(int accNo, String accHolder, double balance)
   {
    this.acNo = acNo;
    this.balance = accbalance;
    this.accholder = accholder;
   }
  void deposit (double amount)
   {
    balance += amount;
   System.out.println("Deposit of" + amount + "successfull");
  }
void display()
  {
  System.out.println("Acc no." + accno);
  System.out.println("Acc holder" + accholder);
  System.out.println("Acc balance" + balance);
}

void withdraw(double amount)
  {
   if (balance >= amount)
   {
```

```
    else
    System.out.println("Insufficient balance");
    }
}
class Saving Account extends Account
    {
    Saving Account(int accNo, string accholdr, double balance)
    {
    super(accNo, accHoldr, balance);
    }
  vad calculate Interest()
    {
    double interest = balance*0.05;
    deposit(interest);
    System.out.println("Interest added: ₹" + interest);
    }
}
class current Account extends Account{
   private double minimal balance = 1000;
   private double service charge = 50;
current Account (int accNo, string accholdr, double balance)
    {
     super(accNo, accholdr, balance);
    }
   void withdraw (double amount)
    {
```

```
System.out.println("Withdraw of ₹" + amount + "Succesful");
}
else
{
System.out.println("Depost not balance. Service charge of ₹: + Service charge +
                   "Imposed");
        balance -= Service Charge;
}
}
}
```

Output:

Menu

Enter your choice: 1

Enter amount to deposit in saving accant: 1000

Deposit of ₹ 1000.0 succesful

5

Enter your choice: 5

Accant Number: 1001

Accant Holder: Ram

Balance: ₹ 6000.0

3

Accant no. to withdraw from saving: 6001

Insufficient balance.

1

Enter amount to deposit in Current amount: 0001

Insufficient balance. Service charge of ₹ 50 is imposed

Bank balance: ₹ -41.0

6 WEEK | 16-01-2024

- Demonstrate various string constructor with proper program

```
class X
  {
  public static void main(String args[])
    {
    byte ascii={ 65,66,67,68,69,70};
    String s1 = new String(ascii);
    System.out.println(s1);
    String s2 = new String(ascii,2,3);
    System.out.println(s2);
  }
}
```

Output:
ABCDEF

- Demonstrate str length, string literal, string concatenate

```
char chars[] = {'a','b','c'};
String s = new String(chars); String arg = "9"
System.out.println(s.length());
System.out.println("abc".length());
System.out.println("He is " + age + " years old.")
```

Output = 3
Output = 3
Output = He is 9 years old.

```
int a=10;
```

7 Week/30-01-24

```
import java.util.*;
class WrongAge extends Exception{
    WrongAge (String message){
        super(message);
    }
}
class Father {
    int age;
    Father (int age) throws WrongAge {
        if (age <0) {
            throw new WrongAge("Age cannot be negative");
        }
        this.age = age;
    }
}
class Son extends Father {
    int sage;
    Son (int FatherAge, int sonAge) throws WrongAge {
        super(FatherAge);
        if (sonAge >= FatherAge) {
            throw new WrongAge("Son's age should be less than Da
        }
        this.sage = sonAge;
    }
}
public class Error {
    public static void main(String args[]){
    Scanner sc = new Scanner(System.in);
        try {
    System.out.println("Enter dads age:");
        int a = sc.nextInt();
```

```
class
pu
            Son son = new Son(a, b);
        }
        catch (WrongAge e) {
            System.out.println("Exception:" + e.getMessage());
        }
    }
}
```

Output:

Enter Dad's age: 40

Enter son's age: 50

Exception: Son's age should be less than father.

Lab 8 | 6-02-2024

Threads

```
class displayThreads extends Thread {
    private String message;
    private int interval;
    private boolean running = true;
    public DisplayThread(String message, int interval){
        this.message = message;
        this.interval = interval;
    }
    public void run(){
        while (running){
            System.out.println(message);
            try{
            Thread.sleep(interval);
            } catch (InterruptedException e){
                e.printStackTrace();
            }
        }
    }
    public void stopThread(){
        running = false;
    }
}
public class Thread6{
    public static void main(String[] args){
        DisplayThread bmsThread = new DisplayThread("BMS CG", 10000);
        DisplayThread cseThread = new DisplayThread("CSE", 2000);
    bmsThread.start();
    cseThread.start();
```

```
        System.in.read();
    }
    catch (Exception e){
        e.printStackTrace();
    }
    bms Thread.stop Thread();
    csc Thread.stop Thread();
  }
}
```

Output:

Press ctrl to stop

BMS College of Engineering

CSE
CSE
CSE
CSE

BMS college of Engineering

CSE
CSE
CSE

Lab 9 | 20-02-24

IPC

```
class Q {
int n;
boolean valueSet = false;
synchronized int get(){
while(!valueSet)
try{
wait();
} catch (InterruptedException e){
System.out.println("InterruptedException caught");
}
System.out.println("Got:" +n);
valueSet = false;
notify();
return n;
}
synchronized void put(int n){
while(valueSet)
try {
wait();
} catch (InterruptedException e){
System.out.println("InterruptedException caught");
}
this.n = n;
```

```
class pu   class producer implem
           Q q;
           Produces (Q q){
           this.q = q;
           new Thread (this, "Producer").start();
           }
           public void run(){
           int i = 0;
           while (i<15){
           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<15){
           int r = q.get();
           i++;
           }
           }
           }
```

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

```
Press Control-C to stop.
Put: 0
Got: 0
Put: 1
Got: 1
Put: 2
Got: 2
Put: 3
Got: 3
Put: 4
Put: 4
Put: 5
Got 5
Put: 6
Got: 6
Put: 7
Got: 7
```

LAB 1

Develop a Java program that prints all real solutions to the quadratic equation ax2 +bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2 -4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;


class Quadratic{
        int a, b, c;
        double r1, r2, d;
        void getd(){
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter coeffs a, b, c: ");
                a = sc.nextInt(); b = sc.nextInt(); c = sc.nextInt();


        }


        void compute(){
                while(a == 0){
                        System.out.println("Invalid coeff, enter new one: ");
                        Scanner sc = new Scanner(System.in);
                        a = sc.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("r1 = r2 = " + r1);
                }
                else if (d > 0){
                        r1 = (-b + Math.sqrt(d))/(2*a);
                        r2 = (-b - Math.sqrt(d))/(2*a);
```

```
                System.out.println("Roots are real and distinct");
                System.out.println("r1 = " + r1 + "; " + "r2 = " + r2);
        }
        else if (d < 0){
                r1 = -b/(2*a);
                r2 = Math.sqrt(-d)/(2*a);
                System.out.println("Roots are imaginary");
                System.out.println("r1 = " + r1 + " + i" + r2 + "r2 = " + 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 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[];
        String name, usn;
        double sgpa;
        Scanner sc;
        int n;

        Student(){
                subject = new Subject[10];
                sc = new Scanner(System.in);
                System.out.println("Enter no.  of subjects: ");
                n = sc.nextInt();
                for(int i = 0; i < 9; i++){
                        subject[i] = new Subject();
                }
                sc.nextLine();

        }

        void getStudentDetails(){

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

```
            name = sc.nextLine();
            System.out.println("Enter usn: ");
            usn = sc.nextLine();
        }

        void getMarks(){
            System.out.println("\n");
            for(int i = 0; i < n; i++){
                System.out.println("Enter no. of credits: ");
                subject[i].credits = sc.nextInt();
                System.out.println("Enter marks obtained: ");
                subject[i].SubjectMarks = sc.nextInt();
                System.out.println("\n");
                if (subject[i].SubjectMarks > 100) subject[i].SubjectMarks = 100;
                else if (subject[i].SubjectMarks < 40) subject[i].SubjectMarks = 0;
                subject[i].grade = (subject[i].SubjectMarks / 10) + 1;
                if (subject[i].grade == 11) subject[i].grade = 10;
                if (subject[i].SubjectMarks >= 40 && subject[i].SubjectMarks < 50)
subject[i].grade = 4;
                else if (subject[i].SubjectMarks >= 50 && subject[i].SubjectMarks < 55)
subject[i].grade = 5;
                else if (subject[i].SubjectMarks >= 55 && subject[i].SubjectMarks < 60)
subject[i].grade = 6;

            }
        }

        double computeSGPA(){
            int effective = 0, credits = 0;
            for(int i = 0; i < n; i++){
                effective += (subject[i].grade * subject[i].credits);
                credits += subject[i].credits;
```

```
                }

                sgpa = effective/credits;
                return sgpa;

        }
}

class StudentMain{
        public static void main(String args[]){
                Student student = new Student();
                System.out.println("Pranav Y - 1BM22CS204");
                student.getStudentDetails();
                student.getMarks();
                System.out.println("Name of student is: " + student.name);
                System.out.println("USN of student is: " + student.usn);
                System.out.println("SGPA of student is: " + student.computeSGPA());
        }

}
```

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, no_pages;

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

	public String toString(){
		System.out.println("Name: " + this.name);
		System.out.println("Author: " + this.author);
		System.out.println("Price: " + this.price);
		System.out.println("Pages: " + this.no_pages);
		return this.name + this.author + this.price + this.no_pages;
	}
}


class BookMain{
	public static void main(String args[]){
		System.out.println("Roshni P - 1BM22CS223");
		Book books[] = new Book[10];
		Scanner sc = new Scanner(System.in);
```

```
        System.out.println("Enter no. of book objects: ");
        int n = sc.nextInt();
        sc.nextLine();
        for(int i = 0; i < n; i++){
            String name, author;
            int price, no_pages;
            System.out.println("Enter name: ");
            name = sc.next();
            System.out.println("Enter author: ");
            author = sc.next();
            System.out.println("Enter price: ");
            price = sc.nextInt();
            System.out.println("Enter no. of pages: ");
            no_pages = sc.nextInt();
            books[i] = new Book(name, author, price, no_pages);
        }
        System.out.println("\n");
        for(int i = 0; i < n; i++){
            System.out.println("Book " + (i+1) + " Details:\n");
            books[i].toString();
            System.out.println("\n");
        }

    }
}
```

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;


class InputScanner{
	int d1, d2;
	Scanner sc = new Scanner(System.in);
	InputScanner(){
		if(this.getClass() == Circle.class){
			System.out.println("Enter d1: ");
			d1 = sc.nextInt();
		}
		else{
			System.out.println("Enter d1 and d2: ");
			d1 = sc.nextInt();
			d2 = sc.nextInt();
		}
	}
}


abstract class Shape extends InputScanner{
	abstract void printArea();
}


class Triangle extends Shape{
	void printArea(){
		System.out.println("Area of triangle is: " + (double)(d1*d2)/2);
	}
}
```

```
class Rectangle extends Shape{
	void printArea(){
		System.out.println("Area of rectangle is: " + (double)(d1*d2));
	}
}


class Circle extends Shape{
	void printArea(){
		System.out.println("Area of circle: " + (double)(3.14*d1*d1));
	}
}

class AreaMain{
	public static void main(String args[]){
		System.out.println("Roshni P - 1BM22CS223");
		Rectangle r = new Rectangle();
		Triangle tr = new Triangle();
		Circle c = new Circle();
		r.printArea();
		tr.printArea();
		c.printArea();
	}
}
```

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. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks: a) Accept deposit from customer and update the balance. b) Display the balance. c) Compute and deposit interest d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;

import java.lang.*;


class Account {

        String name;

        int acc_no;

        boolean current;

        double balance = 0;

        int min_balance = 100;

        Scanner sc = new Scanner(System.in);


        Account(){

                if(this.getClass() == CurrentAcc.class){

                        current = true;

                } else {

                        current = false;

                }

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

                name = sc.next();

                System.out.print("Enter account no.: ");

                acc_no = sc.nextInt();

        }
```

```
	void deposit() {
		System.out.print("Enter deposit amount: ");
		balance += sc.nextDouble();
	}

	void withdraw() {
		System.out.print("Enter withdraw amount: ");
		double withdraw = sc.nextDouble();
		while (withdraw > balance) {
			System.out.print("Withdraw amount greater than balance, enter new
amount: ");
			withdraw = sc.nextDouble();
		}
		balance -= withdraw;
		if (current && balance < min_balance) {
			System.out.println("Below min balance of 100, removing remaining money
in account");
			balance = 0;
		}
	}

	void withdraw(double withdraw) {
		if (withdraw > balance) {
			System.out.println("Withdraw amount greater than balance");
		}
		if (current && balance < min_balance) {
			System.out.println("Below min balance of 100, removing remaining money
in account");
			balance = 0;
		}
	}
```

```
        void showBalance() {
                System.out.print("balance = " + balance);
        }
}

class CurrentAcc extends Account {
        void cheque(){
                System.out.print("Enter cheque amount: ");
                double cheque = sc.nextDouble();
                withdraw(cheque);
                System.out.println("Cheque created...");

        }
}

class SavingsAcc extends Account {
        void compound(int t, int r) {
                balance = balance * (Math.pow((1 + ((double) r / 100)), t));
                System.out.print("Balance after given rate and time = " + balance);
        }
}

class Bank {
        public static void main(String args[]) {
                SavingsAcc john = new SavingsAcc();
                CurrentAcc smith = new CurrentAcc();
                Account ref = null;
                Scanner sc = new Scanner(System.in);
                int acc, choice;
                System.out.println("------MENU------\n");
                System.out.println(
```

```
                    "1.Deposit\n2.Withdraw\n3.Compute intrest for Savings
Acc\n4.Display account details\n5. Create cheque\n6.Exit\nChoice:");
            choice = sc.nextInt();
            System.out.println("Enter account no.: ");
            acc = sc.nextInt();
            if (acc == 1) {
                ref = john;
            } else {
                ref = smith;
            }
            while (choice != 6) {
                if (choice == 1) {
                    ref.deposit();
                } else if (choice == 2) {
                    ref.withdraw();
                } else if (choice == 3) {
                    if (acc == 1) {
                        john.compound(1, 5);
                    } else {
                        System.out.println("Not a savings account");
                    }
                } else if (choice == 4) {
                    ref.showBalance();
                } else if (choice == 5) {
                    if (acc == 2) {
                        smith.cheque();
                    } else {
                        System.out.println("Not a current account");
                    }
                }
                System.out.println("Enter account no.: ");
```

```
                    acc = sc.nextInt();
                    System.out.println("------MENU------\n");
                    System.out.println(
                                        "1.Deposit\n2.Withdraw\n3.Compute intrest for Savings
Acc\n4.Display account details\n5. Create cheque\n6.Exit\nChoice:");
                    choice = sc.nextInt();
                }

        }
}
```

Lab 6

Demonstrate the utilization of String and StringBuffer functions as well as the usage of abstract classes

```
abstract class Bird{
   abstract void fly();
   abstract void makeSound();
}
class Eagle extends Bird{
   void fly(){
      System.out.println("Eagle fly method");
   }
   void makeSound() {
      System.out.println("Eagle sound method");
   }
}
class Hawk extends Bird{
   void fly() {
      System.out.println("Hawk fly method");
   }
   void makeSound() {
      System.out.println("Hawk sound method");
   }
}
class BirdMain {
   public static void main(String[] args) {
      Eagle e = new Eagle(); Hawk h = new Hawk();
      e.fly();h.fly();e.makeSound();h.makeSound();
   }
}


import java.util.Scanner;
```

```
import java.lang.Math;

class InputScanner{
        int d1, d2, d3;
        Scanner sc = new Scanner(System.in);
        InputScanner(){
                if(this.getClass() == Circle.class){
                        System.out.println("Enter d1: ");
                        d1 = sc.nextInt();
                }
                else{
                        System.out.println("Enter a, b, c: ");
                        d1 = sc.nextInt();
                        d2 = sc.nextInt();
      d3 = sc.nextInt();
                }
        }
}

abstract class Shape extends InputScanner{
        abstract void calculateArea();
  abstract void calculatePerimeter();
}

class Triangle extends Shape{
        void calculateArea(){
    double s = (d1+d2+d3)/2;
                System.out.println("Area of triangle is: " + (double)Math.sqrt(s*(s-d1)*(s-d2)*(s-d3)));
        }
  void calculatePerimeter(){
    System.out.println("Perimeter of triangle is: " + (double)(d1+d2+d3));
```

```
    }
}

class Circle extends Shape{
        void calculateArea(){
                System.out.println("Area of circle: " + (double)(3.14*d1*d1));
        }
    void calculatePerimeter(){
       System.out.println("Perimeter of circle: " + (double)(3.14*2*d1));
    }
}

class ShapeMain{
        public static void main(String args[]){
                System.out.println("Pranav Y - 1BM22CS204");
                Triangle tr = new Triangle();
                Circle c = new Circle();
                tr.calculateArea(); tr.calculatePerimeter();
                c.calculateArea(); tr.calculatePerimeter();
        }
}
import java.util.*;

class StringMain {
        public static void main(String args[]) {
                /* 1 */ char arr[] = { 'B', 'M', 'S', 'C', 'E' };
                String s1 = new String(arr);
                String s2 = new String("bmsce");
                String s3 = new String(s2);
                /* 2 */ String s4 = "some";
```

```
System.out.println("String length: " + s4.length() + "\n" + "Concatenated string: " +
s4.concat(s2));

/* 3 */ int d = 55;

String sd = Integer.toString(d);

System.out.println("Converting Integer to string: " + d + " -> " + sd);

/* 4 */ char res[] = new char[20];

String str = new String("Welcome to BMSCE College");

str.getChars(10, 16, res, 0);

/* 5 */ byte byte_arr[] = s4.getBytes();

for (int i = 0; i < 4; i++) {

        System.out.print(byte_arr[i] + " ");

}

/* 6 */ System.out.println("BMSCE equals BMSCE: " + s1.equals("BMSCE"));

System.out.println("BMSCE equals some: " + s1.equals(s4));

System.out.println("BMSCE equalsIC Bmsce: " + s1.equalsIgnoreCase(s2));

/* 7 */ System.out.println(str.regionMatches(11, "BMSCE College", 0, 11));

/* 8 */ System.out.println(str.startsWith("Welcome"));

/* 9 */ System.out.println(str.endsWith("College"));

/* 10 */ String s5 = new String("BMSCE");

System.out.println("Reference equal b/w s1 and s5 (==): " + s1 == s5);

System.out.println("Value equal b/w s1 and s5 (equals()): " + s1.equals(s5));

/* 11 */ String str_arr[] = { "van", "watch", "ball", "cat", "xmas", "yatch", " zee",
"apple", "ice", "jug",

        "kite", "lift", "man", "net", "orange", "dog", "ent", "free", "gun",
"hen", "parrot", "queen", "ring",

        "star", "tree", "umbrella" };

for (int i = 0; i < str_arr.length; i++) {

        for (int j = i + 1; j < str_arr.length; j++) {

                if (str_arr[i].compareTo(str_arr[j]) > 0) {

                        String temp;

                        temp = str_arr[i];

                        str_arr[i] = str_arr[j];
```

```
                                        str_arr[j] = temp;
                                }
                        }
                }
                for (int i = 0; i < str_arr.length; i++) {
                        System.out.print(str_arr[i] + " ");
                }
                /*11*/ String num_arr[] = {"1", "4", "3", "2", "5"};
                for (int i = 0; i < num_arr.length-1; i++) {
                        for (int j = i + 1; j < num_arr.length; j++) {
                                if (num_arr[i].compareTo(num_arr[j]) > 0) {
                                        String temp;
                                        temp = num_arr[i];
                                        num_arr[i] = num_arr[j];
                                        num_arr[j] = temp;
                                }
                        }
                }
                System.out.println("\n");
                for (int i = 0; i < num_arr.length; i++) {
                        System.out.print(num_arr[i] + " ");
                }

        }
}
```

Lab 6

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

```
package see;


import cie.Student;

import java.util.Scanner;


public class Externals extends Student{

	public int marks[] = new int[5];


	public void inputMarks() {

		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 displayMarks() {

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

			System.out.println("Subject " + (i + 1) + " marks: " + marks[i]);

		}

	}

}

package cie;


import java.util.Scanner;
```

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

	public void inputMarks() {
		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 displayMarks() {
		for (int i = 0; i < 5; i++) {
			System.out.println("Subject " + (i + 1) + " marks: " + marks[i]);
		}
	}
}
import cie.Student;
import cie.Internals;
import see.Externals;
import java.util.Scanner;

class Main{
	public static void main(String args[]){
		int no = 2;
		Externals finalmarks[] = new Externals[no];
		Internals intmarks[] = new Internals[no];
		for (int i = 0; i < no; i++){
			finalmarks[i] = new Externals();
			intmarks[i] = new Internals();
			finalmarks[i].inputMarks();
```

```
                intmarks[i].inputMarks();
            }

            for(int i = 0; i < no; i++){
                System.out.println("CIE: ");
                intmarks[i].displayMarks();
                System.out.println("SEE: ");
                finalmarks[i].displayMarks();
            }
        }
}
package cie;

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

}
```

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=father's age

```
import java.util.Scanner;


class WrongAge extends RuntimeException {
public WrongAge() {
super("Age cannot be negative");
}

public WrongAge(String message) {
super(message);
}
}

class InputScanner {
protected Scanner scanner;

public InputScanner() {
scanner = new Scanner(System.in);
}

public int nextInt() {
return scanner.nextInt();
}
}

class Father extends InputScanner {
protected int fatherAge;
```

```
public Father() {
System.out.println("Enter father's age:");
fatherAge = super.nextInt();

if (fatherAge < 0) {
throw new WrongAge("Age cannot be negative");
}
}

public void display() {
System.out.println("Father's Age: " + fatherAge);
}
}

class Son extends Father {
private int sonAge;

public Son() {
super();
System.out.println("Enter son's age:");
sonAge = super.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");
}
}
public void display() {
super.display();
System.out.println("Son's Age: " + sonAge);
```

```
}
}

public class InheritanceException {
public static void main(String[] args) {
try {
Son son = new Son();
son.display();
} catch (WrongAge e) {
System.out.println("Exception: " + e.getMessage());
}
}
}
```

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.

```
import java.io.*;


class B extends Thread{
	public void run(){
		try{
			for(int i = 0; i < 3; i++){
				System.out.println("BMS");
				Thread.sleep(10000);
			}
		} catch (InterruptedException e){
			System.out.println(e);
		}

	}
}

class C extends Thread{
	public void run(){
		try{
			for(int i = 0; i < 3; i++){
				System.out.println("CSE");
				Thread.sleep(2000);
			}
		} catch (InterruptedException e){
			System.out.println(e);
		}
```

```
        }
}

class ThreadMain{
        public static void main(String args[]){
                B b = new B();
                C c = new C();
                b.start();
                c.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 Arithmetic Exception 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 dividend:");


    // 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 (ArithmeticException 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 {
        wait();
      } catch (InterruptedException e) {
        System.out.println("InterruptedException caught");
      }
    System.out.println("Got: " + n);
    valueSet = false;
    notify();
    return n;
  }

  synchronized void put(int n) {
    while (valueSet)
      try {
        wait();
      } catch (InterruptedException e) {
        System.out.println("InterruptedException caught");
      }
    this.n = n;
    valueSet = true;
    System.out.println("Put: " + 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();
      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.");
    }
  }
```