INDEX

Name Ankith Gowda H.S Std 3 Sem Sec A

Roll No. 053 Subject OOJP ~~School~~/College BMSCE

~~School~~/College Tel. No. 9945235985 Parents Tel. No. 8073945430

Sl. No.	Date	Title	Page No.	Teacher Sign / Remarks
1	9/10/24	Program 1: Quadratic Equation		
2	9/10/24	Program 2: Student Marks		
3	23/10/24	Program 3: Book Demo		
4	23/10/24	Program 4: Length & Area For Shapes		
5	13/11/24	Program 5: Bank Account		
6	13/11/24	Program 6: Packages		
7	20/11/24	Program 7: Exception Handling		
8	27/11/24	Program 8: Threads		
9	27/11/24	Program 9: SwingDemo		
10	27/11/24	Program 10: Deadlock		

Week - 1

Date 09/10/24
Page

lab Program - 1

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

```
import java.util.Scanner;
 class quadratic
 {
 public static void main (String args[])
  {
double/~~float~~ a, b, c;
  Scanner s= new Scanner (System.in);
  System.out.println ("Enter a, b, c:");
a=s.b.next~~Float~~double();
b= b.next~~Float~~double();
c= b.next~~Float~~double();
  if (a==0)
   {
  System.out.println (" Not quadratic");
   }
   else
   {
double/~~float~~ d= b*b - 4*a*c;
  } if (d==0) {
double/~~float~~ r= -b/(2*a);
  System.out.println ("Equal roots. r1=r2="+r);
  }
   else if (d>0)
    {
double/~~float~~ r1= -b+(Math.sqrt(d))/(2*a);
```

Date / / /
Page

```
double r2 = (-b - (Math.sqrt(d)))/(2*a);
System.out.println("r1 = "+r1+" "+"r2 = "+r2);
}
else {
System.out.println("Roots are imaginary");
float r1 = (-b)/(2*a);
float r2 = math.sqrt(-d)/(2*a);
System.out.println("r1 = "+r1+" "+"r2 = "+r2);
}
else }
System.out.println("Arpith Gowda 1BM23CS053");
}
}
```

Output:- root 1 = ~~-0.67+1.30i~~

root 2 = ~~-0.67-1.30i~~

$$(a+3)(a-4)=0$$

$$a^2+3a-4a-12=0$$

$$a^2-a-12=0$$

Date 09/10/24
Page

Week-2 Lab Program 2

2) Develop a Java program to create a class student with members usn, name, an array credits & an array marks. Include methods to accept & display details & a method to calculate SGPA of a student.

A.
```
import java.util.Scanner;
class Student
{
 String usn;
 String name;
 int [] credits;
 int [] marks;
 void acceptDetails ()
 {
  Scanner mark = new Scanner (System.in);
  System.out.println("Enter USN:");
   usn = mark.nextLine();
  System.out.println("Enter name:");
   name = mark.nextLine();
  System.out.println("Enter number of subjects:");
  int n = mark.nextInt();
  credits = new int[n];
  marks = new int[n];
  for (int i=1; i<=n; i++)
  {
   System.out.println("Enter credit and marks:");
   credits[i] = mark.nextInt();
   marks[i] = mark.nextInt();
  }
    void sgpa cal()
    {
```

```
{
double sgpa;
int cred=0, mark=0;
for cred=0, mark=0;
for (int i=0; i<n; i++)
  {
    cred += credits[i];
    mark += marks[i];
  }
System.out.println("Total credits :"+cred);
System.out.println("Total marks: "+mark);
for (int i=0; i<n; i++)
  {
double sgpa += ((marks[i]/10.0)) * credits[i]);
  }
System.out.println("SGPA is :"+sgpa);
void displDetails()
{
System.out.println("Name:"+mark.name);
System.out.println("USN:"+mark.USN);
} for (int i=1; i<=n; i++)
  {
System.out.println("credit for"+i+credits[i]);
System.out.println("marks for"+i+marks[i]);
}
public static void main (String [] args)
  {
    Student obj = new Student ();
    obj.acceptDetails();
    obj.displDetails();
    obj.sgpaCal();
  }
}
}
```

Output: Enter name: arpith gowda
Enter usn: 1BM23CS053
Enter marks of subject:1 -
80
Enter credits for subject :1
8
Enter marks of subject : 2
90
Enter credits for subject : 2
9
Enter marks of subject : 3
70
Enter credits for subject:3
7
Enter marks of subject : 4
85
Enter credits for subject:4
8
Enter marks of subject:5
95
Enter credits for subject:5
9
Enter marks of subject:6
75
Enter credits for subject:6
7
Enter marks of subject :7
60
Enter credits for subject:7
6
Enter marks of subject:8
88
Enter credits of subject:8
8

o/p seen
sk
16/10/24

Student-1
Name : arpith gowda
USN: 1BM23CS053
SGPA : 8.0

Lab Program 2

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

```
import java.util.Scanner;
class book{
    String name, author;
    int num_pages;
    double price;

    void setDetails(){
    Scanner sc= new Scanner(System.in);
    System.out.print("Enter Name:");
    this.name = sc.next();
    System.out.print("Enter Author:");
    this.author = sc.next();
    System.out.print("Enter Pages:");
    this.num_pages= sc.nextInt();
    System.out.print("Enter Price:");
    this.price = sc.nextDouble();
    return;
    }
    void getDetails(){
    System.out.println("Name:"+name+"\n Author:"+author+
"\n Pages:"+num_pages+"\n Price:"+price);
    return;
    }
    public String toString(){
    return "Name:"+name+"\n Author:"+author+"\n Pages:"
    +num_pages+"\n Price:"+price;
    }
}
public class BookDemo{
    public static void main(String args[]){
    Scanner sc= new Scanner(System.in);
    System.out.print("Enter the number of books:");
    int bookNum = sc.nextInt();
    Book bookArray[]= new Book[bookNum];
    for(int i=0; i<bookNum; i++){
    bookArray[i] = new Book();
    bookArray[i].setDetails();
    System.out.println();
    }
    for(int i=0; i<bookNum; i++){
    bookArray[i].getDetails();
    }
    }
}
```

→ Output:-

```
Enter the number of books:3
Enter Name:aman
Enter Author:dinesh
Enter Pages:30
Enter Price:20.0
Enter Name:arpith
Enter Author:gowda
Enter Pages:40
Enter Price:20.0
Enter Name:rahul
Enter Author:shetty
Enter Pages:50
Enter Price:40.0
Name:aman
Author:dinesh
Pages:30
Price:20.0
Name:arpith
Author:gowda
Pages:40
Price:20.0
Name:rahul
Author:shetty
Pages:50
Price:40.0
```

Seen

Date 23/10/24
Page

Lab Program 4

1) 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;
abstract class Shape{
int dimension1;
int dimension2;
public Shape(){
this.dimension1=0;
this.dimension2=0;
}
public Shape(int dimension1, int dimension2)
{
this.dimension1= dimension1;
this.dimension2 = dimension2;
}
public abstract void printArea();
}
class Rectangle extends Shape{
    public Rectangle(int length, int width){
    dimension1=length;
    dimension2= width;
    }
    public void printArea(){
    int area= dimension1 * dimension2;
    S.o.pln("Area of Rectangle: "+ area);
}
}
class Triangle extends Shape{
    public Triangle(int base, int height){
        dimension1 = base;
        dimension2 = height;
    }
    public void printArea(){
    double area = 0.5 * dimension1 * dimension2;
    S.o.pln(" Area of Triangle : "+ area);
    }
}
class Circle extends Shape{
    public Circle(int radius){
        dimension1 = radius;
        dimension2=0;
    }
    public void printArea(){
    double area = Math.PI * dimension1 * dimension1;
    System.out.println("Area of Circle :"+area);
    }
}
class ShapeArea{
    p.s.v.m(String args[]){
    Scanner scan = new Scanner(System.in);
    S.o.pln("Enter length and width for Rectangle:");
    int radius = scanner.nextInt();
    Shape circle = new Circle(radius);
    circle.printArea();
    scanner.close();
    }
}
```

Date __/__/__
Page

Date / /
Page

Output :- Enter length and width for Rectangle :
20
30
Area of Rectangle : 600
Enter base and height for Triangle :
10
20
Area of Triangle : 100.0
Enter radius for Circle :
20
Area of Circle : 1256.637061

O/P seen
Sk
23/10/24

Date 13/11/24
Page

Lab Program 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 2 packages in a file that declares the final marks of n students in all five courses.

```
package com.CIE;
public class Student {
    public String usn;
                  name;
    public int sem;
public Student (String usn, String name, int sem)
{
    this.usn = usn;
    this.name = name;
    this.sem = sem
}
public void displayDetails (){
    S.O.pln ("USN:" + usn);
    S.O.pln ("Name:" + name);
    S.O.pln ("Sem:" + sem);
}
public class Internals {
public int [] internalMarks;
public Internals (int [] marks) {
```

```
if (marks.length != 5){
S.O.pln("Enter 5 marks!");
}
this.internalMarks = marks;
}
public void displayIMarks(){
S.O.pln("Internal marks:");
for(int i=0;i<5;i++)
{
S.O.pln(mark + " ");
}
S.O.pln();
}
}
```

```
package com.SEE;
import com.CIE.Student;
public class Externals extends Student{
public int[] externalMarks;
public Externals(String name, String usn, int sem,
int[] marks){
super(name, usn, sem);
if(marks.length != 5){
S.O.pln("Enter 5 subjects!");
this.externalMarks = marks;
}
public void displayMarks(){
S.O.pln("SEE Marks:");
for(int i=0; i<5; i++)
S.O.pln(marks + " ");
S.O.pln();
}
import com.CIE.*;
import com.SEE.*;
```

```
import java.util.Scanner;
public class Main {
public static void main(String[] args){
Scanner sc = new Scanner(System.in);
S.O.pln("Enter no of students:");
int n = sc.nextInt();
Externals[] students = new Externals[n];
Internals[] int Marks = new Internals[n];
for(int i=0; i<n; i++){
S.O.pln("Enter usn:");
String usn = sc.nextLine();
S.O.pln("Enter name:");
String name = sc.nextLine();
S.O.pln("Enter sem:");
int sem = sc.nextInt();
S.O.pln("Enter internals marks:");
int[] IMarks = new int[5];
for(int j=0; j<5; j++){
IMarks[j] = sc.nextInt();
}
S.O.pln("Enter external marks:");
int EMarks = new int[5];
for(int k=0; k<5; k++){
EMarks[k] = sc.nextInt();}
int Marks[i] = new Internals(IMarks);
students[i] = new External(name, usn, sem,
EMarks);
}
S.O.pln("Final marks of students");
for(int i=0; i<n; i++){
students[i].displayIMarks();
int Marks[i].displayEMarks();
S.O.pln("Final marks:");
```

```
for (j=0; j<5; j++)
  {
int final = int Marks[j]. I Marks[j]+ (students[i]
.E Marks[j];
S.O.println( final + "  ");
 }
  S.O.println("\n");
 }
}
}
```

Output: Enter usn: 1BM23CS053
Enter Name: Ankith
Enter Semester: 3
Name: Akshay
USN: 1BM23CS053
Semester: 3
Enter CIE marks of subject 1: 50
Enter CIE marks of subject 2: 49
Enter CIE marks of subject 3: 48
Enter CIE marks of subject 4: 47
Enter CIE marks of subject 5: 46
SEE marks 1: 96
SEE marks 2: 97
SEE marks 3: 98
SEE marks 4: 99
SEE marks 5: 100
Final marks in subject 1 is: 49
Final marks in subject 2 is: 48
Final marks in subject 3 is: 49
Final marks in subject 4 is: 49
Final marks in subject 5 is: 50

Date 13/11/24
Page

Lab Program 5

a) Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account & the other current account. The savings account provides compound interest & withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest.

b) Create a class Account that stores customer name, acc no. & type of account. From this derive the classes Cur-acct & Sav-acct to make them more specific to their requirements

a) Accept deposit from customer & update the balance.
b) Display the balance.
c) Compute & deposit interest
d) Permit withdrawal & update the balance
e) Check for the minimum balance, impose penalty if necessary & update the balance.

```
import java.util.Scanner;
class Account {
    String customerName;
    String accountNum;
    double balance;
    public Account(String customerName, String accountNum,
    double balance){
        this.customerName= customerName;
        this.accountNum = accountNum;
        this.balance= balance;
    }
        public void deposit (double amount){
            if (amount >0){
                balance += amount;
                S.o.println("Deposited :" + amount);
```

```
} else {
    s.o.pln("Invalid amount");
  }
 }
public void withdraw(double amount){
 if(amount>0 && amount <= balance){
  balance -= amount;
  s.o.pln("Withdrew: " +amount);
 } else {
    s.o.pln("Invalid request or insufficient funds");
  }
 }
 public void display(){
    s.o.pln(" The balance is:" + balance);
  }
 }

class SavAcc extends Account{
double interestRate;
public SavAcc(String customerName, String
accountNum, double balance, double interestRate){
super(customerName, accountNum, balance);
this.interestRate = interestRate;
 }
  public void compoundInterest(){
    double interest = balance * (interestRate/100);
deposit(interest);
  s.o.pln("Interest compounded:"+ interest);
 }
}

class CurAcc extends Account {
  static final double MIN_BAL = 1000;
  static final double SERVICE_CHARGE =100;
   }
   }
}

public class Bank{
  p.s.v.m(String[] args){
  Scanner scanner= new Scanner(System.in);
  S.o.p(" Enter name:");
  String SavName= scanner.nextLine();
  S.o.p(" Enter acc no:");
  String savAccNum= scanner.nextLine();
  S.o.p(" Enter initial balance for savings account:");
  double savBalance= scanner.nextDouble();
  S.o.p(" Enter interest rate for savings account:");
  double SavInterestRate= scanner.nextDouble();
  scanner.nextLine()
  SavAcc savings = new SavAcc(savName, savAccNum,
  savBalance, SavInterestRate);
   savings.display();
  S.o.p ("Enter the amount to deposit:");
  savings.deposit(scanner.nextDouble());
  savings.display();
  S.o.p("Enter the amount to withdraw:");
  savings.withdraw(scanner.nextDouble());
  savings.display();
  S.o.p("\n Enter customer name for current acc:");
  String currName= scanner.nextLine();
   S.o.p ("Enter account number for current acc:");
  String currAccNum= scanner.nextLine();
  S.o.p("Enter initial balance for current account:");
  double currBalance= scanner.nextDouble();
  scanner.nextLine();
   CurAcc current= new CurAcc(currName, currAccNum);
  current.display();
```

Date __/__/__
Page _____

```
S.o.p("\n Enter the amount to deposit in
current account:");
current.deposit(scanner.nextDouble());
current.display();
scanner.close();
    }
}
```

Output:- Enter customer name for savings account: Arpith
Enter acc number for savings account : 100
Enter initial balance for savings account: 10000
Enter interest rate for savings account: 2
The balance is : 30000.0
Enter the amount to withdraw from account: 10000
Withdrew : 10000.0
The balance is 20000.0
Deposited : 400.0
Interest compounded: 400.0
The balance is 20400.0
Enter name for current account: gowda
Enter Initial balance for current account: 10000
Deposited : 10000.0
The balance is : 30000.0
Enter the amount to withdraw from current account: 10000
Deposited : 10000
The balance is: 30000.0
Enter the amount to withdraw from current acc: 2000
The balance is: 28000.0

20/11/24

Date __/__/____
Page _____

Lab Program 7

2) Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called as "Son" which extends the base class. In father's class implement a constructor which takes the age and throws the exception wrongAge() when the input age is less than zero. In son's class implement a constructor that uses father and son's age and throws an exception if son's age greater than or equal to father's age.

```
import java.util.Scanner;
class WrongAgeException extends Exception{
  public WrongAgeException(String message){
    super(message);
  }
}
class SonAgeException extends Exception {
  public SonAgeException(String message){
    super(message);
  }
}
class Father {
    private int age;
    public Father(int age) throws WrongAgeException {
      if (age < 0){
        throw new WrongAgeException("Father's age cannot be
        negative or zero");
      }
      this.age = age;
    }
```

```
public int getAge(){
return age;
  }
}
class Son extends Father{
private int SonAge;
public Son(int FatherAge, int SonAge) throws
WrongAgeException, SonAgeException{
super(FatherAge);
if (SonAge >= FatherAge){
throw new SonAgeException("Son's age cannot be
greater than or equal to father's age");
  }
  this.SonAge = SonAge;
}
public int getSonAge(){
  return SonAge;
  }
}
public class FatherSon{
public static void main(String[] args){
while(true){
Scanner sc = new Scanner(System.in);
try{
    S.o.p("Enter Father's age :");
    int FatherAge = sc.nextInt();
Father father = new Father(FatherAge);
    S.o.p("Enter Son's age:");
    int SonAge = sc.nextInt();
    Son son = new Son(FatherAge, SonAge);
    S.o.pln("Accepted Successfully");
    }
    catch(WrongAgeException e){
    S.o.pln(e.getMessage());
    }
    catch(SonAgeException e){
      S.o.pln(e.getMessage());
      }
      S.o.pln("Would you like to re-enter details(Y/N)");
      String input = sc.next();
      if(input.equalsIgnoreCase("n")){
      break;
        }
      }
S.o.pln("Name: Arpith Gowda H S");
S.o.pln("USN: 1BM23CS053");
  }
}
```

Output:- Enter Father's age: -1
Father's age cannot be negative or zero
Would you like to re-enter details (Y/N): Y
Enter Father's age: 30
Enter Son's age : 35
Son's age cannot be greater than or equal to Father's age.
Would you like to re-enter details (Y/N): Y
Enter Father's age: 30
Enter Son's age: 5
Accepted Successfully
Would you like to re-enter details (Y/N): N
Name: Arpith Gowda H S
USN: 1BM23CS053

27/11/24

Date __/__/__
Page

Lab Program 8
Threads

```
class BMS extends Thread {
  public void run () {
    try {
      while (true) {
      S.o.println ("BMS college of engineering");
        Thread.sleep (10000);
      }
    } catch (Interrupted exception e) { }
  }
}

  class CSE extends Thread {
  public void run () {
    try {
      while (true) {
      S.o.println ("CSE");
        Thread.sleep (2000);
      }
    } catch (Interrupted exception e) { }
    }
  }

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

Output : BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE

27/11/24

Date __/__/__
Page ______

Lab Program 9

Swing Demo

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo {
SwingDemo(){
JFrame jfrm = new JFrame ("Divider App");
jfrm.setSize(275, 150);
jfrm.setLayout(new FlowLayout());
jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
JLabel namelabel = new JLabel("Name Arshith Gowda, USN : 1BM23CS058");
JLabel jlab = new JLabel("Enter the divider and divident:");
JTextField ajtf = new JTextField(8);
JTextField bjtf = new JTextField(8);
JButton button = new
JButton ("Calculate");
JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();
JLabel anslab = new JLabel();
jfrm.add(err);
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){
  S.o.pln ("Action event from a text field");}
  };
```

Date __/__/__
Page ______

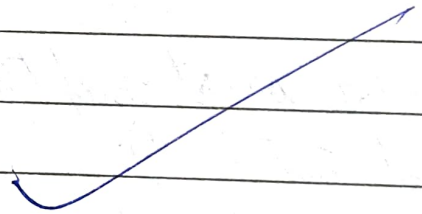
```
ajtf.addActionListener (l);
bjtf.addActionListener (l);
button.addActionListener (new ActionListener(){
public void
 actionPerformed (ActionEvent evt){ try {
int a = Integer.parseInt( ajtf.getText()); int
b = Integer.parseInt (bjtf.getText()); int ans = a/b;
alab.setText ("\n A = " + a);
blab.setText ("\n B = " + b);
  anslab.setText ("\n Ans = " + 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!"); }
   }
   public static void main (String args[]){
    SwingUtilities.invokeLater(new Runnable (){
      public void run()
      {
        new SwingDemo ();
      }
      });
    }
```

Date __/__/__
Page _____

Output : Name Arpith Gowda USN : 1BM23CS053
Enter the divisor & divident : [22] [11] [Calculate]
A=22, B=11, Result : 2

Name : Arpith Gowda USN : 1BM23CS053
Enter the divisor & dividend : [ABC] [5] [Calculate]
Error: please enter valid integers!

Name: Arpith Gowda USN : 1BM23CS053
Enter the divisor & dividend : [10] [0] [Calculate]
Error : B should be NON zero



27/11/24

Date __/__/__
Page _____

Lab Program 10
Deadlock

```
class A
{
synchronized void foo (B b)
{
String name = Thread.currentThread().getName();
SOP(name + "entered A.foo");
try {
Thread.sleep(1000);
}
Catch (Exception e)
{
SOP("A Interrupted");
}
SOP(name + "trying to call B.last()");
b.last();
}
void last()
{
SOP("Inside A.last");
}
}
class B
{
synchronized void bar(A a)
{
String name = Thread.currentThread().
getName();
SOP(name + "entered B.bar");
try
{
```

```
        thread.sleep(1000);
    }
    catch (Exception e)
    {
      sop(" B interrupted ");
    }
    sop( name + " trying to call A.last()");
     a.last();
    }
    void last()
    {
      sop (" Inside B.last ");
    }
    }
class Deadlock implements runnable
    {
      A a = new A();
      B b = new B();
     Deadlock()
     {
thread.currentThread().setName(" Main
Thread t = new Thread (this,      Thread ");
                                  " Racing Thread ");
t.start();
a.foo(b);
sop(" Back in main thread ");
}
public void run()
 {
  b.bar(a);
sop(" back in other thread ");
 }
  public static void main (String args[])
 {
sop(" Name: Ankith Gowda, USN: 1BM23CS053 ");
 new Deadlock();
 }
}
```

Output: Name: Ankith Gowda, USN: 1BM23CS053
Main Thread entered A.foo
Racing Thread entered B.bar
Main Thread trying to call B.last()
Inside B.last
Back in main thread
Racing thread trying to call A.last()
Inside A.last
Back in other thread