18/12/23

OOJ Observation Book

Lab - 1

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

Ans

```
import java.util.Scanner;
public class Q_1
{
    public static void main (String xx[])
    {
        Scanner input = new Scanner (System.in);
        System.out.println("Enter the value of a:");
        double a = input.nextDouble();
        System.out.println("Enter the value of b:");
        double b = input.nextDouble();
        System.out.println("Enter the value of c:");
        double c = input.nextDouble();
        double d = b*b - 4.0*a*c;
        if (d>0.0)
        {
            double r1 = (-b + Math.pow(d, 0.5))/(2.0*a);
            double r2 = (-b - Math.pow(d, 0.5))/(2.0*a);
            System.out.println("r1 = " + r1 + "\n r2 = " + r2);
        }
        else if (d=0)
        {
            double r = (-b) + Math.p/(2.0*a);
```

```
        System.out.println("r1 = " + r1 + "\nr2 = " + r2);
    }
    else
    {
        System.out.println("There are no real solutions");
    }
  }
}
```

Output:

1. Enter the value of a:
1
Enter the value of b:
2
Enter the value of c:
1
r1 = -1.0
r2 = -1.0

2. Enter the value of a:
1
Enter the value of b:
6
Enter the value of c:
10
There are no solutions

3. Enter the value of a:
3
Enter the value of b:

Date Page

2
Enter the value of c:
2
r1 = -0.27924
r2 = -2.38724

LAB-2

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

import java.util.Scanner;
class Student
{
    int usn;
    String name;
    int[] cred = new int[8];
    int[] marks = new int[8];
    int tcred = 0;
    double sgp = 0.0;
    void accept()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter your usn:");
        usn = s.nextInt();
        System.out.println("Enter your name:");
        name = s.next();

```
for(int i=0; i<8; i++)
{
    System.out.println("Enter number of
    credits of " + (i+1) + " subject:");
    cred[i] = s.nextInt();
    System.out.println("Enter your marks
    of " + (i+1) + " subject:");
    marks[i] = s.nextInt();
}
}
int grade(int m)
{
    if (m>=90 && m<=100)
        return 10;
    else if (m>=80 && m<90)
        return 9;
    else if (m>=70 && m<80)
        return 8;
    else if(m>=60 && m<70)
        return 7;
    else if(m>=50 && m<60)
        return 6;
    else
        return 0;
}
void SGPA()
{
    tcred = 0;
    for(int i=0; i<8; i++)
    {
        tcred += cred[i];
    }
```

```
        for(int i=0; i<8; i++)
            sgp = sgp + (cred[i] * grade(marks[i]));
        sgp = sgp/tcred;
    }
    void display()
    {
        System.out.println("Your usn : " + usn);
        System.out.println("Your name : " + name);
        System.out.println("Your sgpa : " + sgp);
    }
}
class Main
{
    public static void main(String args[])
    {
        Student st = new Student();
        st.accept();
        st.SGPA();
        st.display();
    }
}
```

Output

```
Enter your usn:
126
Enter your name:
Kedar
Enter number of credits of 1 subject:
4
Enter your marks of 1 subject:
95
:
:
```

Enter number of credits of 8 subject
1
Enter your marks of 8 subject
100

Your usn : 126
Your name : Kedar
Your sgpa : 10.0

11/1/24

Date Page

LAB - 4

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called 'Father' and derived class called 'Son' which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age < 0. In Son Class, implement a constructor that cases both father and son's age and throws an exception if son's age is >= father's age

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

```
        this.age = age;
    }
    public int getAge()
    {
        return age;
    }
}
class Son extends Father
{
    private int sonage;
    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");
        }
        this.sonage = sonAge;
    }
    public int getSonAge()
    {
        return sonAge;
    }
}
public class Main
{
    public static void main(String[] args)
    {
```

```
try
{
    Scanner s = new Scanner(System.in);
    S.O.P("Enter father's age:");
    int fatherAge = s.nextInt();
    S.O.P("Enter son's age:");
    int sonAge = s.nextInt();
    Son son = new Son(fatherAge, sonAge);
    S.O.P("Father's age:" + son.getAge());
    S.O.P("Son's age:" + son.getSonAge());
}
catch (WrongAgeException e)
{
    S.O.P("Error:" + e.getMessage());
}
catch (Exception e)
{
    S.O.P("An unexpected error
        occured:" + e.getMessage());
}
}
private static int getPositiveIntegerInput
(Scanner s)
{
while(true)
{
    try
    {
        int value = Integer.parseInt
            (Scanner.nextLine());
        if (value < 0)
        {
            S.O.P("Please enter a non-
            negative integer");
        }
```

```
        else
        {
            return value;
        }
    }
    catch (NumberFormatException e)
    {
        S.o.P("Please enter a valid
              integer!:");
    }
  }
 }
}
```

Output:

Enter Father's age: 50
Enter Son's age: 70
Error: Son's age cannot be greater than or equal to Father's age

2. 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 internals 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.

Student.java

```
package CIE
public class Student
{
    int usn;
    String name;
    int sem;
    public void accept()
    {
        Scanner s = new Scanner(System.in);
        S.O.P("Enter usn:");
        usn = s.nextInt();
        S.O.P("Enter name:");
        name = s.next;
        S.O.P("Enter sem:");
        sem = s.nextInt();
    }
    public void display()
    {
        S.O.P("Enter usn:" + usn);
        S.O.P("Name:" + name);
        S.O.P("Sem:" + sem);
    }
}
```

Internals.java

```
package CIE;
public class Internals extends Student
{
    int m[] = new int[5];
    public void accept();
    {
        super.accept();
    }
    public void display()
    {
        super.display();
        for(int i=0; i<5; i++)
        {
            S.O.P("Marks (i+1)" + m[i]);
        }
    }
}
```

Externals.java :-

```
package SEE;
import CIE.Student;

public class Externals extends Student
{
    int ms[] = new int[5];
    public void accept()
    {
        super.accept();
        S.O.P("Enter marks in 5 subjects\n");
```

```
for (int i=0; i<5; i++)
{
    ms[i] = s.nextInt();
}
}
```

Marks.java

```
import CIE.student;
import CIE.Internals;
import SEE.Externals
class Main
{
    p s v m (String xx[])
    {
        Scanner s = new Scanner(System.in);
        int n;
        S.O.P("Enter the number of students:");
        n = s.nextInt();
        Internals in[] = new Internals[n];
        Externals ext[] = new Externals[n];
        for(int i=0; i<n; i++)
        {
            in[i] = new Internals();
            ext[i] = new Externals();
            in[i].accept();
            ext[i].accept();
            in[i].display(); ext[i].display();
        }
    }
}
```

Lab - 7

5. Create a Bank Class that maintains two kinds of account for the customers.

```
class Account
{
    protected String customerName;
    protected String accountNumber;
    protected double balance;
    public Account(String CustomerName,
                   String accountNumber)
    {
        this.customerName = customerName;
        this.customerNumber = accountNumber;
        this.balance = 0;
    }
    public void deposit(double amount)
    {
        balance += amount;
        S.O.P("Deposit of $:" + amount
              + " successful");
    }
    public void displayBalance()
    {
        S.O.P("Account Number:" +
        accountNumber + "\n Balance :"
        + balance);
    }
}
```

```
public void computeInterest()
{
    double interestRate = 0.05;
    double interest = balance * interestRate;
    balance += interest;
    S.O.P("Interest of $" + interest +
          " computed & added to the
          balance");
}

class Bank
{
    public static void main(String aa[])
    {
        SavingsAccount SA = new SavingsAccount
                            ("Ramya", "SA100");
        CurrentAccount CA = new CurrentAccount
                            ("Radha", 1A2002);

        SA.deposit(5000);
        SA.displayBalance();
        SA.computeInterest();
        SA.displayBalance();
        SA.withdraw(2000);
        SA.displayBalance();
        CA.deposit(8000);
        CA.displayBalance();
        CA.withdraw(5000);
        CA.displayBalance();
    }
}
```

SURYA Gold
Date Page

Output:
Deposit of $5000.0 successful
Account Number: SA1001
Balance: 5000.0
Interest of $256.0 computed and
added to the balance.

1.

2

3

SURYA Gold
Date Page

Lab - 9

Report on AWT Programs

1. TextFieldDemo

The program displays text field and labels. When "Enter" button is pressed, the program updates the display to show the entered name and password. Also, it handles window closing events.

2. Division Main

It provides input fields, a division button and a result display label similar to previous program.

3. ButtonList D

A window with three buttons "Yes", "NO" and "Undecided" will be displayed. When any of these three buttons are clicked, a dialog window pops up displaying a corresponding message based on the button clicked.

4. ButtonList

This program is similar to previous program, but the only difference is that once any of those buttons are clicked.

5 ButtonDrag:

Upon compiling, a window appears with number blocks from 0 to 9 along with reset, start and restart button. The objective is to arrange the numbers in ascending order by swapping adjacent buttons