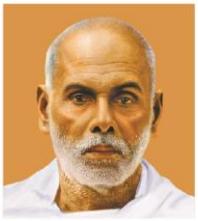
Sree Narayana Guru College
of Engineering & Technology
CHALAKKODE P.O., KOROM, PAYYANUR, KANNUR-670 307



SAMPLES OF TUTORIALS SHEET

SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY

PAYYANNUR, KANNUR

Tutorial	**1**	*Academic Year / Semester*	**2022-23/5**
Subject name with code	**CST 205 OBJECT ORIENTED PROGRAMMING USING JAVA**	*Branch*	**CSE**
Date of Issue	14/09/2022	*Date of submission*	29/09/2022

Q.No	*QUESTIONS*	*Mark*	*CO*	*Level*
1.	Illustrate the following java program: 1. Check the given number is prime or not 2. Check the given number is odd or even 3. Check the given string is palindrome or not 4. Print Fibonacci series 5. Matrix Addition 6. Sum of elements in an array. 7. Print a pyramid series	35	I	2

CO - Course Outcome [CO]

CO 1: Implement programs in Java which use data types, operators, control statements, built in packages & interfaces, Input/output streams and Files.

LEVEL - Bloom's Taxonomy Level

Level 2: Understanding

Dr. LEENA A. V.
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY, PAYYANUR
KANNUR

SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY

PAYYANNUR, KANNUR

ANSWER KEY/VALUATION SCHEME

Tutorial	**1 (SCHEME)**	*Academic Year / Semester*	**2022-23/5**
Subject name with code	**CST 205 OBJECT ORIENTED PROGRAMMING USING JAVA**	*Branch*	**CSE**
Date of Issue	14/09/2022	*Date of submission*	29/09/2022

Q.No	*QUESTIONS*	*Mark*	*CO*	*Level*
1.	Illustrate the following java program: 1. Check the given number is prime or not 2. Check the given number is odd or even 3. Check the given string is palindrome or not 4. Print Fibonacci series 5. Matrix Addition 6. Sum of elements in an array. 7. Print a pyramid series Each programs – 5 marks	35	I	2

CO - Course Outcome [CO]

CO 1: Implement programs in Java which use data types, operators, control statements, built in packages & interfaces, Input/output streams and Files.

LEVEL - Bloom's Taxonomy Level

Level 2: Understanding

Dr. LEENA A. V.
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY, PAYYANUR
KANNUR

SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY

PAYYANNUR, KANNUR

Tutorial	**2**	*Academic Year / Semester*	**2022-23/5**
Subject name with code	**CST 205 OBJECT ORIENTED PROGRAMMING USING JAVA**	*Branch*	**CSE**
Date of Issue	1/10/2022	*Date of submission*	10/10/2022

Q.No	*QUESTIONS*	*Mark*	*CO*	*Level*
1.	Illustrate the following UML diagram: 1. Use case for Movie Reservation System. 2. Use case for Online Music Player System. 3. Class diagram for a Library Management System. 4. Class diagram for Online Movie Ticket Booking. 5. Activity diagram for Food Ordering System.	25	I	2

CO - Course Outcome [CO]

CO 1: Implement programs in Java which use data types, operators, control statements, built in packages & interfaces, Input/output streams and Files.

LEVEL - Bloom's Taxonomy Level

Level 2: Understanding

Dr. LEENA A. V.
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY, PAYYANNUR
KANNUR

SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY

PAYYANNUR, KANNUR

ANSWER KEY/VALUATION SCHEME

Tutorial	**2 (SCHEME)**	*Academic Year / Semester*	**2022-23/5**
Subject name with code	**CST 205 OBJECT ORIENTED PROGRAMMING USING JAVA**	*Branch*	**CSE**
Date of Issue	1/10/2022	*Date of submission*	10/10/2022

Q.No	*QUESTIONS*	*Mark*	*CO*	*Level*
1.	Illustrate the following UML diagram: 1. Use case for Movie Reservation System. 2. Use case for Online Music Player System. 3. Class diagram for a Library Management System. 4. Class diagram for Online Movie Ticket Booking. 5. Activity diagram for Food Ordering System. **Diagram- 5 marks each**	25	I	2

CO - Course Outcome [CO]

CO 1: Implement programs in Java which use data types, operators, control statements, built in packages & interfaces, Input/output streams and Files.

LEVEL - Bloom's Taxonomy Level

Level 2: Understanding

Dr. LEENA A. V.
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY, PAYYANUR
KANNUR

SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY

Tutorial	**3**	*Academic Year / Semester*	**2022-23/ 3**
Subject name with code	**CST 205 OBJECT ORIENTED PROGRAMMING USING JAVA**	*Branch*	**CSE**
Date of Issue	**07/10/2022**	*Date of submission*	**14/10/2022**

Q.No	*QUESTIONS*	*Mark*	*CO*	*Level*
1.	Demonstrate the following control statements in Java: i. Decision Making /Selection Statements ii. Looping Statements iii. Jump Statements	10	2	2

CO - Course Outcome [CO]

CO 2 : To get thorough knowledge of java languages and to utilize the features of java like datatypes, operators, control statements etc and how to use the object oriented concepts - classes, objects ,constructors, data hiding, inheritance and polymorphism.

LEVEL - Bloom's Taxonomy Level

Level 2: Understanding

Dr. LEENA A. V.
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY, PAYYANUR
KANNUR

SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY

Answer key /Valuation Scheme

Tutorial	**3(Scheme)**	*Academic Year / Semester*	**2022-23/ 3**
Subject name with code	**CST 205 OBJECT ORIENTED PROGRAMMING USING JAVA**	*Branch*	**CSE**
Date of Issue	**07/10/2022**	*Date of submission*	**14/10/2022**

Q.No	*QUESTIONS*	*Mark*	*CO*	*Level*
1.	Demonstrate the following control statements in Java: i. Decision Making /Selection Statements ii. Looping Statements iii. Jump Statements **Syntax and Flowchart – 5 marks for each , Programming Example – 5 marks for each**	30	2	2

CO - Course Outcome [CO]

CO 2 : To get thorough knowledge of java languages and to utilize the features of java like datatypes, operators, control statements etc and how to use the object oriented concepts - classes, objects ,constructors, data hiding, inheritance and polymorphism.

LEVEL - Bloom's Taxonomy Level

Level 2: Understanding

Dr. LEENA A. V.
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY, PAYYANUR
KANNUR

SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY

Tutorial	4	*Academic Year / Semester*	**2022-23/ 3**
Subject name with code	**CST 205 OBJECT ORIENTED PROGRAMMING USING JAVA**	*Branch*	**CSE**
Date of Issue	**17/10/2022**	*Date of submission*	**25/10/2022**

Q.No	*QUESTIONS*	*Mark*	*CO*	*Level*
1.	Contrast the difference between Method overloading and Method Overriding.	5	2	2
2.	Define abstract class.	5	2	1
3.	Compare and contrast class and abstract class.	5	2	2

CO - Course Outcome [CO]

CO 2 : To get thorough knowledge of java languages and to utilize the features of java like datatypes, operators, control statements etc and how to use the object oriented concepts - classes, objects ,constructors, data hiding, inheritance and polymorphism.

LEVEL - Bloom's Taxonomy Level

Level 1: Remenmbering
Level 2: Understanding

Dr. LEENA A. V.
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY, PAYYANUR
KANNUR

SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY

Answer Key/Valuation Scheme

Tutorial	4(Scheme)	*Academic Year / Semester*	**2022-23/ 3**
Subject name with code	**CST 205 OBJECT ORIENTED PROGRAMMING USING JAVA**	*Branch*	**CSE**
Date of Issue	**17/10/2022**	*Date of submission*	**25/10/2022**

Q.No	*QUESTIONS*	*Mark*	*CO*	*Level*
1.	Contrast the difference between Method overloading and Method Overriding. **Any 5 points – 5 marks**	5	2	2
2.	Define abstract class. **Definition -1 mark, Explanation – 4 marks**	5	2	1
3.	Compare and contrast class and abstract class. **Comparison – 5 marks**	5	2	2

CO - Course Outcome [CO]

CO 2 : To get thorough knowledge of java languages and to utilize the features of java like datatypes, operators, control statements etc and how to use the object oriented concepts - classes, objects ,constructors, data hiding, inheritance and polymorphism.

LEVEL - Bloom's Taxonomy Level

Level 1: Remenmbering
Level 2: Understanding

Dr. LEENA A. V.
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY, PAYYANUR
KANNUR

SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY

Tutorial	5	*Academic Year / Semester*	**2022-23/ 3**
Subject name with code	**CST 205 OBJECT ORIENTED PROGRAMMING USING JAVA**	*Branch*	**CSE**
Date of Issue	**31/10/2022**	*Date of submission*	**11/11/2022**

Q.No	*QUESTIONS*	*Mark*	*CO*	*Level*
1.	Develop a java package named primepackage, with a class Prime containing a static method that check whether a number is prime or not and returns that information. Import this package in another class and use to check a number is prime or not.	5	3	3
2.	Illustrate a java package named "even" package with a class Even containing a static method to check whether a given number is even or not.	5	3	2
3.	Contrast the difference between packages and interface.	5	3	2

CO - Course Outcome [CO]

CO 3: To understand the utilization of built in packages & interfaces and to illustrate how robust programs can be written in Java using exception handling mechanism , Input/ Output Streams and Files in Java to develop programs

LEVEL - Bloom's Taxonomy Level

Level 2: Understanding
Level 3: Applying

Dr. LEENA A. V.
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY, PAYYANUR
KANNUR

SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY PAYYANNUR, KANNUR

Answer Key / Valuation Scheme

Tutorial	5(Scheme)	*Academic Year / Semester*	**2022-23/ 3**
Subject name with code	**CST 205 OBJECT ORIENTED PROGRAMMING USING JAVA**	*Branch*	**CSE**
Date of Issue	**31/10/2022**	*Date of submission*	**11/11/2022**

Q.No	*QUESTIONS*	*Mark*	*CO*	*Level*
1.	Develop a java package named primepackage, with a class Prime containing a static method that check whether a number is prime or not and returns that information. Import this package in another class and use to check a number is prime or not. **Syntax and logic -2.5 mark , Program – 2.5 marks.**	5	3	3
2.	Illustrate a java package named "even" package with a class Even containing a static method to check whether a given number is even or not. **Syntax and logic -2.5 mark , Program – 2.5 marks.**	5	3	2
3.	Contrast the difference between packages and interface. **Any 5 points -5 marks.**	5	3	2

CO - Course Outcome [CO]

CO 3: To understand the utilization of built in packages & interfaces and to illustrate how robust programs can be written in Java using exception handling mechanism , Input/ Output Streams and Files in Java to develop programs

LEVEL - Bloom's Taxonomy Level

Level 2: Understanding
Level 3: Applying

Dr. LEENA A. V.
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY, PAYYANUR
KANNUR

SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY

Tutorial	6	*Academic Year / Semester*	**2022-23/ 3**
Subject name with code	**CST 205 OBJECT ORIENTED PROGRAMMING USING JAVA**	*Branch*	**CSE**
Date of Issue	**22/11/2022**	*Date of submission*	**30/11/2022**

Q.No	*QUESTIONS*	*Mark*	*CO*	*Level*
1.	Explain in detail following file handling operations in java: i. Create a file ii. Get information of a file iii. Read from a file iv. Write to a file v. Delete a file	10	3	2

CO - Course Outcome [CO]

CO 3: To understand the utilization of built in packages & interfaces and to illustrate how robust programs can be written in Java using exception handling mechanism , Input/ Output Streams and Files in Java to develop programs

LEVEL - Bloom's Taxonomy Level

Level 2: Understanding

Dr. LEENA A. V.
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY, PAYYANUR
KANNUR

SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY

Valuation Key/ Answer Scheme

Tutorial	6(Scheme)	*Academic Year / Semester*	**2022-23/ 3**
Subject name with code	**CST 205 OBJECT ORIENTED PROGRAMMING USING JAVA**	*Branch*	**CSE**
Date of Issue	**22/11/2022**	*Date of submission*	**30/11/2022**

Q.No	*QUESTIONS*	*Mark*	*CO*	*Level*
1.	Explain in detail following file handling operations in java: i. Create a file ii. Get information of a file iii. Read from a file iv. Write to a file v. Delete a file **Each operation with simple java program – 10 marks.**	10	3	2

CO - Course Outcome [CO]

CO 3: To understand the utilization of built in packages & interfaces and to illustrate how robust programs can be written in Java using exception handling mechanism , Input/ Output Streams and Files in Java to develop programs

LEVEL - Bloom's Taxonomy Level

Level 2: Understanding

Dr. LEENA A. V.
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY, PAYYANUR
KANNUR

SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY

Tutorial	**7**	*Academic Year / Semester*	**2022-23/ 3**
Subject name with code	**CST 205 OBJECT ORIENTED PROGRAMMING USING JAVA**	*Branch*	**CSE**
Date of Issue	**02/12/2022**	*Date of submission*	**15/12/2022**

Q.No	*QUESTIONS*	*Mark*	*CO*	*Level*
1.	Illustrate different event classes and event listener interface in java.	10	4	2

CO - Course Outcome [CO]

CO 4: To provide basic exposure for the application of programs in java using multithreading, string handling mechanisms, collection framework and event handling mechanisms.

LEVEL - Bloom's Taxonomy Level

Level 2: Understanding

Dr. LEENA A. V.
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY, PAYYANUR
KANNUR

SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY

Valuation Key/ Answer Scheme

Tutorial	7(Scheme)	*Academic Year / Semester*	**2022-23/ 3**
Subject name with code	**CST 205 OBJECT ORIENTED PROGRAMMING USING JAVA**	*Branch*	**CSE**
Date of Issue	**02/12/2022**	*Date of submission*	**15/12/2022**

Q.No	*QUESTIONS*	*Mark*	*CO*	*Level*
1.	Illustrate different event classes and event listener interface in java. **List out the event class and event listener interface- 2 marks, Explanation – 8 marks.**	10	4	2

CO - Course Outcome [CO]

CO 4: To provide basic exposure for the application of programs in java using multithreading, string handling mechanisms, collection framework and event handling mechanisms.

LEVEL - Bloom's Taxonomy Level

Level 2: Understanding

Dr. LEENA A. V.
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY, PAYYANUR
KANNUR

SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY

Tutorial	**8**	*Academic Year / Semester*	**2022-23/ 3**
Subject name with code	**CST 205 OBJECT ORIENTED PROGRAMMING USING JAVA**	*Branch*	**CSE**
Date of Issue	**12/12/2022**	*Date of submission*	**21/12/2022**

Q.No	*QUESTIONS*	*Mark*	*CO*	*Level*
1.	Explain in detail different Swing layout managers in Java swing.	10	5	2

CO - Course Outcome [CO]

CO 5: To impart the techniques of creating GUI based applications and database connectivity.

LEVEL - Bloom's Taxonomy Level

Level 2: Understanding

Dr. LEENA A. V.
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY, PAYYANUR
KANNUR

SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY

Valuation Key/ Answer Scheme

Tutorial	**8(Scheme)**	*Academic Year / Semester*	**2022-23/ 3**
Subject name with code	**CST 205 OBJECT ORIENTED PROGRAMMING USING JAVA**	*Branch*	**CSE**
Date of Issue	**12/12/2022**	*Date of submission*	**21/12/2022**

Q.No	*QUESTIONS*	*Mark*	*CO*	*Level*
1.	Explain in detail different Swing layout managers in Java swing. **List out the Swing layout manager - 2 marks, Explanation with figures– 8 marks.**	10	5	2

CO - Course Outcome [CO]

CO 5: To impart the techniques of creating GUI based applications and database connectivity.

LEVEL - Bloom's Taxonomy Level

Level 2: Understanding

Dr. LEENA A. V.
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY, PAYYANUR
KANNUR

Tutorial - 1

20/9/22

Q1) check given number is prime or not

Q2) check given number is odd or even

Q3) check given string is palindrome or not

Q4) print fibonacci

Q5) print a pyramid pattern

Q6) matrix addition

Q7) sum of elements in a array.

Q1) Prime or not

```
import java.util.scanner;
class prime
{
public static void main (string args[])
 {
int n, c=1, i=2;
 Scanner sc = new scanner (system.in);
system.out.print("\n enter a number=");
  n= sc.next Int();
  for (i=2; i<=2; i++)
  {
  if (n%.2==0 || n%.3==0)
  {
  c=0;
  break;
  }
  }
  if (c==1)
```

```
System.out.println (n+" is prime number");
else
System.out.println (n+" is not a prime number");
}
}
```

Q2) odd or even

```
import java.util.scanner;
Public class odd or even
{
Public static void main (String [] args)
{
int n;
Scanner sc= new scanner (System.in);
System.out.print ("Enter the number");
n= sc.nextInt();
if (n%2==0)
{
System.out.print In ("the given number is even");
}
else
{
System.out.println ("the given number is odd")
}
}
}
```

Q4) fibonacci

```
import java.util.*;
Public ~~dac~~ class first
{
Public static void main (String [] args)
{
scanner sc= new scanner (system.in);
int a=0, b=1, c, l=3, n;
```

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

```
system.out.print("enter the number");
n=sc.nextInt();
system.out.print(a+" "+b);
while(i<=n)
{
c=a+b;
system.out.print(" "+c);
a=b;
b=c;
i++;
}
}
```

Q5) Pyramid pattern

```
import java.io.*;
Public class pyramid
{
Public static void main(string[]args)
int rows=5;
for(inti=1;i<rows;i++)
{
for(intj=i;j<=i;j++)
{
System.out.print(" "+j);
}
System.out.println();
}
}
```

Q7) Sum of elements in array

```
import java.io.*;
Public class sum of array
{
Public static void main(String args[])
{
int[]arr=new int[]{1,2,3,4,5};
int sum=0;
```

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

```
for (int i=0 ; i
{
sum = sum + arr[i];
}
System.out.println ("sum of all the elements of
    an array:" + sum);
}
}
```

Q6) Matrix addition

```
import java.util.scanner;
public class matrix sum
{
public static void main (String [] args)
{
int a[][], b[][], c[][], i, j
a = new int [3][3];
b = new int [3][3];
c = new int [3][3];
scanner sc = new scanner (system.in);
system.out.println ("enter the 1st matrix");
for (i=0 ; i<3; i++)
for (j=0; j<3; j++)
a[i][j] = sc.nextInt();
system.out.println ("enter 2nd matrix");
}
for (i = 0; i<3; i++)
{
for (j=0; j<3; j++)
b[i][j] = sc.nextInt();
System.out.println ("addition ...");
}
for (i=0; i<3; i++)
{
for (j=0; j<3; j++)
{
c[i][j] = a[i][j] + b[i][j]
```

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

```
}
System.out.print(c[i][j]+"\t");
}
System.out.println();
}
}
}
```

Q3) palindrome

```
import java.util.Scanner;
public class palindrome string
{
Public static void main (String[] args)
{
Scanner sc=new Scanner (System.in);
System.out.println("enter your string");
String str=sc.next();
String org_str = str;
String rev = "";
int ln= str.length();
for (int i= ln-1; i>=0; i--)
{
rev= rev + str.charAt(i);
}
if (org_str.equals(rev))
{
System.out.println( org_str + "is palindrome
string");
}
else
{
System.out.println(org_str + "is not a palindrome
string");
}
}
```

Nimi 10

10/10

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

Tutorial-2

1) Illustrate the following UML diagram

a) use case for movie reservation system

b) use case for online music playing system

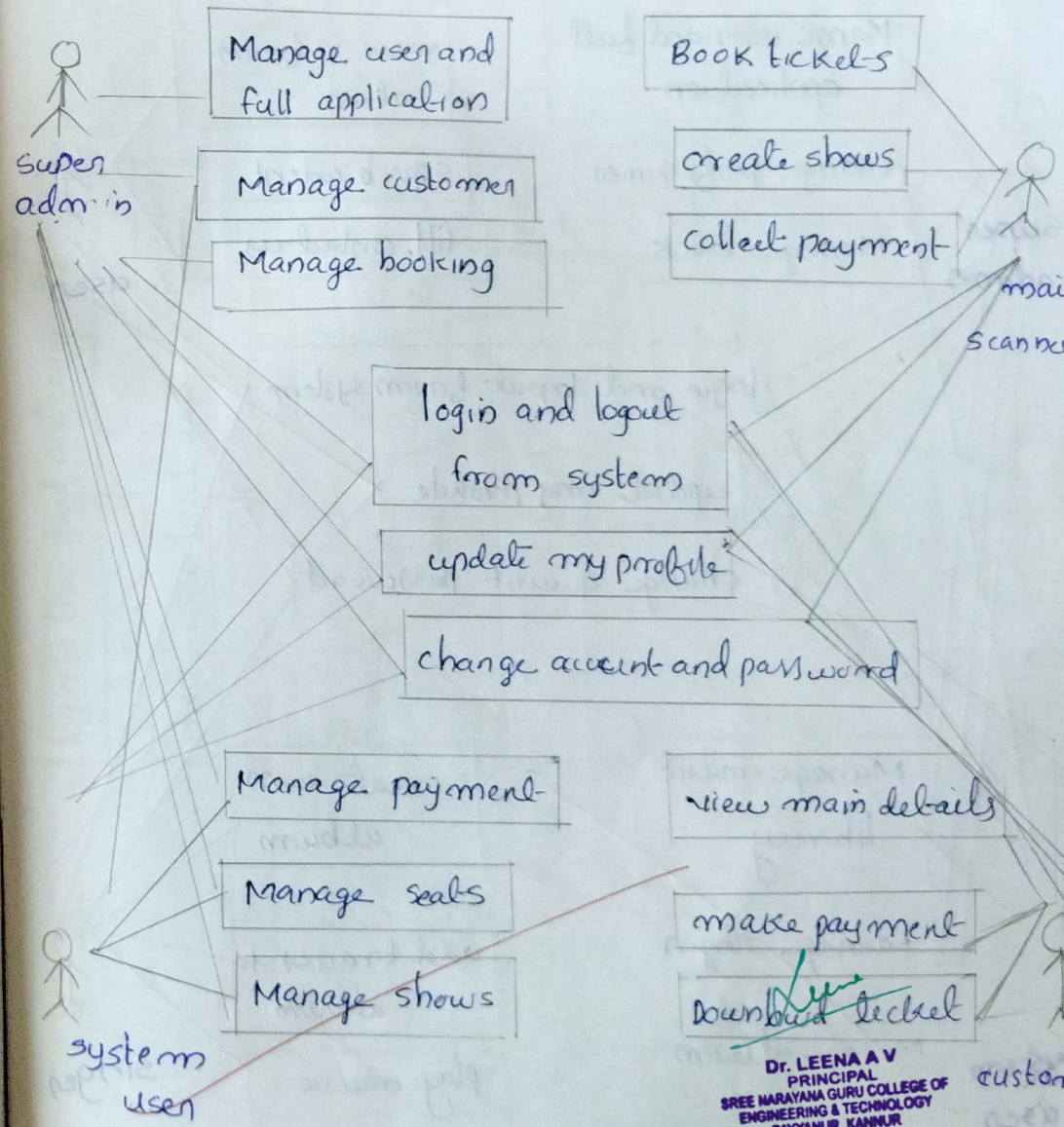
c) use case for movie reservation system

d) class diagram for a book and student

e) class diagram for online movie ticket booking system which contains classes like admin, registered user, visitor and guest user.

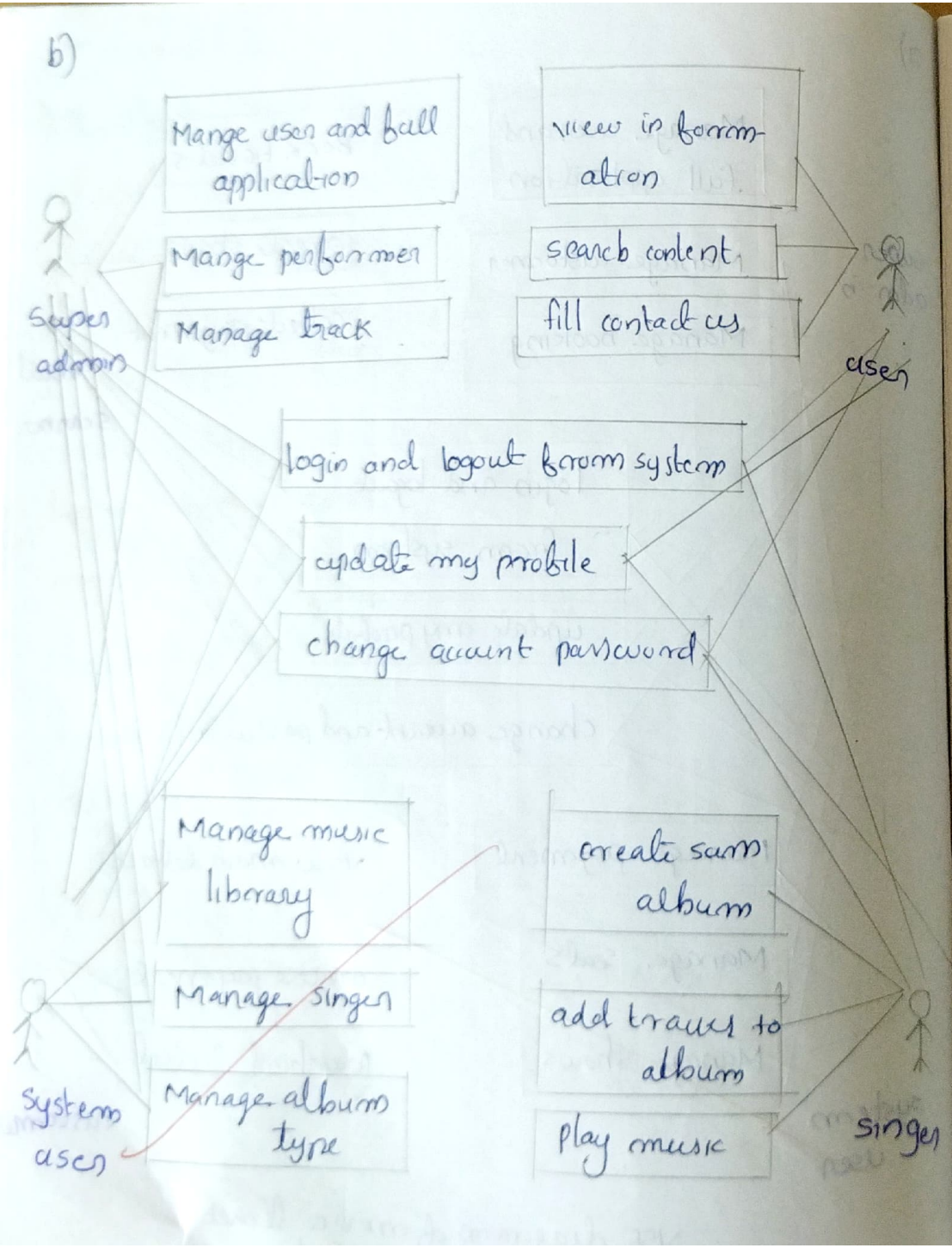
f) activity diagram for food ordering system which contains activities like order, delivery-food item, category and payment.

a)

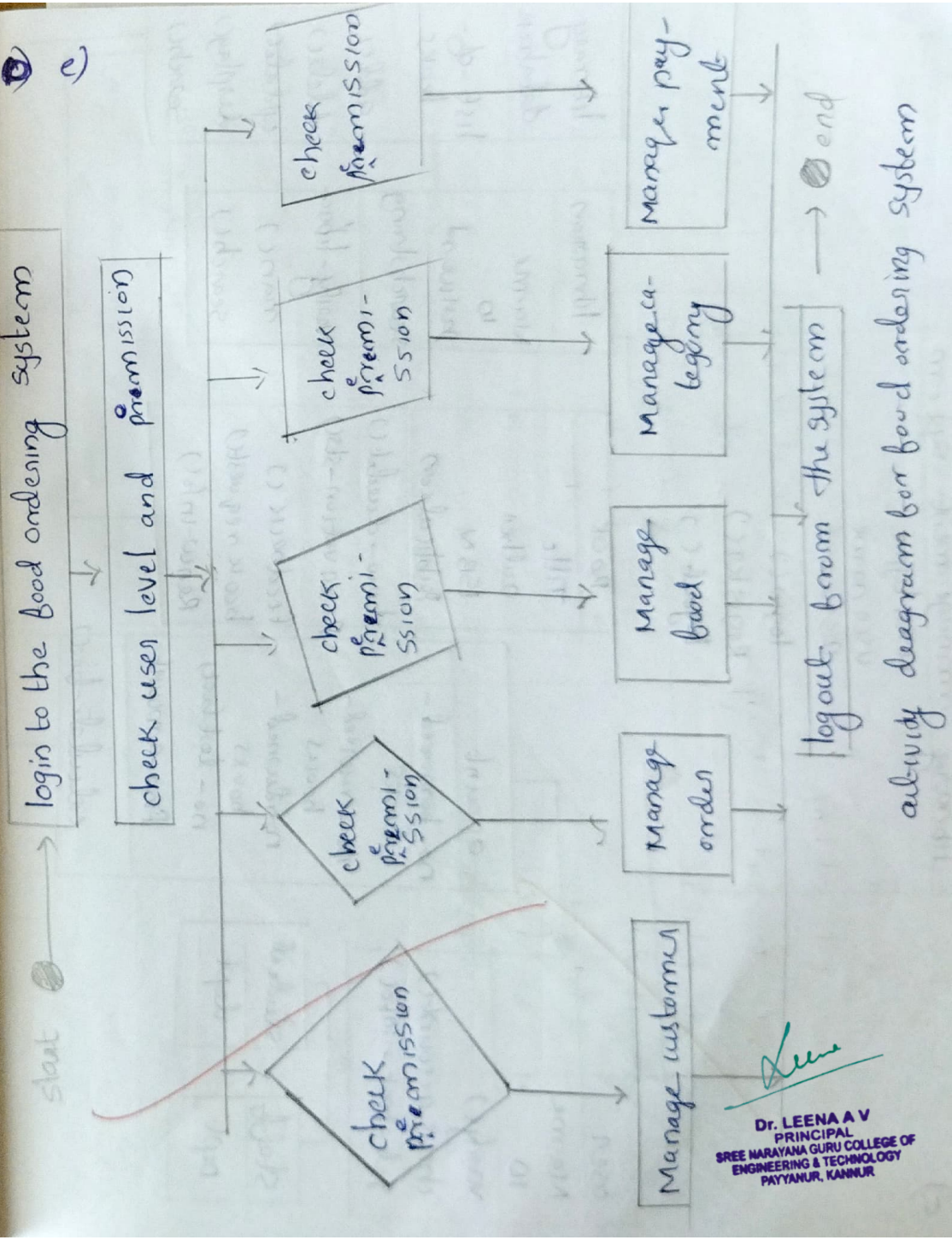


VSE diagram of movie ticket booking system

b)

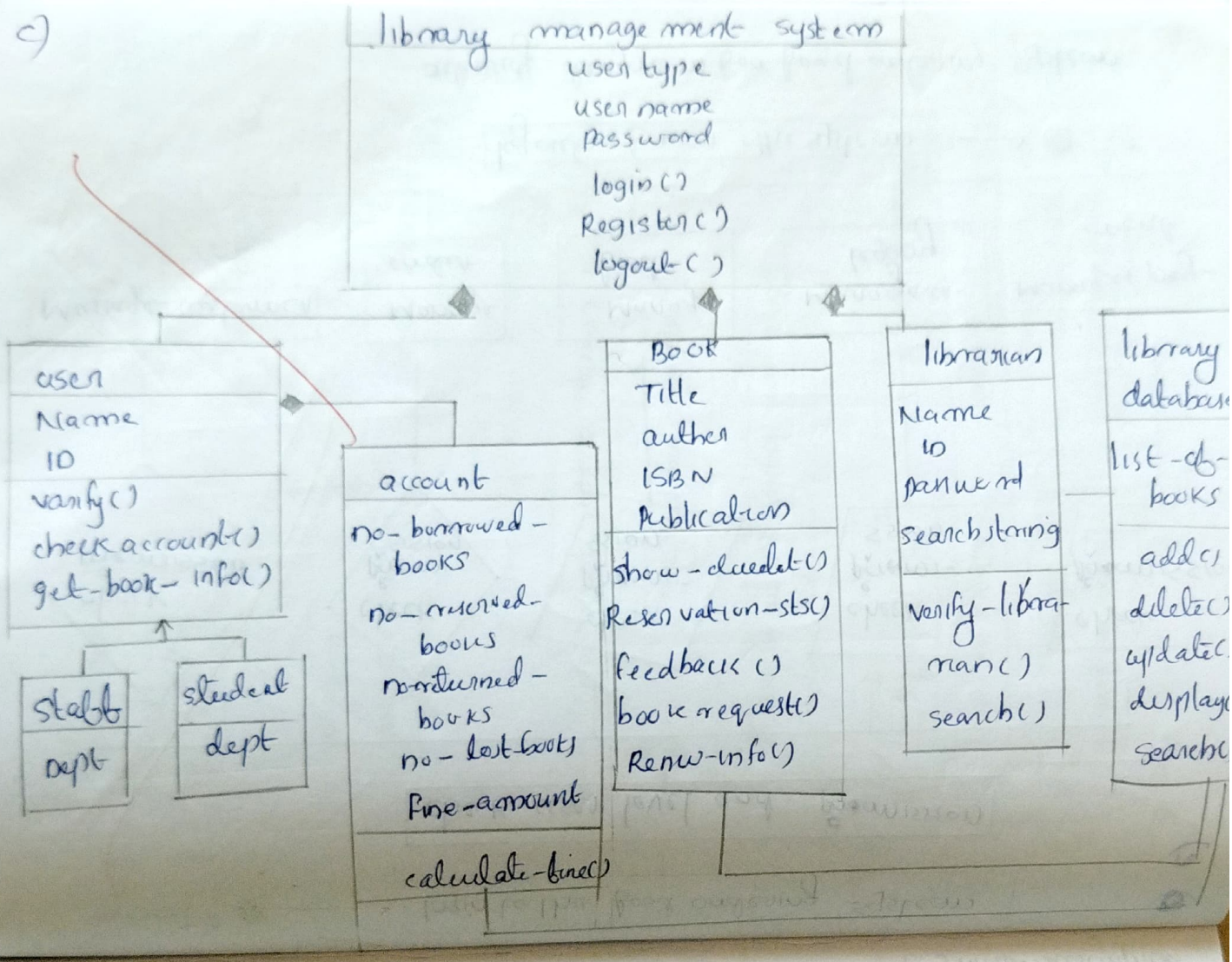


c)



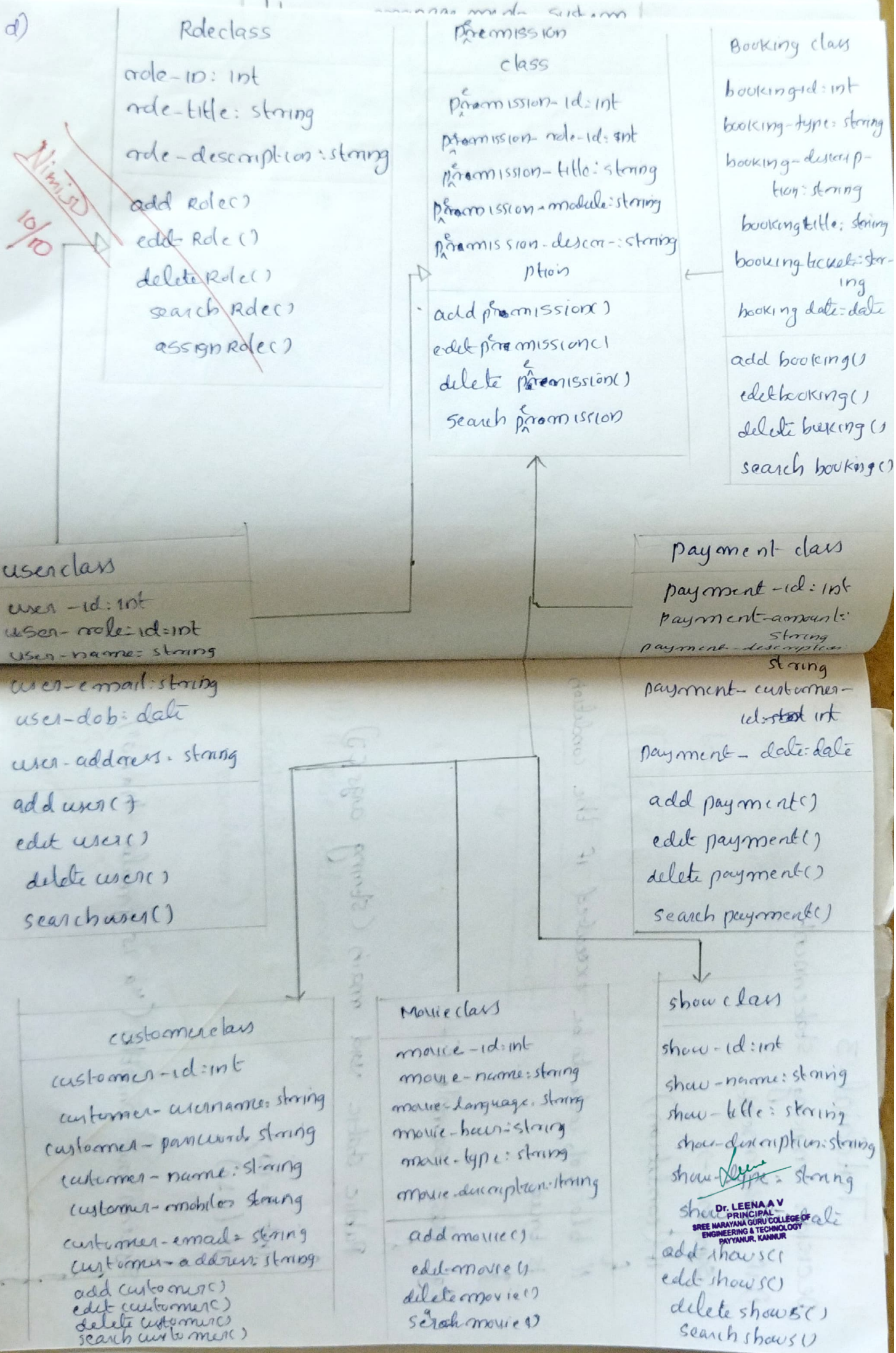
Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

c)



Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

d)



Tutorial-3

Decision making statement

i) if statement

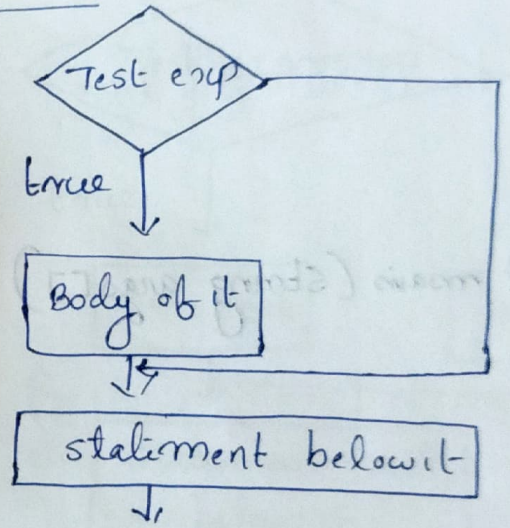
Syntax :→

```
if (condition)
{
// block of code to be executed if the condition
is true
}
```

eg: class sample if

```
{
Public static void main (String args[])
{
int a=10;
if (a>0)
{
System.out.println("a is greater than 0");
}
}
}
```

Flow chart



ii) if else statement

Syntax :→

```
if (condition)
{
// block of code to be executed if true
}
else
{
```

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

// block of code to be executed if false

}

eg: class sample

{

Public static void main (String args[])

{

int a,b;

if(a>b)

{

System.out.println ("a is greater than b");

}

else

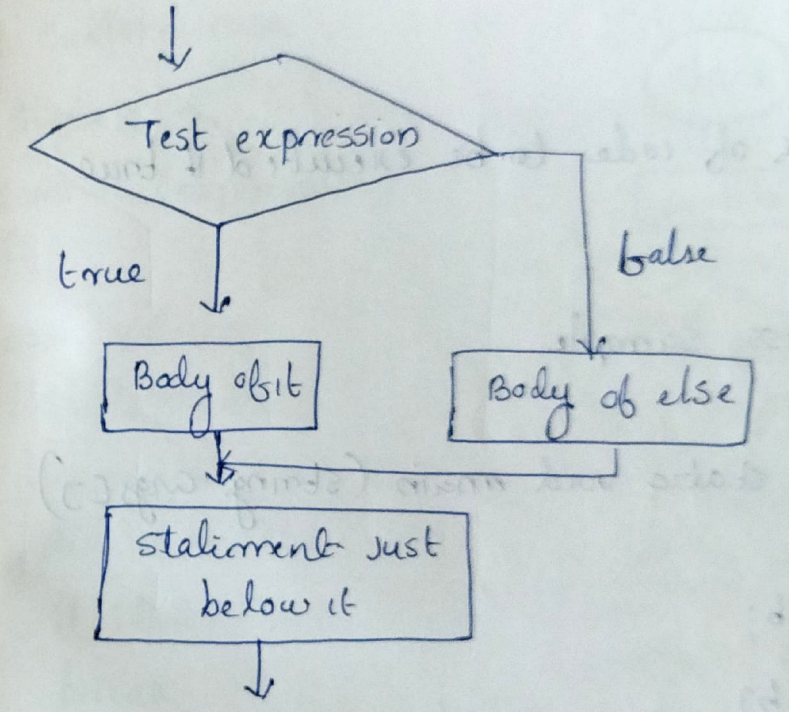
{

System.out.println ("b is greater than a");

}

}

}



iii) If else if ladder

Syntax

if (condition)

{

// block of code to be executed if true

}

else if (condition)

{

// block of code to be executed if true

}

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

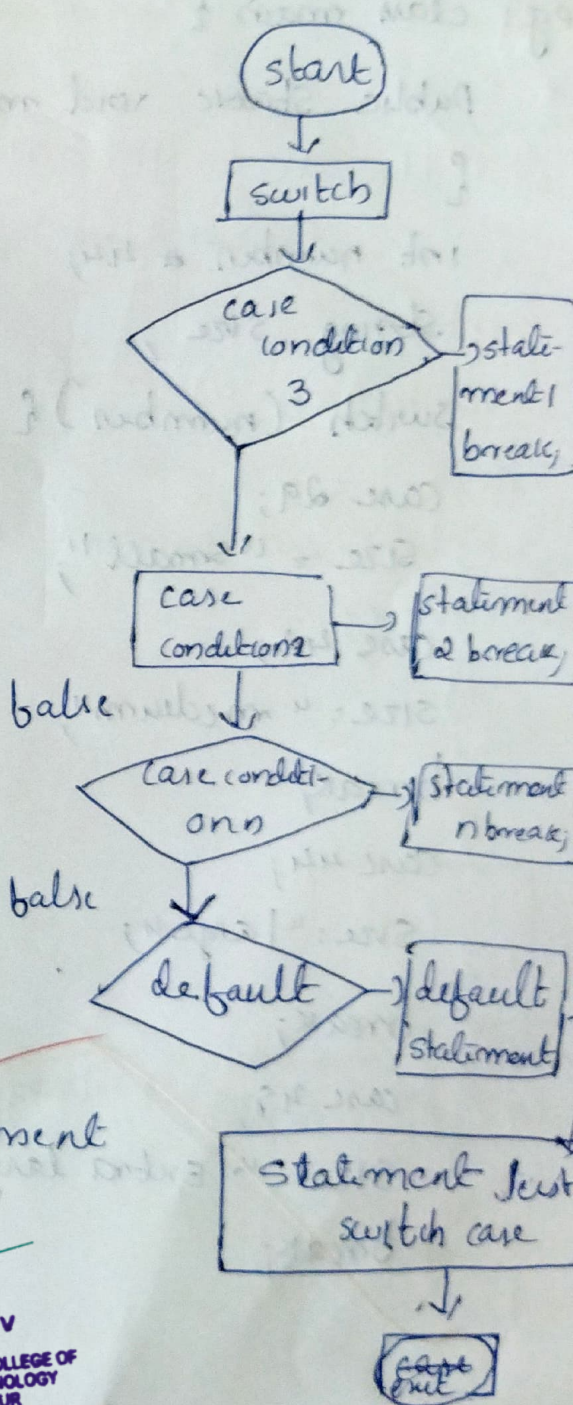
```
else
{
// block of code to be executed if true
}

eg: class sample
  {
  Public static void main (String args[])
  {
  int a,b;
  if(a>b)
   {
  System.out.println("a is greater than b");
   }
   else if (b>a)
   {
   System.out.println("b is greater than a");
   else
   {
   System.out.println("a is equal to b");
   }
   }
   }
```

IV) switch case

syntax :-

```
switch (expression)
{
case values;
// statement
break;
 case value 2;
 // statement
 break;
 .
 .
 .
 case value N;
 // statement
 break;
 default;
 // default statement
 }
```



Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

eg: class main {

```
Public Static void main (String [] args)
{
int number = 44;
String size;
Switch (number) {
Case 29;
 Size = "small";
Case 40;
Size: " medium";
break;
Case 44;
 Size: "large";
 break;
 case 48;
 Size: " Extra large";
 break;
```

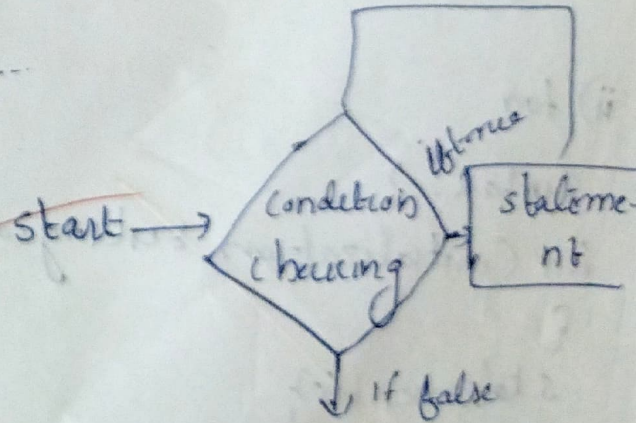
```
default
Size = "unknown";
break;
}
System.out.println ("size:"+size);
}
}
```

looping statement:-

i) ~~do~~ while

syntax:-

```
while (condition)
{
 loop statements...
 }
```



Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

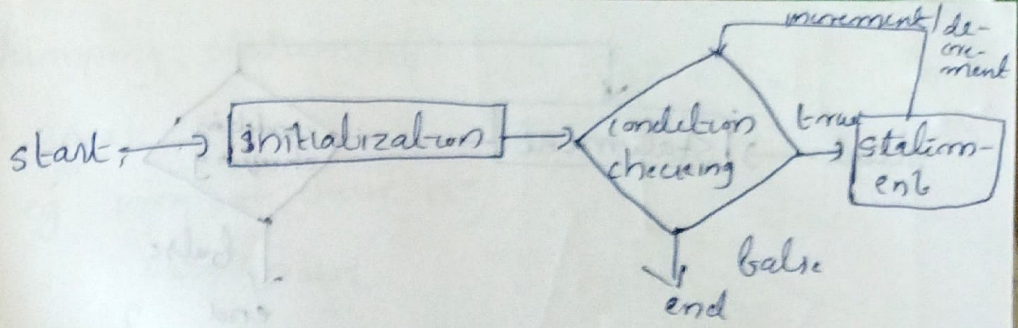
eg: class while loop Demo

```
{
 Public static void main (string args[])
{
int x=1;
while (x<=4)
{
system.out.println ("value of x"+x);
 x++;
 }
 }
 }
```

ii) for

Syntax:->

```
for (initialization, testing condition)
{
statement(s);
}
```



eg: class for loop demo

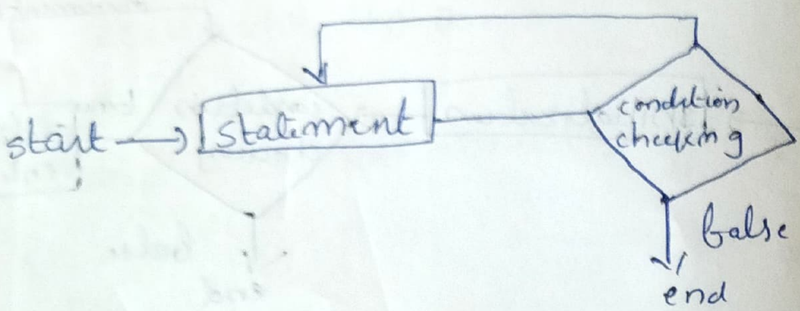
```
 {
 Public static void main (String args[])
 {
 for (int x= 2; x<=4; x++)
 System.out.println ("value of x: "+x);
  }
  }
```

iii) do

Syntax:->

```
do
{
statements...
}
while (condition);
```

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR



eg: class dowhile loop Demo

```
{
public static void main (String args[])
{
int x=21;
 do
{
system.out.println ("value of x:"+x);
x++;
}
while (x<20)
}
}
```

Jumping statement

1) Break

eg: import java.io.*;

```
    class Break.
    {
  Public static void main (String []args)
  {
  int n=10;
  for (int i=0; i<n; i++){
  If (i==6)
    break;
   System.out.println(i);
   }
   }
   }
```

syntax:-

break;

Arathi. T. V
Roll No: 14

25/10/22

Tutorial-4

1) Difference b/w method overriding and method overloading with example
2) write a note on abstract class
3) Diff b/w abstract class and normal class
4) Diff b/w JVM and JRE
5) features of oop or concept of oop

1) Method overloading

- Method overloading is used to increase the readability of the program.
- Method overloading is performed within class
- in case of method overloading, parameter must be different
- Method overloading is the example of compile time polymorphism

Method overriding

- Method overriding is used to provide the specific implementation of the method that is already provided by its super class.
- Method overriding occurs in two classes that ~~is already provided~~ have IS-A
- incase of method overriding parameter must be same
- method overriding is the example of runtime polymorphism.

2) abstract class:-> is restricted class that cannot be used to create objects

Rule

- an abstract class must be declared with an abstract keyword
- it can have abstract and non-abstract method

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

- it cannot be instantiated
- it can have final method
- it can have constructors and static method also.

eg: abstract class animal

```
{
Public abstract void animal sound();
Public void sleep(){
System.out.println("zzzz");
}
}
class pig extends animal{
Public void animal sound(){
system.out.println ("the pig says: Wee Wee");
}
}
class My main class{
Public static void main (string [] args){
Pig my pig = new pig();
my pig.animal sound();
My pig.sleep();
}
}
```

4) JRE

JRE creates a runtime environment in which the program can run effectively.

JRE becomes operational at the moment when the application program is executed

JRE is installed into JDK by default.

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

~~JRE creates~~
JVM becomes operational when the byte code needs to be interpreted into machine language

JVM is sub bundled with JRE, whose main function is to read the generated bytecode

5) . easy to learn and understand
- it is object oriented programming language
- object oriented concept

- object
- class
- data abstraction and data encapsulation
- polymorphism
- inheritence
- platform independent
- powerful
- Robust
- Java is a ~~block~~ WORA programming language

Nimisha 25/10/22

19/10

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

Tutorial-5

S3 CSE	Roll NO:

1) Develop a Java package prime and use it in a program

```
Package prime
{
int x=0, n;
 Public void check()
{
for (int i=2; i<n/2; i++)
if (n%2==0)
  x++
}
if (x==0)
System.out.println("the no. is prime numbe
 else
System.out.println("the no. is not a prime
                    number");
}
}
import java.prime package.*;
```

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

```
import java.util.Scanner;
class find
{
Scanner sc= new Scanner (System.in)
System.out.println ("Enter the value");
int n=sc.nextInt();
Public static void main (String [] args)
{
prime pr=new prime();
pr.check();
}
}
```

a) Illustrate a java package "even" with a class Even containing a static method to check even or not

```
package Even odd
class Even
{
int n;
void check ()
{
if (n% 2==0)
System.out.println ("Even number");
else
System.out.println ("odd number");
}
}
import java Even odd.*;
import java.util.*;
class find Even odd
{
Scanner sc= new Scanner (System.in);
System.out.println ("Enter the no:");
int n = sc.nextInt()
Public static void main (String args[])
{
Even e= new Even();
e.check;
}
}
```

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

3) Difference b/w package and interface

Package

- Group of classes/interface
- created using 'package' keyword
- its can be imported
- uses import keyword to import

Interface

- Group of abstract method and constant fields
- created using 'interface' keyword
- its can only be extended by another interface and implemented
- uses 'implement' keyword to implement

Nimmi 16/11/22 16/10

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

22/11/'22

Tutorial 6

Arathi. T. V
Roll No: 14

Create a file

- use createNewFile() method to create a file.
- This method return true if the file was successfully created else false if file already exists.
- The method is enclosed in a try ... catch block.
- Its throws an IO Exception if an error occurs

Example

```
import java.io.File;
Import java.io.IoException;
Public class createfile {
Public static void main (String [] args) {
try {
File myobj = new file ("filename.txt");
if (myobj.createNewfile()) {
System.out.println ("file created:" + myobj.getName());
}else{
system.out.println ("file already exists.");
}
} catch (IoException e) {
```

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

```
System.out.println("An error occured.");
e.printStackTrace();
}
}
}
```

2) Get file Information

```
import java.io.file;
Public class file information{
Public static void main (String [] args){
file myobj = new file ("NewFile1.txt");
If (myobj.exists()){
System.out.println("file name:" + myobj.getName());
System.out.println("Absolute path:" + myobj.getAbsolute path());
System.out.println("writeable:" + myobj.canwrite());
System.out.println("Readable" + myobj.canRead());
System.out.println("file size in bytes" + myobj.length());
}else{
System.out.println("the file doesnot exist");
}
}
}
```

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

Read from a file

```
Import java.io.file;
Import java.io.fileNotfound Exception;
Import java.util.scanner;
Public static Read from file{
Public static void main (String [] args){
try{
file myobj = new file ("D:\Alehandling\Newfile1.Ext");
Scanner myReader = new scanner (myobj);
while (myReader.hasnextLine()){
string data = myReader.nextLine();
System.out.println(data);
}
myReader.close();
}catch (fileNot found Exception e){
system.out.println("An error occured");
e.printstackTrace();
}
}
```

write to a file

```
import java.io.filewriter
Import java.io.Exceptions
Public class writeTOfile{
Public static void main (String [] args){
try{
```

```
FileWriter myWriter = new FileWriter("filename.txt");
myWriter.write("files in java might be tricky, but it is fun enough");
myWriter.close();
System.out.println("successfully wrote to the file");
} catch (IOException e){
System.out.println("An error occured");
e.printStackTrace();
}
}
}
```

Delete a file

```
Import java.io.File;
Public class Delete file{
Public static void main(String[]args){
file myobj = new file("filename.txt");
If(myobj.delete()){
System.out.println("deleted the file"+myobj.getName());
}else{
System.out.println("failed to delete the file");
}
}
}
}
```

10/10

Nimitha 29/11/22

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

Tutorial-7

Foreground events

- Those events which require the direct interaction of user.
- They are generated as consequances of a person interacting with the graphical components in Graphical user interface.
- For example, clecking on a button, moving the mouse entering a character through keyboard, selecting an item from list, scrolling the page etc.

Background events

- Those events that require the interaction of end user are known as background events.
- operating system interrupts, hard ware or software failure, time expires, an operations completion are the example of background events

- The object is registered using addItemListner() method
- when the action event occurs, the object's itemStateChanged method is invoked
- The interface method:

  void itemStateChanged (ItemEvent e)

→ invoked when an item has been selected or deselected by the user.

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

KeyListener interface

- The Java KeyListner is notified whenever you change the state of key
- The class which processes the key event
- implements the KeyListner interface
- The object of the class must be registered with a component
- The object is registered using addKeyListner() method
- The interface method:

  i. void keyPressed (KeyEvent e)

  invoked when a key has been pressed

ii) void keyReleased (KeyEvent e)

invoked when a key has been released

iii) void keyTyped (KeyEvent e)

invoked when a key has been typed

MouseListner interface

- Java mouseListner is notified whenever you change the state of mouse
- The class which processes the mouse event
- implements the mouseListner interface
- The object of the class must be registered with a component
- The object is registered using addMouseListener() method
- The interface method

i) void mouseClicked (MouseEvent e)

invoked when the mouse button has been clicked on a component

ii) void mouse entered (mouse event e)

invoked when mouse ents a component

iii) void mouseexited (mouse Event e)

invoked when mouse exits a component

iv) void mousepressed (Mouse Event e)

invoked when mouse button been pressed on a component

v) void mouse Released (Mouse Event e)

invoked when mouse button has been released on a component

5 Mouse motion Listner interface

- The java mouse motion listner is notified whenever you move or drag mouse
- it is notified against Mouse event
- used for receiving mouse events on a component
- The class that process mouse motion events needs to implements MouseMotionListener interface
- The interface method:

i) void mouseDragged (MouseEvent e)

invoked when mouse button is pressed on a component and then dragged

ii) void mouseMoved (MouseEvent e)

invoked when mouse cursor has been moved onto a component but no buttons have been pushed

6) WindowListner interface

- The Java windowListner is notified when even you change the state of window
- It is notified against windowevent
- The windowListner interface is found in java.awt.event package
- It have seven methods

i) void windowActivated (windowEvent e) - invoked when the window is set to be active window

ii) void windowclosed (windowevent e) - invoked when window has been closed as the result of calling dispose on the window

iii) void windowclosing (windowEvent e) when the user attempts to close the window from

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

the window's system menu.

iv) void windowDeactivated (windowEvent e) - invoked when a window is no longer the active window

v) void windowDeiconified (windowEvent e) - invoked when a window is changed from a minimized to a normal state

vi) void windowIconified (windowEvent e) - invoked when a window is changed from a normal to a minimized state.

vii) void windowOpened (windowEvent e) - invoked the first time a window is made visible.

10/10 Nimisha

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

Tutorial-8

Explain different swing layout managers

- layout refers to the arrangement of components with in containers
- The task of laying out the controls is done automatically by the layout managers
- properties like size, shape and arrangement varies from one layout manager to other

The class that the layout managers are

1. Java.awt. Border layout
2. Java.awt. card layout
3. Java.awt. flow layout
4. Java.awt. Grid Layout
5. Java.awt Grid bag layout
6. Javax swing. Box layout
7. Javax swing. Group Layout
8. Javax swing. Scroll panel layout
9. Javax swing. spring layout

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

1) Border Layout

- The border layout is used to arrange the components in five regions
  north, south, west, east, center
- each region may contain one component only
- The border layout provides ~~two~~ five constants for each region

1) public static final int NORTH
2) public static final int SOUTH
3) public static final int EAST
4) public static final int WEST
5) public static final int CENTER

constructors:-

- Border layout ()
- Border layout (int hgap, int vgap)

eg: import java.awt.*;
import javax.swing.*;
public class Border
{
JFrame f;
Border()
{

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

```
f= new JFrame();
JButton b1= new JButton ("NORTH");;
JButton b2= new JButton ("SOUTH");;
JButton b3= new JButton ("EAST");;
JButton b4= new JButton ("WEST");;
JButton b5= new JButton ("CENTER");;
 f.add (b1, BorderLayout.NORTH);
 f.add (b2, BorderLayout.SOUTH);
 f.add (b3, BorderLayout.EAST);
f.add (b4, BorderLayout.WEST);
 f.add (b5, BorderLayout.CENTER);
 f.setSize (300,300);
 f.setVisible (true);
}
Public static void main (String [] args)
 {
 New Border();
 }
 }
```

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

2) Grid layout

- The Java Grid layout class is used to arrange the components in rectangular grid
- one component is displayed in each rectangle

constructors:-

- Grid layout()
- Grid layout(int rows, int columns)
- Grid layout(int rows, int columns, int hgap, int vgap)

3) Flow layout

- The java flow layout, class is used to arrange the components in a line, one after another
- it is the default layout of the applet or panel

field of flow layout class

1) public static final int LEFT
2) public static final int RIGHT
3) Public static final int CENTER
4) public static final int LEADING
5) public static final int TRAILING

constructors:-

- flow layout()
- flow layout(int align)
- flow layout(int align, int hgap, int vgap)

eg:- import java.awt.*;
import javax.swing.*;
Public class flowlayout.example

JFrame

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

```
JFrame frameobj;
flow layout Example()
 {
 frame obj= new JFrame();
JButton b1= new JButton("1");
JButton b2= new JButton("2");
JButton b3= new JButton("3");
JButton b4= new JButton("4");
JButton b5= new JButton("5");
 frame obj.add(b1);
 frame obj.add(b2);
 frame obj.add(b3);
 frame obj.add(b4);
 frame obj.add(b5);
 frame obj.set Layout (new flow layout);
frame obj.set size(300,300);
 frame obj.set visible(true);
 }
public static void main(string [] args)
{
new flow Layout example();
}
}
```

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

Box layout

- The Java Box layout class is used to arrange the components either vertically or horizontally
- it is available in javax. swing package

Field of Box layout class

public static final int X. AXIS

public static final int Y. AXIS

public static final int LINE. AXIS

public static final int PAGE. AXIS

constructor:- Box layout (container (int axis);

eg:- import java. awt. *;

```
    import javax. swing.*;
public class Box layout example extends Frame
  {
  Button buttons[];
  Public Box layout Example ()
   {
   buttons = new Button [5];
   for(int i=0; i<5; i++)
   {
   buttons [i]= new Button ("Button"+(i+1));
   add ( buttons [i]);
    }
   set layout (new Box layout ( this. Box layout . Y. Axis))
```

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

```
setVisible(true);
}
public static void main(String args[])
{
BoxLayoutExample ex = new BoxLayoutExample();
}
}
```

5) Card Layout

- The Java CardLayout class manages the components in such a manner that only one component is visible at a time

Leena

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR

constructors:- card layout()
card layout(int hgap, int vgap)

common used method of card layout class

1) public void next(container parent)
2) public void previous(container parent)
3) public void first(container parent)
4) public void last(container parent)
5) public void show(container parent, String name)

6) Grid bag layout

- The java, Grid bag layout class is used to align components vertically, horizontally or along their baseline
- The components may not be of same size
- each Grid bag layout object maintains a dynamic, rectangular grid of cell
- each components associated an instance of Grid bag constraints

constructors:- Grid bag layout()

7) Group layout

- Group layout groups its components and place them in a contain hierarchically
- The grouping is done by instance of the Group class
- Group is an abstract class, and two concrete classes which implements this Group class are sequential Group and parallel Group

8) spring layout

- A spring layout positions the children of its associated container according to a set of constraints.

10/10

Nimisha

Dr. LEENA A V
PRINCIPAL
SREE NARAYANA GURU COLLEGE OF
ENGINEERING & TECHNOLOGY
PAYYANUR, KANNUR