	a	b
1	6	2
2	5	6
3	1	3

23/30

Verified by me.

12/13

G.V.P. COLLEGE OF ENGINEERING FOR WOMEN

Name: KORUKONDA DEDEEPYA Date: 09-05-2023

Roll No.: 20761A0550 Branch: CSE-1 Year 3

Sem II Subject Compiler Design

No of Addl. Sheets 02

Signature of Invigilator

Ans 1(a).

Basic block is a set of statements that are executed in a sequence.

The characteristics of Basic Blocks are:

1. It does not contain jump statements.
2. It will execute the sequence of steps one after the other as they appear.
3. It does not alter control flow.

Determining Leader of the block:

* Leader is the first statement of the block.
* A target statement to conditional or unconditional goto is also a Leader.
* The statement immediately after goto is a Leader.

Determing basic block/ main block:

* The statement from leader to the next leader is considered as a basic block. If the leader appears for first time it is called first leader.
* The block containing first leader is called as main block.

Matrix multiplication Flow graph:

```
Start
for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        for(k=0;k<n;k++){
            c[i][j] = a[i][k] * b[k][j];
        end k in for
    end j in for
end i in for
Stop.
```

```
Start
1  P=0.                    (L)
2  int i=0, j=0, k=0, n=4;
3  if i<n goto (4) else L9   L
4  L4: if j<n goto L5 else L8   L
5  L5: if k<n goto L7 else L7   L
6.  c[i][j] = a[i][k] * b[k][j];
7.  L7: end k
8.  L8: end j
9.  L9: end i
10. goto (3)
Stop
```

L: Leader statements.

be relocated.

* At compile time when a variable is declared or create space for the variables and store its data here.

Answer 1(b) Peephole optimization techniques:

* A peephole is a small window of active code.
* Peephole optimization involves replacing the short code with shorter/faster code.
* According to basic code generation strategy, there is a chance that the code we end up with will be redundant and less efficient.
* In peephole optimization, we will consider small blocks of code. and in each block evaluate the codes and search if any improvizations can be made. If we can improvise then we will do it and move to the next block of code.

Some of the peephole optimization techniques include:

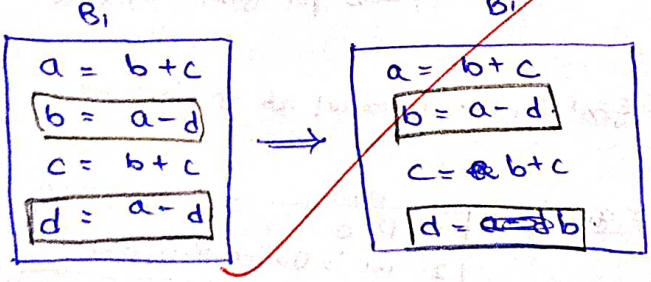
1. Redundant code elimination / subexpress° elimination
2. Dead code elimination.
3. Instruction selection 4. Renaming temporary variables.
5. Interchange the independent adjacent statements in a blocks of code.
6. Algebraic transformations.

In brief,

Redundant codes elimination ie, Redundant subexpression elimination:

Removing of the repetitive codes in sub expression

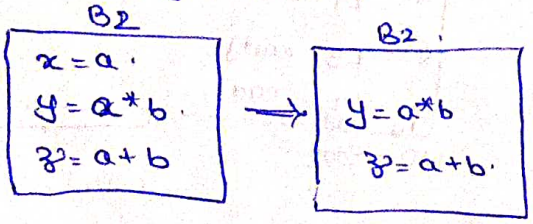
Eg:



Dead code elimination:

A variable is said to be live if it is subsequently used in the program. Otherwise it is dead. Removing dead codes is a way of optimizing program for better efficiency.

Eg:



x=a is dead here since no where else we are using x.

3. Instruction selection:

The space occupied by registers is crucial in comparision to mem. occupied space. Thus instructions are wisely selected and repetitive ones are ignored.

$X = a+b+c$.

```
MOV  a, R0
MOV  b, R0
ADD  a, b
MOV  X, R0
MOV  R0, X
ADD  c, X
MOV  X, R0
```

R0

a	
ab	R0
a+b	R0
X	R0
a+b	R0
a+b+c	X
X	R0

a+b+c

→

$X = a+b+c$

```
MOV  a, R0
ADD  b, R0
ADD  c, R0
MOV  R0, X
```

4. Renaming temporary variables:

to improve

It renames temp. variables for better understanding and code optimizability.

$t_1 = a + z$	$t_1 = a + z$
$t_2 = t_1 + y$	$t_2 = t_1 + y$
$t_3 = t_1 + t_2$	$t_3 = t_1 + t_2$
$t_3 = t_1$	$t_4 = t_3$

5. Interchange independent adjacent statements in a block.

$a = b+c$	$a = b+c$
$d = x-a$	$e = y-a$
$e = y-a$	$d = x-a$
$f = d-e$	$f = d-e$

6. Algebraic transformations:

The following form of statements can be ignored since there values don't really change

$x = x * 1$

$x = x + 0$

$x = x / 1$

be relocated.

* At compile time when a variable is declared we create space for the variables and store its data here.

Answer 2(a). Register allocation and assignment in target code generation:

The register memory is more efficient than the memory for normal use. Thus, the effective utilization of registers is a must.

Basic register allocation is a combination of 2 sub problems:

1. Register allocation.
2. Register assignment.

In brief,

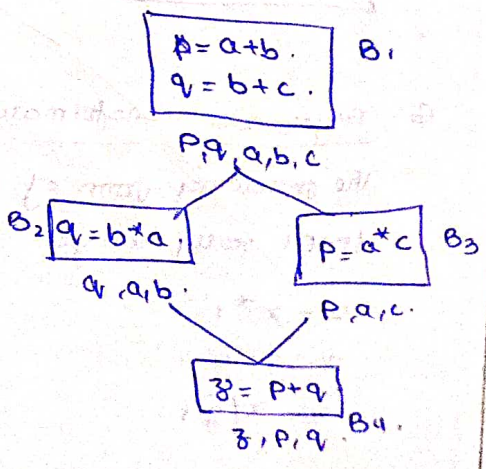
Register allocation: Registers are allocated to a set of variables.
Register assignment: Specific registers are assigned to variables.

Some strategies for efficient register allocation and assignment in target code generation are:

1. The specific values for register are stored specifically.
   1. Base addresses.
   2. Top of stack.
2. The global allocation of variables:
   1. If a variable is to be used throughout the program then it is allocated memory only once at the begining. So that everytime we make use of it, we don't have reallocate memory.
   2. Even in loops to we only initialize once, instead of initializing everytime. This makes the code more efficient.
3. use.
   ^ Count:

The value of for count of variable used is taken using the formula use(x, B) + 2* live(x, B) where x is the variable and B is the block.

Based on the results of use count we will determine register allocations to make use of least possible registers.



use(P, B1) = 1+2. =3.

use(q, B1) ....

Variable	B_1	B_2	B_3	B_4	Σ
p	1+2=3	0.	1+2*1=3	1+2=3	9.
q	1+2=3	1+2=3	~~1+2=3~~ 0	1+2=3.	9.
a	1+2=3.	1+2=3	1+2=3	0.	9
b.	1+2=3	1+2=3	0	0	6.
c	1+2=3	0	1+2=3	0	6.

Thus, register allocation is as b & c, p, q & a as..

b, c & p, q, a.

4. Loop optimization:

If we have two loops L_1 and L_2 nested. Say Loop L_2 is inside L_1 i.e.,

```
L1{
   L2{
   }
}
```

Then, whatever variable declarations are done for L_2 we can use the same for L_1 instead of reassign & creating register memories location again. This way we can make efficient use of register memory.

5. Through graph colouring:

In graph colouring we try to assign different colours to adjacent nodes using some principle we will optimize the code for efficient performance.

Answer 2(b) There are three storage environment allocation strategies:

1. static allocation
2. Stack allocation
3. Heap allocation.

In brief,

Static allocation:

* The variables are stored right at compile time.
* Once the variables are stored they are static fixed and won't be relocated.
* At compile time when a variable is declared we create space for the variable and store its data here.

* The drawback is it cannot be used for recursive procedures.

Stack allocation:

* In static allocation, if memory organization is based on stack data structures then it is called stack storage allocation.
* When the activation begins an activation record is pushed onto the stack.
* When the activation ends, the activation record is popped out of the stack.
* This follows the LIFO principle (Last In First Out principle).

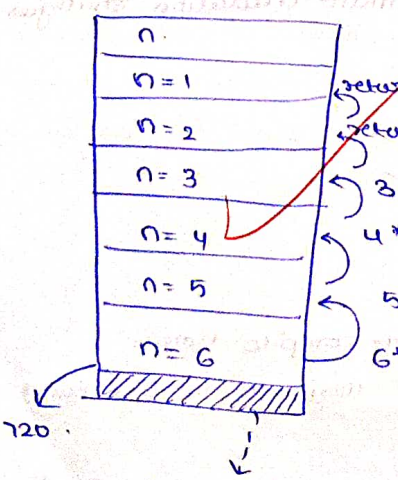
Heap allocation:

* Heap allocation is the most user friendly storage allocation.
* In heap allocation the data is dynamically stored as per the user requirements.
* This storage allocation is very flexible compared to static and stack storage allocations.
* The example for dynamic data is given below:

Program:

```
int fact(n)
{
  if (n<=1)
    { return 1;
    }
  else
    return n*fact(n-1);
}
```

Memory allocation: fact(6) = ?



(please ignore this space drawn by mistake).

12
16
72

Three address code forms analysis in Intermediate code generation for the expression $-(a*b)+(c+d)-(a+b+c+d)$.

Given, $-(a*b)+(c+d)-(a+b+c+d)$.

$t_1 = a*b$	$-(t_1)+(c+d)-(a+b+c+d)$
$t_2 = -t_1$	$t_2+(c+d)-(a+b+c+d)$
$t_3 = c+d$	$t_2+t_3-(a+b+c+d)$
$t_4 = t_2+t_3$	$t_4-(a+b+c+d)$
$t_5 = a+b$	$t_4-(t_5+c+d)$
$t_6 = c+d$	$t_4-(t_5+t_6)$
$t_7 = t_5+t_6$	t_4-t_7
$t_8 = t_4-t_7$	

Thus, three address code for given expression is generated.

Answer 3(b). Back patching in Intermediate code generation:

SDT for boolean expressing using back patching:

1. $E \rightarrow E_1$ or ME_2 { backpatch(E.falselist, M.quad);

E.truelist = merge(E1.true, E2.truelist);

E.falselist = E2.falselist; }.

M: Marker node.

2. $E \rightarrow E_1$ and ME_2 { backpatch (E.truelist , M.quad).

E.truelist = E2.truelist;

E.falselist = merge(E1.falselist , E2.falselist); }.

3. $E \rightarrow$ not E_1 { E.truelist = E1.falselist;

E.falselist = E1.truelist; }

4. $E \rightarrow (E_1)$ { E.truelist = E1.truelist;

E.falselist = E1.falselist; }

5. id1 relop id2. { E.truelist = id1 in (quad); E.falselist = id2 in (quad+1); }.

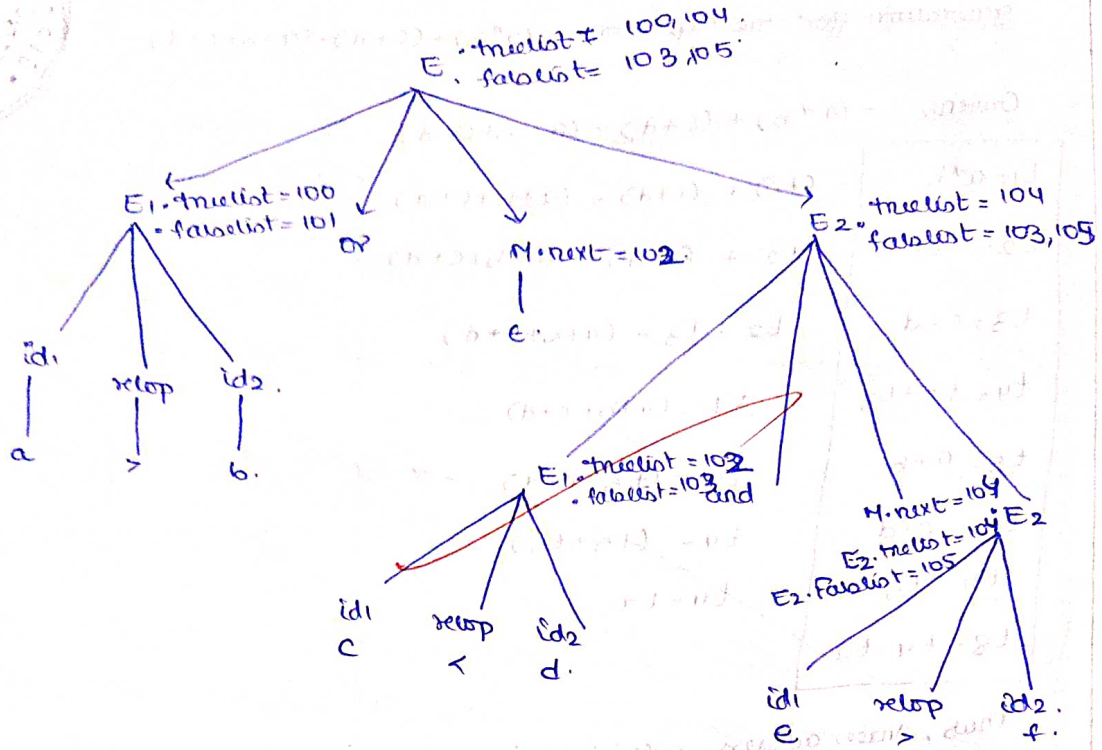
Gen (if 'id1' relop id2 'goto ___')

Gen ('goto ___')

$M \rightarrow \epsilon$.

5. M.next = M.nextquad().

Eg: a>b or c<d and e>f.



3. if a > b goto 102

goto 102.

if c < d goto 107

goto 103.

if e > f goto 106.

goto 104.

Final true and false locations are 106, 107 respectively.

There are 4 functions used in Back patching:

1. gen() To generate a list of statements.
2. merge(l_1, l_2) To merge lists l_1 and l_2.
3. backpatch(l, L) To place the three address code statements in list l inside the target label 'L'.
4. nextquad : To place the next quadruple value.

Verified by me
P. Sulumitha

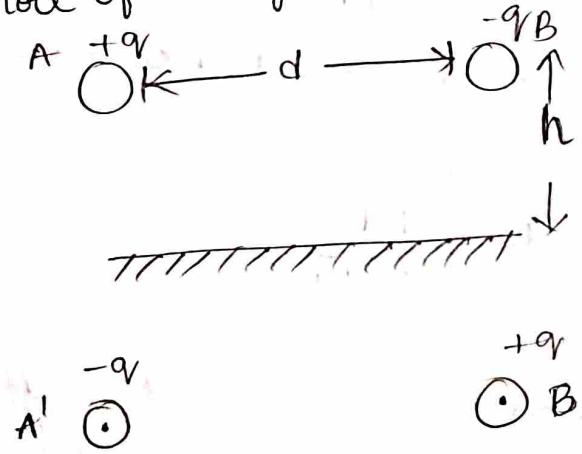
① -6
② -5
③ -6
17/30

G.V.P. COLLEGE OF ENGINEERING FOR WOMEN - J...
NAME: P. Sai Sulumitha DATE: 26-09-2022
Roll No.: 20JQ1A0224 BRANCH: EEE Year :
SEM: III - I SUBJECT: Power Systems 2
No of Addl. Sheets:
26/9/2022
Signature of Invig...

1(b) Effect of Earth on line capacitance:

Earth is a good conductor. Then, the effect of earth on capacitance of overhead lines is considered. Hence, there is a formation of capacitance between two conductors. In-addition to this, there is found to be the capacitance between the earth and the conductors. This can be improved by Method of Images.

Method of Images:-



Consider two conductors A and B, having charges +q and -q, and their images are A' and B' having opposite charges, at a distance of "h" from the ground. The ~~first~~ potential of charges are given by V_A and V_B.

$\Rightarrow V = L \frac{dI}{dt}$

$\therefore V = LIV$ (3)

G.V.P. COLLEGE OF ENGINEERING FOR WOMEN - J...

NAME: V. Anitha DATE: 26/9/22

Roll No.: 20TQ1A0231 BRANCH: EEE Year: III

SEM: I SUBJECT: Power systems 2

No of Addl. Sheets: ____

① -3
② -12
③ -6

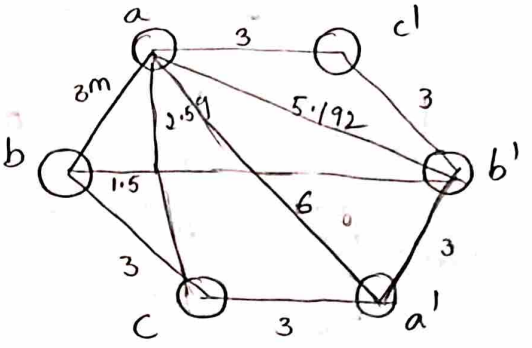
21/30

Verified by V. Anitha

26/9/20

1a) Given the distance between two conductors $d = 3m$

Diameter = 31.77 mm.



$$D_m = \sqrt[3]{D_{ab} D_{bc} D_{ca}}$$

$$D_{ab} = \sqrt[4]{D_{ab} D_{ab'} D_{a'b} D_{a'b'}}$$

$$= \sqrt[4]{(3)(5.192)(5.192)(3)} = 3.946$$

$$D_{bc} = \sqrt[4]{D_{bc} D_{bc'} D_{b'c} D_{b'c'}}$$

$$= \sqrt[4]{(3)(5.192)(5.192)(3)} = 3.946$$

$$D_{ca} = \sqrt[4]{D_{ca} D_{ca'} D_{c'a} D_{c'a'}}$$

$$= \sqrt[4]{5.18 \times 3 \times 3 \times 5.18} = 3.942$$

$$D_m = \sqrt[3]{3.946 \times 3.946 \times 3.942} = 3.944$$

$$D_{SA} = \sqrt[4]{(r' D_{a'a})(r' D_{aa'})} = \sqrt[4]{[0.7788 \times 0.03177 \times 6]^2}$$

$$= 0.385$$

$$D_{SB} = \sqrt[4]{(r' D_{bb'})(r' D_{b'b})} = \sqrt[4]{(0.7788 \times 0.03177 \times 6)^2}$$

$$= 0.385$$

---

through L_1 so there is no charging ... is no current through L_2. this process is commutative. After some time, there is a current through

G.V.P. COLLEGE OF ENGINEERING FOR WOMEN - -

NAME: L. Gayathri DATE:

Roll No.: 21JG5A0214 BRANCH: EEE Year :

SEM SUBJECT: PS-II

No of Addl. Sheets: 4 Signature of Invigilator

(1) -9
(2) -9
(3) -6

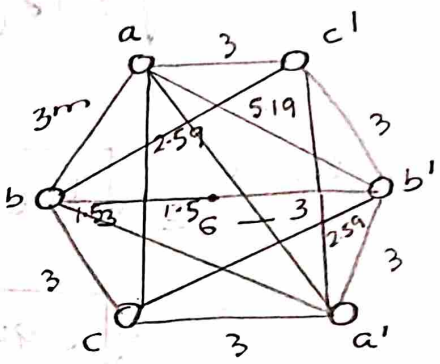
24/30

L. Gayathri

1(a) Given that

3-ϕ

double circuit has six conductors



side = 3m

diameter = 31.77 mm = 0.03177m

Radius = 0.01588 m

Find

(i) inductive reactance

(ii) capacitive reactance to neutral

if one of the circuit is removed

By the problem

line is completely transposed,

$$L/ph = 0.2 \ln\left[\frac{Dm}{Ds}\right] \quad mH/km$$

$$Dm = \sqrt[3]{Dab\, Dbc\, Dca}$$

$$Dab = \sqrt[4]{ab(ab')(a'b)(a'b')}$$

$$= \sqrt[4]{3(5.19)(5.19)(3)} = 3.94$$

30/30

P. Sulumitha
Verified by me

G.V.P. COLLEGE OF ENGINEERING FOR WOMEN

Name: P. Sai Sulumitha Date: 7-3-23

Roll No. 20JQ1A0224 Class: EEE Year: III

Sem: II Subject: EMI 4

No of Addi. Sheets: _____

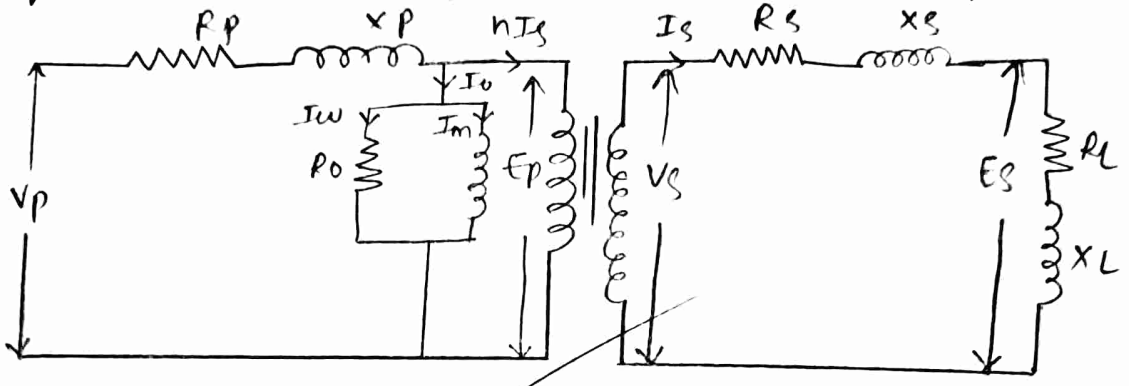
12 + 12 + 6

Signature of Invigilator

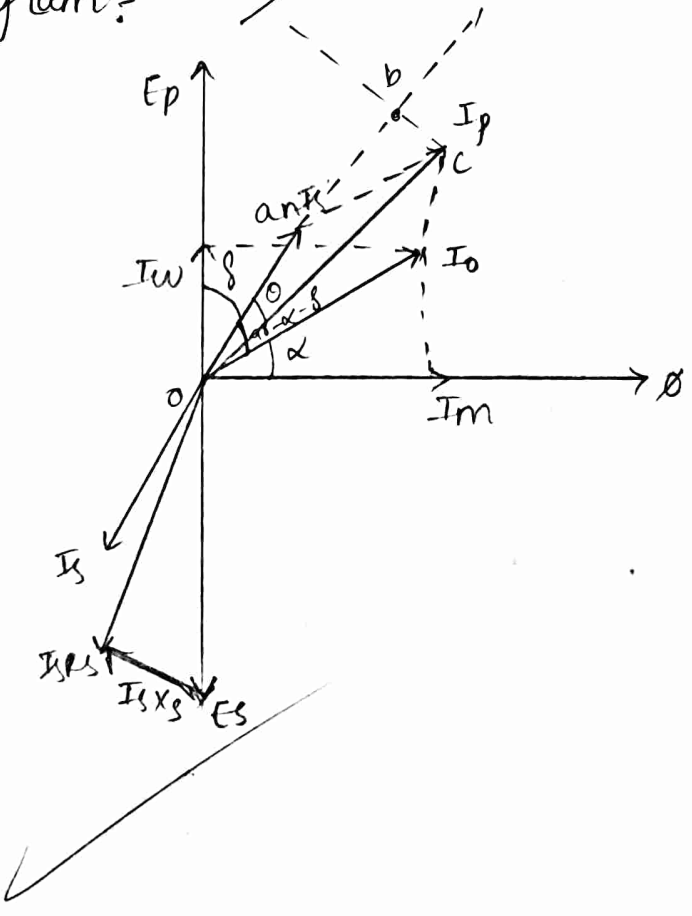
P. Sec 14/3/23

1a) Expression for the current Transformer :-

Equivalent circuit for the current transformer :-



Phasor Diagram :-



28/30

P.S. 13/3/23

G.V.P. COLLEGE OF ENGINEERING FOR WOMEN

Name. A. N.V. varalakshmi Date: 7/03/2023

Roll No. 20JG1A0201 Branch: EEE Year III

Sem 2nd sem Subject Electrical Measurements and Instrumentation

No of Addi. Sheets 3

11 + 11 + 6 = 28 Verified by me A. Varalakshmi Signature of Invigilator ASRao

① b) current Transformer

Given $N_S = 300$

$r_s = 1.0\,\Omega$

$x_s = 1.5\,\Omega$

$I_s = 5A$

MMF = 100 AT

$I_w = 40A$

$n = \frac{N_s}{N_p} = \frac{300}{1} = 300$

Burden Impedance = $\sqrt{(1.5)^2 + (1.0)^2}$

$= 1.8$

$\cos\delta = \frac{1.5}{1.8} = 0.833$

$\sin\delta = \frac{1.0}{1.8} = 0.555$

$E_s = 5 \times 1.8 = 9.0$

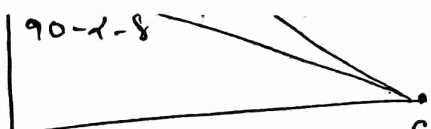
$E_p = \frac{E_s}{n} = \frac{9.0}{300} = 0.03.$

$I_w = 40A.$

$I_m = \frac{MMF}{\text{No. of ~~secondary~~ primary turns}} = \frac{100}{1} = 100.$

$R = n + \frac{I_m \sin\delta + I_w \cos\delta}{I_s}$

$= 300 + \frac{100 \times 0.555 + 40 \times 0.833}{5}$



29/30

G.V.P. COLLEGE OF ENGINEERING FOR WOMEN

Name: P. Shyamili Date: 7-3-23

Roll No. 20JG1A0222 Branch: EEE Year: III rd

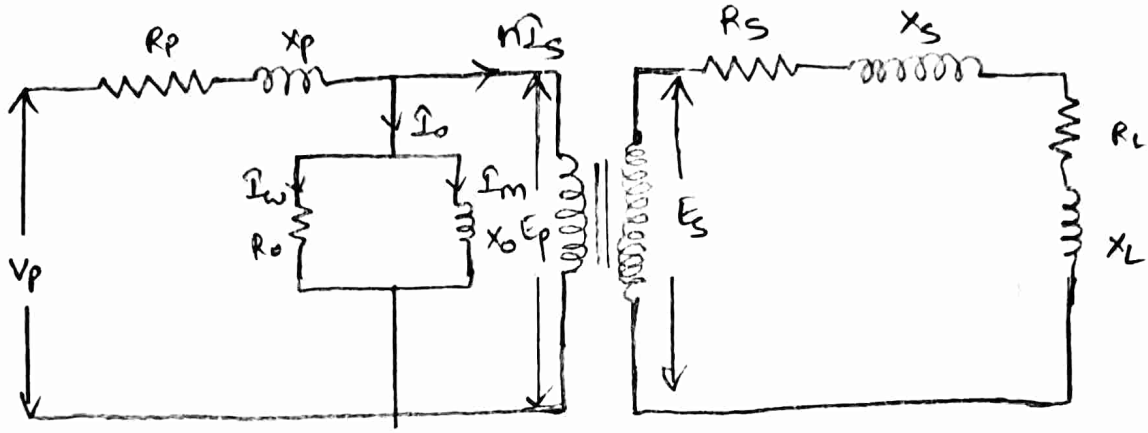
Sem: 1st Mid Subject: Electrical Measurement and Instrumentation

No of Addl. Sheets: 4

12+1+6 Verified by Signature of I/C 7/3/23

I Answer all the Questions P. Shyamili Max Marks: 30.

1a) Current transformer:-



In this current transformer there are having.

Rp = Primary Side Resistance

Xp = Primary Reactance

Rs = Secondary Resistance

Xs = Secondary Reactance.

Io = No load current

Ro = No load Resistance

Xo = No load Reactance

Iw = Working current

Im = Magnetizing Current

Is = Secondary Side current

Ip = Secondary Side current.

Is

G.V.P. COLLEGE OF ENGINEERING FOR WOMEN

Name: M.M.L. Prasanna Chandra Date: 21-03-23

Roll No.: 21JG1A1231 Branch: IT Year: II nd

Sem: II nd Subject: PSE

No of Addl. Sheets: 2

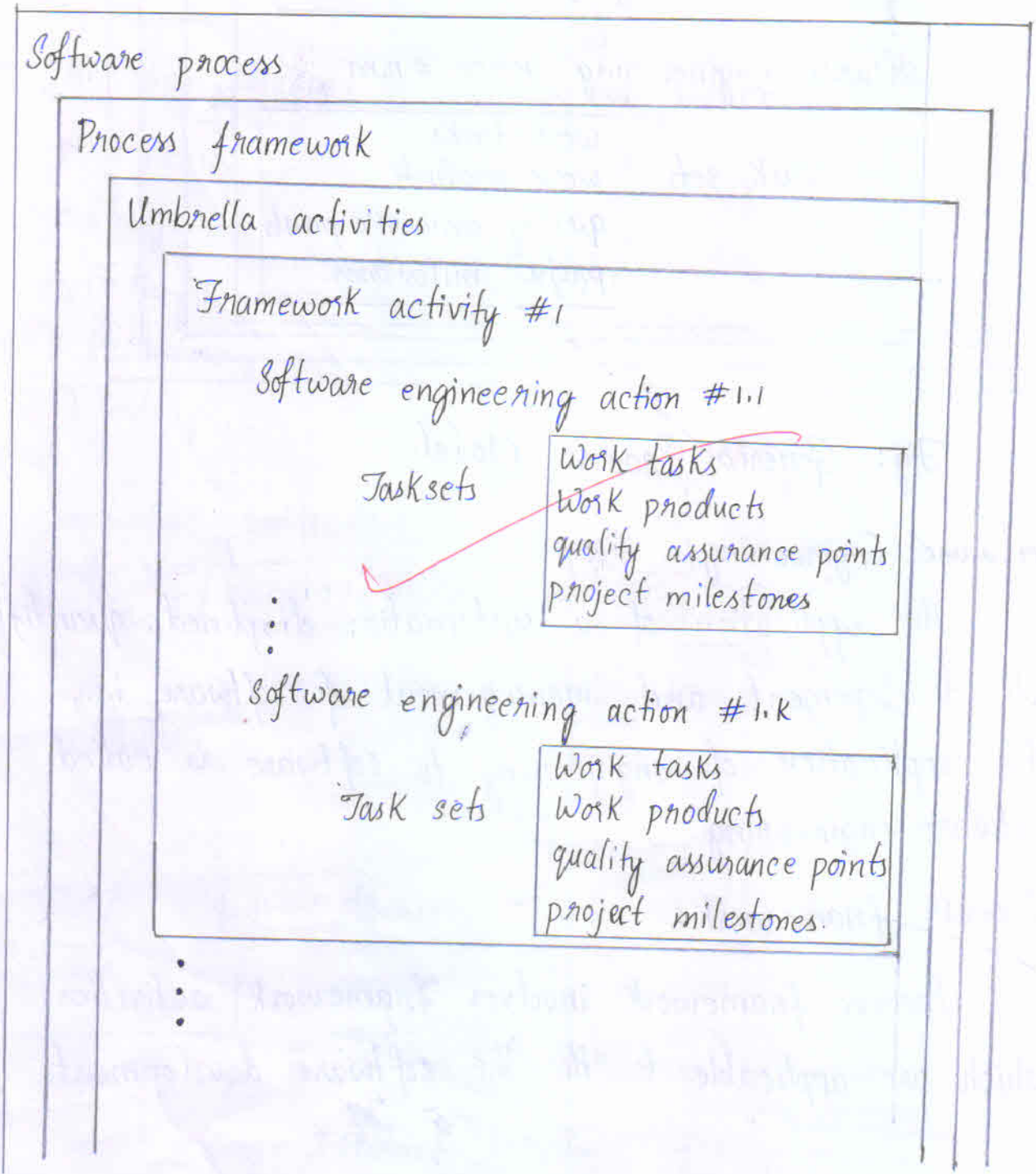
Signature of Invigilator

11 + 12 + 5 = 28/30 (14)

Verified by me Prasanna 06/04/23

1.a) Generic Process Model:

A.



[PTO]

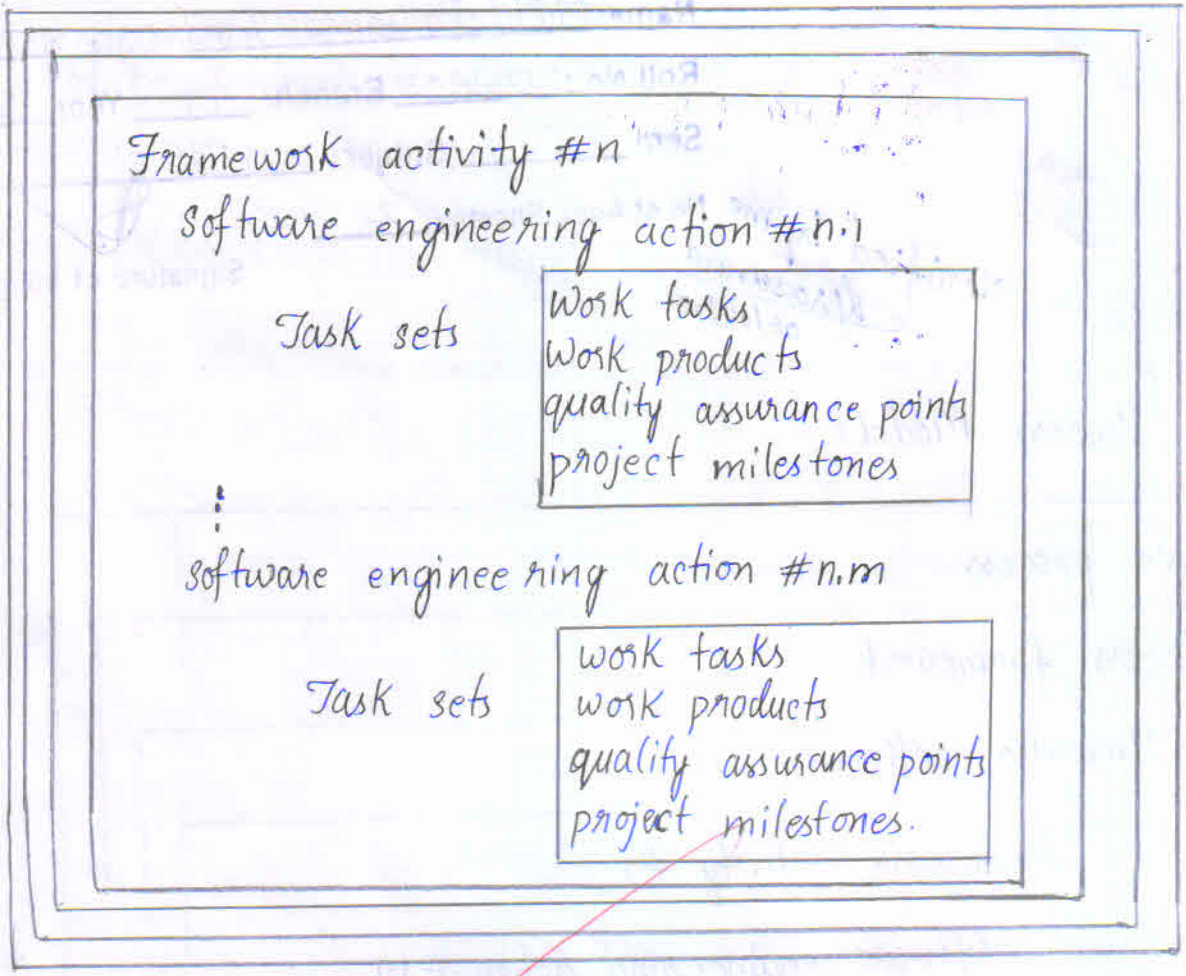


Fig: Generic Process Model.

Software Engineering:

The application of a systematic, displined, quantifia-ble developement and management of software i.e., the application of engineering to software is called software engineering.

*Process framework:

Process framework involves framework activities which are applicable to the the software development.

2,a) A) Extreme programming:

Extreme programming is denoted by XP.

→ XP involves to the developement of framework activities in the software development.

→ XP programming includes commincation, simplicity, repeat ...

→ The key concepts of extreme programming are:

- Planning
- Design
- Coding
- Refactoring
- Testing.

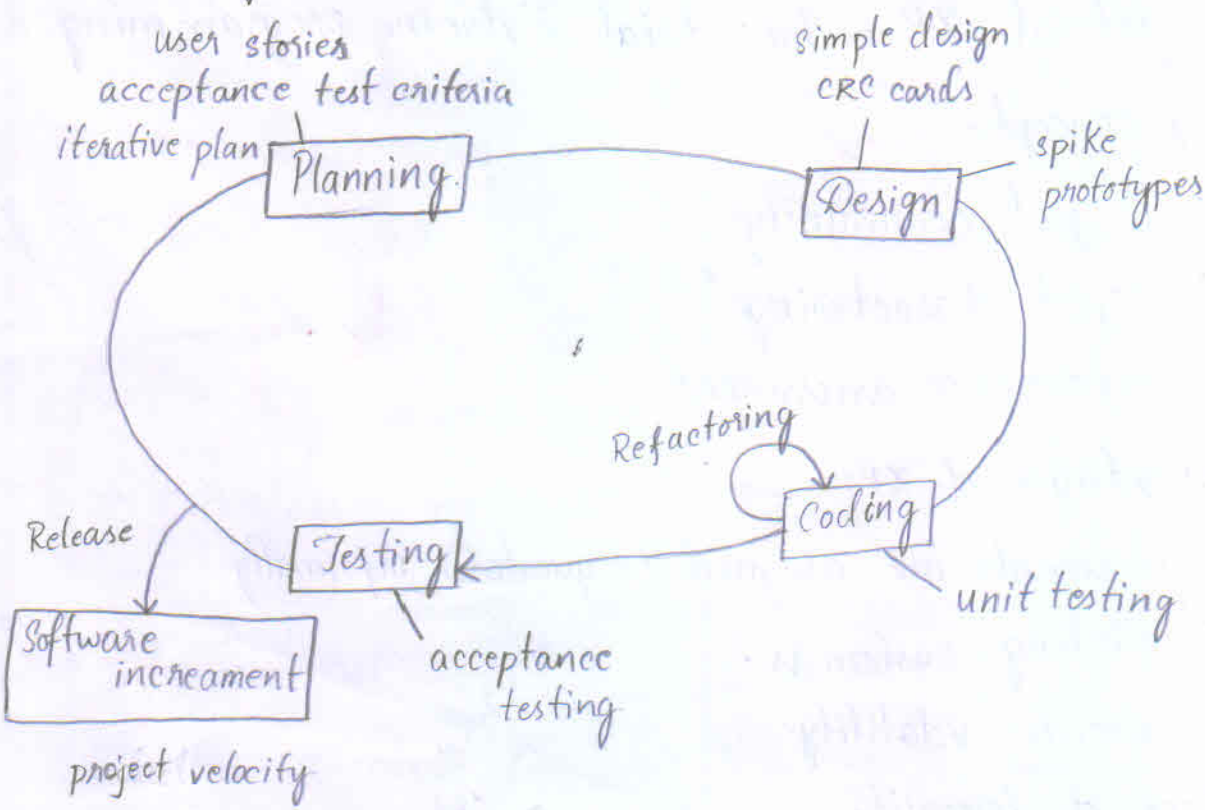


Fig: Extereme Programming.

1) Project velocity is described as the number of customer stories involved during the first release.

2) Extreme programming also includes CRC cards.
[class - relevant collaboration].

3) Refactoring:

This refers to the software developement internally. At the part of coding, the internal structure is development but the cycle is not disturbed externally.

→ Development of design takes place until a refined product is released.

Industrial XP:

→ Industrial XP - Industrial Extreme programming

→ Key concepts:

- Project community.
- Project chartering.
- Performance assessment.

Disadvantages of XP:

→ Requirements are assigned (requested) informally.

→ Conflicting customers

→ Requirents volatility.

→ Lack of formality.

2,b) Requirements Engineering:

A) As per the customer requirements, these functions are categorised into three domains:

1) Functionality domain.
2) Information domain
3) Behavioural domain.

The board spectrum of tasks and techniques that are implemented to meet the requirements of software development is called Requirements Engineering.

Elements of requirement engineering:

→ Scenario-based elements:

The elements which are implemented based on the user's view.

→ Class-based elements:

The set of objects of a class define the funtionality of an actor.

→ Behavioural elements:

Based on the activity performed or the role of the element.

→ Flow-oriented elements:

The elements refers to the transition of the process flow in software developement.

3, Scenario - Based Modelling:

A, As per the flow of activities which are represented in detail, the scenario of the elements is described.

→ One of the example of scenario - based modelling is activity diagram.

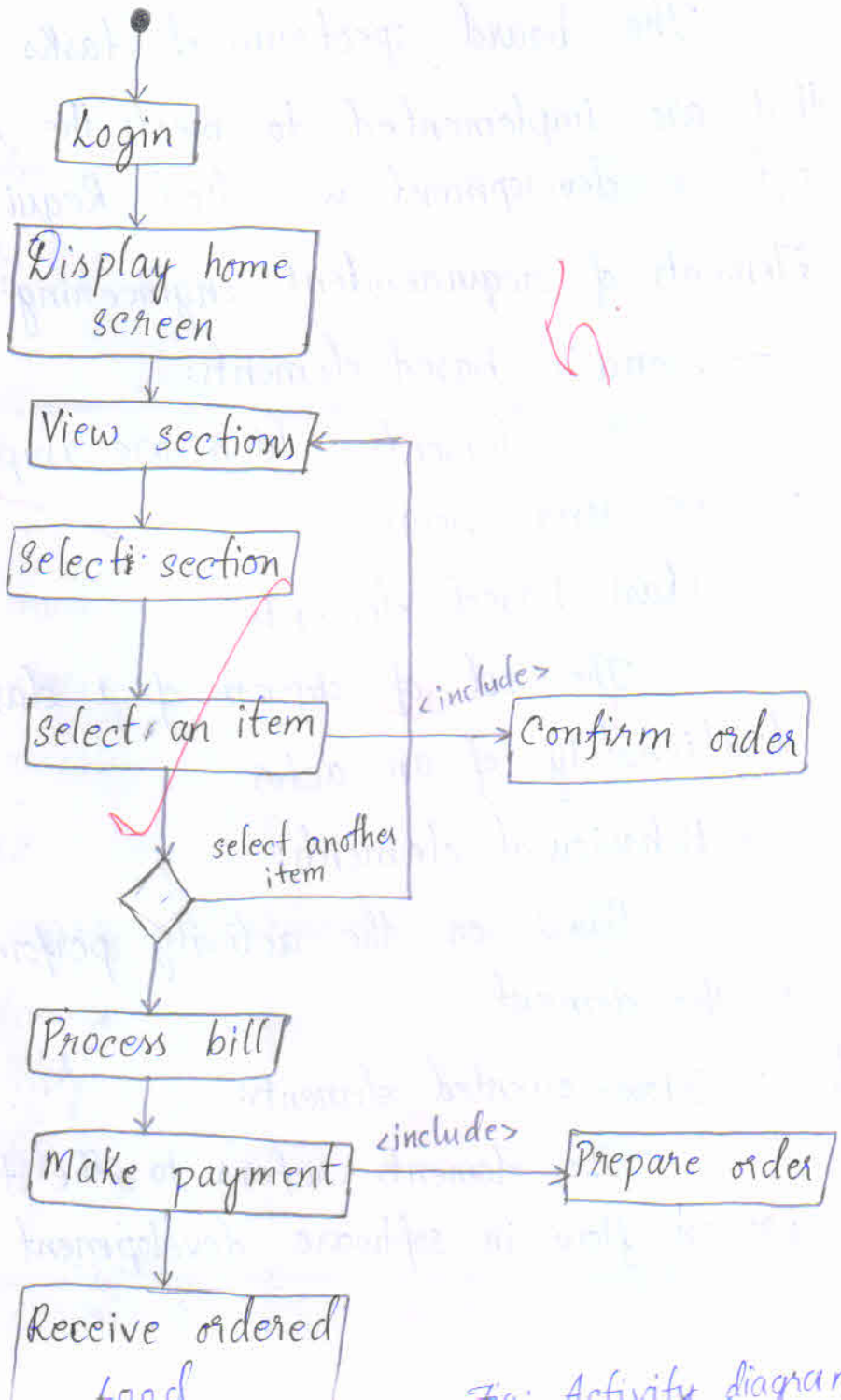


Fig: Activity diagram

1,b) Unified Process:

A,

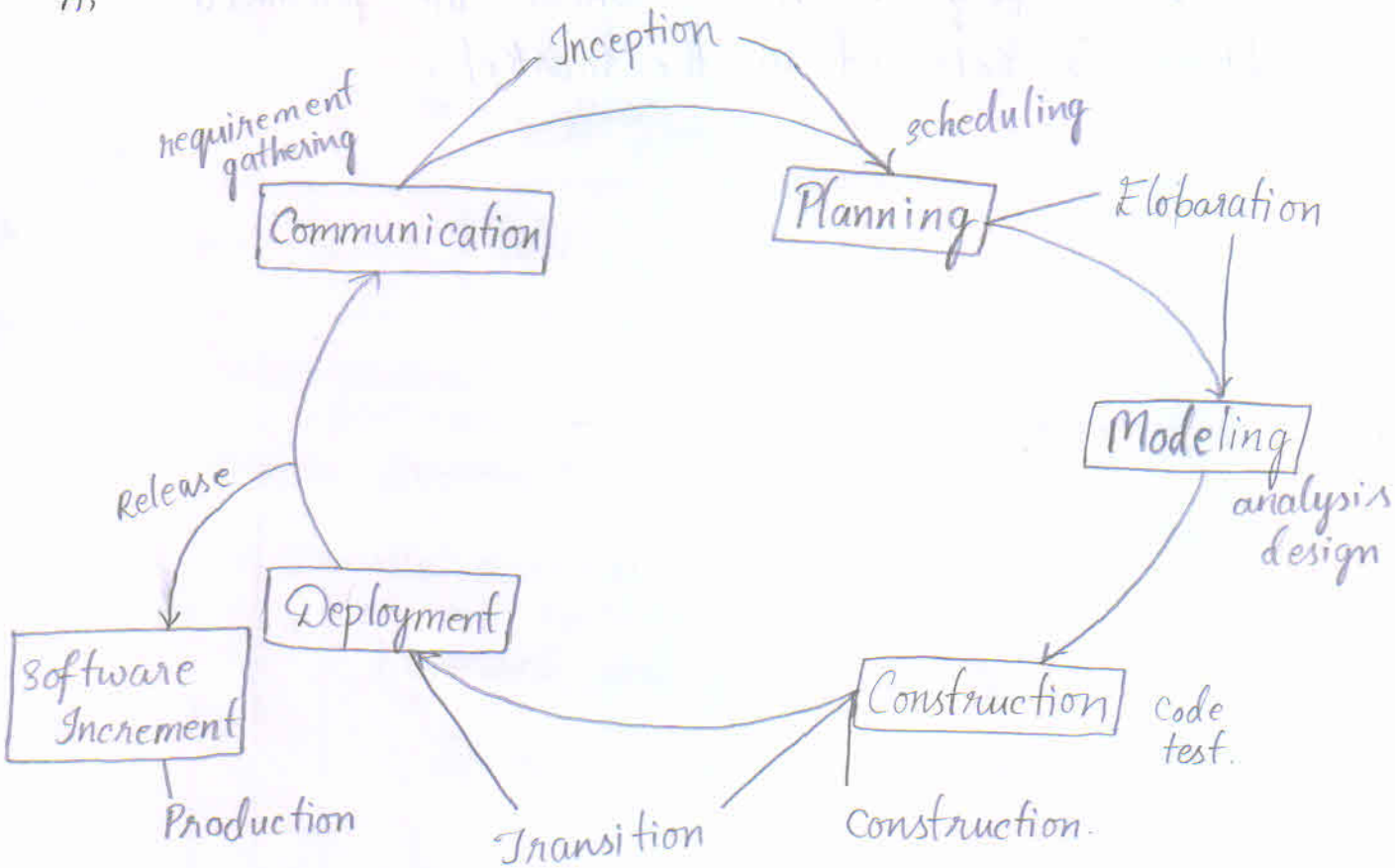


Fig: Unified Process.

1) Inception:

Requirement gathering for the software and planning for it.

2) Elobaration:

Planning and modeling for the software developement.

3) Construction:

Elobarated software is constructed (coded) and tested.

4) Transition:

5) Production :

The deployment state where the produced software is released in the market.

G.V.P. COLLEGE OF ENGINEERING FOR WOMEN

Name: S. Likhitha Date: 21-3-23

Roll No.: 21JG1A1255 Branch: IT Year 2

Sem 2 Subject Software Engineering

No of Addl. Sheets ____ Signature of Invigilator 21/3/23

12 + 11 + 4. (14)

Verified by 27/30

1. (a) 1. A process is a collection of work activities, actions and tasks that are performed to create a work product.

Generic process Model :

→ The Software process of a generic model is shown in the Systematic representation below.

→ The Software process consists of the process Controll of Generic process model.

→ The Requirements are then in the Umberella activities in the process controll.

→ The Umbrella activities Consits of the frame work activities which contains the task sets that define

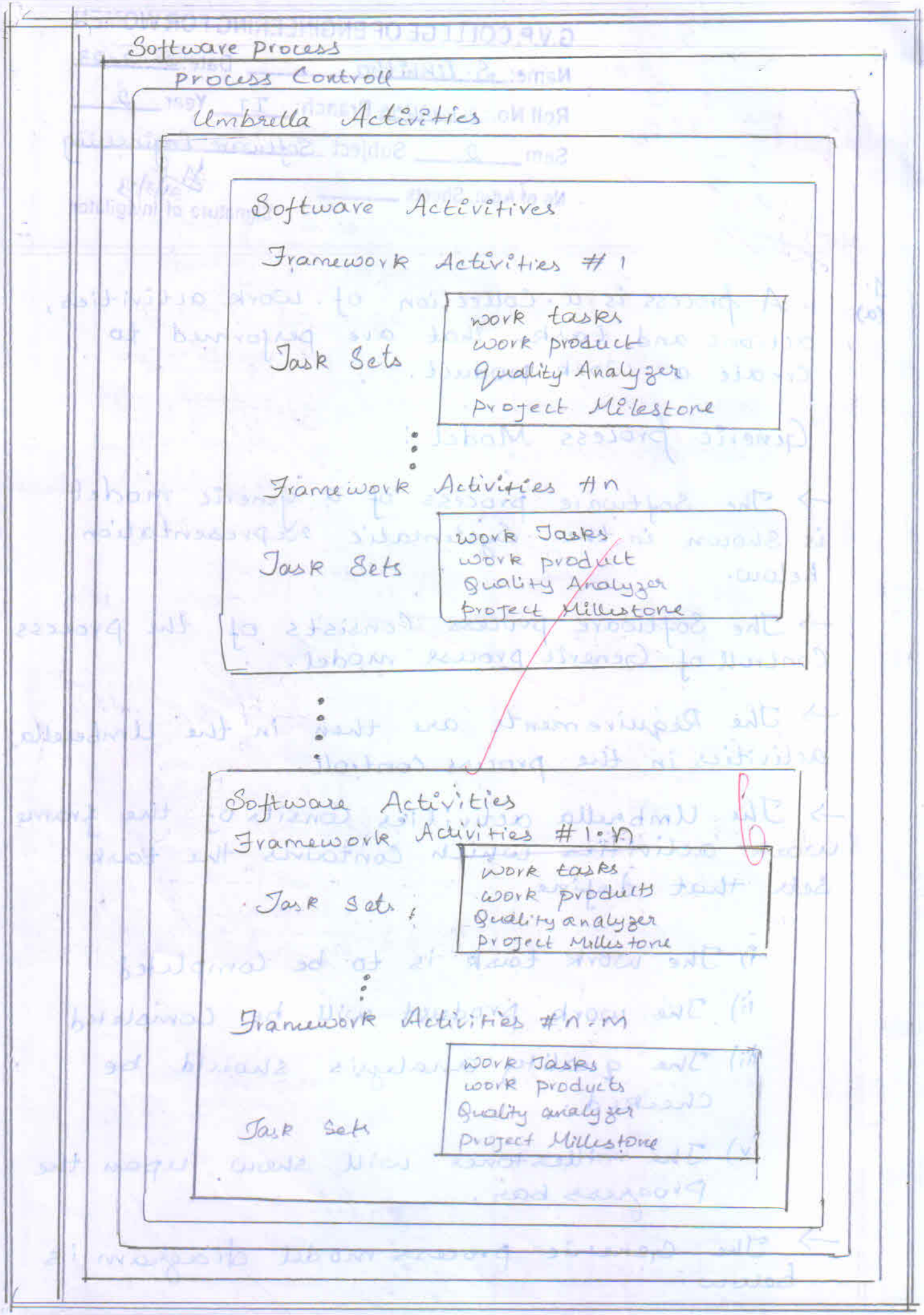
i) The work task is to be completed

ii) The work product will be completed

iii) The quality analysis should be checked.

iv) The milestones will show upon the Progress bar.

→ The Generic process model diagram is below:


Software Process
Process Controll
Umbrella Activities
Software Activitives
Framework Activities # 1
Task Sets
work tasks
work product
Quality Analyzer
Project Milestone
Framework Activities #n
Task Sets
work Tasks
work product
Quality Analyzer
Project Millestone
Software Activities
Framework Activities # 1.m
Task Sets :
work tasks
work products
Quality analyzer
Project Millestone
Framework Activities #n.m
Task Sets
work Tasks
work products
Quality analyzer
Project Millestone

1. (b)

→ The Unified process Model is about the Comunication between the end users.

→ It emphasises about the Communication.

→ The Unified Model will deal majorly about Communication.

→ It also deals about planning and Modeling. The Unified model role is of Software deployment.

→ It also plays a roles in planning and constructing the respective Software for the End User.

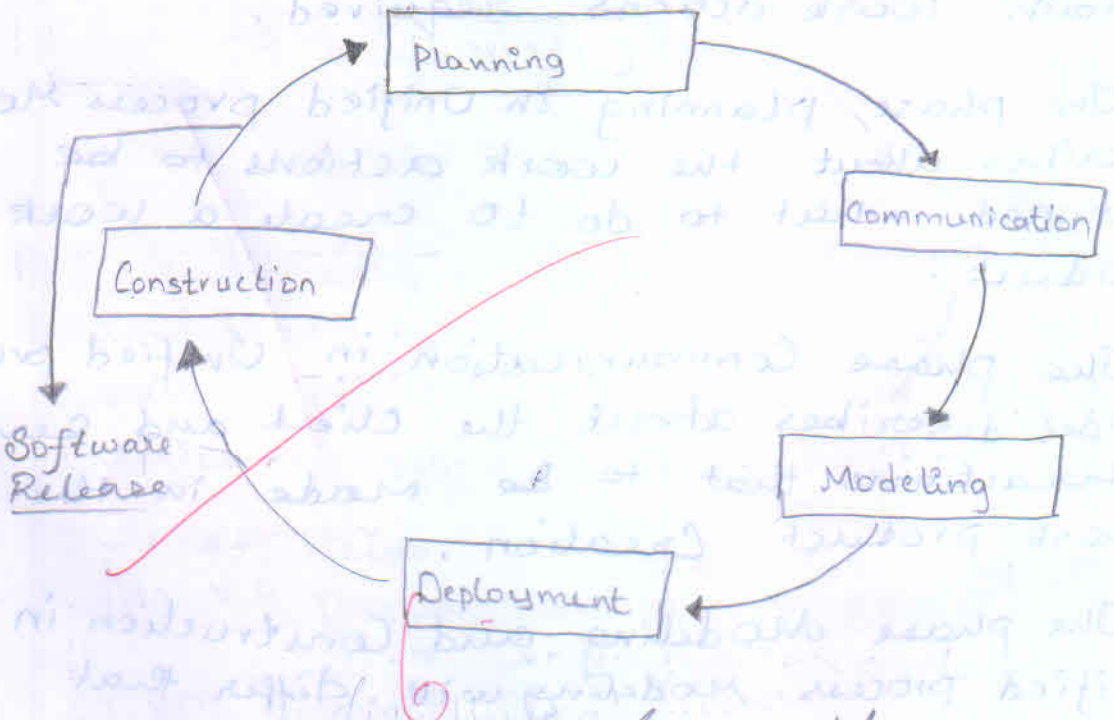


fig: Unified process Modell

→ A process is a Collection of work activities, actions and tasks that are performed to Create a ~~web page~~. work product

→ Their are 4 major phases for any Creation of a work product like planning, design, Coding and Testing.

→ But whereas in Unified process Model we have 5 major phases where a work product is to be Created as planning, Communication, deployment, Modeling, Construction.

→ In Unified process Model Communication and Modeling plays a major role where a work product has to be Created with Certain work actions required.

→ The phase planning in Unified process Model describes about the work actions to be Planned what to do to create a work product.

→ The phase Communication in Unified process Model describes about the client and server interactions that to be made in the work product Creation.

→ The phase Modeling and Construction in Unified process Modeling will differ that the work actions and activities or tasks that need to be take place in the Creation of a work product.

→ At the End Finally it goes with the phase of deployment which clicks upon the Creation of work product.

2. (a) # Extreme Programming (XP):

→ Extreme programing is also known as XP.

→ The XP is related with Object Oriented programming also.

→ The XP has a Value, Process and 9XD which is the origin of XP.

→ Their are 3 States:

- → XP Value
- → XP Process
- → Industrial XP

1) XP Value:

The Extreme programing has a Base value which is fundamental fundation of set of values of the XP. Their will be a value which is used in this Extreeme programming. The XP Value is choosen as a driver in the XP programming.

2) XP Process.

The XP process is the fundamental set of process from Extreme programming. The XP process Involves some phases like planning, design, Coding and testing.

Systematic diagram to represent some important phases of XP.

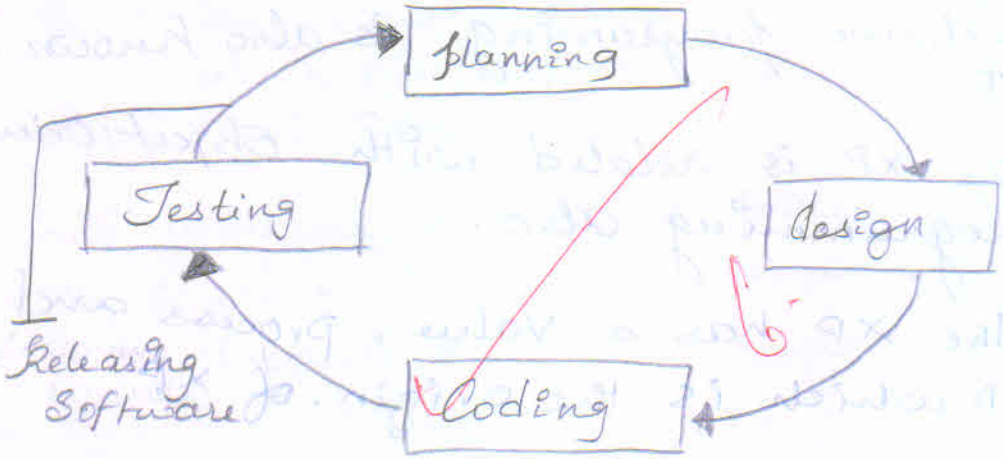


fig: XP process

3) Industrial XP:

The Industrial XP is also known as IXP. The IXP is a Origin of the XP. The origin of XP is said to be IXP.

The Industrial XP will differ about the Coorporate databases and Bussiness applications. The Industrial Extreme programing will also involve in phases of Extreme programming.

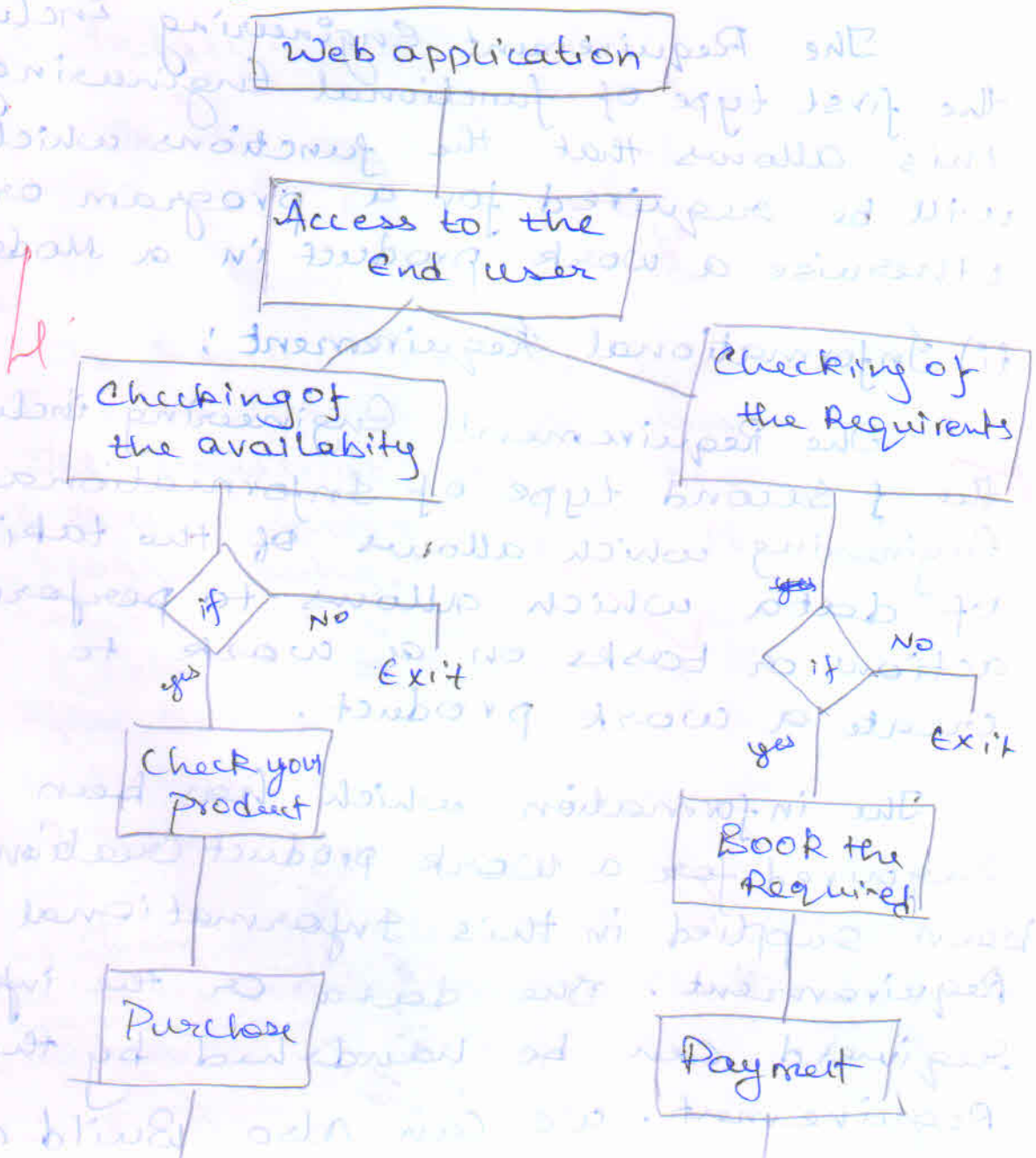
Extrememe is programing is a object-~~Oriee~~ Oriented programing so that it will have this IXP Industrial Extreme programing)

→ Their will also be some phases like Radio activeness, project charging, project Bahaancer etc...

→ The Role of Extreme programing is ~~the~~

3. Scenario - Based Modeling :

The Web apps will be always depended upon the Modeling. Their are many diffent Modelings but this Scenario - Based Modeling is depended upon the situation. It will be depended on the End User. According to the input of the End User the modeling will be working upon. So. it is known as Scenario - Based Modeling.



24/30 = 12/15

Verified by me
A. Bhavana

a	b	c
7	11	6

G.V.P. COLLEGE OF ENGINEERING FOR WOMEN

Name: A. Bhavana Date: 23/3/23

Roll No.: 21761A0424 ECE Year II nd

Sem II Subject AC

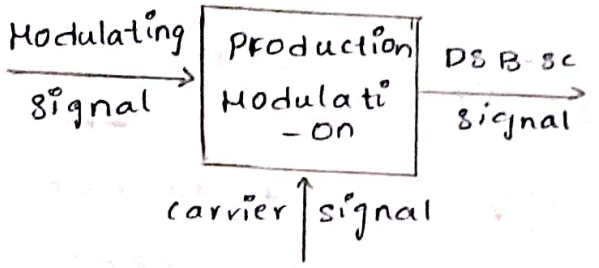
No of Addl. Sheets (2)

Signature of Invigilator

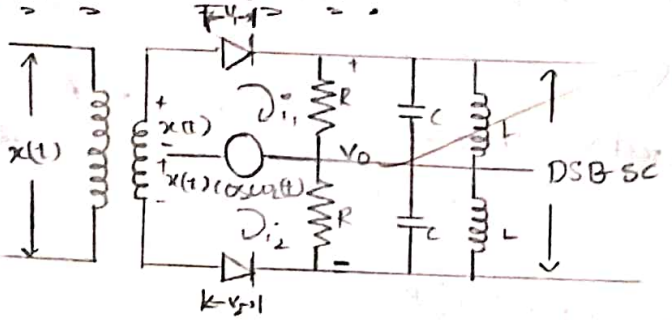
2a) Ans

Generation of DSB-SC signal:-

DSB-SC : Double side Band Suppression carrier



Balanced modulator:-



voltage across the resistors

$$-V_0 + i_1 R - i_2 R = 0$$

$$V_0 = (i_1 - i_2) R$$

→ To find the i_1 & i_2 currents in the circuit we use Square law modulation technique. i.e $I = a + bV + cV^2$ 'V' is nothing but the total voltage i.e voltage across the diode $D_1[V_1]$ and voltage across the diode $D_2[V_2]$

case(i): If the diode 1 [D_1] is ON, $V_1 \geq 0$

i.e $i_1 = AE$ i.e $i_1 = A\cos\omega_c t + x(t)$

case(ii): If the diode 2 [D_2] is ON, $V_2 \geq 0$

i.e $i_2 = A\cos\omega_c t - x(t)$

the frequency ...

→ The frequency signal should not combined

27½/30 = 14/15

verified by me
G. Naga Sai Vani
12-12-2022

G.V... COLLEGE OF ENGINEERING FOR W... EN - JU

NAME: G. Naga Sai Vani DATE: 18-11-2022

Roll No.: 21JGIA0441 BI...NCH: ECE-I Year II nd Year

SEM 2-1 SUBJECT: Random Variables and Stochastic Processes

No. of Addl. Sheets: 2

27½/30

1) 6 6
2) 3½ 6
3) 6

1(a)

(i) **Discrete Sample Space:-**

If the number of outcomes in the sample space are countable then the Sample space is called the Discrete Sample Space.

Eg:- Pack of Cards

$n(S) = 52$

(ii) **Conditional Probability:**

If A and B are two joint events, the occurence of event A given event B is called the Conditional Probability

$$P\left(\frac{A}{B}\right) = \frac{P(A \cap B)}{P(B)}$$

(or)

$$P\left(\frac{A}{B}\right) = \frac{P(AB)}{P(B)}$$

Eg:- Given that you drew a red card, the probability of occurence of four

$$P\left(\frac{\text{four}}{\text{red}}\right) = \frac{2}{26} = \frac{1}{13}$$

(iii) **Continuous Random Variable:**

If all the outcomes of the Sample Space are

2) $F_{XY}(\infty, \infty) = 1$

Proof:- The Joint CDF of X, Y is $F_{XY}(x, y) = P(X \le x \cap Y \le y)$

$F_{XY}(\infty, \infty) = P(X \le \infty, Y \le \infty)$