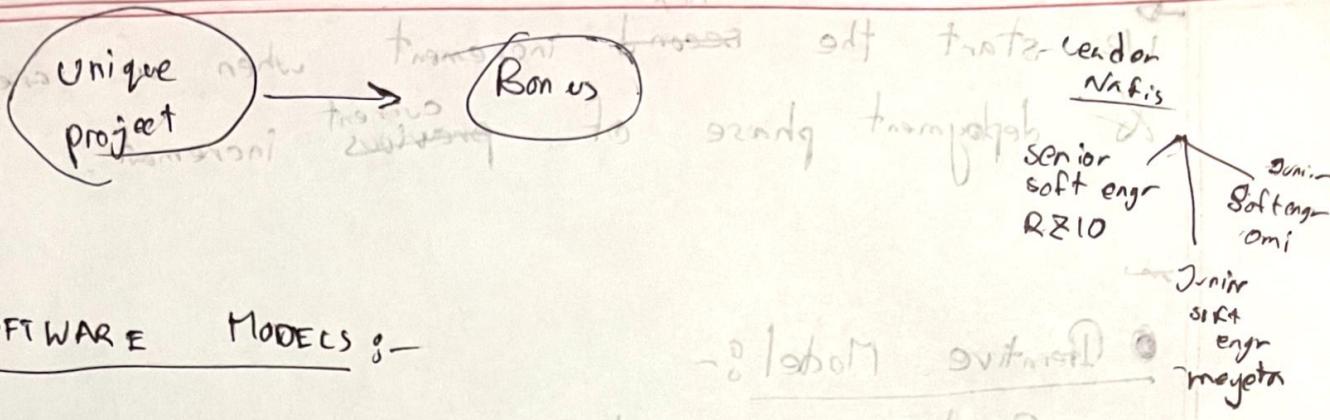

Prepared and Written By
RAZEEN AHMED

CSE-470
[SOFTWARE ENGINEERING]

HANDWRITTEN NOTE

Computer Science and Engineering
BRAC University

Github: github.com/razeen
LinkedIn: linkedin.com/in/razeenahmed



SOFTWARE MODELS :-

(i) Waterfall Model :-

- static → not suitable for mass level project.

- requirement cannot be added later.

- Not suitable for dynamic softwapp.

* Scenario based q. → which model you will choose in this scenario.

V-Model

#Evolutionary process model :

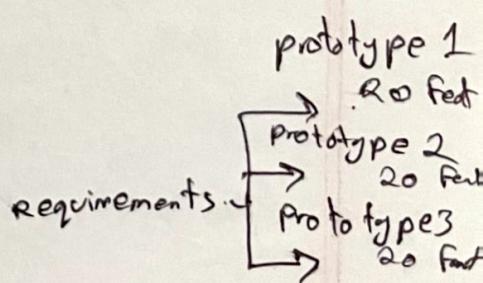
- Combination of iterative & incremental

Incremental
Iterative Model

- major requirements are fixed

- provides the user small blocks

- suitable for small & medium projects.



17/01/80

→ We start the second increment when we are const. in
next & deployment phase of current previous increment.

① Iterative Model :-

- Customer is not sure about the requirements.
- Quality of the features is refined with time.
- suitable for mass level project.

Evolutionary Model :-

- Can change requirements at any time.

Isom - V

Evolutionary Model :-

Iteration & iteration To maintain compatibility

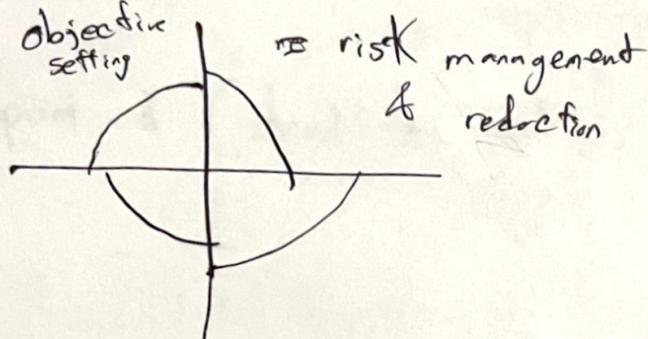
short subcycle

exit the streamer system

internal loop now off schedule

SPIRAL MODEL :-

- Unlimited Budget.
- Huge risks.
- Every step analysis of risk.
- No limitation to time.



CMMI → Capability Maturity Model Integration

Process & behavioral model

Postman
Docker
Apache

} Tools

Application layer back plan from

and tubing of prime of in access

Layer

Why we need Agile?

→ Frequent changes

→ ~~base~~ backup and quick response → respond to change swiftly.

Agile are three type :-

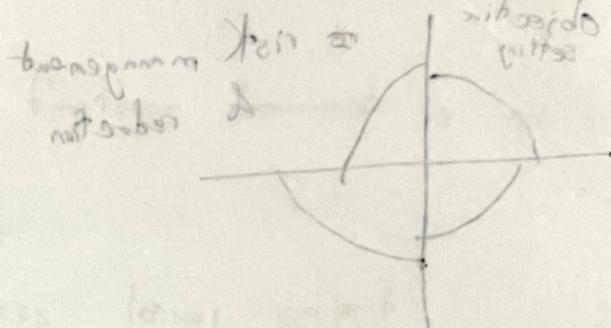
(i) Extreme Programming

(ii) XP

(iii) Pair Programming.

(iv) S

• ~~Complex gets easier~~
• ~~Metamorphosis~~
• ~~Evolution~~



AUP Phases

(i) AUP Inception Phases

(ii) Elaboration

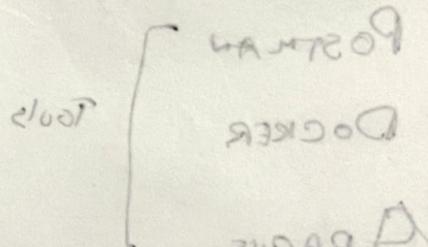
(iii) Construction

(iv) Transition

AUP Discipline:-

& Configuration management

Project management



Scrum:-

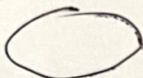
most widely used agile methodology -

Product

Backlog To arrange the product by its priority from top to bottom

Use Case Diagram

4 elements:-

- (i) system 
 - (ii) Actors 
 - (iii) Use case 
 - (iv) Relationships:-
- (a) Primary actor
 - (b) Secondary actor
 - (c) Association (connection)
 - (d) include (must)
 - (e) Generalization (subdivision)
 - (f) extend (not always)

sprint - 1 (deadline - 10th July)

Iterative Model

- (i) Customer is not sure about the requirements.
- (ii) Customer wants prototype version of the software (from the beginning of project)
- (iii) Better for small projects (million \$ to billion \$)

Incremental Model:-

- iterations P

method (i)

protoA (ii)

- (i) customer is well known about the requirements.
- (ii) Software is divided into fixed no. of increments to be delivered to customer.
- (iii) Customer wants prototype version of the software (from the beginning of the project).
- (iv) Major changes not welcomed.
- (v) Increments needs to be prioritized by customer feedback.
- (vi) Better for small & medium.

Iterative Model:-

- (i) Customer is not sure about the requirements.
- (ii) Even the development team may not be sure (about which technology or algorithms might be needed).
- (iii) Time is unlimited.
- (iv) Tools & requirements not identified yet.
- (v) Better for mass level project.
- (vi) Quality of features are prioritized (and refined with time).
- (vii) Customer feedback (with repetitive iterations) are welcomed.

Evolutionary Model:-

- (i) Can change requirement anytime
(ii) Customer involvement & taking feedback to make more efficient & reliable
(project may expand beyond its initial scope)

Spiral Model:-

- (i) risk driven project model.
(ii) long term project commitment & budget.
(iii) Users/developer not sure about the requirement.
(iv) complex requirements.
(v) significant changes are expected.
(vi) work in small & early top base

CMML

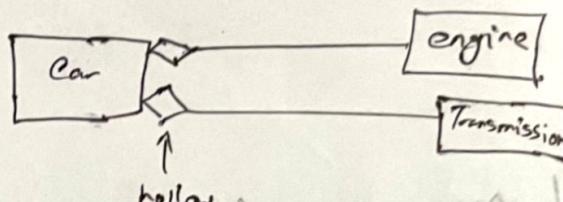
- (i) A process & behavioral model
(ii) Helps organizations streamline process improvement and encourage productive, efficient behaviors that decrease risks in software maturing database management system
(iii) It measures the maturity of the organization
(iv)

Aggregation relationship:-

words like "is a part of" or "is made up of".

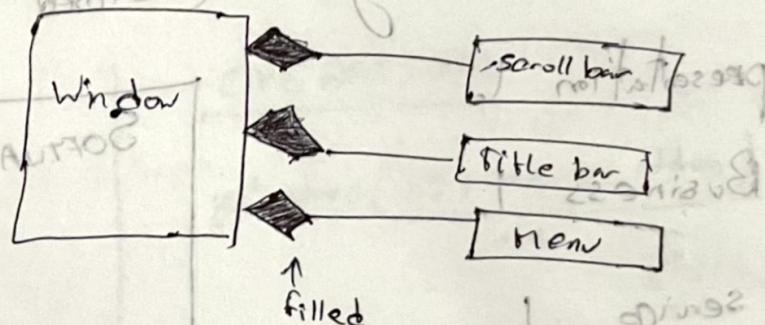
Association with a diamond.

→ Can exist independently from the aggregate



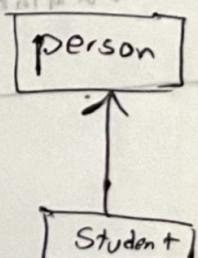
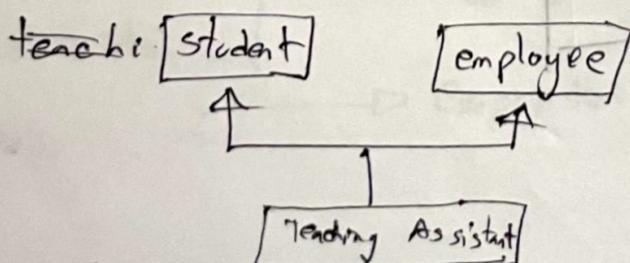
Composition relationship:-

composition indicates a strong ownership. (They live and die as a whole)



Generalization relationship:-

creates a superclass contains the basic attributes that will be used in subclasses,



DESIGN PATTERN

23 design patterns classified in three categories:-

(i) Creational

(ii) Structural

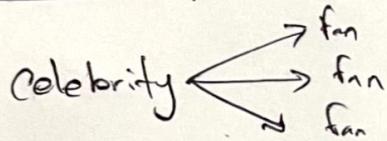
(iii) Behavioral

Database connection (singleton pattern)

[Creational]

Observer pattern:

[Behavioral]



(youtube subscription bell button)

Telus
Nabil

Robot
Wing

Hand

→ Face
Dome

Singleton Pattern

DESIGN PATTERN

Intent:- Ensure a class has only one instance, and provide a global access to it.

Example:- USGS database.

Falls on creational pattern.

Adapter Pattern :-

new HelpDesk() [creation of new object - (i) (class)]

static variable [class variable - (ii)]

instance (iii)

def __new__(cls):

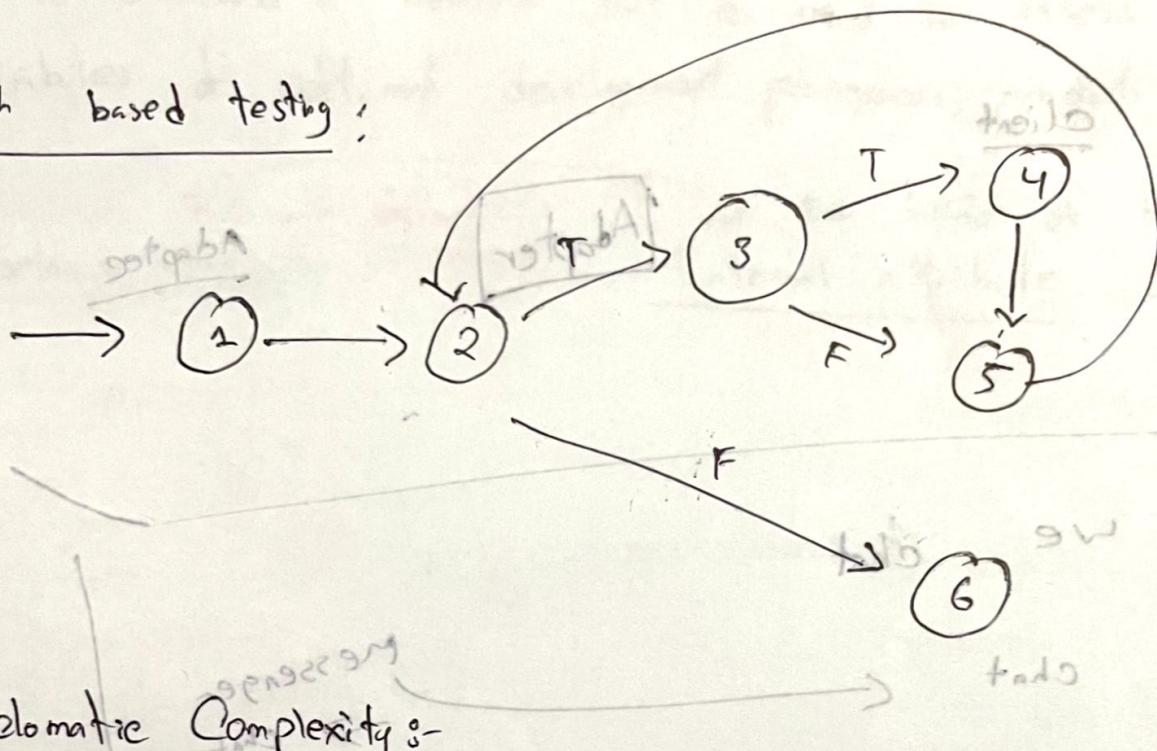
cls.__init__(cls) [takes class reference]

[instance] : metting revised

metting revised
and ← → finalised
and to

White box testing :- A technique which is used to find out the number of paths in a graph.

Path based testing:



Cyclomatic Complexity :-

$$C.C. = E - N + 2P$$

Predicted ^{a few} Nodes = T/F nodes.

Tested path $\leq C.C.$

The program is

- Q. types?
 - Find out the C.C.
 - Find out the possible test paths

METRICS

SOFTWARE MEASUREMENT:-

It is used of finding how a software is performing.

Software Metrics:-

It is standard of measure that is used to assess various attributes of software development, processes, products & projects.

External attribute This will automatically get affected.

Internal attribute If the inside of this attribute gets affected

PRODUCT METRICS:-

Dynamic metric :-

→ helps to identify the efficiency and reliability of a system. Collected by measurements made of a program in execution.

Static metric :- It is collected from a static situation where we represent the system. It is usually collected from case study.

→ help assess

- Complexity.
- understandability.
- maintainability.

Cyclomatic Complexity

→ It is a source code complexity measurement that is being correlated to a number of coding errors.

It is used to measure the complexity of a program

$$\text{Cyclomatic Complexity, } M = E - N + 2P$$

Here,

E = number of edges of the graph

N = " nodes "

P = " connected components

Done by quantifying the number of linearly independent paths through a program source code

Lower the program's cyclomatic complexity, lower the risk to modify and easier to understand.

Again,

$$M = R + 1$$

Here,

R = number of regions in the graph

if, else if, while, for, case

do while

No. of times these keywords appears that is the no. of regions.

and,

$$M = P + 1$$

Here,

P = number of predicate nodes in the graph.

SPECIALIZATION INDEX :-

- It measures the extent to which subclasses override their ancestors class.
- This metric provides a percentage, where the class contains at least one operation.
[Normal range is 0% to 120 %]

For a root class
Specialization indication
is zero.

$$SIX = \frac{NMO \times DIT}{NMO + NMA + NMI}$$

Where,

DIT = Depth of inheritance.

NMO = Number of overloaded / modified operations

NMA = Number of operations added to the inheritance.

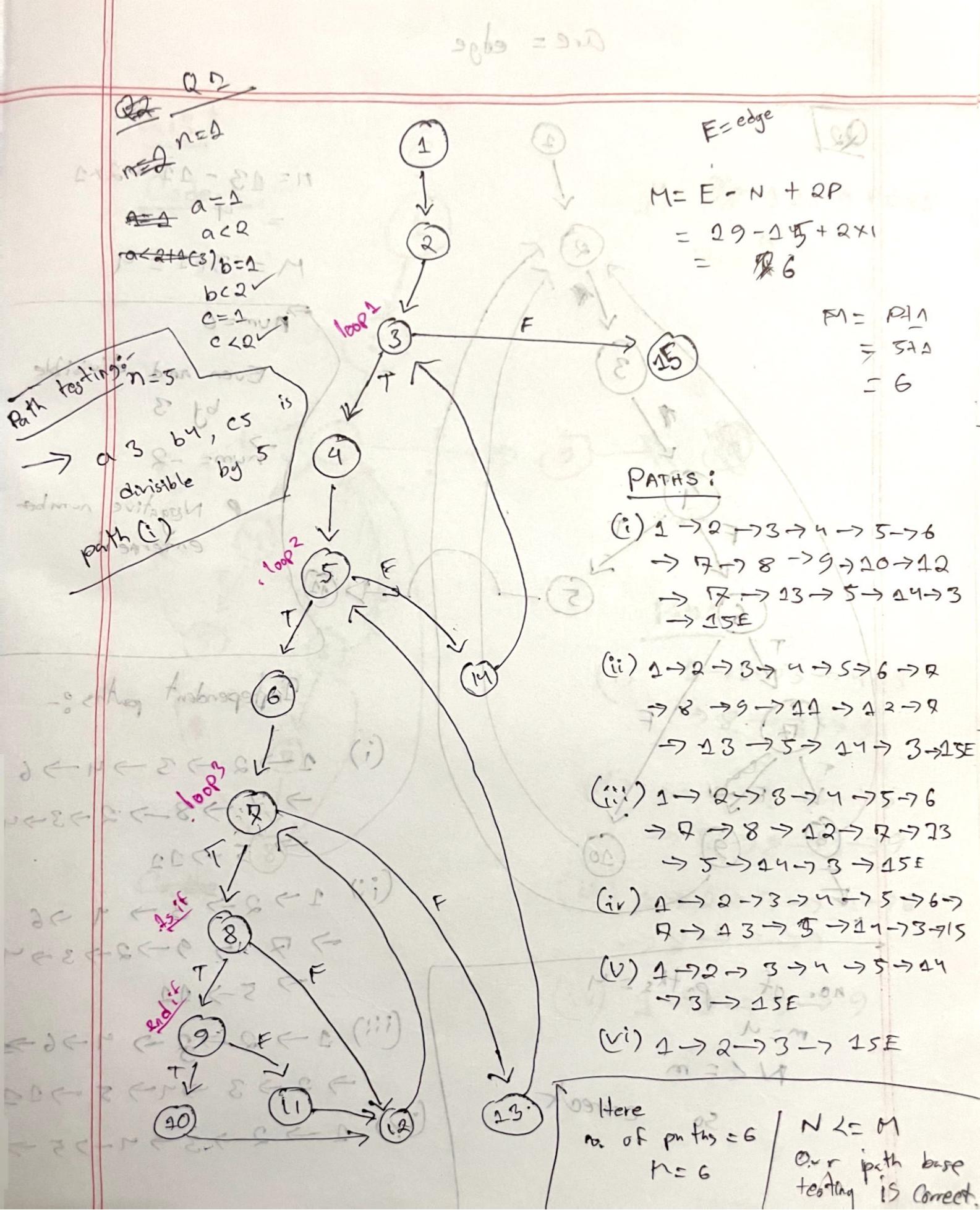
NMI = The number of inherited operations.

DIRECT REMOVAL EFFICIENCY % -

$$DRE = \frac{E}{E+D}$$

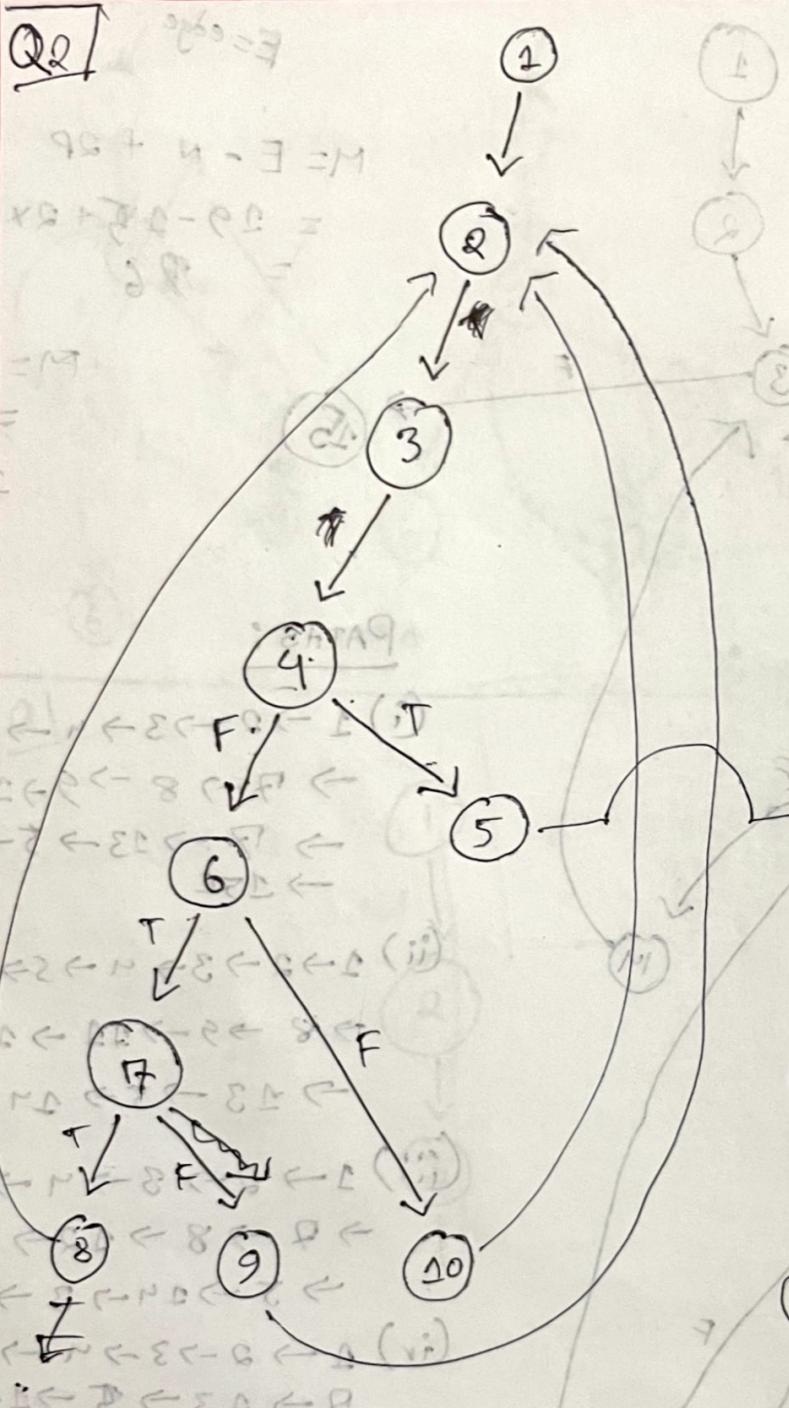
E = Number of errors before delivery of the software.

D = " " defects found after delivery of "



arc = edge

Q2



$$\text{no. of paths} = 4$$

$$N \leq m$$

so correct

$$n = 13 - 12 + 2 \times 1$$

$$= 4$$

$$m = 8 + 1 = 9$$

$$\text{num} = 6$$

Even and divisible
by 3
 $\rightarrow \text{num} = -2$

\varnothing Negative number
entered.

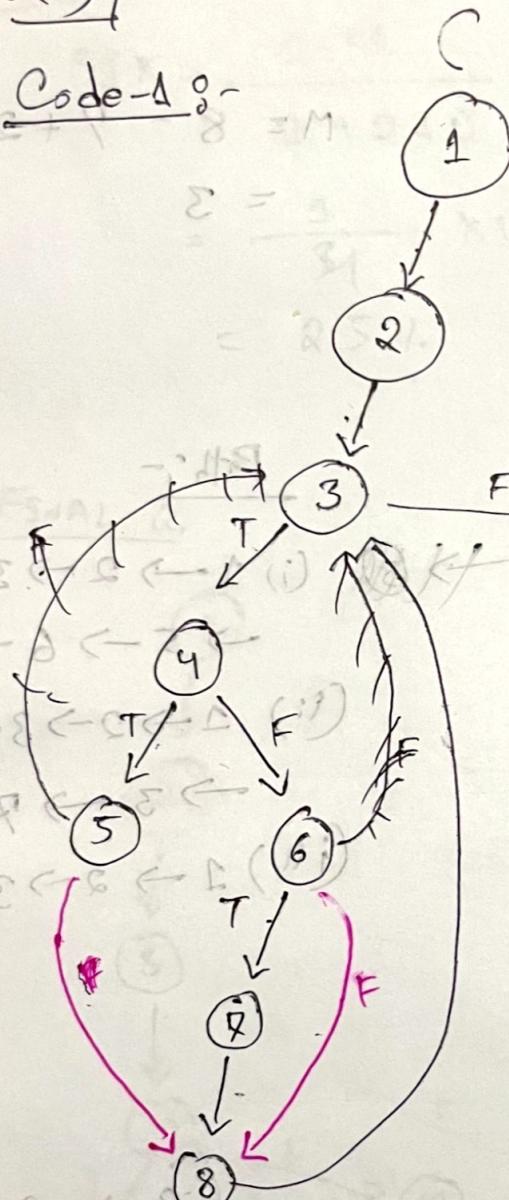
Independent paths :-

- (i) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6$
 $\rightarrow 7 \rightarrow 8 \rightarrow 2 \rightarrow 3 \rightarrow 4$
 $\rightarrow 5 \rightarrow 11$
- (ii) $1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 6$
 $\rightarrow 8 \rightarrow 9 \rightarrow 2 \rightarrow 3 \rightarrow 4$
 $\rightarrow 5 \rightarrow 11$
- (iii) $1 \rightarrow 2 \rightarrow 8 \rightarrow 9 \rightarrow 6 \rightarrow 10$
 $\rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 5 \rightarrow 11$
- (iv) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 11$

$$10 \leftarrow 80 - 38$$

Q.3/

Code-1 :-



Code

W19 S-sho

$$M = \emptyset \neq 10 - 9 + 2 \times 1$$

$$= 4$$

Path

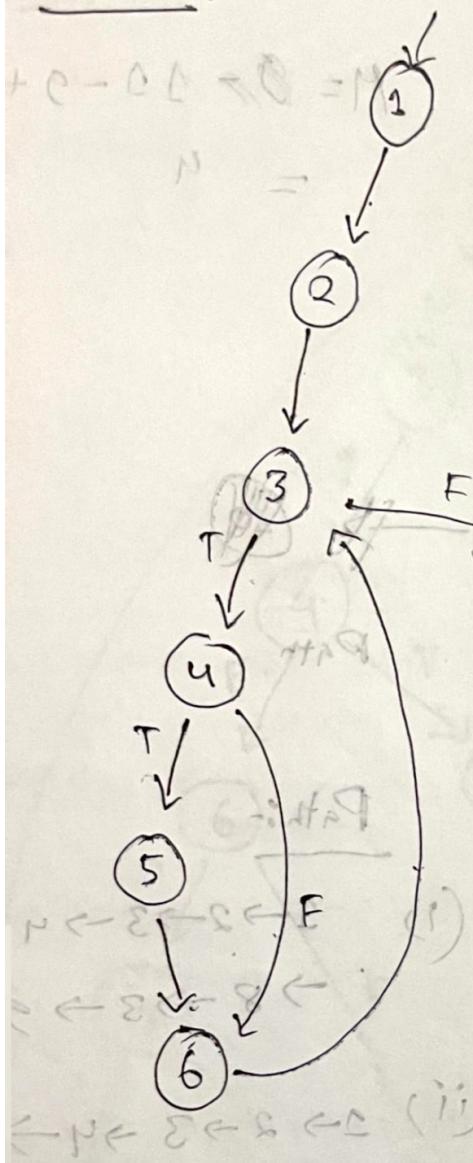
Path :-

(i) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$
 $\rightarrow 8 \rightarrow 3 \rightarrow 9$

(ii) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6$
 $\rightarrow 7 \rightarrow 8 \rightarrow 3 \rightarrow 9$

(iii) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6$
 $\rightarrow 8 \rightarrow 3 \rightarrow 9$

(iv) $1 \rightarrow 2 \rightarrow 3 \rightarrow 9$

Code-2 :

(ii)

→ 8 ← 7 ←

n ← ε ← g ← c (iii)

ε ← ε ← 8 ←

n ← ε ← c ← l (iv)

~~NAO XORT
Nro+num+num~~

$$M = 8 - 7 + 2 \times 1 \\ = 3$$

18.10

Path :-

(i) 1 → 2 → 3 → 4
→ 5 → 6 → 3 → 7

(ii) 1 → 2 → 3 → 4 → 6
→ 3 → 7

(iii) 1 → 2 → 3 → 7

(Code 2 better)

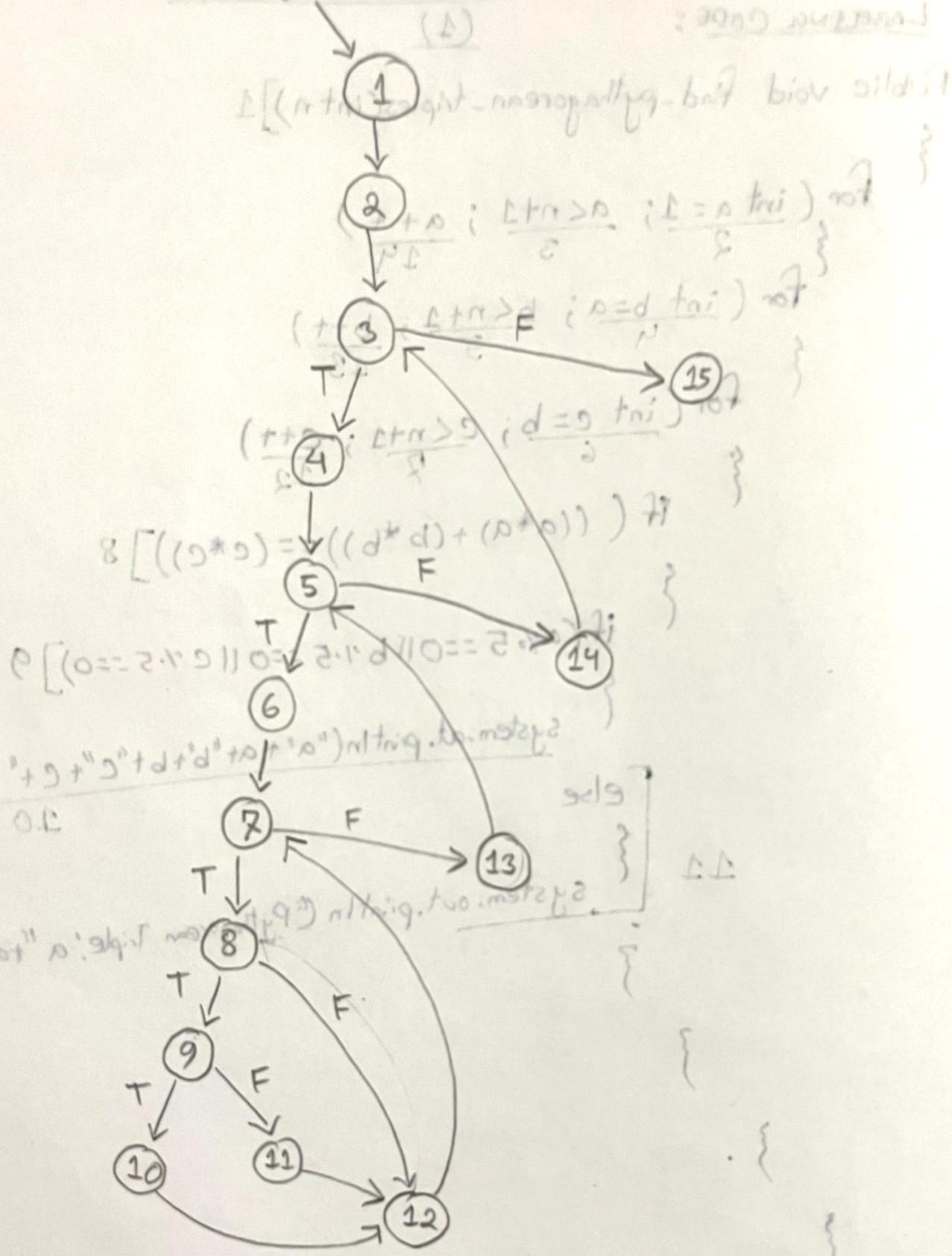
Answer to the q. no. - 01

LABELING CODE:

(1)

```
Public void find-pythagorean-triples(int n)] 1
{
    for (int a=1; a<n+1; a++) ] 2
    {
        for (int b=a; b<n+1; b++) ] 3
        {
            for (int c=b; c<n+1; c++) ] 4
            {
                if ((a*a)+(b*b)) == (c*c)) ] 5
                {
                    if (a%5 == 0 || b%5 == 0 || c%5 == 0) ] 6
                    {
                        System.out.println("a"+a+b+c+" is divisible by 5"); ] 7
                    }
                    else ] 8
                    {
                        System.out.println("Pythagorean Triple, a "+a+b+c); ] 9
                    }
                }
            }
        }
    }
} . ] 10
} ] 11
```

DO - Start CFG₂: at main



(2)

~~-: 5 nodes fast alg~~

$$\text{Cyclomatic complexity, } M = E - N + 2P \quad \text{Edges, } E = 19$$

$$= 19 - 15 + 2 \times 1 \quad \text{Nodes, } N = 15$$

~~"2nd addition of 20 pd 8 NO" = 6 - P :- tqnC best~~

~~Connected Nodes, P = 1~~

(3)

(4)

(2) & (3) most

Independent Paths :-

(i) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 7 \rightarrow 13$
 $\rightarrow 5 \rightarrow 14 \rightarrow 8 \rightarrow 15$ ~~grouping by 5 nodes~~

(ii) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 11 \rightarrow 12 \rightarrow 7 \rightarrow 13 \rightarrow 5$
 $\rightarrow 14 \rightarrow 3 \rightarrow 15$ ~~M = 10~~

(iii) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 12 \rightarrow 7 \rightarrow 13 \rightarrow 5 \rightarrow 14 \rightarrow 3$
 $\rightarrow 15$ ~~initiate bond diag~~

(iv) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 13 \rightarrow 5 \rightarrow 14 \rightarrow 3 \rightarrow 15$

(v) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 14 \rightarrow 3 \rightarrow 15$

(vi) $1 \rightarrow 2 \rightarrow 3 \rightarrow 15$

Path test case:-

(3)

Taking path no. + (i) from the independent paths;

Test case :- $N = 5$

Expected Output:- "~~a3b~~" "a3 by c5 is divisible by 5."

(4) (2)

From (2) & (3),

- i.e. testing the program

Number of independent paths, $n = 6$

Cyclomatic Complexity, $M = 6$

Over we get, $P \leq M - 1 = 6 - 1 = 5$ (ii)

$n \leq M - 1 = 6 - 1 = 5$

$\therefore P \leq 5$ (iii)

So, our path based testing done correctly. (iv)

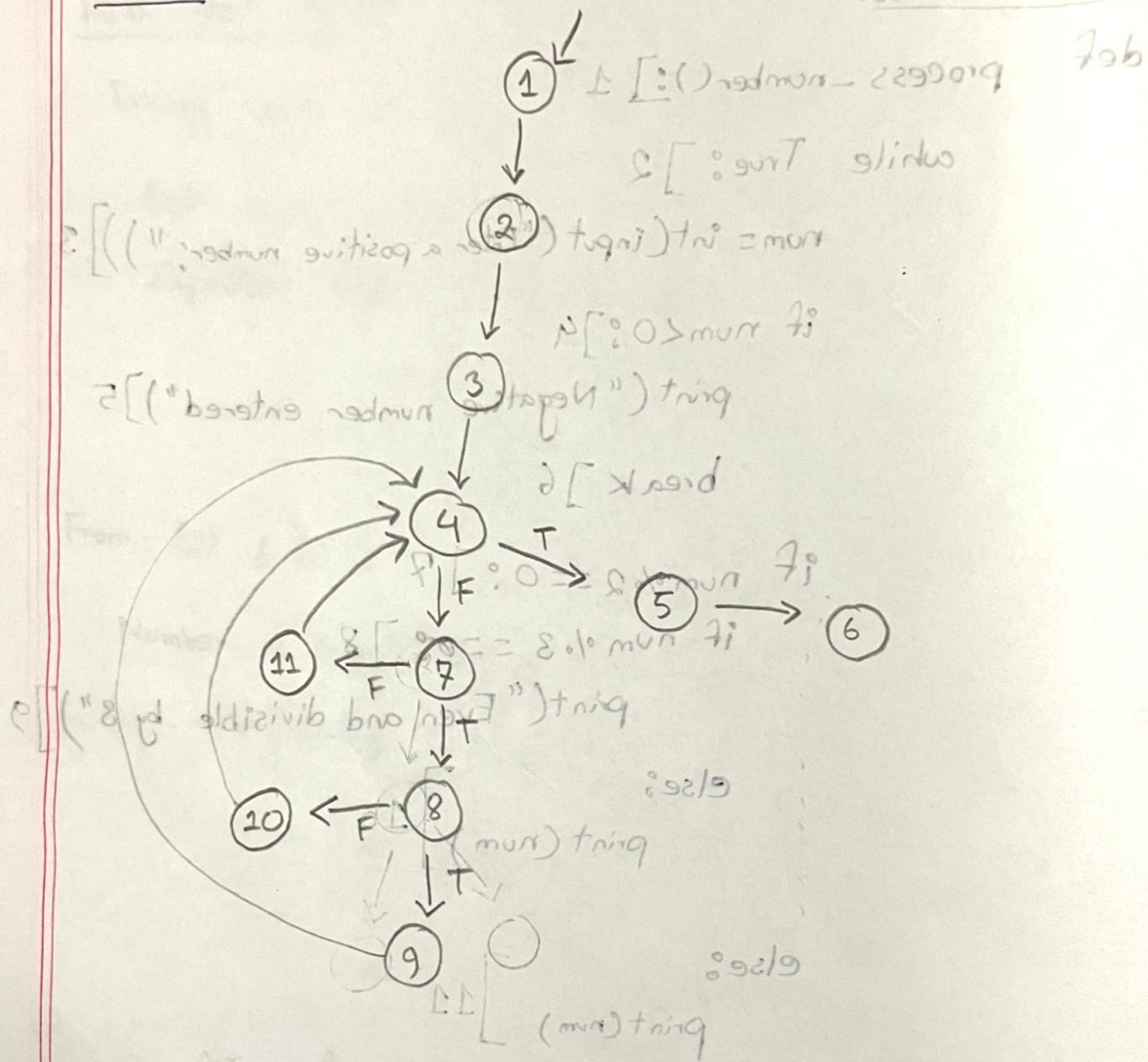
$P \leq 5$ (v)

$P \leq 5$ (vi)

LABELLING CODE:Answer to the q. no.- 02

CEP:

```
def process_number():] 1
    while True:] 2
        num = int(input("Enter a positive number:"))] 3
        if num < 0:] 4
            print("Negative number entered")] 5
            break] 6
        if num % 2 == 0:] 7
            if num % 3 == 0:] 8
                print("Even and divisible by 3")] 9
            else:
                print(num)] 10
        else:
            print(num)] 11
```

CFG:Answers of all of yourquestions here

Ans (2) (ii) The q no. 23

Cyclomatic complexity, $M = E - N + 2P$

Edges, $E = 13$

$N = 11$, $P = 1$

Nodes, $N = 11$

$P = M$

Connected Nodes, $P = 1$

$\text{if } (A[i] * 2 == 0) \{ \}$

$\sum \text{ of top } 5 = A[1] + A[2] + A[3] + A[4] + A[5]$

$$\frac{(3)}{(N)} M \Rightarrow n$$

(P)

Independent paths :-

(i) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 2 \rightarrow 9 \rightarrow 4 \rightarrow 5 \rightarrow 6$

(ii) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 7 \rightarrow 8 \rightarrow 10 \rightarrow 4 \rightarrow 5 \rightarrow 6$

(iii) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 7 \rightarrow 14 \rightarrow 4 \rightarrow 5 \rightarrow 6$

(iv) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$

Path test case :-

Taking path no.-(i) from I.P. \Rightarrow

Test case :- $\text{num} = 6$

Expected

Output :- "Even and divisible by 3"

Taking path no.-(iv) from I.P.

Test case : $\text{num} = -2$

Output : "Negative number entered"

(u) (v)

From (2) and (3) :-

$$\Sigma E = E \text{ (edges)}$$

$$\Sigma L = V - 3 \quad \text{or} \quad \Sigma L = M \quad (\text{fixing conditions})$$

Number of independent paths, $n = 4$

Cyclomatic complexity, $M = 4$

We get,

$$(4) \quad n \leq M_{(\Sigma)}$$

∴ testing the program

So, our path based testing done correctly.

~~$\Sigma \leftarrow N \leftarrow P \leftarrow S \leftarrow Q \leftarrow R \leftarrow \Sigma$~~

$\Sigma \leftarrow N \leftarrow P \leftarrow S \leftarrow Q \leftarrow R \leftarrow \Sigma \quad (i)$

$\Sigma \leftarrow N \leftarrow P \leftarrow S \leftarrow Q \leftarrow R \leftarrow \Sigma \quad (ii)$

$\Sigma \leftarrow N \leftarrow P \leftarrow S \leftarrow Q \leftarrow R \leftarrow \Sigma \quad (iii)$

$\Sigma \leftarrow N \leftarrow P \leftarrow S \leftarrow Q \leftarrow R \leftarrow \Sigma \quad (vi)$

∴ second test

(vi) - on diag point

$\Sigma \leftarrow N \leftarrow P \leftarrow S \leftarrow Q \leftarrow R \leftarrow \Sigma$: second test

"Even number divisible by 4" - : $\Sigma \leftarrow N \leftarrow P \leftarrow S \leftarrow Q \leftarrow R \leftarrow \Sigma$

<9.1 mark (i) - on diag point

$\Sigma \leftarrow N \leftarrow P \leftarrow S \leftarrow Q \leftarrow R \leftarrow \Sigma$: second test

Excessive

Excessive

Answer to the q. no.- 03 - Ques

CODE-1 :-

```

int sum=0; ] 1
for (int i=0; i<A.length, i++) ] 2
{
    if (A[i] % 2 == 0) { ] 3
        sum += A[i]; ] 4
    }
    else
    {
        if (A[i] % 2 != 0) ] 5
            System.out.println(A[i] + " is odd, skipping..."); ] 6
    }
}
System.out.println(sum); ] 7

```

(1)

CODE-2 :-

```

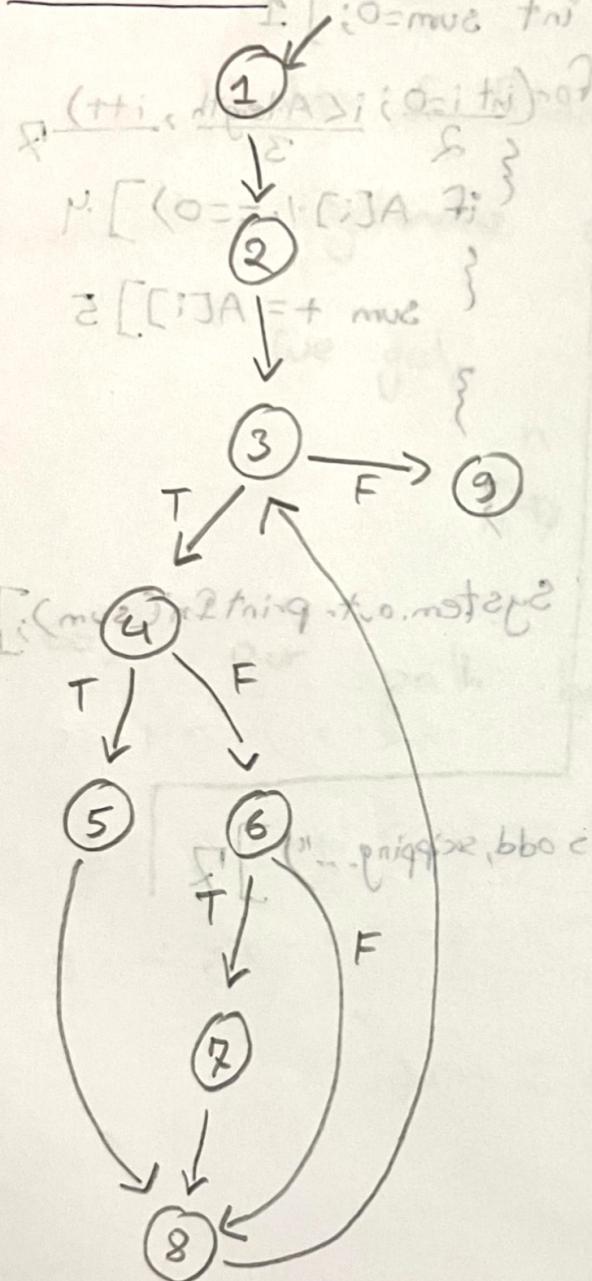
int sum=0; ] 1
for (int i=0; i<A.length, i++) ] 2
{
    if (A[i] % 2 == 0) { ] 3
        sum += A[i]; ] 4
    }
}
System.out.println(sum); ] 5

```

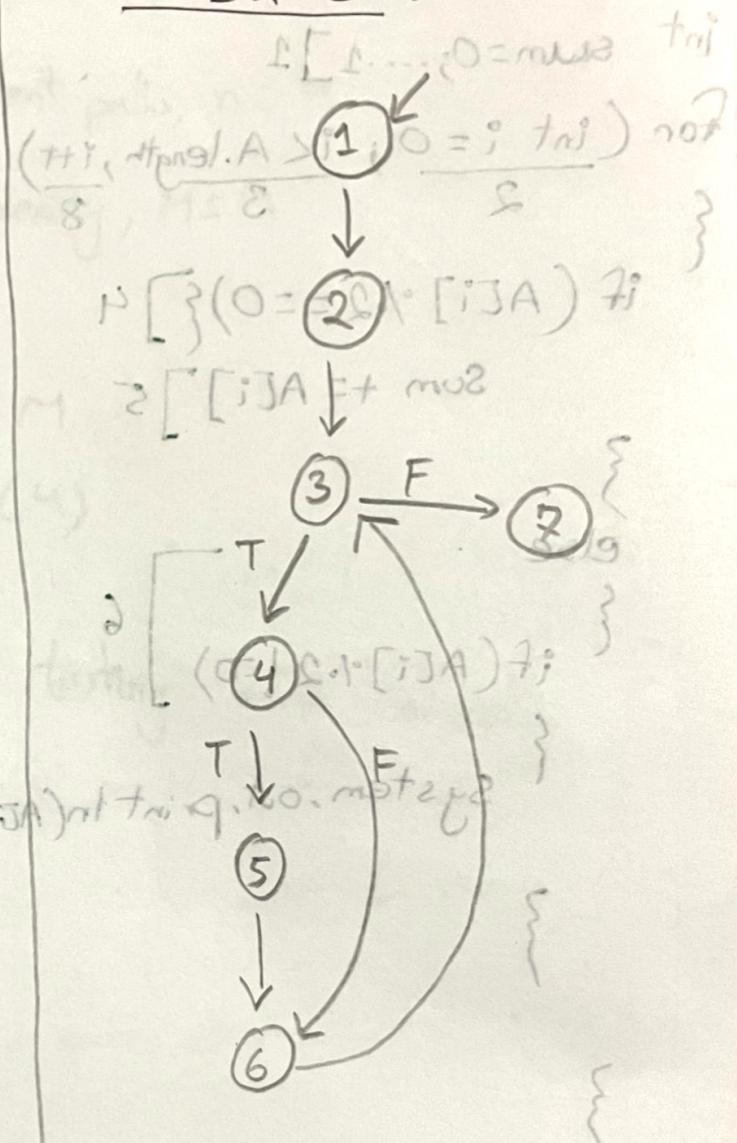
CFG₂ :- 20 - on . p set at zero

(5)

CODE1 CFG :-



CODE2 CFG :-



Code 1 cyclomatic complexity :-

edges, nodes, branches (Labs)	
$M_1 = 11 - 9 + 2 \times 1$	Edges, $E = 11$
$\therefore M_1 = 4$	Nodes, $N = 9$
	Connected nodes, $P = 1$
	(i) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \leftarrow 2 \leftarrow 1 - E \leftarrow 8 \leftarrow 1$
	(ii) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \leftarrow 2 \leftarrow 1 - E \leftarrow 8 \leftarrow 1$
	(iii) $1 \rightarrow 2 \rightarrow 3 \leftarrow 8 \leftarrow 2 \leftarrow 1 - E \leftarrow 8 \leftarrow 1$
	(iv) $e \leftarrow 8 \leftarrow 2 \leftarrow 1 - E \leftarrow 8 \leftarrow 1$
	(v) $e \leftarrow 2 \leftarrow 1 - E \leftarrow 8 \leftarrow 1$
	(vi) $e \leftarrow 8 \leftarrow 2 \leftarrow 1 - E \leftarrow 8 \leftarrow 1$

Code 2 Cyclomatic complexity :-

edges, nodes, branches (Labs)	
$M_2 = 8 - 7 + 2 \times 1$	Edges, $E = 8 - 7 + 2 \times 1$ test diag 11A
$\therefore M_2 = 3$	Nodes, $N = 7$
(ii) \leftarrow diag	Connected nodes, $P = 1$
$[1] = A$ \because good test	Edges, $E = 8$ (i) \leftarrow diag prior
$O = m \nu 2$	Nodes, $N = 7$
$M_1 > M_2$	Connected nodes, $P = 1$
Here, $\therefore L = 0$ \therefore good	$\partial = m \nu 2 \therefore$ \therefore $L = 0$ \therefore good

Code-2 cyclomatic complexity is better. $[2, 3, 4] = A$ \therefore good test

$[1] = A$ \therefore good test

$O = m \nu 2 \therefore$ $L = 0$ \therefore good

$O = m \nu 2 \therefore$ $L = 0$ \therefore good

(2)

Code 2: print sum of odd numbers

Code-1 Independent paths:-

(i) $1 \rightarrow 2 \rightarrow 3 - 4 \rightarrow 5 \rightarrow 8 \rightarrow 3 \rightarrow 9$

(ii) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 3 \rightarrow 9$

(iii) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 8 \rightarrow 3 \rightarrow 9$

(iv) $1 \rightarrow 2 \rightarrow 3 \rightarrow 9$

All path test case for each independent path :-

Using path \rightarrow (i),

Test case :- $A = [2, 4]$

Expected

Output :- Sum = 6

Using path \rightarrow (ii),

Test case :- $A = [1]$

Expected

Output :- "1 is odd, skipping..."

Sum = 0

Using path \rightarrow (iii),

Test case :- $A = [1.2, 1.5]$

Expect

Output :- Sum = 0

Using path \rightarrow (iv),

Test case :- $A = []$

Expected

Output :- Sum = 0

Code-2 Independent paths :-

(i) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 3 \rightarrow 7$

(ii) $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 3 \rightarrow 7$

(iii) $1 \rightarrow 2 \rightarrow 3 \rightarrow 7$

Path test case for each independent path:-

Using path \rightarrow (i)

Test case :- $A = [2, 4]$

Output :- sum=6

Using path \rightarrow (ii)

Test case :- $A = [1]$

Output :- sum=0

Using path \rightarrow (iii)

Test case : $A = []$

Output : sum=0

Answer to the q. no. - 09

[For bird class]

$$SIX = \frac{NMO \times DIT}{NMO + NMA + NMQ}$$

(i) $\rightarrow S \leftarrow D \leftarrow Z \leftarrow N \leftarrow E \leftarrow Q \leftarrow R$
 (ii) $NMO = 1$
 (iii) $DIT = 1$
 (iv) $NMA = 2$
 (v) $NMQ = 2$

$$= \frac{1 \times 1}{1+2+2} \times 100\%$$

done for 2nd test step

$$= \frac{1}{4} \times 100\% \quad (i) \leftarrow \text{tag prior}$$

(ii) $\leftarrow \text{tag prior}$

$$[E] = A \quad SIX_{2nd} = 25\%$$

using path $\rightarrow (i)$

$O = mva \therefore tag two$

Test case :- $A = 1$

$$[E] = A \quad \because \text{2nd test}$$

$$O = mva \quad \therefore \text{tag two}$$

(ii) $\leftarrow \text{tag prior}$

$$[E] = A \quad \because \text{2nd test}$$

$$O = mva \quad \therefore \text{tag two}$$

Expect
Output

observer + Adapter + Singleton 7/9b

Messi → class
+
10 minutes school.

Class : AI-software

Student → class

def student-track(self, student):

print(student.getname(), "has been tracked")

def payment(self, student):

(else) → if student.getpay() == "due":

: print(student.getname(), "payment is due")

else:

print(student.getname(), "payment is completed")

Class Messi

7/9c) etnampq 7/9b

instance = None

def __new__(cls):

if cls.__instance == None:

print("Messi the goat has arrived")

els. __instance = super(Messi, els).__new__(cls)

cls.teacher = "Messi"

return cls.__instance

def __init__(self):

(tnebutz) brqgq self.__student = []

```

def upload_video(self, video)
    for i in self._student:
        print(f"cls. teacher, {i}")
        i.notify(cls._teacher, video)

def tracker(self):
    "sub" == () take = AF-soft-adapter(self)
    for i in self._student:
        take.student_track(i)

def payments(self):
    take = AF-soft-adapter(self)
    for i in self._student:
        if i["balance"] <= 0:
            take.payment_student_pay(i)

def add_students(self, student):
    self._student.append(student)

```

class student

def __init__(self, name, payment):

self.name = name

self.payment = payment

def notify(self, teach, video):

print("teach", "has uploaded a new lecture on", "video")

def getpay(self):

return self.payment

class

AI-soft-adapter

def student-track(self, student):

[] = track = AI-software()

track.student-track(student)

def payment(self, student):

student.getpay() == "due":

student.payment

print("student.getname()", "payment is due")

pay = AI-software()

: (do) else

st pay

print("student.getname()", "payment is completed")

singleton + observer \rightarrow

Class Registrar
____instance = None
def __new__(cls):
 if cls.__instance == None:
 (cls.__instance = super(Register, cls).__new__(cls))
 cls.__name = "Register"
 return cls.__instance

def __init__(self)

self.__student = []
self.__teacher = []

(student) \rightarrow self.__student.append

(teacher) \rightarrow self.__teacher.append

def ~~course~~ schedule_teacher(self, teacher):

"sub" == (teacher, teacher)
for var in self.__teacher:

(var, teacher) \rightarrow var

(sub, teacher) \rightarrow teacher.notification(teacher)

def add_teacher(self, teacher):

(teacher, teacher) \rightarrow teacher.append(teacher)

$$\frac{E}{E+D} = 0.8$$

$$0.8 \times E = E \\ 0.8E = E \\ 0.8 = 1$$

Functional

```

def course-schedule-student (self, -detal):
    for var + in + self.-student:
        var. notification(detal)

def add-student (self, p stud):
    self.-student.append(stud)

```

Class Student:

```

def __init__(self, name, id):
    self.-student = name
    self.-id = id

def notification (self, note):
    print( self.-student, "your c schedule is", note )

```

Class Teacher:

```

def __init__(self, name, id):
    self.-student = name
    self.-id = id

def notification(self, note):
    print( self.-student, "your c. schedule is", note )

```

Functional

- (i) Should be able to send messages to other users.
- (ii) Images & file attachments along with text messages.
- (iii) Integrates chat.

MVC :-

- Advantage :-
- Separately work (multiple ways to view and interact)
 - User experience better.
 - Unknown future requirements.

disadvantage :-

can include additional code and
Complexity of data model when interactions
are simple.

- Non functional
- (i) provide instant response
 - (ii) If accessible to users 24/7

- disadvantage :-
- allows the data to change independently
 - changes made in one site shown in all of them

SOFTWARE PATTERN :-

Repository pattern:-

USES :-

- (i) Large volumes of information are generated that has to be stored for a long time.
- (ii) Data driven - system.

(inclusion of data in the app times out)

Advantage :-

- (i) Components can be independent.
- (ii) Changes made by one component can be propagated/transferred to all components.

Limitations

Disadvantage: (i) large amount of memory loss of stored blocks.

(ii) Repository is a single point failure, so problems in the repository affect the whole system.

(iii) Inefficiencies in organizing all communication through the repository.

-: SVM

Client server pattern:-

USES:

- (i) Data in a shared database accessed from range of locations.
- (ii) load on a system is variable.

Advantage:-

- (i) servers can be distributed across a network.
- (ii) Cost efficient as requires less maintenance cost.
- (iii) Data recovery is possible.

Disadvantage:-

- (i) Each service is a single point failure, denial of service attacks or service failure.
- (ii) Performance may be unpredictable as depends on the network and system.

Pipeline filter pattern

Pipeline

USES :- (Commonly used in data processing applications)

- (i) When inputs are processed in separate stages to generate related outputs.

Advantages :-

- (i) Easy to understand and supports transformation reuse.
- (ii) Workflow style matches the structure of many business processes.

Disadvantage :-

- (i) The format for data transfer has to be agreed upon between source & sink.
- (ii) Each transformation must parse its input and unparse its output to the agreed form.

SCRUM :-

ROLES :-

(A) Product Owner :-

- Product Manager or Product sponsor
- Decides features, release date prioritization.
- Conveys clear vision to scrum team.

(B) Scrum Master

- (i) Project Manager / Team Lead
- (ii) Remove politics.
- (iii) Keep everyone productive.

Project T.

simulating with existing PBL

→ Teams are self organizing in form of scrum

→ QA, programmers, UI designer working in tight roles (i) et alios

ARTIFACTS:

(A) Product Backlog:-

- (i) A list of all desired work on project.
- (ii) List of user stories along with story points.
- (iii) Prioritized by product owner

(B) SPRINT BACKLOG :-

- (i) High priority features are selected based on product backlog
- (ii) Features will be covered within 14 days time slot.
- (iii) Estimated work is updated daily

(C) BURNDOWN CHART :

- (i) A display of what work has been completed and what is left to complete.

(ii) One for each developer / work team

(iii) Updated every day

base point (regular took off) (i)

costing around (ii)

newly arriving (iii)

average took off is equal to total cost

stab sales constant working

not increasing

more of main work equal to cost

work

CEREMONY:

(A) Daily Scrum Meeting

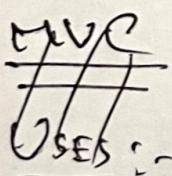
- (i) Only T.M., P.O., S.M. can talk. spontaneous
- (ii) Helps avoid other unnecessary meeting. spontaneous

(B) Sprint Review

- (i) Team presents what is accomplished during the sprint mandatory
- (ii) Ad informal.
- (iii) whole team participates.

(C) Sprint Retrospective

- (i) Identify the scopes of improvement for better result in next sprints.
- (ii) what worked well, what went wrong in the next are also discussed.



LAYERED ARCHITECTURE

USES

- (i) building new facilities on top of existing facilities.
- (ii) development is spread across several teams with each team with separate layer
- (iii) requirement for multilevel security.

Routes → map URL's to specific function.

Advantage:-

- (i) Easy replacement or addition of entire layers.
- (ii) Testing is easy (as components are isolated).

Disadvantage:-

- (i) Difficulties in higher-level in direct interaction with lower level.
- (ii) Performance problem, (because of multiple levels of service request).

- (iii) Use of even stories with story

Problems

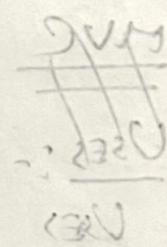
- Inheritance

- User need to convert all entities to JSON

• JSON

- User need to know how to handle both tables

• Entity also give



API'S

• Entity entities to get no additional was provided

• User must have certain base in transactions

• Using

• CSV

• Entities conversion not translatable