

**Program Name :** Masters of Computer Application

**Semester :** 2

**Paper Title :** Software Engineering

**Submitted by :** Rishi Raj Chaurasia

**Examination Roll Number :** 19234757031

**Department Roll Number :** 31

# Software Engineering Assignment

Submitted By - Rishi Raj Chauhan  
Roll No. - (31)

Ans 1

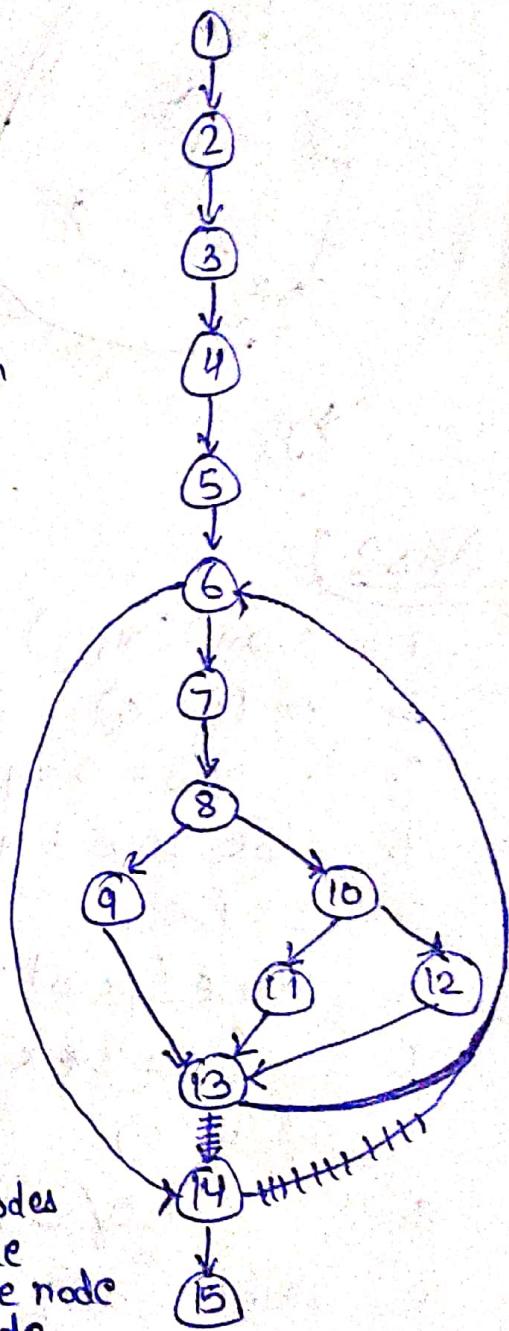
```

1. low , high , mid , i , j : integer ;
2. found : boolean ;
3. low = 1 ;
4. high = n ;
5. found = false ;
6. while (low <= high) & (!found) begin
7.     mid = (low + high)/2 ;
8.     if (e < a[mid]) then
9.         high = mid - 1 ;
10.    else if (e > a[mid]) then
11.        low = mid + 1 ;
12.    else found = true ;
13. end ;
14. bin-search = found ;
15. end ;

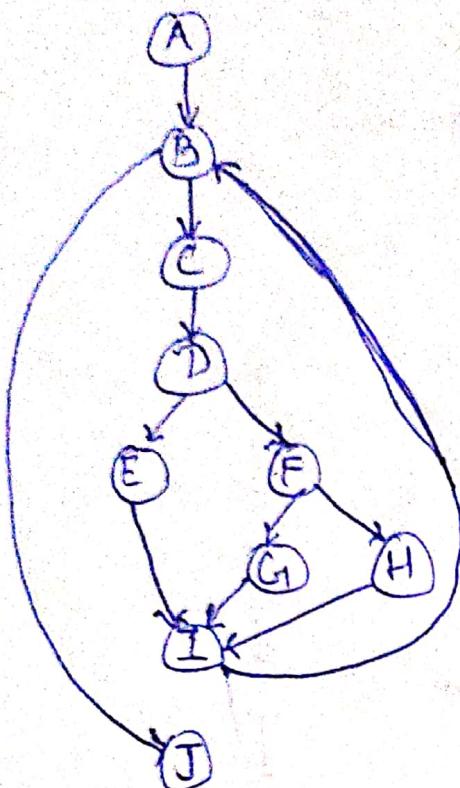
```

Flow Graph Nodes      DD Path Graph Node      Remark

1 - 5	A	Sequential nodes
6	B	Decision node
7	C	Intermediate node
8	D	Decision node
9	E	Intermediate node
10	F	Decision node
11	G	Intermediate node
12	H	Intermediate node
13	I	3 nodes are joined
14,15	J	Sequential node



## DD Graph



## Cyclomatic complexity

$$\text{Edges} - \text{Vertices} + 2 \\ \Rightarrow 12 - 10 + 2 \\ = 4$$

## Independent paths:-

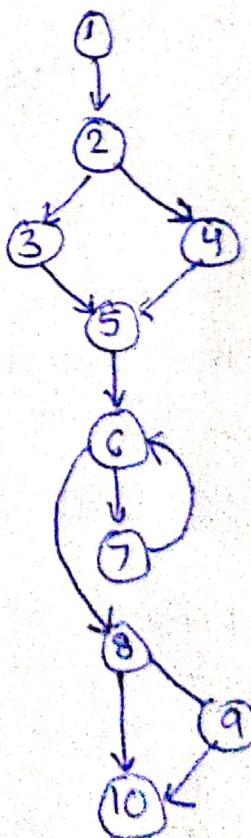
- 1) ABJ
- 2) ABCDEIBJ
- 3) ABCDFGIBJ
- 4) ABCDFHIBJ

## Ans 2)

```

1. scanf("%d");
2. if(y<0)
3.   pow=0-y;
4. else pow=y;
5. z=1.0;
6. while(pow!=0)
7.   {z=z*x; pow=pow-1}
8. if(y<0)
9.   z=1.0/z;
10. printf(z);
    
```

<u>Variable</u>	<u>Defined at Node</u>	<u>Used at Node</u>
x	1	7
y	1	2, 3, 4, 8
z	5, 7, 9	10
pow	3, 4	6, 7



<u>Variable</u>	<u>Path(beginning, end) nodes</u>	<u>Definition Clear?</u>
x	(1, 7)	Yes
y	(1, 2)	Yes
	(1, 3)	Yes
	(1, 4)	Yes
	(1, 8)	Yes
z	(5, 10)	No
	(7, 10)	No
	(9, 10)	Yes
pow	(3, 6)	Yes
	(3, 7)	Yes
	(4, 6)	Yes
	(4, 7)	Yes

Ans.3) Integers = { ..., -2, -1, 0, 1, 2, ... }

### Boundary Value Analysis

$$n=1$$

$$\text{Test Case} = 4n+1 = 5$$

Since, we don't have defined domain for the integer we can take any number from set of integer ( $\mathbb{Z}$ )

<u>Test Case</u>	<u>Input</u>	<u>Expected Output</u>
1	0	Number is Even
2	100	Number is Even
3	50	Number is Even
4	1	Number is Odd
5	99	Number is Odd

### Equivalence Class

Output domain

$$O_1 = \{ a : \text{Even} \quad \text{if } a \in \mathbb{Z} \text{ & } a \% 2 = 0 \}$$

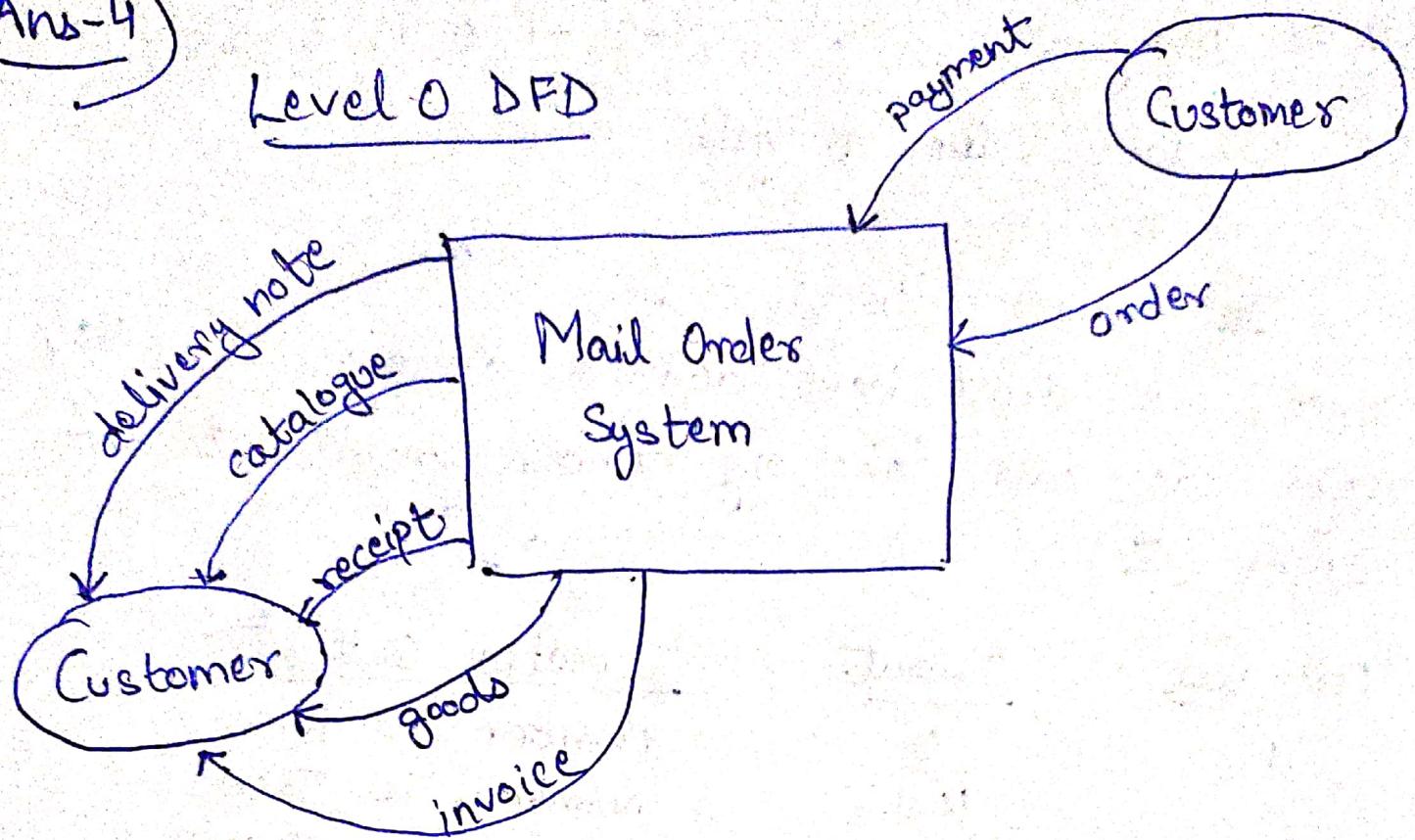
$$O_2 = \{ a : \text{Odd} \quad \text{if } a \in \mathbb{Z} \text{ & } a \% 2 \neq 0 \}$$

$$O_3 = \{ a : \text{Invalid} \quad \text{if } a \notin \mathbb{Z} \}$$

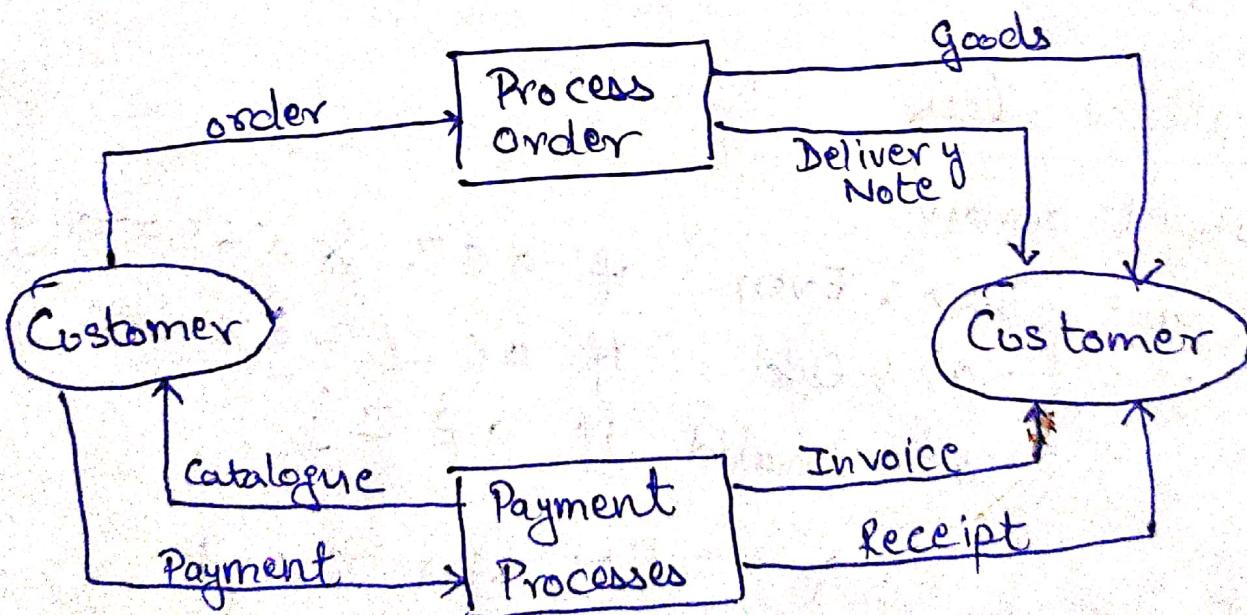
<u>Test Case</u>	<u>Input</u>	<u>Expected Output</u>
1	-100	Number is Even
2	99	Number is odd
3	27.5	Invalid.

Ans-4)

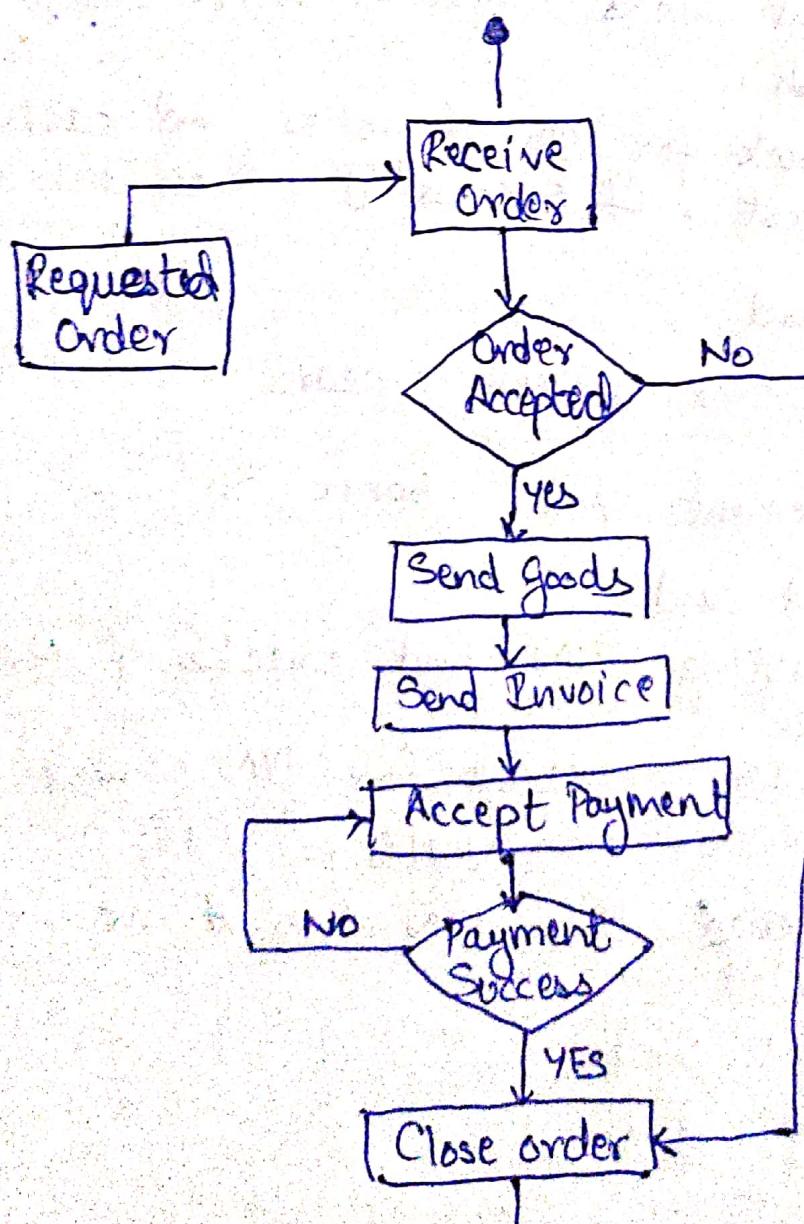
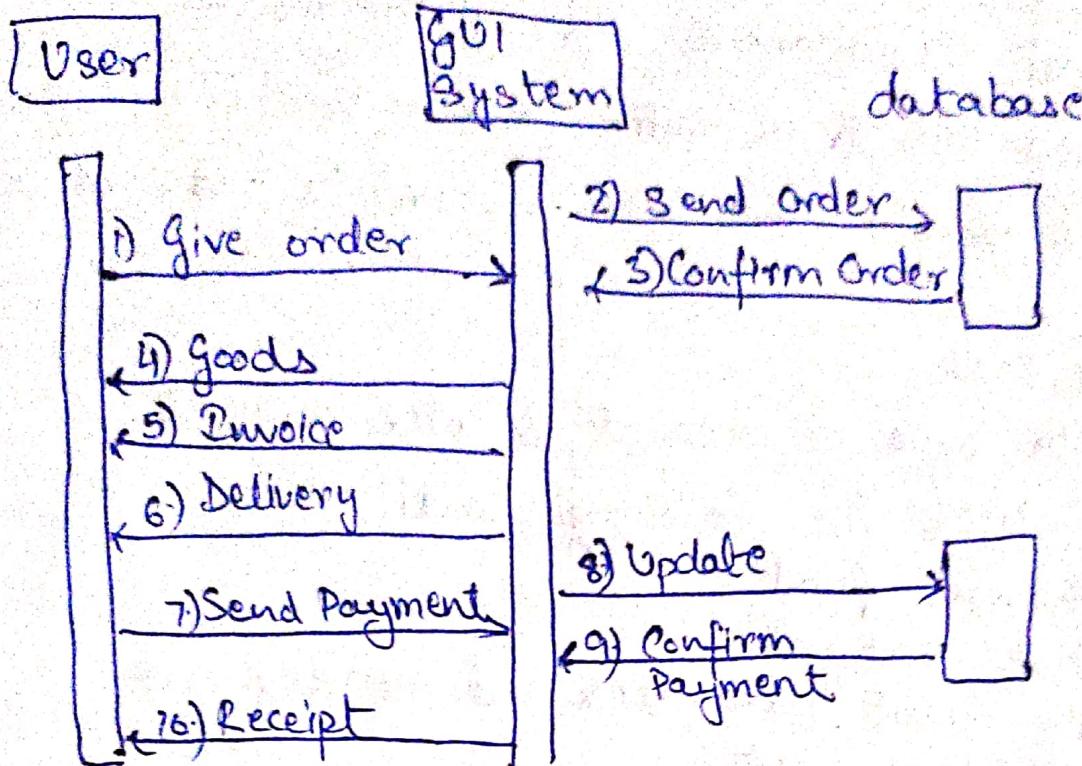
Level 0 DFD



Level 1 DFD



## Sequence Diagram



## Ques 5)

⇒ Functional Requirement for 'Withdraw' -

- Debit card should be valid and in working condition.
- Debit Card should have been successfully ~~not~~ read by ATM.
- Validate PIN.
- Select the type of account & enter amount.
- System check for following conditions.
  - a) Sufficient fund
  - b) Active Account
  - c) Daily limit parameters and monthly free transaction limit.
- After ~~at~~ successful checks, system authorizes dispensing of cash.
- ATM machine check for no. of notes and cash limit. in its chest. If successful, then dispenses cash.
- System ejects card.
- Again initial welcome screen appears.

⇒ Functional Requirement for 'Change Pin' -

- Insert valid ATM card.
- Authenticates existing PIN. and checks if the account is active.
- Enter new pin after selecting for PIN change.
- Re-enter new pin for confirmation.
- If both pin matched, then PIN changed successfully, else transaction failed.
- System ejects SIM card.
- Welcome screen appears.

Ans 6

### 1) Selenium:-

It is a testing framework to perform web application testing across various browsers and platforms like Windows, Mac and Linux. Selenium helps the tester to write tests in various programming languages like Java, PHP, C#, Ruby, etc. It offers record and playback features to write tests without learning Selenium IDE. It is the base for most of the other software testing tools in general.

### 2) Ranorex :-

Ranorex Studio offers various testing automation tools that cover testing all desktop, web, and mobile applications.

It offers following features —

- 1) GUI recognition.
- 2) Reusable test codes.
- 3) Bug detection.
- 4) Integration with various tools.
- 5) Record & Playback.

### 3) Test Complete:-

It is a functional testing platform that offers various solutions to automate testing for desktop, web and mobile applications by SmartBear Software.

Test Complete offers the following features.

- GUI testing
- Scripting language support - JavaScript, Python, VBScript, C# Script & C++ Script.
- Test Visualizer
- Scripted testing
- Test recording and playback.

## 4) Watir

Watir is an open source testing tool made up of Ruby libraries to automate web application testing. It is pronounced as 'water'.

Watir offers the following features:-

- Tests any language based web application.
- Cross-browser testing.
- Compatible with business-driven development tools like RSpec., Cucumber and Test/Unit.
- Tests webpage's buttons, forms, links and their responses.

Ans 7)

```
1. #include <studio.h>
2. int main()
3. {
4.     int a, b, c;
5.     cin >> a >> b >> c;
6.     if (a == 0)
7.     {
8.         cout << "Invalid";
9.         return;
10.    }
11.    int d = b*b - 4*a*c;
12.    double sqrt_d = sqrt(abs(d));
13.    if (d > 0)
14.    {
15.        cout << "Roots are real & unique\n";
16.        cout << (double)(-b + sqrt_d)/(2*a)
17.            << " and ";
18.            << (double)(-b - sqrt_d)/(2*a);
19.    }
20.    else if (d == 0)
21.    {
22.        cout << "Roots are real and equal \n";
23.        cout << -(double)b/(2*a);
24.    }
25. }
```

```

23     else
24     {
25         cout << "Roots are complex \n";
26     }
27     return 0;
28 }
```

### Cross Reference List

a -	6	5	6	11	16	16	21
b -	4	5	11	11	16	16	21
c -	4	5	11				
d -	11	12	13	18			
sqrt-d -	12	16	16				

<u>Line</u>	<u>Line Variable</u>	<u>Count</u>
-------------	----------------------	--------------

1	-	0
2	-	0
3	-	0
4	a, b, c	1
5	a, b, c	1
6	a	0
7	-	0
8	-	0
9	-	0
10	a, b, c, d	2
11	d, sqrt-d	0
12	d	0
13	-	0
14	-	0
15	a, b, sqrt-d	0
16	-	0
17	d	0
18	-	0
19	-	0
20	a, b	0
21	-	0
22	-	0
23	-	0
24	-	0
25	-	0
26	-	0
27	-	0
28	-	0

Avg. No. of Live Variables (LV) =  $\frac{\text{sum of Count of LV}}{\text{Count of Executive Statement}}$  =  $\frac{20}{24} = 0.8334$

Avg. Life of variables (x) =  $\frac{\text{sum of Count of LV}}{\text{Total no. of variables}}$  =  $\frac{20}{5} = 4$

Program Weakness =  $LV * \gamma$  =  $0.8334 * 4$  =  $3.3334$