PRACTICAL 1

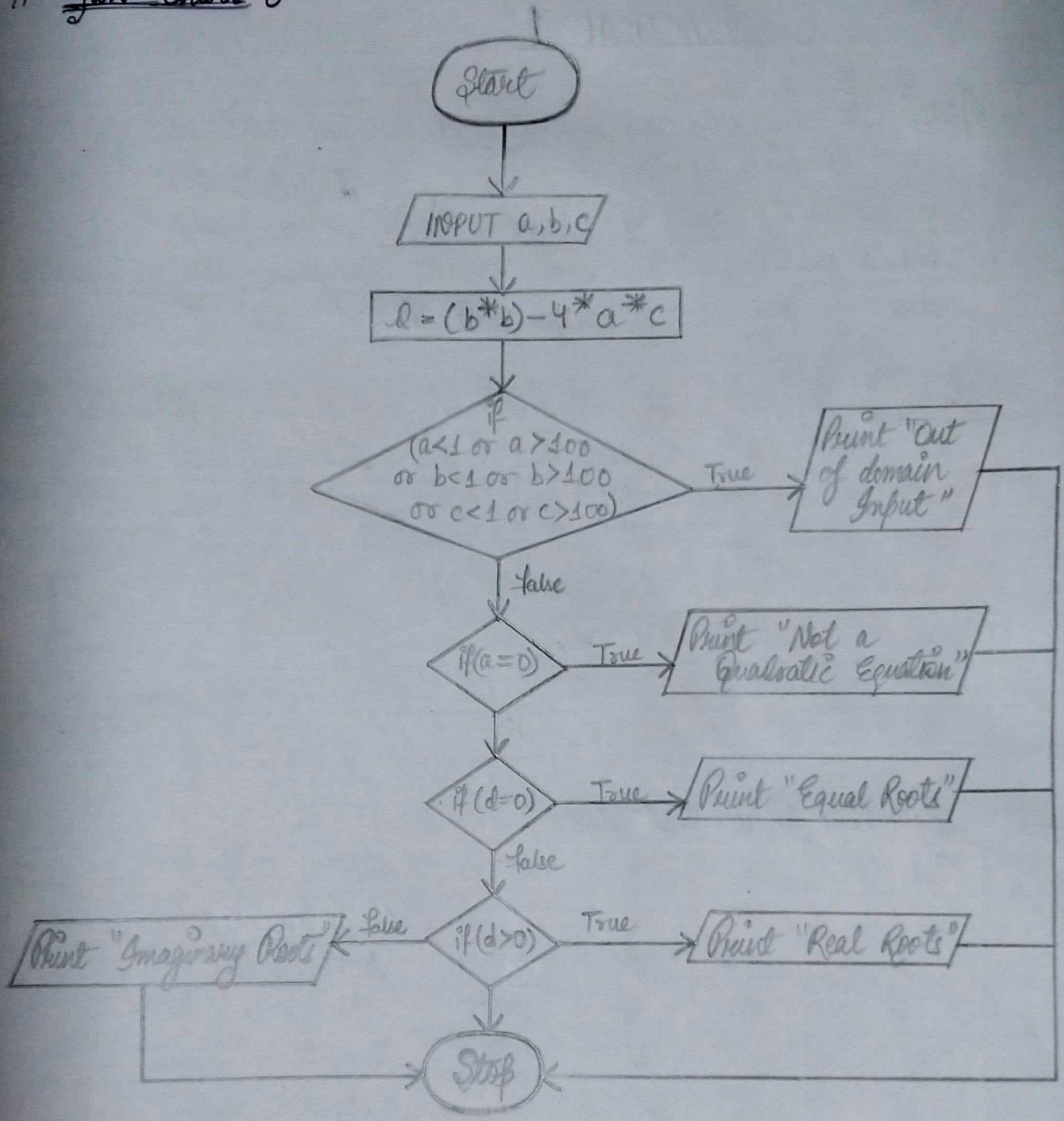
Aim: To determine the nature of roots of quadratic equation, its input is triple of positive integers (say x, y, z) and values may be from interval [1, 100] the program output may have one of the following [→ Not a Quadratic equation, Real roots, Imaginary roots, Equal roots]. Perform Boundary Value Analysis and Robust Case Testing.

Theory: Boundary Value Analysis is a software testing technique in which tests are designed to include representatives of boundary values in a range. A BVA has a total of $4*n+1$ distinct test cases, where n is the no. of variables in a problem. Robust Case Testing has software given invalid inputs. It is generally used to test exception handling. The total test cases are derived using $6*n+1$, where n is the no. of variables in a problem.

Boundary Value Test Cases:

Test Case	a	b	c	Expected Output	Output
1	50	50	1	Real Roots	✓
2	50	50	2	Real Roots	✓
3	50	50	50	Imaginary Roots	✓
4	50	50	99	Imaginary Roots	✓
5	50	50	100	Imaginary Roots	✓
6	50	1	50	Imaginary Roots	✓
7	50	2	50	Imaginary Roots	✓
8	50	99	50	Imaginary Roots	✓

Flow Chart:


Start
INPUT a,b,c
d = (b*b) - 4*a*c
if (a<1 or a>100 or b<1 or b>100 or c<1 or c>100)
True
Print "Out of domain Input"
false
if(a=0)
True
Print "Not a Quadratic Equation"
if(d=0)
True
Print "Equal Roots"
false
if(d>0)
False
Print "Imaginary Roots"
True
Print "Real Roots"
Stop

9	50	100	50	Equal Roots	✓
10	1	~~100~~50	50	Real Roots	✓
11	2	50	50	Real Roots	✓
12	99	50	50	Imaginary Roots	✓
13	100	50	50	Imaginary Roots.	✓

Robust Case Test Cases :

Test Case	a	b	c	Expected Output	Output
1	0	50	50	Not a Quadratic Equation	✓
2	1	50	50	Real Roots	✓
3	2	50	50	Real Roots	✓
4	50	50	50	Imaginary Roots	✓
5	99	50	50	Imaginary Roots	✓
6	100	50	50	Imaginary Roots	✓
7	101	50	50	Out of Domain Input	✓
8	50	0	50	Out of Domain Input	✓
9	50	1	50	Imaginary Roots	✓
10	50	2	50	Imaginary Roots	✓
11	50	99	50	Imaginary Roots	✓
12	50	100	50	Equal Roots	✓
13	50	101	50	Out of Domain Input	✓
14	50	50	0	Out of Domain Input	✓
15	50	50	1	Real Roots	✓
16	50	50	2	Real Roots	✓
17	50	50	99	Imaginary Roots	✓
18	50	50	100	Imaginary Roots	✓
19	50	50	101	Out of Domain Input	✓

*

SOURCE CODE:

```
#include<iostream>
using namespace std;

int main()
{

int a,b,c,d;

cout<<"The quadratic equation is of the type a(x^2) + bx + c = 0"<<endl;
cout<<"Enter a : "<<endl;
cin>>a;
cout<<"Enter b : "<<endl;
cin>>b;
cout<<"Enter c : "<<endl;
cin>>c;

d = (b*b)- (4*a*c);

if((a<1)|| (b<1)|| (c<1)|| (a>100)|| (b>100)|| (c>100)){
   if(a==0)
      cout<<"Not a Quadratic Equation"<<endl;
   else
      cout<<"Out of domain Input"<<endl;
}
else if(d==0)
   cout<<"Equal Roots"<<endl;

else if(d>0)
   cout<<"Real Roots"<<endl;

else     //if d<0
   cout<<"Imaginary Roots"<<endl;

return 0;
}
```

OUTPUTS:

```
The quadratic equation is of the type a(x^2) + bx + c = 0
Enter a :
50
Enter b :
0
Enter c :
50
Out of domain Input
```

```
The quadratic equation is of the type a(x^2) + bx + c = 0
Enter a :
50
Enter b :
1
Enter c :
50
Imaginary Roots
```

```
The quadratic equation is of the type a(x^2) + bx + c = 0
Enter a :
50
Enter b :
100
Enter c :
50
Equal Roots
```

```
The quadratic equation is of the type a(x^2) + bx + c = 0
Enter a :
0
Enter b :
50
Enter c :
100
Not a Quadratic Equation
```

```
The quadratic equation is of the type a(x^2) + bx + c = 0
Enter a :
1
Enter b :
50
Enter c :
99
Real Roots
```

PRACTICAL 2

Aim: To determine the type of triangle. Its input is triple of positive integers (say x, y, z) and the values may be from interval [1, 100]. The program output may be one of the following [→ Scalene, Isosceles, Equilateral, Not a Triangle]. Perform BVA and RCT.

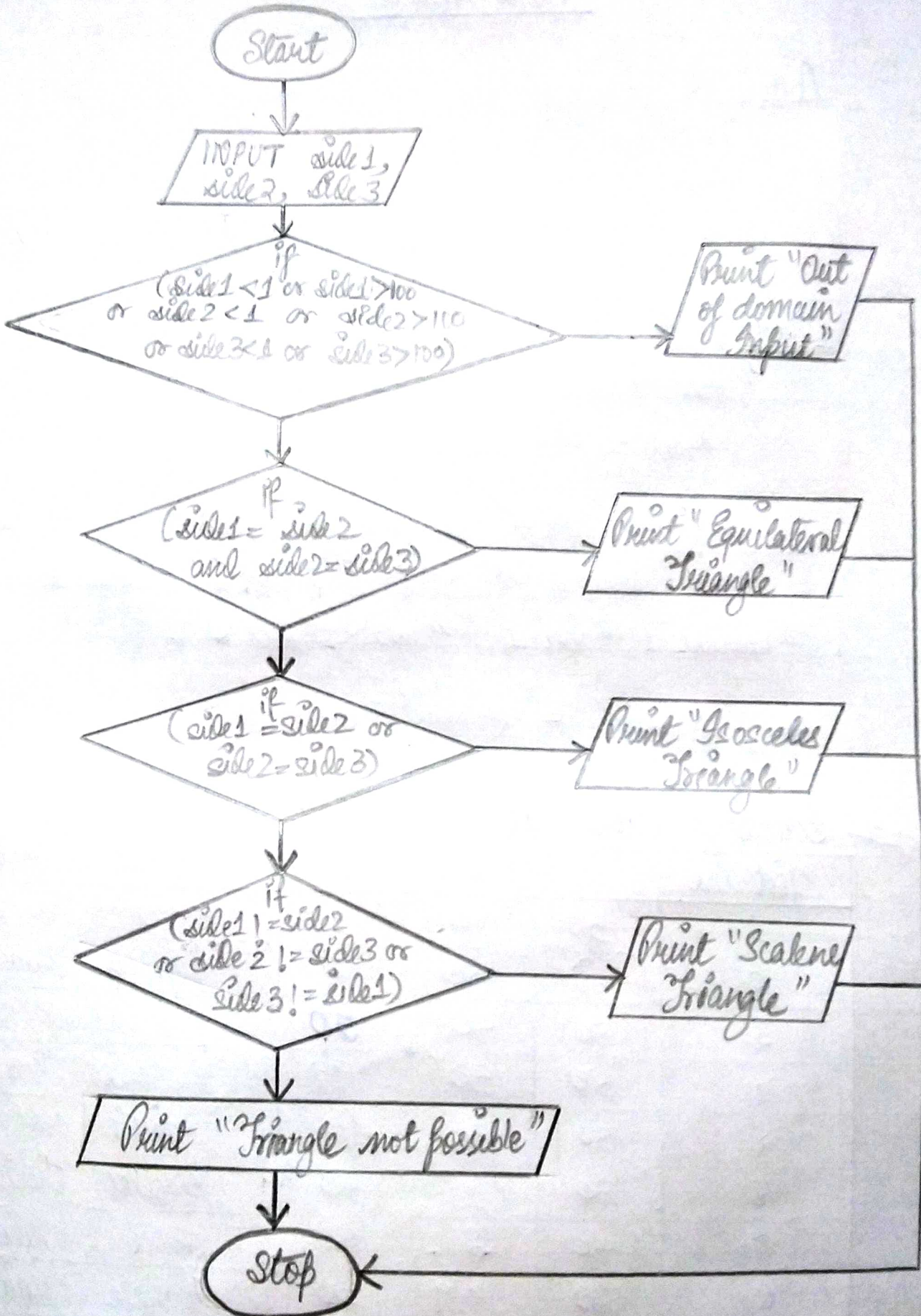
Theory: for BVA → Total no. of test cases = $4*n+1$ i.e, $4 \times 3 + 1 = 13$
Each variable we give 5 cases: a_{min}, a_{min+1}, $a_{nominal}$, a_{max-1}, a_{max}.

for RCT → Total no. of test cases = $6*n+1$ i.e, $6 \times 3 + 1 = 19$
Each variable we give 7 cases: a_{min-1}, a_{min}, a_{min+1}, $a_{nominal}$, a_{max-1}, a_{max}, a_{max+1}.

* Nominal = $(a_{min} + a_{max}) / 2$

Boundary Value Test Cases:

Test Case	a	b	c	Expected Output	Output
1	50	50	1	Isosceles Triangle	✓
2	50	50	2	Isosceles Triangle	✓
3	50	50	50	Equilateral Triangle	✓
4	50	50	99	Isosceles Triangle	✓
5	50	50	100	Triangle not possible	✓
6	50	1	50	Isosceles Triangle	✓
7	50	2	50	Isosceles Triangle	✓
8	50	99	50	Isosceles Triangle	✓


Start
INPUT side1, side2, side3
if (side1 < 1 or side1 > 100 or side2 < 1 or side2 > 110 or side3 < 1 or side3 > 100)
Print "Out of domain Input"
IF (side1 = side2 and side2 = side3)
Print "Equilateral Triangle"
if (side1 = side2 or side2 = side3)
Print "Isosceles Triangle"
if (side1 != side2 or side2 != side3 or side3 != side1)
Print "Scalene Triangle"
Print "Triangle not possible"
Stop

9	50	100	50	Triangle not possible	✓
10	1	50	50	Isosceles Triangle	✓
11	2	50	50	Isosceles Triangle	✓
12	99	50	50	Isosceles Triangle	✓
13	100	50	50	Triangle not possible	✓

Robust case Testing cases:

Test case	a	b	c	Expected Output	Output
1	50	50	0	Out of domain Input	✓
2	50	50	1	Isosceles Triangle	✓
3	50	50	2	Isosceles Triangle	✓
4	50	50	50	Equilateral Triangle	✓
5	50	50	99	Isosceles Triangle	✓
6	50	50	100	Triangle not possible	✓
7	50	50	101	Out of domain Input	✓
8	50	0	50	Out of domain Input	✓
9	50	1	50	Isosceles Triangle	✓
10	50	2	50	Isosceles Triangle	✓
11	50	99	50	Isosceles Triangle	✓
12	50	100	50	Triangle Not possible	✓
13	50	101	50	Out of domain Input	✓
14	0	50	50	Out of domain Input	✓
15	1	50	50	Isosceles Triangle	✓
16	2	50	50	Isosceles Triangle	✓
17	99	50	50	Isosceles Triangle	✓
18	100	50	50	Triangle not possible	✓
19	101	50	50	Out of domain Input	✓

*

SOURCE CODE:

```
#include<iostream>
using namespace std;

int main()
{

int side1, side2, side3;

cout<<"Determining the type of Triangle"<<endl;
cout<<"Enter side1 : "<<endl;
cin>>side1;
cout<<"Enter side2 : "<<endl;
cin>>side2;
cout<<"Enter side3 : "<<endl;
cin>>side3;

if((side1<1)||(side2<1)||( side3<1)||(side1>100)||(side2>100)||(side3>100))
   cout<<"Out of domain Input"<<endl;

else if ((side1 + side2 > side3 && side1 + side3 > side2 && side2 + side3 > side1) && (side1 >
0 && side2 > 0 && side3 > 0))
{
   if (side1 == side2 && side2 == side3)
      cout<<"Equilateral Triangle"<<endl;
   else if (side1 == side2 || side2 == side3 || side1 == side3)
      cout<<"Isosceles Triangle"<<endl;
   else
      cout<<"Scalene Triangle"<<endl;
}
else {
  cout<<"Triangle not possible"<<endl;
}
 return 0;
}
```

OUTPUTS:

```
Determining the type of Triangle
Enter side1 :
50
Enter side2 :
50
Enter side3 :
50
Equilateral Triangle
```

```
Determining the type of Triangle
Enter side1 :
50
Enter side2 :
1
Enter side3 :
50
Isosceles Triangle
```

```
Determining the type of Triangle
Enter side1 :
50
Enter side2 :
99
Enter side3 :
100
Scalene Triangle
```

```
Determining the type of Triangle
Enter side1 :
50
Enter side2 :
100
Enter side3 :
50
Triangle not possible
```

```
Determining the type of Triangle
Enter side1 :
0
Enter side2 :
50
Enter side3 :
99
Out of domain Input
```

PRACTICAL 3

Aim: Consider the program of classification of triangle. It's input is a triple of positive integers (say a, b, c) and values for each of these may be from interval [1, 100]. The output may have one of the options given below:-
Obtuse-angled, acute-angled, right-angled or invalid triangle & input values out of range. Find all the du-paths & identify those du-paths that are definition clear. Also, find all du-paths, all uses and all definitions & generate test cases for them.

Theory: Triangle can be classified by their angles. The following are the conditions:-

→ Acute-angled triangle → When all three angles are less than 90°. Then the triangle is called acute-angled triangle.

$\angle a < 90^\circ$, $\angle b < 90^\circ$, $\angle c < 90^\circ$
where a, b, c are the angles.

→ Right-angled triangle → When a triangle contains one right-angle & two acute angles. Then, the triangle is known as a right-angled triangle.

$\angle a = 90^\circ$ or $\angle b = 90^\circ$ or $\angle c = 90^\circ$
$\angle b < 90^\circ$ or $\angle a < 90^\circ$ or $\angle b < 90^\circ$
$\angle c < 90^\circ$ or $\angle c < 90^\circ$ or $\angle a < 90^\circ$
where a, b, c are the angles.

→ Obtuse-angled triangle → When one angle of a triangle is greater

than 90° and two acute angles. Then the triangle is known as obtuse-angled triangle.

→ Invalid/out of range value → When the sum of all the angles of a triangle is greater than 180°. Then, the triangle is known as an "Invalid Triangle". And when any of the angle is greater than 180°, then the value is considered out of range.

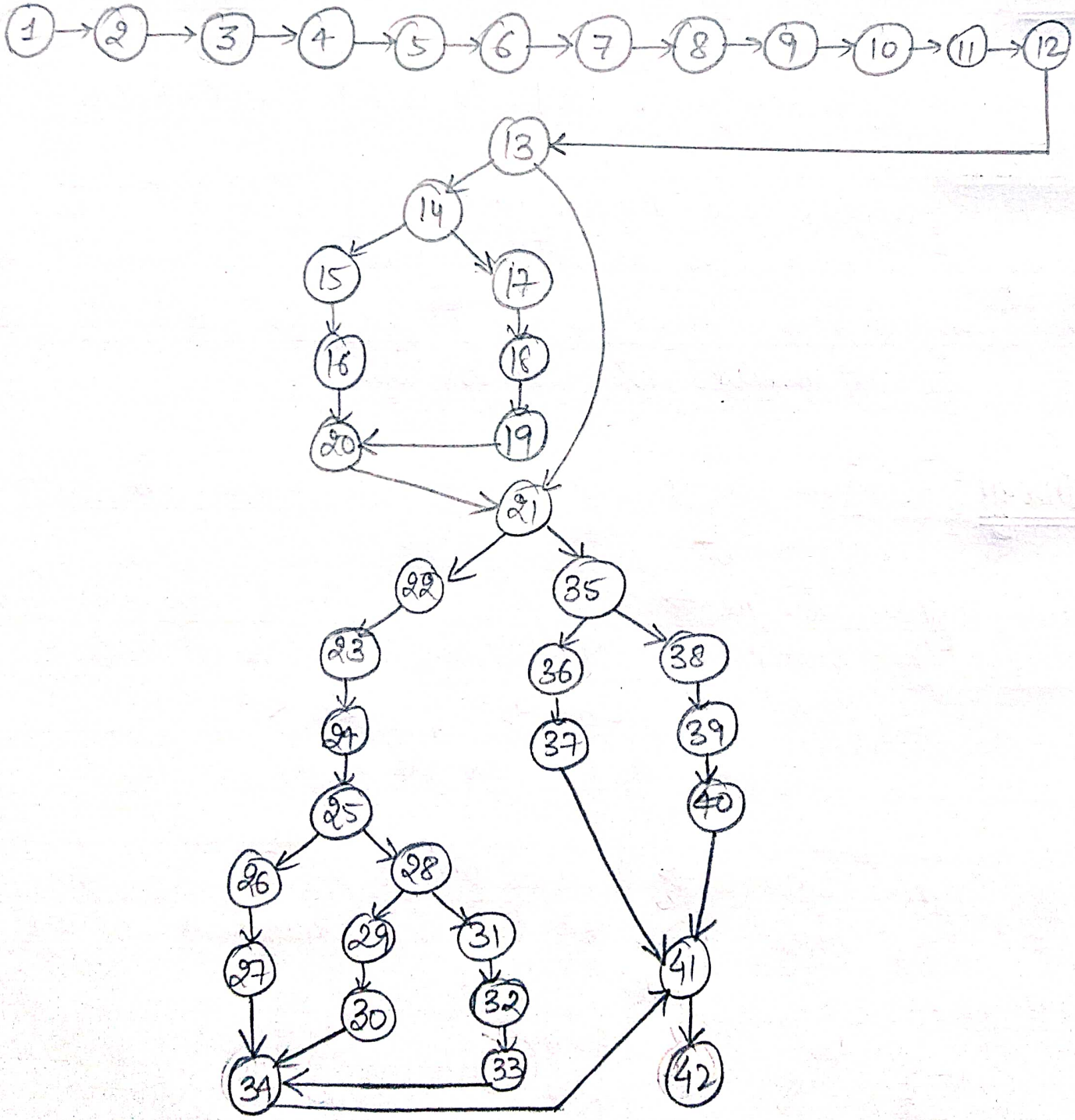
Solution :-

Variable	Define at node	Used at node
a	8	13, 14, 22, 23, 24
b	10	13, 14, 22, 23, 24
c	12	13, 14, 22-24
a1	22	25, 28
a2	23	25, 28
a3	24	25, 28
valid	5, 15, 18	21, 35

Expected Output :-

S.No.	a	b	c	Output	Remark
1	30	20	40	Obtuse	8-13
2	30	20	40	Obtuse	8-14
3	30	20	40	Obtuse	8-16, 20-22
4	–	–	–	–	8-14, 17-22
5	–	–	–	–	8-13, 21, 22
6	30	20	40	Obtuse	8-16, 20-23
7	–	–	–	–	8-14, 17-23

Program Graph :



8	–	–	–	–	8-13, 21-23
9	30	20	40	Obtuse	8-16, 20-24
10	–	–	–	–	8-14, 17-21
11	–	–	–	–	8-13, 21-24
12	30	20	40	Obtuse	10-13
13	30	20	40	Obtuse	10-14
14	30	20	40	Obtuse	10-16, 20-22
15	–	–	–	–	10-14, 17-22
16	–	–	–	–	10-14, 17-23
17	30	20	40	Obtuse	10-16, 20-23
18	–	–	–	–	10-14, 17-24
19	–	–	–	–	10-13, 21-24
20	30	20	40	Obtuse	10-16, 20-24
21	–	–	–	–	10-14, 17-24
22	–	–	–	–	10-13, 31-24
23	30	20	40	Obtuse	12, 13
24	30	20	40	Obtuse	12-14
25	30	20	40	Obtuse	12-16, 20-22
26	–	–	–	–	12-16, 20-22
27	–	–	–	–	12, 13, 21-22
28	30	20	40	Obtuse	12-16, 20-23
29	–	–	–	–	12-14, 17, 23
30	–	–	–	–	12, 13, 21-23
31	30	20	40	Obtuse	12-16, 20-24
32	–	–	–	–	12-14, 17-24
33	–	–	–	–	12, 13, 21-24
34	30	20	40	Obtuse	22-25

35	30	40	50	Right	23-25, 28
36	30	20	40	Obtuse	24, 25
37	30	40	50	Right	23, 25, 28
38	30	20	40	Obtuse	24, 25
39	30	40	50	Right	24, 25, 28
40	30	20	40	Obtuse	5-16; 20, 21
41	30	10	15	Out of range	5-14, 17-21
42	102	-1	6	Out of range	5-13, 21
43	30	10	15	-	5-16, 20, 21, 25
44	30	10	15	Out of range	5-14, 17-21, 35
45	102	-1	6	Out of range	5-13, 21, 35
46	30	20	40	Obtuse	15, 16, 20, 21
47	-	-	-	-	15, 16, 20, 21, 35
48	30	10	15	Out of range	18-21
49	30	10	15	Out of range	18-21, 39

All du-paths :-

All du-paths	Definition clear	All du-paths	Definition clear
8-13	Yes	8-14, 17-22	Yes
8-14	Yes	8-13, 21-22	Yes
8-16, 20-22	Yes	8-16, 20-23	Yes
8-14, 17-23	Yes	10-13	Yes
8-13, 21-23	Yes	10-14	Yes
8-16, 20-24	Yes	10-16, 20-22	Yes
8-14, 17-24	Yes	10-14, 17-22	Yes
8-13, 21-24	Yes	10-13, 21, 22	Yes
12-14, 17-22	Yes	10-16, 20-23	Yes

12, 13, 21, 22	Yes	10-14, 17-23	Yes
12, 13	Yes	12-14	Yes
12-16, 20-23	Yes	12-16, 20-22	Yes
12-14, 17-23	Yes	23-25, 28	Yes
12, 13, 21-23	Yes	24, 25	Yes
12-16, 20-24	Yes	24, 25, 28	Yes
12-14, 17-24	Yes	5-16, 20, 21	No
12, 13, 21-24	Yes	5-14, 17-21	No
22-25	Yes	5-13, 21	Yes
22-25, 28	Yes	5-16, 20, 21, 35	No
23, 25	Yes	5-14, 17-26, 35	No
8-13, 21, 25	Yes	15, 16, 20, 21	Yes
15, 16, 20, 21	Yes	18-21, 35	Yes

Test cases for all definitions :-

S.No.	a	b	c	Output	Remark
1	30	20	40	Obtuse	8-13
2	30	20	40	Obtuse	10-13
3	30	20	40	Obtuse	12, 13
4	30	20	40	Obtuse	22-25
5	30	20	40	Obtuse	23-25
6	30	20	40	Obtuse	24-25
7	30	20	40	Obtuse	5-16, 20, 21
8	30	20	40	Obtuse	5-16, 20, 21
9	30	10	13	Invalid	18-21

DELTA Pg No. 10
Date / /

All use paths :-

All uses	Definition clear	All uses	Definition clear
8-13	Yes	23-25, 28	Yes
8-14	Yes	24, 25	Yes
8-16, 20-22	Yes	24, 25, 28	Yes
8-16, 20-23	Yes	5-16, 20, 21	No
8-16, 20-24	Yes	5-14, 17-21, 35	No
10-13	Yes	15, 16, 20, 21	Yes
10-14	Yes	15, 16, 20, 21, 35	Yes
10-16, 20-24	Yes	12-14	Yes
10-13, 21-23	Yes	12-16, 20, 21, 22	Yes
10-16, 20-24	Yes	12-16, 20-23	Yes
12, 13	Yes	18, 21	Yes
12-16, 20, 24	Yes	18-21, 35	Yes
22-25	Yes	22-25, 28	Yes
23-25	Yes	-	-Yes-

Test cases for all use cases :-

S.No.	a	b	c	Output	Remark
1	30	20	40	Obtuse	8-13
2	30	20	40	Obtuse	8-14
3	30	20	40	Obtuse	8-16, 20-22
4	30	20	40	Obtuse	8-16, 20-23
5	30	20	40	Obtuse	8-16, 20-24
6	30	20	40	Obtuse	10-13
7	30	20	40	Obtuse	10-14
8	30	20	40	Obtuse	10-16, 20-22

9	30	20	40	Obtuse	10-13-21-23
10	30	20	40	Obtuse	10-16, 20-24
11	30	20	40	Obtuse	12, 13
12	30	20	40	Obtuse	12-14
13	30	20	40	Obtuse	12-16, 20, 21, 22
14	30	20	40	Obtuse	12-16, 20-23
15	30	20	40	Obtuse	12-16, 20-24
16	30	20	40	Obtuse	22-25
17	30	40	50	Obtuse	22-25, 28
18	30	20	40	Obtuse	23-25
19	30	40	50	Right	23-25, 28
20	30	20	40	Obtuse	24, 25
21	30	40	50	Right	24, 25, 28
22	30	20	40	Obtuse	5-16, 20-21
23	-	-	-	-	5-16, 20, 21, 35
24	30	10	15	Invalid	18-21
25	30	10	15	Invalid	18-21, 35

Result: The aim has been performed & then implemented successfully.

SOURCE CODE:

```
#include<iostream>
using namespace std;
int main()
{
double a,b,c,a1,a2,a3;
int valid = 0;
cout<<"Enter first angle of the triangle:  ";
cin>>a;
cout<<"Enter second angle of the triangle:  ";
cin>>b;
cout<<"Enter third angle of the triangle:  ";
cin>>c;

if(a>0 && a<=100 && b>0 && b<=100 && c>0 && c<=100)
        {
        if((a+b) > c && (b+c) > a && (a+c) >b)
                valid = 1;
        else
                valid = -1;
        }

if(valid==1)    {
        a1 = (a*a + b*b)/(c*c);
        a2 = (b*b + c*c)/(a*a);
        a3 = (c*c + a*a)/(b*b);

if(a1<1 || a2<1 || a3<1)
        cout<<"Obtuse-angled triangle";
else if(a1==1||a2==1||a3==1)
        cout<<"Right-angled triangle";
else
        cout<<"Acute-angled triangle";
}
else if(valid == 1)
        cout<<"Invalid triangle";
else
        cout<<"Out of range";
return 0;
}
```

OUTPUTS:

```
Enter first angle of the triangle: 40
Enter second angle of the triangle: 50
Enter third angle of the triangle: 30
Right-angled triangle
```

```
Enter first angle of the triangle: 1
Enter second angle of the triangle: 50
Enter third angle of the triangle: 50
Acute-angled triangle
```

```
Enter first angle of the triangle: 20
Enter second angle of the triangle: 30
Enter third angle of the triangle: 40
Obtuse-angled triangle
```

```
Enter first angle of the triangle: 20
Enter second angle of the triangle: 40
Enter third angle of the triangle: 60
Out of range
```

PRACTICAL 4

Aim: Consider an automated banking application. The user can dial the bank form on PC, provide a 6 digit password with a series of keyboard commands that activate the banking application. The software for the application accepts the data in the following form:

Area code: Blank / 3 digit number
Prefix: 3 digit number not beginning with 1 or 0
Suffix: 4 digit number.
Password: 6 character, alphanumeric
Commands: Check status, Deposit, withdrawal

Design adhoc test case to test the system.

Theory: Adhoc testing is an informal or unstructured software testing that aims to break the testing process to find possible defects or errors at any possible stage.

Adhoc testing is done randomly & it is usually an unplanned activity to create that, following are types of adhoc testing:

Buddy testing: Two buddies from development team & one from test team mutually work on identifying defects in the same module. It helps testing develop better test cases while development team can also make design changes.

Pair testing: Two testers are assigned to same modules & they share ideas & work on the same systems to find defects. One tester executes the test while other one records the findings.

DELTA Pg No. 13
Date / /

Monkey Testing: Testing is performed randomly w/o any test cases in order to break the system.

Test Cases:-

Input					Expected output	Actual output
Area code	Prefix	Suffix	Password	Command		
123	222	2323	sdjaes	Check status	Invalid	Invalid
123	122	3232	3xfcsf	withdraw	Invalid	Invalid
	221	323	dxd2ds	Deposit	Invalid	Invalid
123	011	3332	Dfd2ss	Check status	Invalid	Invalid
123	4567	2232	VgThgs	deposit	Invalid	Invalid
123	523	43332	Sd55sf	withdraw	Invalid	Invalid
123	234	3456	32232	withdraw	Invalid	Invalid
123	23	3457	Edjjod	withdraw	Invalid	Invalid
	234	12ab	3d333d	deposit	Invalid	Invalid
012	345	2222	abc12a	withdraw	Invalid	Invalid
97	345	2221	abcde12	deposit	Invalid	Invalid
321	657	7865	quer34	withdraw	Accepted	Accepted
	345	6565	fsword1	check status	Accepted	Accepted
123	234	3333	DjEw3T	deposit	Accepted	Accepted
	203	3562	Aser415	deposit	Accepted	Accepted

Result: Created a automated banking application & generated its test cases.

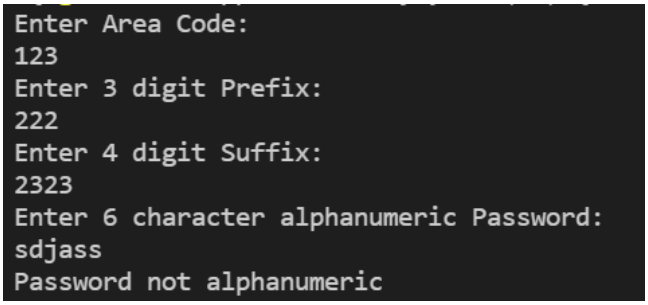
SOURCE CODE:

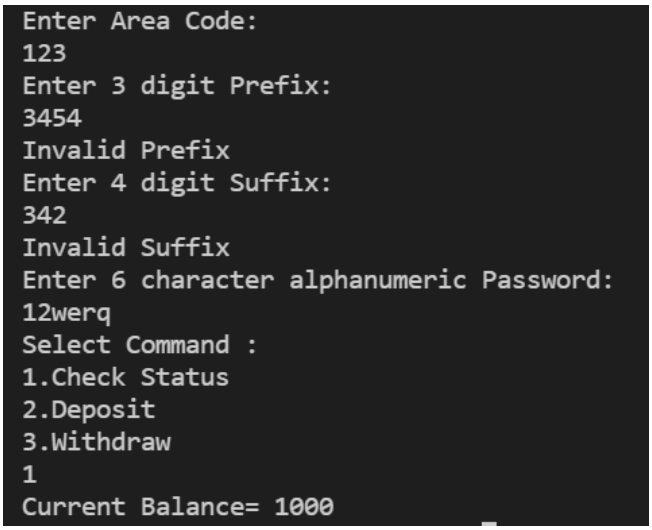
```
#include<bits/stdc++.h>
using namespace std;
static int balance=1000;

bool isAlphaNum(string s)
 {
    for(int i=0;i<s.size();i++)
    {
       if(!((s[i]>='A' && s[i]<='Z') || (s[i]>='a' && s[i]<='z') && (s[i]>='0' && s[i]<='9' )))
       {
          return false;
       }
    }
    return true;
 }

void withdraw()
        {
                int amnt;
                cout<<"Enter amount to be withdrawn: "<<endl;
                cin>>amnt;

                if(balance<amnt)
                {
                        cout<<"Not Enough balance!"<<endl;
                }
                else
                {
                        balance -= amnt;
                        cout<<"Withdrawn Successfully"<<endl;
                }
        }

void check()
        {
                cout<<"Current Balance= "<<balance<<endl;
        }

void deposit()
        {
                int dep;
                cout<<"Enter amount to be deposited: "<<endl;
                cin>>dep;
                if(dep>=0)
                {
                        balance += dep;
                        cout<<"Amount deposited Successfully"<<endl;
                }
                else
```

```
		{
			cout<<"Enter positive amount"<<endl;
		}
	}

int main()
	{

		string alphapass="";
		string areacode;
		cout<<"Enter Area Code: "<<endl;
		getline(cin, areacode);

		if(!(areacode.length()==0 || (areacode.length()==3)))
		{
			cout<<"Invalid Area Code"<<endl;
		}

		string pre;
		cout<<"Enter 3 digit Prefix: "<<endl;
		getline(cin,pre);

		int prefix = stoi(pre);
		if(!(prefix/100>1 && prefix/1000==0))
		{
			cout<<"Invalid Prefix"<<endl;
		}

		string suffix;
		cout<<"Enter 4 digit Suffix: "<<endl;
		getline(cin, suffix);

		if(suffix.length()!=4)
		{
			cout<<"Invalid Suffix"<<endl;
		}

		cout<<"Enter 6 character alphanumeric Password: "<<endl;
		getline(cin, alphapass);

		if(alphapass.length()!=6)
		{
			cout<<"Invalid Password"<<endl;
		}
		else if(!(isAlphaNum(alphapass)))
		{
			cout<<"Password not alphanumeric"<<endl;
		}

		char ch='y';
		while(ch=='y'||ch=='Y')
```

```
        {
                int command;
                cout<<"Select Command : \n1.Check Status\n2.Deposit\n3.Withdraw"<<endl;
                cin>>command;
                switch(command)
                {
                        case 1: check();
                                break;
                        case 2: deposit();
                                break;
                        case 3: withdraw();
                                break;
                        default: cout<<"Choice not Available"<<endl;
                }
                cout<<"Do you wish to continue? (y/n):";
                cin>>ch;
        }
}
```

OUTPUTS:





DELTA Pg No. 14
Date / /

PRACTICAL 5

Aim: Prepare a test plan document for Library Management System.

TEST PLAN LIBRARY MANAGEMENT SYSTEM

→ Table of Contents:

1) Introduction

Every college/school has their library both for teachers & students use the traditional system to manage them is either keeping track of them in a register or keeping track of a similar entry in computer. Its very time consuming. Online Library Management System helps in solving this issue.

2) Purpose

The ~~Libr~~ Library Management system is an online application ~~Testing~~ for testing a librarian in managing book library in a university. This test plan document support the following objective:

→ Identify existing project information & software that should be tested.
→ List the recommended tools requirements.
→ Recommend & describe the testing strategies to be employed.
→ List the deliverable elements of the test activities.

3) Scope

The system that is to developed provides the related information on students & system administration.

→ Creating a system administrator who will be the role user managing the system on the backend.
→ System admin can add/delete/view/~~edit~~ the ~~books~~.
→ Admin can add/delete/view/edit the books issued.
→ Admin can search for the books issued.

4) Testing Strategies

The aim of the system testing process is to determine all defects in the project.

(4.1) Unit Testing :-

In order to test a single module, we need to provide a complete environment & besides the module we could require.

→ The procedure belonging to other modules.
→ Non local data structures that module accesses.

→ A procedure to call the functions of the module or for test.

Unit testing was done on each of every modules that it describes under the module description:

(i) Test for admin module:
  (a) Testing admin for login
  (b) Student account registration

(ii) Test for student login module:
  (a) Test for student login interface
  (b) Test for account creation.

(iii) Test for teacher login module:
  (a) Test for teacher login interface.

(4.2) System & Integration Testing :-
The primary objective is to test the module interfaces
→ UI user interface module, w/c is visible to end user.
→ DBMS is the database management system w/c has all data.
→ VAL is the validation module.
→ CNT : these contents are displayed in reports.

(4.3) Performance & Stress Testing :-
Stress testing involving beyond normal operation capacity.

(4.4) User Acceptance Testing :-
There are different types of acceptance testing. The most common among them is the User Acceptence (UA).

8) Test Schedule

S.No.	Task	Days	St. time	End time	Responsibility
1	Understanding and analysis	5	2 July	7 July	Team
2	Generating test cases	10	7 July	17 July	Member 1
3	Test case documentation	40	7 July	17 July	Member 2
4	Verify env. setup	1	17 Aug	17 Aug	Member 3
5	Unit testing	10	18 Aug	28 Aug	Member 4
6	IUT testing	15	7 Sept	22 Sept	Member 5
7	Final testing	15	21 Sept	24 Sept	End user 1
8	Eval. test criteria	2	22 Sept	24 Sept	Member 1
9	Summary report	1	25 Sept	25 Sept	Team

5) Features to be tested
→ GUI testing
→ Database testing
→ Basic operations add/delete/etc.
→ Advance operations
→ BIU.

6) Hardware requirements / Software requirements
→ Windows - OS
→ Python - language
→ MySQL - database
→ Visual Studio Code - IDE

7) Environment requirements
(i) Mainframe → Specify both the necessary & designed properties of test acquirement
(ii) Work Stations → Computers provided in the libraries to be used by admins & students.

9) Risk & Mitigation
Keep battery back up & avoid electricity issues.

10) Tools
(i) Selenium
(ii) Appp.

Result: Successfully prepared a test plan for a Library Management System.

PRACTICAL 6

Aim: Study of any testing tool (WinRunner)

Theory: Winrunner is a program that is responsible for the automated testing of software.
Winrunner is a mercury interactives enterprise functional testing tool for Microsoft Windows Softwares.

Importance of Automated testing
- → Reduced testing time
- → Consistent test procedures
- → Reduced QA cost
- → Improved test productivity
- → Proof of adequate testing
- → Good for doing tedious coost.

Winrunner uses:
- → Sophisticated automated tests can be created & run on an application.
- → A series of wizards can be used to create tests in an automated manner.
- → It has ability to record various interactions & transform them into scripts.
- → When the user makes an interaction with the GUI, this interaction can be recorded.

WinRunner testing process:

(i) Create GUI map
(ii) Create tests
(iii) Debug tests
(iv) Run tests
(v) View results
(vi) Report defects

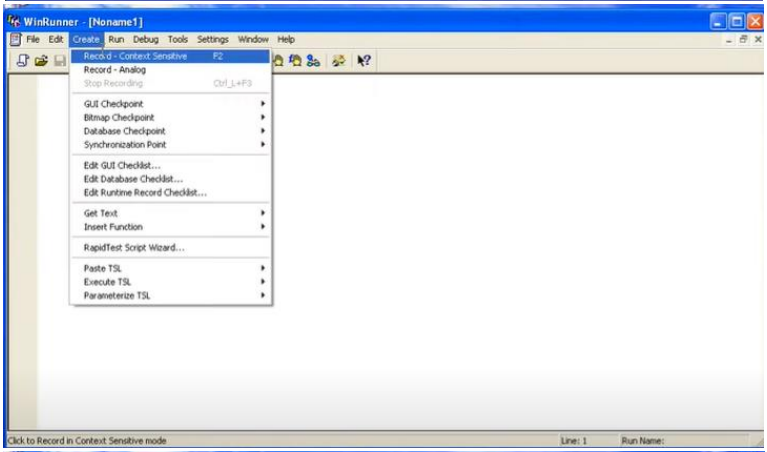
* When the test is completed, it provides detailed information regarding the results.
* It will show errors that were found & will also provide detailed information about them.
* It will test the complete program in a way that is similar to user interactions.
* The recover manager is a powerful tool that can assist users with various scenarios.

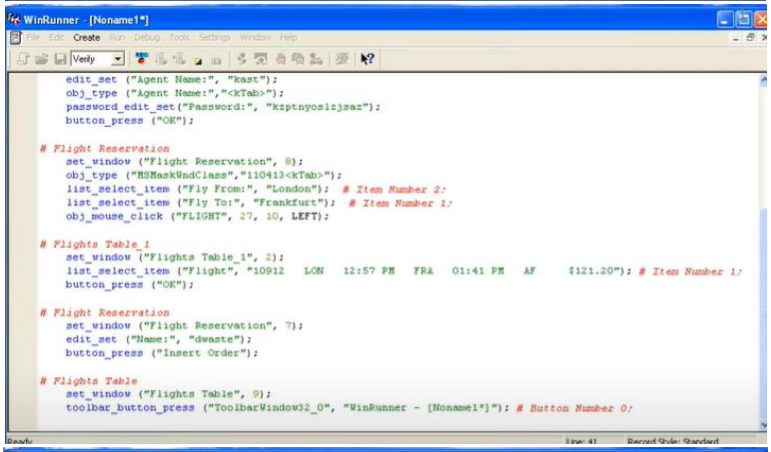
The goal of WinRunner is to make sure business process are properly carried out, it uses TSL (Test Script Language).

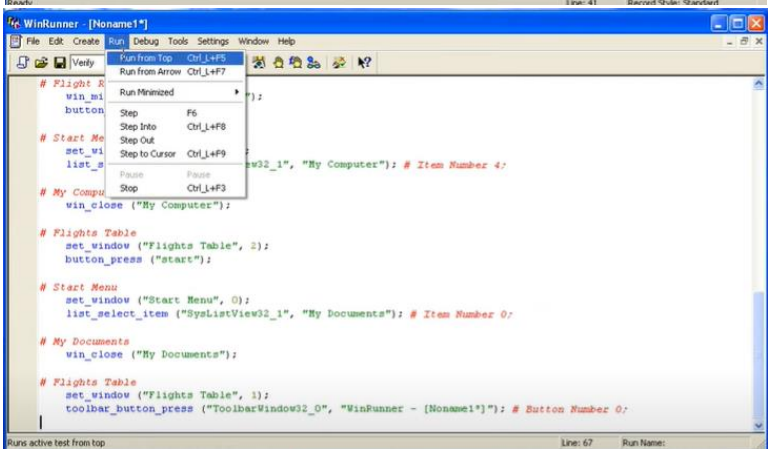
Result: Successfully studied & analyzed the WinRunner testing tool.

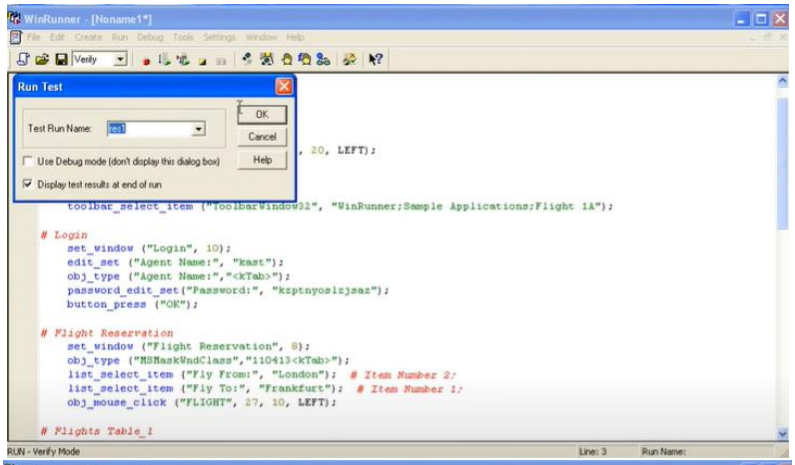
WINRUNNER:

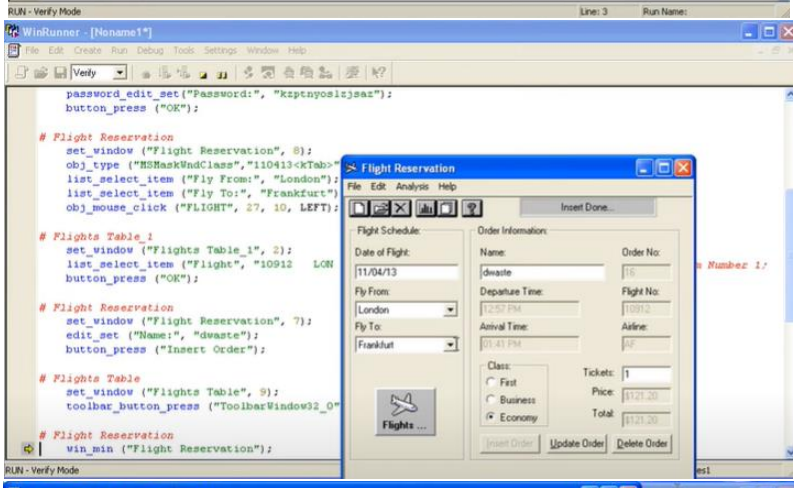

WinRunner - [tstinit]
Welcome To WinRunner
MERCURY INTERACTIVE
WELCOME TO
WinRunner®
New Test
Create a new test script
Open Test
Open an existing test script
Quick Preview
View a Quick Preview of WinRunner
Show on Startup

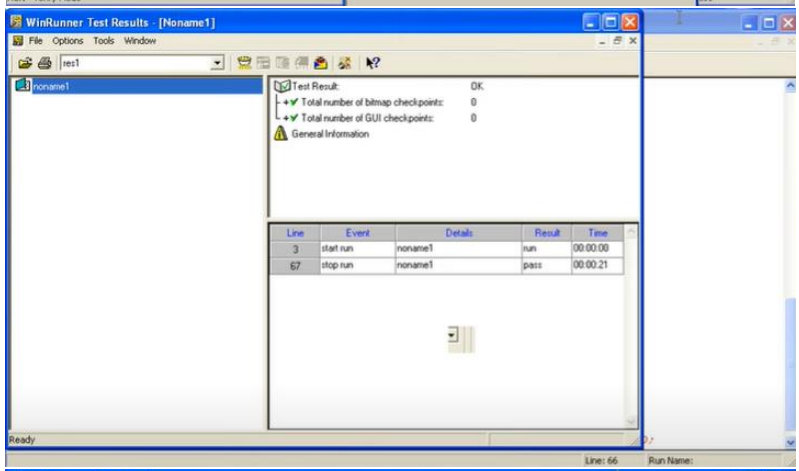

WinRunner - [Noname1]
Record - Context Sensitive F2
Record - Analog
Stop Recording Ctrl_L+F3
GUI Checkpoint
Bitmap Checkpoint
Database Checkpoint
Synchronization Point
Edit GUI Checklist...
Edit Database Checklist...
Edit Runtime Record Checklist...
Get Text
Insert Function
RapidTest Script Wizard...
Paste TSL
Execute TSL
Parameterize TSL
Click to Record in Context Sensitive mode

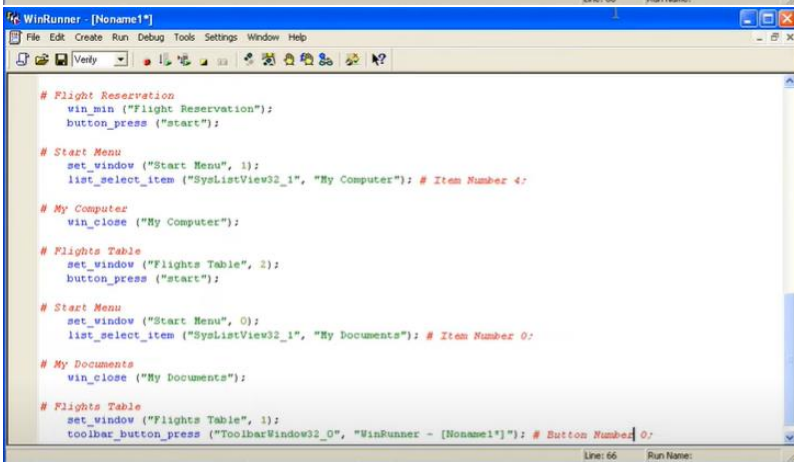

WinRunner - [Noname1*]
edit_set ("Agent Name:", "kast");
obj_type ("Agent Name:","<kTab>");
password_edit_set("Password:", "kzptnyoslzjsaz");
button_press ("OK");
Flight Reservation
set_window ("Flight Reservation", 8);
obj_type ("MSMaskWndClass","110413<kTab>");
list_select_item ("Fly From:", "London"); # Item Number 2;
list_select_item ("Fly To:", "Frankfurt"); # Item Number 1;
obj_mouse_click ("FLIGHT", 27, 10, LEFT);
Flights Table 1
set_window ("Flights Table 1", 2);
list_select_item ("Flight", "10912 LON 12:57 PM FRA 01:41 PM AF $121.20"); # Item Number 1;
button_press ("OK");
Flight Reservation
set_window ("Flight Reservation", 7);
edit_set ("Name:", "dwaste");
button_press ("Insert Order");
Flights Table
set_window ("Flights Table", 9);
toolbar_button_press ("ToolbarWindow32_0", "WinRunner - [Noname1*]"); # Button Number 0;


WinRunner - [Noname1*]
Run from Top Ctrl_L+F5
Run from Arrow Ctrl_L+F7
Run Minimized
Step F6
Step Into Ctrl_L+F8
Step Out
Step to Cursor Ctrl_L+F9
Pause Pause
Stop Ctrl_L+F3
win_close ("My Computer");
Flights Table
set_window ("Flights Table", 2);
button_press ("start");
Start Menu
set_window ("Start Menu", 0);
list_select_item ("SysListView32_1", "My Documents"); # Item Number 0;
My Documents
win_close ("My Documents");
Flights Table
set_window ("Flights Table", 1);
toolbar_button_press ("ToolbarWindow32_0", "WinRunner - [Noname1*]"); # Button Number 0;
Runs active test from top


WinRunner - [Noname1*]
Run Test
Test Run Name:
res1
OK
Cancel
Help
Use Debug mode (don't display this dialog box)
Display test results at end of run
toolbar_select_item ("ToolbarWindow32", "WinRunner;Sample Applications;Flight 1A");
Login
set_window ("Login", 10);
edit_set ("Agent Name:", "kast");
obj_type ("Agent Name:","<kTab>");
password_edit_set("Password:", "kzptnyoslzjsaz");
button_press ("OK");
Flight Reservation
set_window ("Flight Reservation", 8);
obj_type ("MSMaskWndClass","110413<kTab>");
list_select_item ("Fly From:", "London");
list_select_item ("Fly To:", "Frankfurt");
obj_mouse_click ("FLIGHT", 27, 10, LEFT);
Flights Table 1
RUN - Verify Mode
Line: 3
Run Name:


WinRunner - [Noname1*]
Flight Reservation
File Edit Analysis Help
Insert Done...
Flight Schedule:
Date of Flight:
11/04/13
Fly From:
London
Fly To:
Frankfurt
Flights ...
Order Information:
Name:
dwaste
Order No:
Departure Time:
Flight No:
Arrival Time:
Airline:
Class:
First
Business
Economy
Tickets:
Price:
Total:
Insert Order
Update Order
Delete Order
password_edit_set("Password:", "kzptnyoslzjsaz");
button_press ("OK");
Flight Reservation
set_window ("Flight Reservation", 8);
obj_type ("MSMaskWndClass","110413<kTab>");
list_select_item ("Fly From:", "London");
list_select_item ("Fly To:", "Frankfurt");
obj_mouse_click ("FLIGHT", 27, 10, LEFT);
Flights Table 1
set_window ("Flights Table 1", 2);
list_select_item ("Flight", "10912 LON
button_press ("OK");
Flight Reservation
set_window ("Flight Reservation", 7);
edit_set ("Name:", "dwaste");
button_press ("Insert Order");
Flights Table
set_window ("Flights Table", 9);
toolbar_button_press ("ToolbarWindow32_0"
Flight Reservation
win_min ("Flight Reservation");
RUN - Verify Mode


WinRunner Test Results - [Noname1]
File Options Tools Window
res1
noname1
Test Result:
OK
Total number of bitmap checkpoints:
0
Total number of GUI checkpoints:
0
General Information
Line
Event
Details
Result
Time
3
start run
noname1
run
00:00:00
67
stop run
noname1
pass
00:00:21
Ready
Line: 66
Run Name:


WinRunner - [Noname1*]
Flight Reservation
win_min ("Flight Reservation");
button_press ("start");
Start Menu
set_window ("Start Menu", 1);
list_select_item ("SysListView32_1", "My Computer"); # Item Number 4;
My Computer
win_close ("My Computer");
Flights Table
set_window ("Flights Table", 2);
button_press ("start");
Start Menu
set_window ("Start Menu", 0);
list_select_item ("SysListView32_1", "My Documents"); # Item Number 0;
My Documents
win_close ("My Documents");
Flights Table
set_window ("Flights Table", 1);
toolbar_button_press ("ToolbarWindow32_0", "WinRunner - [Noname1*]"); # Button Number 0;
Line: 66
Run Name:

DELTA Pg No. 20
Date / /

PRACTICAL 7

Aim: Study of any testing tool (Selenium)

Theory: Selenium is a robust set of tools that support rapid development of test automation for web based applications.

Selenium Components →

→ Selenium is composed of 4 major tools :-
(i) IDE (ii) RC (iii) WebDriver (iv) Grid

→ Selenium RC provides an API & library for each of its supported languages HTML, Java, C#, Pearl, Ruby.

Selenium grids

It allows the selenium RC solution to scale for largest test cases / suites that must run in multiple environments with Selenium RC are running on various operating system and browser configuration, each of those when launching register with a hub.

Flexibility & Extensibility

Selenium highly flexible. Selenium IDE allows for addition of user defined user extensions. for creating additional commands customized to the user needs.

Test Suite

A test suite is a collection of tests. Often one will run all the tests in a test suite as one continuous batch job.

When using Selenium IDE, test suites also can be defined using a simple HTML file.
Test suites can also be maintained when using Selenium RC.

Few typical Selenium commands are:-

1) Open: Opens a page using URL.

2) Click: Performs a click.

3) Verify title: Verifies an expected page title.

4) Verify text present: Verifies expected text is somewhere on page.

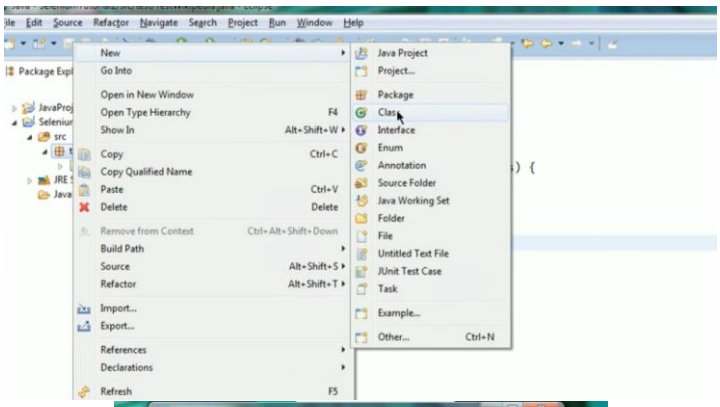
5) Verify table: Verifies a table's contents.

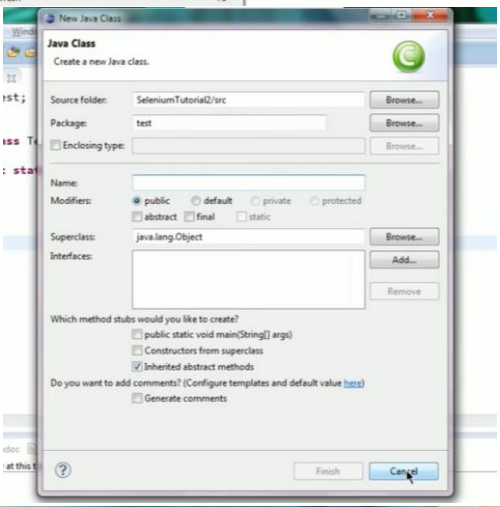
6) Wait for page to load: pauses execution until one expected new page.

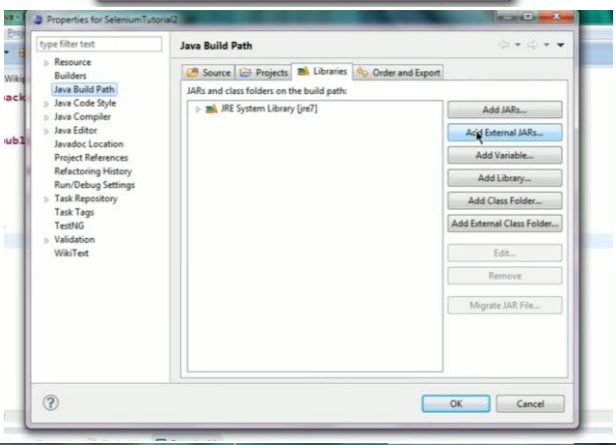
7) Wait for element present: pauses execution until an expected output/ UI is designed by its HTML is present on the page.

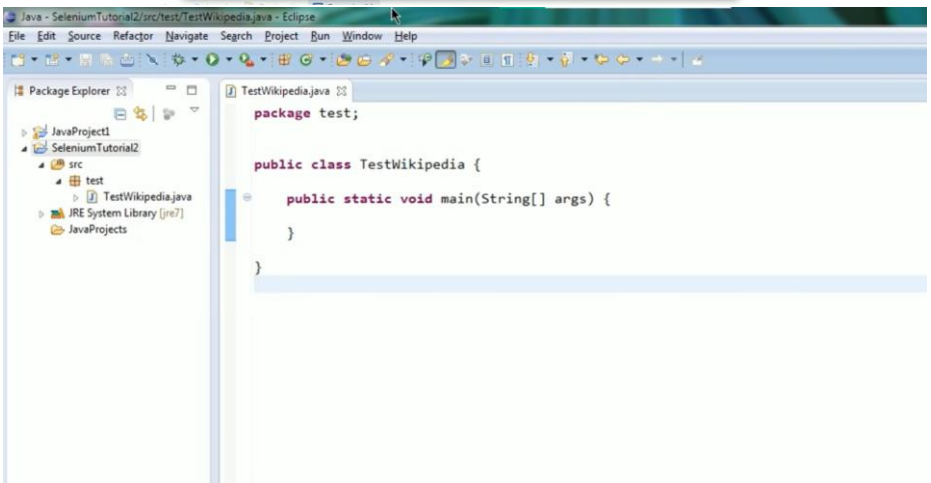
Result: Successfully studied a Web testing Tool - Selenium.

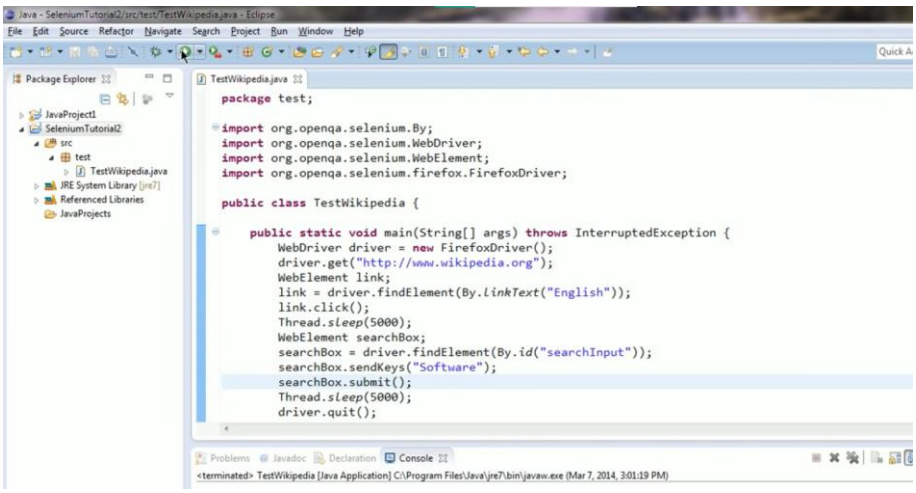
SELENIUM:

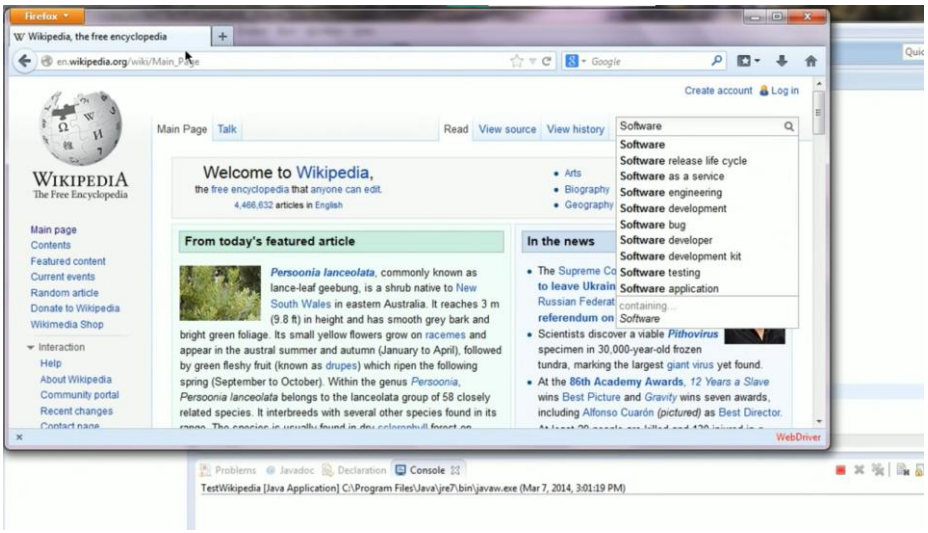

New
Go Into
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
References
Declarations
Refresh F5
Java Project
Project...
Package
Class
Interface
Enum
Annotation
Source Folder
Java Working Set
Folder
File
Untitled Text File
JUnit Test Case
Task
Example...
Other... Ctrl+N

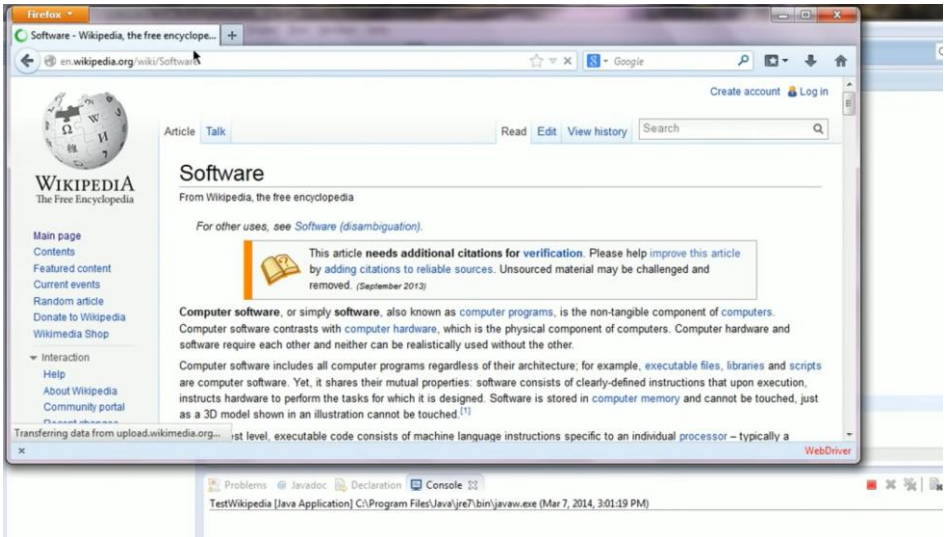

New Java Class
Java Class
Create a new Java class.
Source folder: SeleniumTutorial2/src
Package: test
Enclosing type:
Name:
Modifiers: public default private protected
abstract final static
Superclass: java.lang.Object
Interfaces:
Which method stubs would you like to create?
public static void main(String[] args)
Constructors from superclass
Inherited abstract methods
Do you want to add comments? (Configure templates and default value here)
Generate comments
Finish
Cancel


Properties for SeleniumTutorial2
Java Build Path
Resource
Builders
Java Build Path
Java Code Style
Java Compiler
Java Editor
Javadoc Location
Project References
Refactoring History
Run/Debug Settings
Task Repository
Task Tags
TestNG
Validation
WikiText
Source
Projects
Libraries
Order and Export
JARs and class folders on the build path:
JRE System Library [jre7]
Add JARs...
Add External JARs...
Add Variable...
Add Library...
Add Class Folder...
Add External Class Folder...
Edit...
Remove
Migrate JAR File...
OK
Cancel


Java - SeleniumTutorial2/src/test/TestWikipedia.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JavaProject1
SeleniumTutorial2
src
test
TestWikipedia.java
JRE System Library [jre7]
JavaProjects
TestWikipedia.java
package test;
public class TestWikipedia {
public static void main(String[] args) {
}
}


Java - SeleniumTutorial2/src/test/TestWikipedia.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JavaProject1
SeleniumTutorial2
src
test
TestWikipedia.java
JRE System Library [jre7]
Referenced Libraries
JavaProjects
TestWikipedia.java
package test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class TestWikipedia {
public static void main(String[] args) throws InterruptedException {
WebDriver driver = new FirefoxDriver();
driver.get("http://www.wikipedia.org");
WebElement link;
link = driver.findElement(By.linkText("English"));
link.click();
Thread.sleep(5000);
WebElement searchBox;
searchBox = driver.findElement(By.id("searchInput"));
searchBox.sendKeys("Software");
searchBox.submit();
Thread.sleep(5000);
driver.quit();
Problems Javadoc Declaration Console
<terminated> TestWikipedia [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Mar 7, 2014, 3:01:19 PM)


Firefox
Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Main_Page
Google
Create account Log in
Main Page Talk
Read View source View history
Software
Software
Software release life cycle
Software as a service
Software engineering
Software development
Software bug
Software developer
Software development kit
Software testing
Software application
containing...
Software
WIKIPEDIA
The Free Encyclopedia
Welcome to Wikipedia,
the free encyclopedia that anyone can edit.
4,466,632 articles in English
Arts
Biography
Geography
Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikimedia Shop
Interaction
Help
About Wikipedia
Community portal
Recent changes
From today's featured article
Persoonia lanceolata, commonly known as lance-leaf geebung, is a shrub native to New South Wales in eastern Australia. It reaches 3 m (9.8 ft) in height and has smooth grey bark and bright green foliage. Its small yellow flowers grow on racemes and appear in the austral summer and autumn (January to April), followed by green fleshy fruit (known as drupes) which ripen the following spring (September to October). Within the genus Persoonia, Persoonia lanceolata belongs to the lanceolata group of 58 closely related species. It interbreeds with several other species found in its
In the news
The Supreme Co
to leave Ukrain
Russian Federat
referendum on
Scientists discover a viable Pithovirus specimen in 30,000-year-old frozen tundra, marking the largest giant virus yet found.
At the 86th Academy Awards, 12 Years a Slave wins Best Picture and Gravity wins seven awards, including Alfonso Cuarón (pictured) as Best Director.
WebDriver
Problems Javadoc Declaration Console
TestWikipedia [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Mar 7, 2014, 3:01:19 PM)


Firefox
Software - Wikipedia, the free encyclope...
en.wikipedia.org/wiki/Software
Google
Create account Log in
Article Talk
Read Edit View history
Search
WIKIPEDIA
The Free Encyclopedia
Software
From Wikipedia, the free encyclopedia
For other uses, see Software (disambiguation).
This article needs additional citations for verification. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed. (September 2013)
Computer software, or simply software, also known as computer programs, is the non-tangible component of computers. Computer software contrasts with computer hardware, which is the physical component of computers. Computer hardware and software require each other and neither can be realistically used without the other.
Computer software includes all computer programs regardless of their architecture; for example, executable files, libraries and scripts are computer software. Yet, it shares their mutual properties: software consists of clearly-defined instructions that upon execution, instructs hardware to perform the tasks for which it is designed. Software is stored in computer memory and cannot be touched, just as a 3D model shown in an illustration cannot be touched.[1]
Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikimedia Shop
Interaction
Help
About Wikipedia
Community portal
Transferring data from upload.wikimedia.org...
rst level, executable code consists of machine language instructions specific to an individual processor – typically a
WebDriver
Problems Javadoc Declaration Console
TestWikipedia [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Mar 7, 2014, 3:01:19 PM)

PRACTICAL 8

Aim: To study an test management tool (QA complete).

Theory:

QA complete is a test management tool that can be used for both manual and automated testing. It is a tool with powerful test management capabilities.

Benefits →

→ It can be interpolated with any no. of tools.
→ Customizable with as per testers's need.
→ Requirement and tests can be traced to defect effectively.

Features →

→ **Test case management:** Simple test cases structuring also allows for focussed matrices and clear status report.

→ **Test environment management:** Various environments are linked to individual test cases for effective test coverage.

→ **Defect and issue management:** Mainly tracks the isolation process of bugs.

Steps to setup and work on QA complete.

(i) Login to QA complete tool.
(ii) Create a new release as E-work under the release tab click

add new item.

(iii) Navigate to list management tab → Test Library → Add new folder.

(iv) Navigate to test management → Test set → Create new folder / test using add new item.

(v) To run test sets clicks on the top right corner select the run status and click run option.

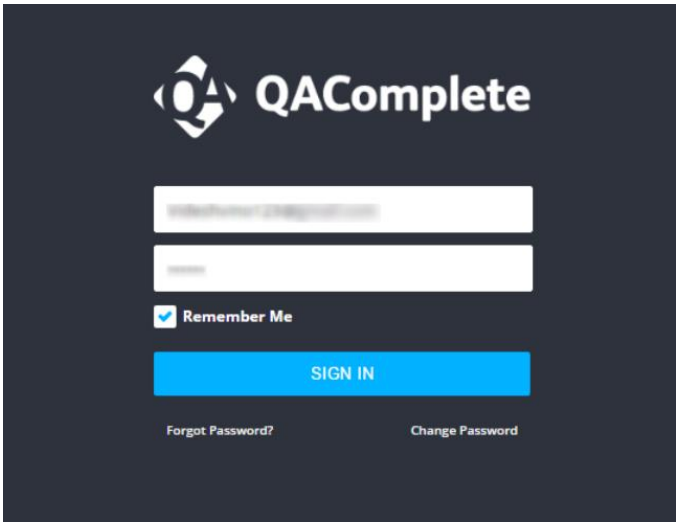
(vi) If any of the steps fail in a test set, it prompts to create a defect.

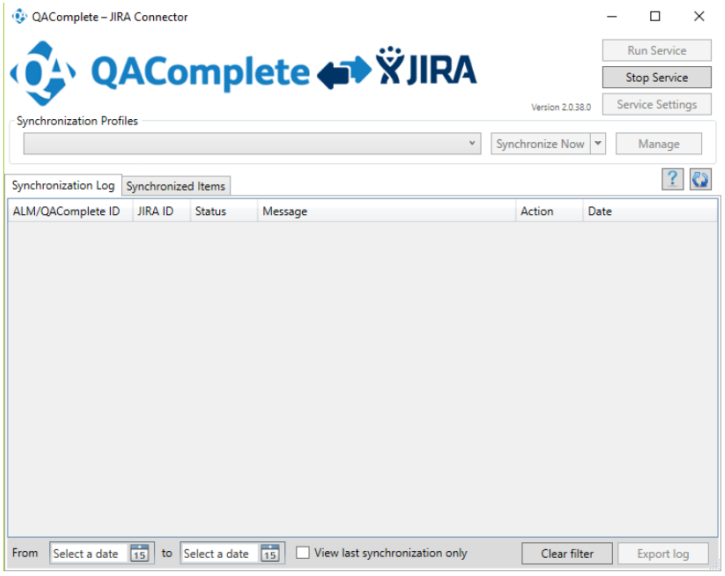
(vii) When the Yes option is selected a defect is created automatically.

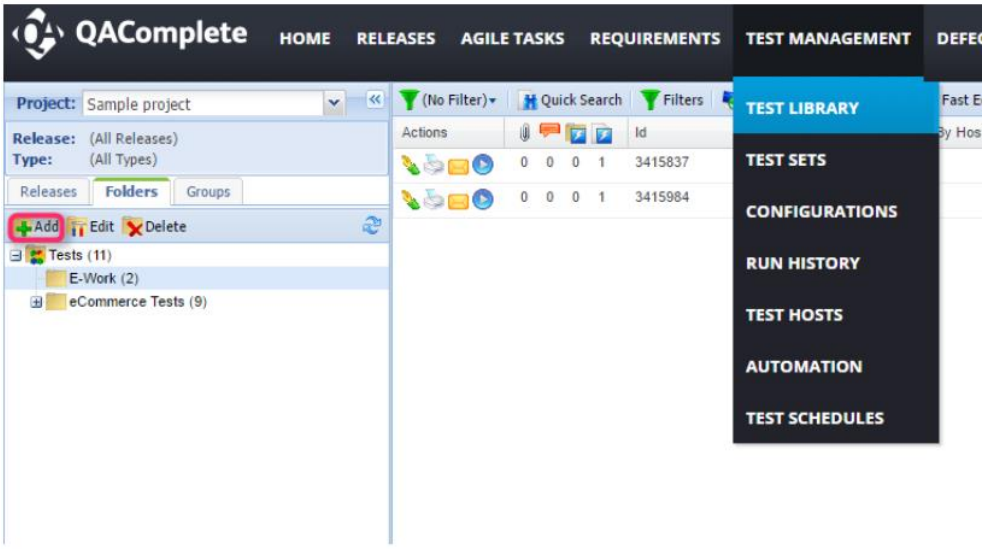
(viii) Navigate to the defect tab & view automatically created bugs.

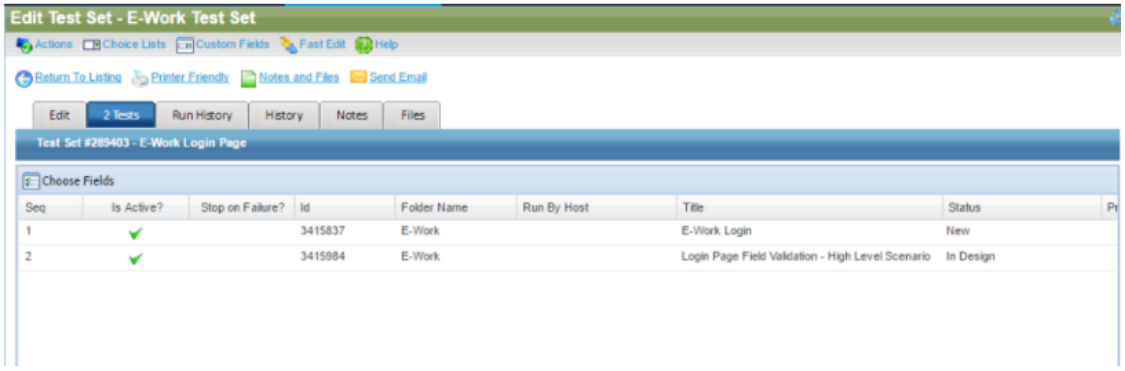
Result: The QA complete tool was successfully studied & explored.

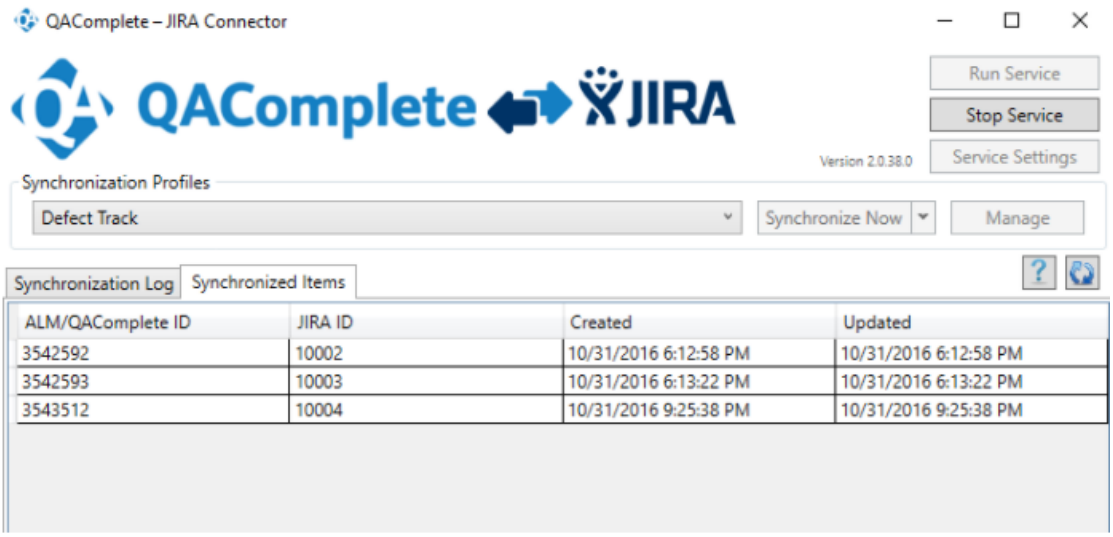
QA COMPLETE:

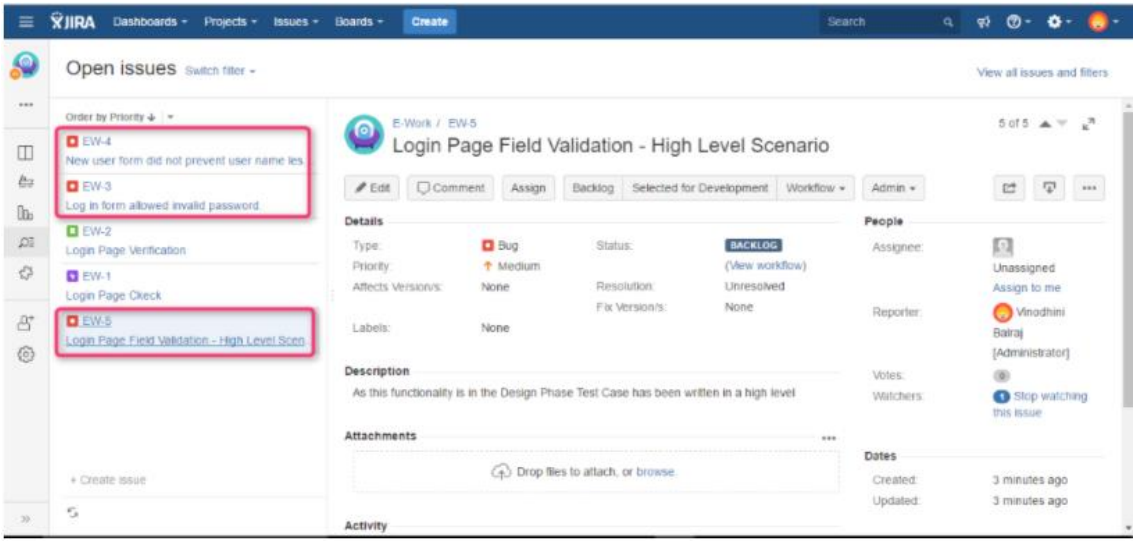

QAComplete
Remember Me
SIGN IN
Forgot Password?
Change Password


QAComplete – JIRA Connector
QAComplete
JIRA
Run Service
Stop Service
Service Settings
Version 2.0.38.0
Synchronization Profiles
Synchronize Now
Manage
Synchronization Log
Synchronized Items
ALM/QAComplete ID
JIRA ID
Status
Message
Action
Date
From
Select a date
to
Select a date
View last synchronization only
Clear filter
Export log


QAComplete
HOME
RELEASES
AGILE TASKS
REQUIREMENTS
TEST MANAGEMENT
Project: Sample project
Release: (All Releases)
Type: (All Types)
Releases
Folders
Groups
Add
Edit
Delete
Tests (11)
E-Work (2)
eCommerce Tests (9)
(No Filter)
Quick Search
Filters
Actions
Id
3415837
3415984
TEST LIBRARY
TEST SETS
CONFIGURATIONS
RUN HISTORY
TEST HOSTS
AUTOMATION
TEST SCHEDULES


Edit Test Set - E-Work Test Set
Actions
Choice Lists
Custom Fields
Fast Edit
Help
Return To Listing
Printer Friendly
Notes and Files
Send Email
Edit
2 Tests
Run History
History
Notes
Files
Test Set #289403 - E-Work Login Page
Choose Fields
Seq
Is Active?
Stop on Failure?
Id
Folder Name
Run By Host
Title
Status
1
3415837
E-Work
E-Work Login
New
2
3415984
E-Work
Login Page Field Validation - High Level Scenario
In Design


QAComplete – JIRA Connector
QAComplete
JIRA
Run Service
Stop Service
Service Settings
Version 2.0.38.0
Synchronization Profiles
Defect Track
Synchronize Now
Manage
Synchronization Log
Synchronized Items
ALM/QAComplete ID
JIRA ID
Created
Updated
3542592
10002
10/31/2016 6:12:58 PM
10/31/2016 6:12:58 PM
3542593
10003
10/31/2016 6:13:22 PM
10/31/2016 6:13:22 PM
3543512
10004
10/31/2016 9:25:38 PM
10/31/2016 9:25:38 PM


JIRA
Dashboards
Projects
Issues
Boards
Create
Search
Open issues
Switch filter
View all issues and filters
Order by Priority
EW-4
New user form did not prevent user name les
EW-3
Log in form allowed invalid password
EW-2
Login Page Verification
EW-1
Login Page Check
EW-5
Login Page Field Validation - High Level Scen
+ Create issue
E-Work / EW-5
Login Page Field Validation - High Level Scenario
5 of 5
Edit
Comment
Assign
Backlog
Selected for Development
Workflow
Admin
Details
Type:
Bug
Priority:
Medium
Affects Version/s:
None
Labels:
None
Status:
BACKLOG
(View workflow)
Resolution:
Unresolved
Fix Version/s:
None
People
Assignee:
Unassigned
Assign to me
Reporter:
Vinodhini Balraj [Administrator]
Votes:
Watchers:
Stop watching this issue
Description
As this functionality is in the Design Phase Test Case has been written in a high level
Attachments
Drop files to attach, or browse.
Dates
Created:
3 minutes ago
Updated:
3 minutes ago
Activity

CONTENT BEYOND SYLLABUS 1

Aim: To verify if various modules of functionalities of ~~the~~ a mobile are working as desired.

Theory:

Test Scenario

A test scenario is a statement describing the functionality of the application to be tested. It is used for end to end testing of a feature and is generally derived from the user cases.

Test scenario can serve as the basis for lower-level test case creation. A single test scenario can cover one or more test cases. Therefore a test scenario has a one-to-many relationship with the test cases.

Scenario testing

Coherent – The test scenario should be based on a coherent story about how the software application is used.

Credible – They should be credible and focus on something that could happen in the real world.

Motivating – They should motivate the stakeholders to get the issues fixed in cases of the failed test scenario.

Complex – The test scenarios normally involve a complex program or

application flow.

Easy to evaluate – The test result of the test scenario should be easy to evaluate as they involve complex logic.

Test Scenario Template

A Test Scenario document can have the below fields :-

Module – The module or the component of the application.

RequirementId – This field is optional and can be linked to the SRS.

TestScenarioId – This field is the identifier of the test ~~sce~~ scenarios.

Description – This description field describes the purpose of the test scenarios.

Module	RequirementId	TestScenarioId	Test Scenario description
Login	[optional]	TS-01	Verify if user is able to login with credentials
	[optional]	TS-02	Verify if user is not able to login with credentials
	[optional]	TS-03	Verify mandatory field, validation on the login fg.

→ Advantages of Test Scenarios

(i) Scenario testing can be carried out relatively faster than using test cases.

(ii) It can ensure good test coverage since the test scenarios are derived from user stories.

(iii) It saves a lot of time. Hence, these are better with projects having time constraints.

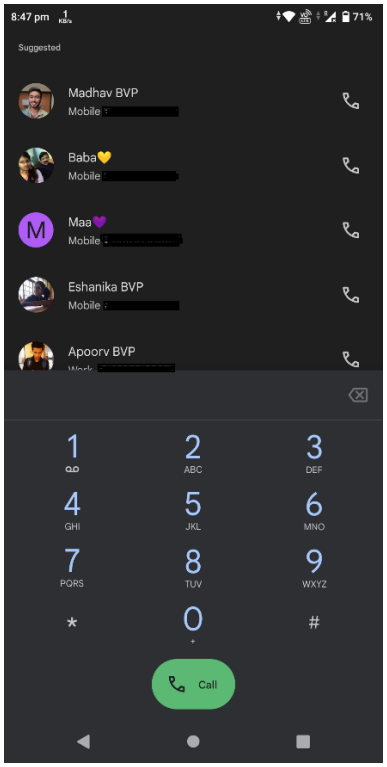
TEST CASES

Test case ID	Test Scenario (functionality)	Checked Yes/No	Successful/ Unsuccessful.
1.	Verify if all the required buttons - no. 0 to 9, calling buttons etc. are present.	Yes	Successful
2.	Verify if user can make a call by selecting a contact person from phone directory.	Yes	Successful
3.	Verify that the user can reject an call.	Yes	Successful
4.	Verify that user can receive an SMS	Yes	Successful
5.	Verify if the user can type & send an SMS	Yes	Successful
6.	Verify the dimension of the mobile.	Yes	Successful
7.	Verify the weight of the mobile.	Yes	Successful
8.	Verify the font type & size of the characters on the keypad.	Yes	Successful
9.	Verify the colour of the mobile phone's body	Yes	Successful
10.	Verify the pressure required to press a key	Yes	Successful
11.	Verify that spacing b/w the keys on the keypad are adequate.	Yes	Successful
12.	Check the type of mobile - smartphone or normal.	Yes	Successful
13.	Check if the mobile is coloured / B&W	Yes	Successful
14.	Check if the mobile phone can be locked out without password or pin.	Yes	Successful
15.	Check the lighting on the mobile screen is adequate - verify it in dark daylight.	Yes	Successful
16.	Verify that the user can receive a call when phone is locked.	Yes	Successful

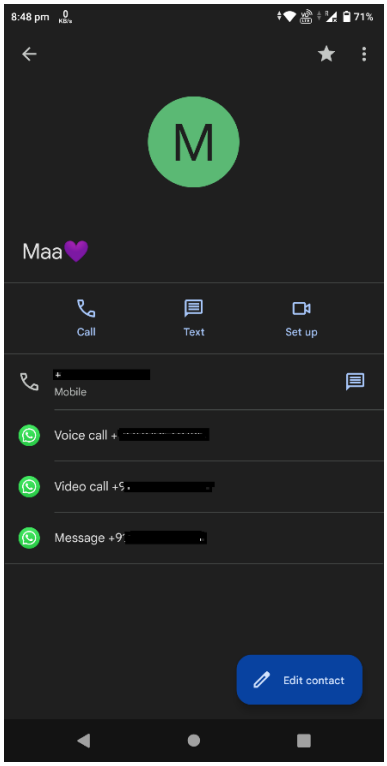
17.	Verify that user can select an incoming call & SMS alert ringtone.	Yes	Successful
18.	Verify that user can make silent / vibrate modes	Yes	Successful
19.	Verify the battery requirement of the mobile	Yes	Successful
20.	Verify the total time taken for complete charge.	Yes	Successful
21.	Verify the total time for mobile to get completely discharged when left idle.	Yes	Successful
22.	Verify the total time to get completely discharged when continuously used in talking	Yes	Successful
23.	Verify the length of charger wire.	Yes	Successful
24.	Verify that mobile can be switched ON or OFF.	Yes	Successful
25.	Verify that user can store contact details on the phone book directory.	Yes	Successful
26.	Verify that user can ~~store~~ delete / update contact details in the phonebook directory.	Yes	Successful
27.	Verify that call logs are maintained	Yes	Successful
28.	Verify sent & received messages are saved in mobile.	Yes	Successful
29.	Verify that user can silent the phone during an incoming call.	Yes	Successful
30.	Verify the auto-reject option can be applied and removed on particular numbers.	Yes	Successful

Result: We have successfully verified various modules & functionalities of mobile and checked that they are working as desired.

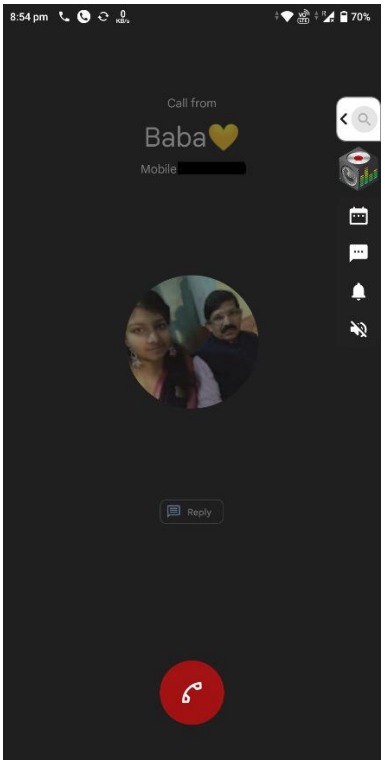
SCREENSHOTS OF TEST SCENARIOS



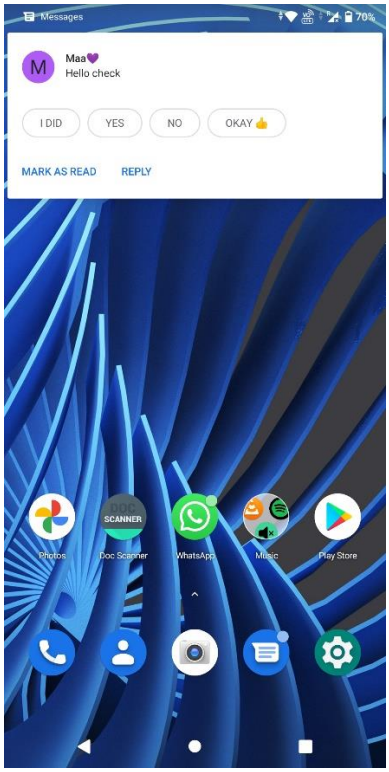
Test Case ID: 1



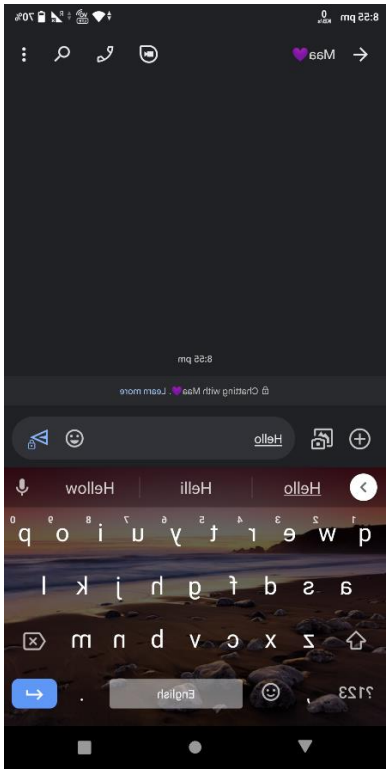
Test Case ID: 2



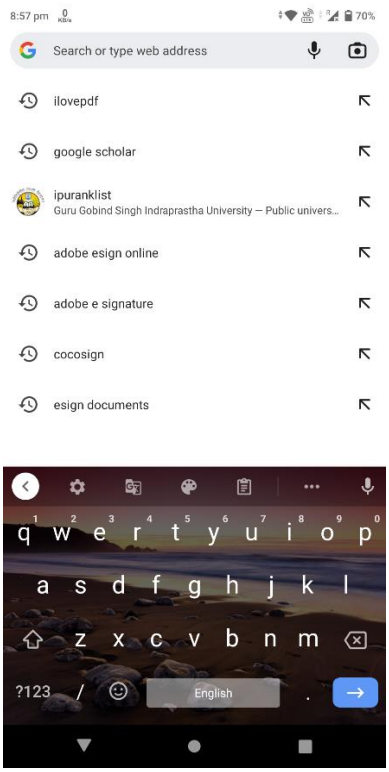
Test Case ID: 3



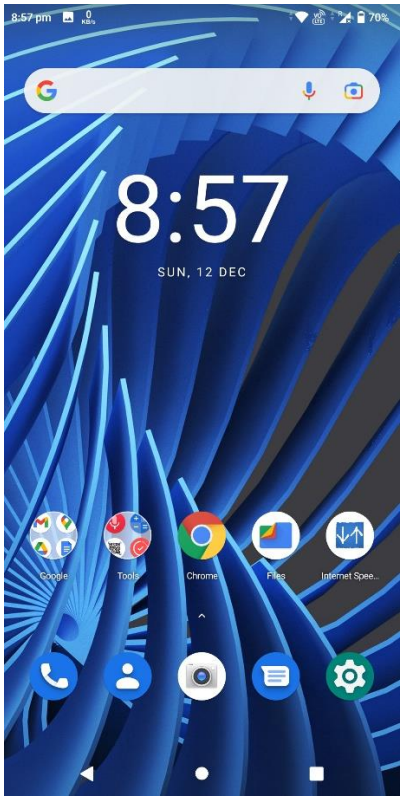
Test Case ID: 4



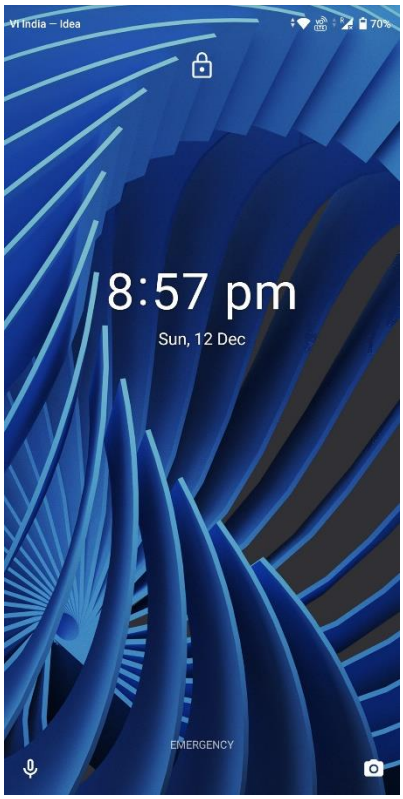
Test Case ID: 5



Test Case ID: 8 & 11

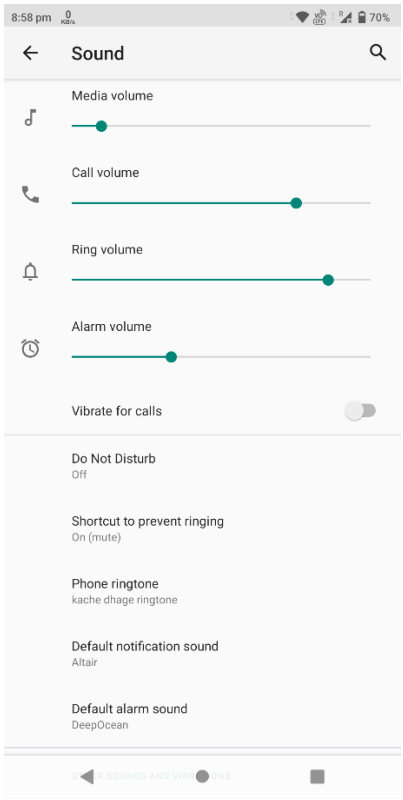


Test Case ID: 12 & 13

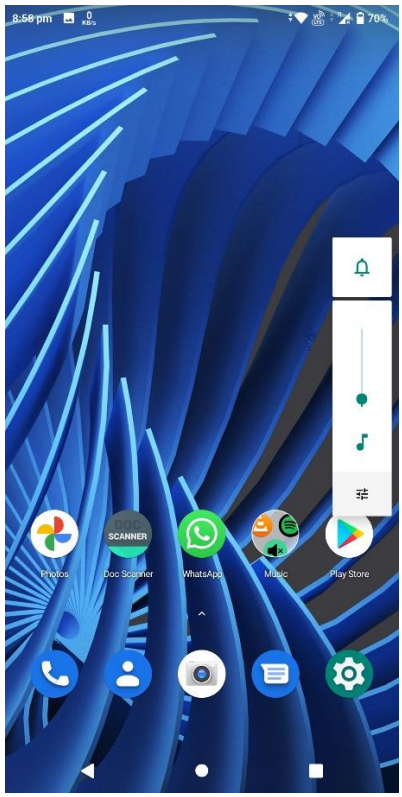


Test Case ID: 14

Test Case ID: 14

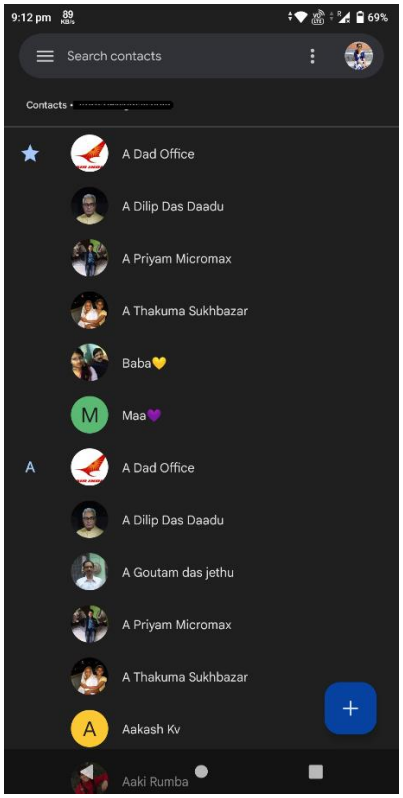


Test Case ID: 17

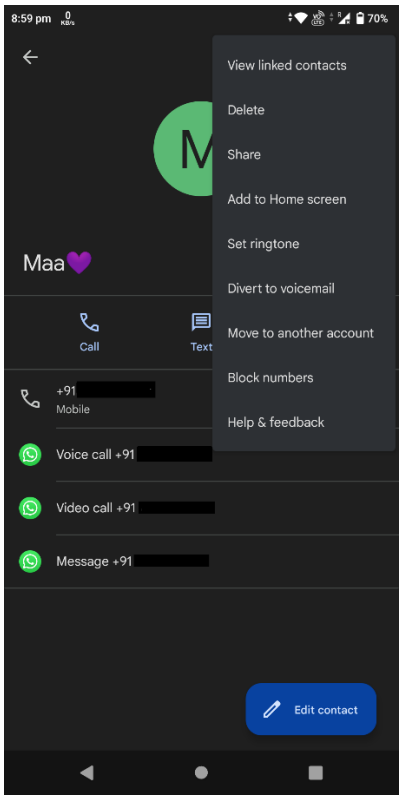


Test Case ID: 18

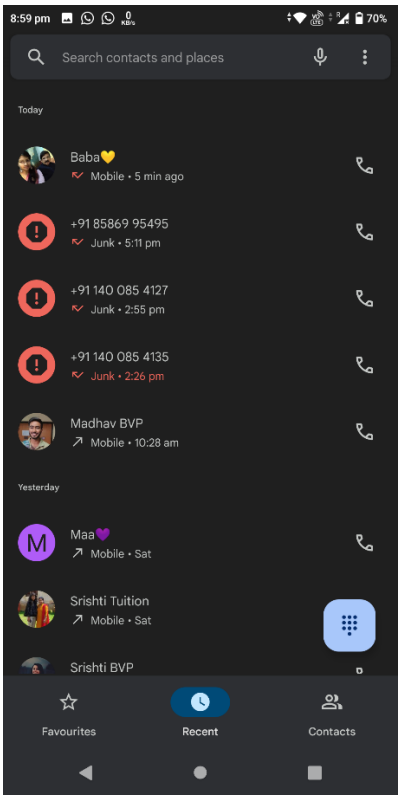
Test Case ID: 24



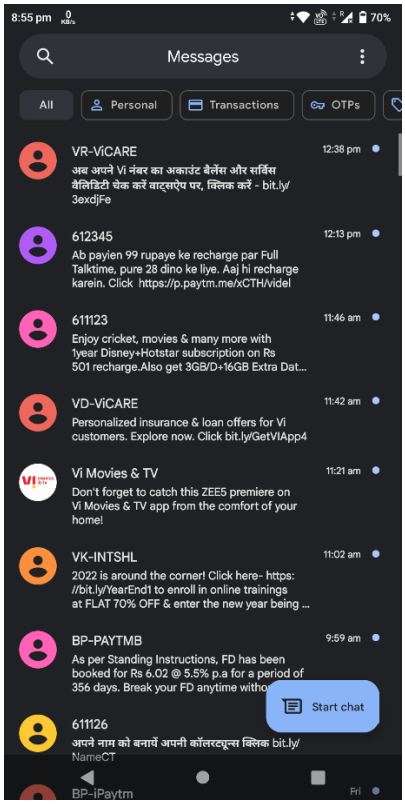
Test Case ID: 25



Test Case ID: 26



Test Case ID: 27



Test Case ID: 28