

# SOFTWARE TESTING LAB (ITD06)

## ASSIGNMENT 1

SHIV KUMAR  
2016UIT2563  
IT-2  
8th Semester

## Practical 5 : Generate the following for experiment number 2.

### 1. Control Flow Graph

### 2. Cyclomatic Complexity.

### 3. Independent Paths.

## SOLUTION:

### code.cpp:

```
#include <bits/stdc++.h>

using namespace std;

int main(){

    int a,b,c;
    cin>>a>>b>>c;
    if(a<=0 || b<=0 || c<=0){
        cout<<"Invalid Input"<<endl;
    }

    else if(a+b<=c || a+c<=b || b+c<=a){
        cout<<"Not Valid"<<endl;
    }

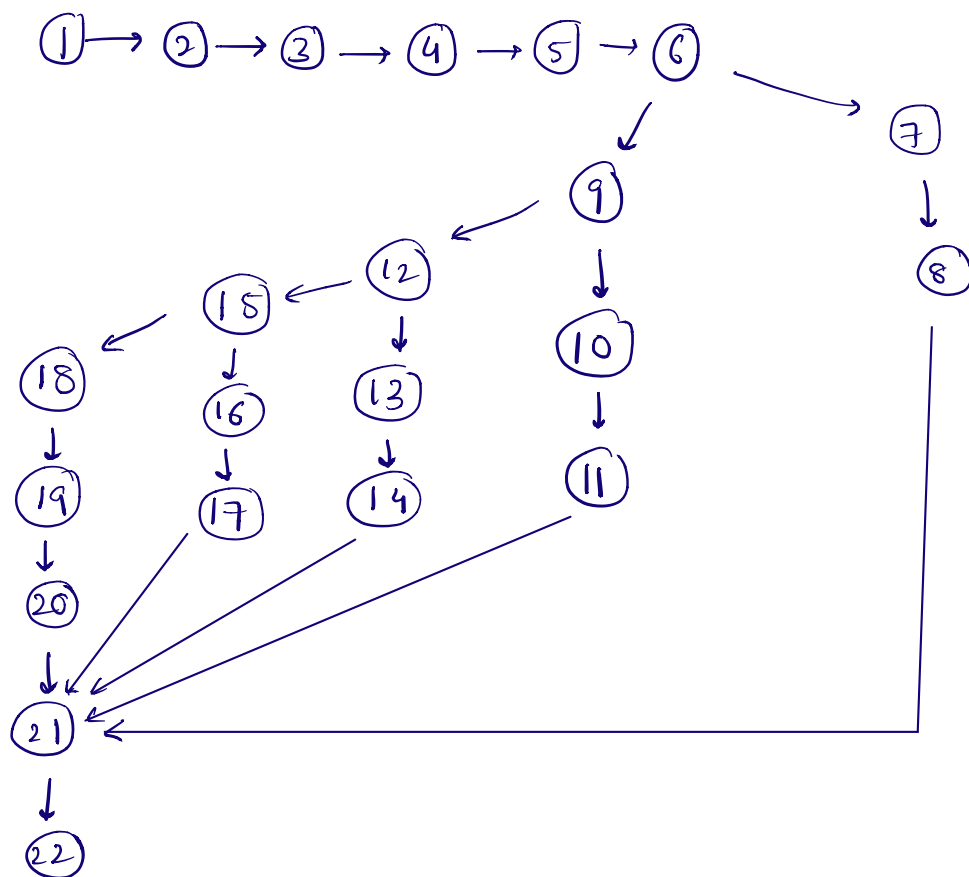
    else if(a==b && a==c && b==c){
        cout<<"Equilateral Triangle"<<endl;
    }

    else if(a==b || b==c || a==c){
        cout<<"Isocetes Triangle"<<endl;
    }
    else{
        cout<<"Scalene Triangle"<<endl;
    }
}
```

```
return 0;  
}
```

Control Flow cycle, cyclomatic complexity, and Independent Paths:

Control Flow graph:



cyclomatic complexity:

No. of edges =  $e = 25$

No. of nodes =  $n = 22$

No. of components =  $p = 1$

$$\begin{aligned}\text{Cyclomatic complexity} &= e - n + 2p \\ &= 25 - 22 + 2 \\ &= 5\end{aligned}$$

Independent graph paths  $\rightarrow$

1)  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 21 \rightarrow 22$

2)  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 21 \rightarrow 22$

3)  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 9 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 21 \rightarrow 22$

4)  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 9 \rightarrow 12 \rightarrow 15 \rightarrow 16 \rightarrow 17 \rightarrow 21 \rightarrow 22$

5)  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 9 \rightarrow 12 \rightarrow 15 \rightarrow 18 \rightarrow 19 \rightarrow 20 \rightarrow 21 \rightarrow 22$

## **Practical 6 : Create a test plan document for any application**

### **SOLUTION:**

#### **INSTITUTE MANAGEMENT SYSTEM TEST PLAN :**

**Prepared by : Shiv Kumar**

**Prepared on : 08/04/2020**

### **INTRODUCTION:**

Institute Management System is designed to automate the management process of an institute from student admission to examination management, from course management to schedule management, from human resource employment to salary management.

By using the latest technology, this Institute Management System automates an institute's operations such as Student Admission Management, Fee Management, Attendance Management, Examination Management, Human Resource Management, Time Table Management, Salary Management and Records & Profiles Management. This system is marked by number of features such as news and events, school calendar, profile search, messages, internal mail, etc.

### **SCOPE:**

1. Admin can add new course and edit exiting course and remove course and also give searching tools which will help him/her to search the course by different field by course Name, course id etc.
2. Admin can create users or Staff & Assign User Rights/Authority to Different Users/Staff.
3. Admin also gives the facility to upload attendance, result, time table, exam papers etc.
4. It also gives the facility to staff/employee view own subject result as well as attendance.
5. Administrator has the status of all members and can give permission to allow the member of this site.
6. Student View own Result & profile
7. Student display fresh news , events, practice set,downloading etc..
8. Student can search semester wise subject details.
9. Student view photo gallery and college information and Faculty info.
10. Students download exam paper, exam schedule, assignment.
11. Faculty Registration , Assign Batch Timing and Role for faculty.

## **ENVIRONMENT REQUIREMENT:**

### **Hardware:**

1. Quad Core processor or above
2. Servers
3. Database Management System

**Software:** Microsoft Windows XP.

## **TESTING STRATEGIES:**

### **1. UNIT TESTING:**

**Definition:** Test the small parts of the programs called units. Each unit is tested independent of other units.

Tested by: Developers

**Methodology:** Test different functionalities like add/remove courses, add/remove professors, find the professor allocated to a course, find the different courses allocated to a student, etc.

### **2. INTEGRATION TESTING:**

**Definition:** Test the integration of different units/modules with each other.

**Tested by:** Tester

**Methodology:** Test the database, queries performed on database while performing add/remove operations on database.

### **3. STRESS TESTING:**

**Definition:** Test how well a system can perform under stress/increase load. It tests whether the system breaks under stress or not.

**Tested by:** Tester

**Methodology:** Tests the Database, Network, etc.

### **4. USER ACCEPTANCE TESTING:**

**Definition:** Tests whether the system is in accordance to the business requirements or not. Tests whether system can fulfil the user needs or not.

**Tested by:** End Users

**Methodology:** Test the whole system together. System is used for few weeks in order to find the flaws/bugs.