COMP 5/6710 Software Quality Assurance

Test 2, April 10, 2015

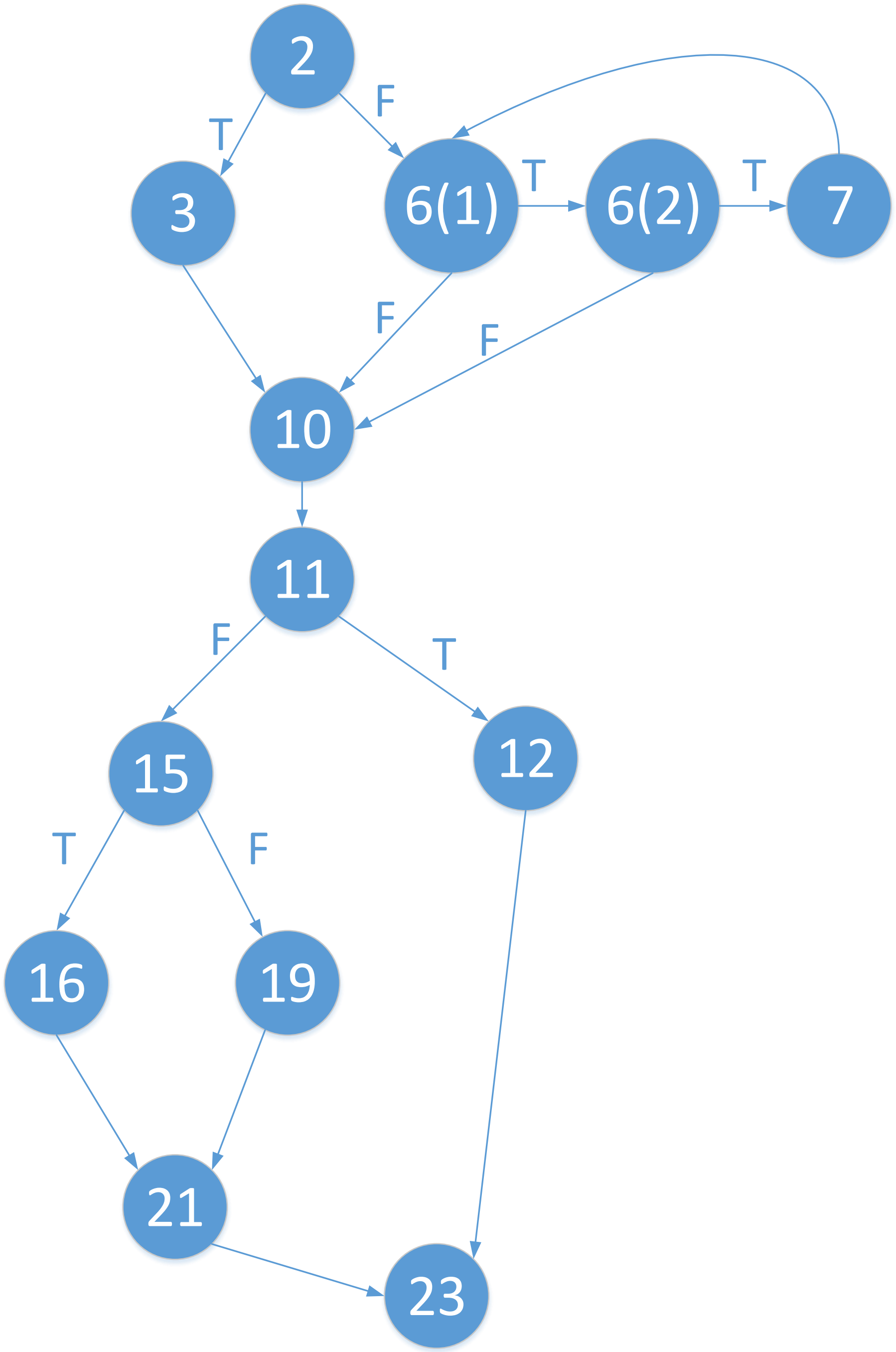
Total: 100 points Name: ______________________

1. (a) Derive the program flow graph for the following program: i) Use line numbers to label nodes in the graph; ii) Mark start and exit points; iii) Mark "T" and "F" on the two branches of a condition. (20)

   (b) What is the Cyclomatic number for the program? (5)

```
1.  void Q2() {
2.        S1();
3.        if ( C1 ) {
4.        S2();
5.       }
6.        else{
7.        if ( C2 ) {
8.           S3();
9.        }
10.        else{
11.            for ( int i = 0; C3; i++ ) {
12.              S4();
13.              if ( C4 ) {
14.                 S5();
15.              }
16.              else{
17.                 S6();
18.              }
19.              S7();
20.           }
21.           S8();
22.       }
24.    }
25.    S9();
26. }
```

2. Given the following graph, derive a set of basis paths and the path predicates for each path. (25)



3. Given the following program.
   a. Find all the Define and Use nodes for variables *kitchen* and *Total*. (10)
   b. Find the DU-paths for variables *kitchen* and *Total*. Use line numbers for node identification. (15)

```
1.  #include<iostream>
2.  using namespace std;
3.  int main()
4.  {
5.      double furniture;
6.      double food;
7.      double kitchen;
8.      double Total;
9.      double FinalMoney;
10.     cout << "How much have you spent on furniture? ";
11.     cin >> furniture;
12.     cout << "How much have you spent on food? ";
13.     cin >> food;
14.     cout << "How much have you spent on kitchen items? ";
15.     cin >> kitchen;
16.     Total = furniture + food + kitchen;
17.     if ( Total < 500 )
18.        FinalMoney = Total;
19.     if ( Total >= 500 && Total < 1000)
20.     {
21.        FinalMoney = 500 + ( Total - 500 ) * 0.95;
22.     }
23.     if ( Total >= 1000 )
24.     {
25.        FinalMoney = 500 + 500 * 0.95 + ( Total - 1000 ) * 0.9;
26.     }
27.     cout << "You should Pay: ";
28.     cout << FinalMoney;
29.     return 0;
30. }
```

4. Functional Testing

We need to develop a ticket fine calculation system. An input to the system contains two numbers, (Type, Number). If Type = 1, it means it is a “speeding” ticket; Type = 2 for a “running a red light" ticket; Type = 3 for a “DUI” ticket; other numbers are invalid input. When Type = 1 (speeding), Number is the exact miles/h you have exceeded the speed limit. When Type = 3 (“DUI”), Number represents the Blood Alcohol Content (BAC). We will explain the details of each input variables in the following statement.

Design test cases to cover all boundaries for this ticket fine calculation system. Some invalid test cases must also be considered. (25)

The ticket fine calculation rules are as follows:

For speeding ticket (Type = 1):

Number (Exceeded Speed) Fine

(0 mile/h, 10 miles/h] $100

(10 miles/h, 20 miles/h] $200

(20 miles/h, 30 miles/h] $300

(30 miles/h, ∞) Number * 15

For running a red light ticket (Type = 2):

Fine

$150

For DUI ticket (Type = 3):

Number (BAC) Fine

(0.00%, 0.04%] $200

(0.04%, 0.08%] $500

(0.08%, ∞) $1000

***Note:** Number ∈ (0 mile/h, 10 miles/h] means 0 < Number <= 10