REG. NO										

MANIPAL INSTITUTE OF TECHNOLOGY
(Constituent Institute of Manipal University)
MANIPAL-576104

VII SEMESTER B.Tech.(COMPUTER SCIENCE AND ENGINEERING) DEGREE END-SEMESTER EXAMINATION-DEC 2014
SUBJECT: SOFTWARE TESTING AND ANALYSIS (CSE 421)
DATE: 01/12/2014

TIME: 3 HOURS MAX.MARKS: 50

Instructions to Candidates

- **Note:** Answer any **FIVE** full questions.

1.A. What causes software to fail? Explain the Y2K problem.

.

1.B. An Insurance Premium Program computes annual car insurance premium based on two parameters:

- Policy Holder's age
- Safe Driving Record. If he is a safe driver, he will pay lesser Premium (thru Safe Driving Reward)

Premium = Base Rate – Safe Driving Reward

The base rate is a function of the policy holder's age. Given by the following table

Age Range	Base Rate
16 <=age < 25	Rs. 2800
25 <=age < 35	Rs. 1800
35 <=age < 45	Rs. 1000
45 <=age < 60	Rs. 800
60 <=age < 100	Rs. 1500

Usually a driver is given penalty point whenever he violates traffic rules (1 point for each violation). Once the driver has 12 points, his license is cancelled. Hence he cannot avail an insurance policy. If the driver (based on his age) has lesser Penalty Points than specified, he's eligible for Reward of premium called Safe Driving Reward. Safe Driving results in deduction of premium (through Safe Driving Reward) for drivers who have fewer penalty points.

If the driver has more penalty points than maximum penalty, safe driving reward will be zero.
E.g.: If a 23 year old driver has 3 penalty point, the his premium is
Base Rate: 2800
Safe Driving Reward: 0
(Since for his age range 16-25 he can have a maximum of 1 penalty point to be eligible for safe driving Reward of Rs. 50)
Premium = Base Rate – Safe Driving Reward
= 2800 – 0 =Rs. 2800
Note: If he had 1 penalty point only, then he'd have been eligible for a safe driving Reward of Rs 50, making his premium to be Rs. 2750

For the above given scenario,

a. Identify the set of valid equivalence classes for this problem.
b. Write the Weak and Strong Normal Equivalence Test cases.
c. Frame an **EFFICIENT** Decision Table for this problem. Derive the test cases out of it.

(3+(2+3+2))

2.A. Explain any THREE principles of Software Testing.

2.B. What are the limitations of BVA techniques? Discuss the situations in which it is not effective?

2.C. Differentiate between the following with a clear example for each.

a. Deliverables and Milestones
b. Static and Dynamic Testing
c. Test cases and Test suite

(4+2+4)

3.A. The ATM system communicates with the bank for any of the three transaction types: deposits, withdrawals, and balance inquiries. These transactions can be performed on two types of accounts: checking and saving. When a bank customer arrives at an ATM station, he or she inserts a card encoded with a personal account number (PAN), which is the key to an internal customer account, containing the customer's name and account information. Then the customer enters his or her personal identification number (PIN). The customer selects the desired transaction from a list of options to complete his or her transaction, further in sequence; the customer's request is processed by the system. For example, if a customer requests a withdrawal transaction and enters the withdrawal amount then the system checks the customer's account balance. If the account balance is sufficient, the money is dispensed. The withdrawal amount is written to the file, and the count of withdrawals per month is incremented. The balance is printed on the transaction receipt and the customer's ATM card is returned.

For the above given scenario,

Draw a DETAILED State Diagram and generate the "*All States*" and "*All Transitions*" test cases from it.

3.B. Consider the following code,

```
function Mystery (x, y: integer): integer;
    var s, z: integer;
    s := 1;
    z := -1;
    if x < 0 then
        s := -1;
        x := -x;
    end if;
    if y < 0 then
        s := -s;
        y := -y;
    end if;
    while x >= 0 do
        x:= x - y;
        z:= z + 1;
    end while;
        z := s * z;
```

```
print(z);
 return(z);
  end Mystery;
```

Write the test cases for branch coverage, statement coverage and path coverage.

(5+5)

4.A. For the code in Q3B,

a. Identify definitions and the uses of all the variables.
b. Identify all the du pairs and all the du-paths.
c. Write the test cases for all uses criteria.

4.B. Explain the concept of Memory analysis and lockset Analysis.

(5+5)

5.A. Consider the following program:

```
main( )
{
int a,b,c,sum,diff,mul;
scanf("%d %d %d", &a,&b,&c);
sum=calsum(a,b,c);
diff=caldiff(a,b,c);
mul=calmul(a,b,c);
printf("%d %d %d", sum, diff, mul);
}
```

```
calsum (int x, int y, int z)
{
   int d;
   return (d); }
```

a. Suppose main() module is not ready for the testing of calsum() module. Design a driver module for main( ).
b. Modules caldiff( ) and calmul ( ) are not ready when called in main ( ).Design stubs for these two modules.

5.B. Consider a program

```
main()
{
char chr;
1. printf("enter the special character");
2. scanf("%c", &chr);
3. if((chr!=48) && (chr!=49) &&(chr!=50) &&(chr!=51) &&(chr!=52) &&(chr!=53)
   &&(chr!=54) &&(chr!=55) &&(chr!=56) &&(chr!=57))
4. {
5. switch(chr)
6. {
7. case '*' : printf("it is a special character");
8. break;
9. case '#' : printf("it is a special character");
10. break;
11. case '@' : printf("it is a special character");
12. break;
13. case '!' : printf("it is a special character");
```

```
14. break;
15. case '%' : printf("it is a special character");
16. break;
17. default: printf( " not special character");
18. break;
19. }// end of switch
20. }//end of if
21.       else
22.             printf(" not entered a valid character");
23. } //end of main
```

Derive the test cases for performing path testing by generating the possible independent paths. (3+7)

6.A. What is mutation testing? A test engineer generates 70 mutants of a program P and 150 test cases to test the program P. After the first iteration of mutation testing, the tester finds 58 dead mutants and 4 equivalent mutants. Calculate the mutation score for this test suite. Is the test case adequate for program P? Should the test engineer develop additional test cases? Justify your answer.

6.B. With an example for each, explain the various Integration Testing strategies.

6.C. Explain the superclass-subclass development scenarios.

(3+3+4)

****

.