Reg. No.										

MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

A Constituent Institution of Manipal University

VII SEMESTER B.TECH. (COMPUTER SCIENCE AND ENGINEERING) MAKE-UP EXAMINATIONS, DEC 2017

SUBJECT: SOFTWARE TESTING AND ANALYSIS [CSE 4020]

REVISED CREDIT SYSTEM
(28/12/2017)

Time: 3 Hours MAX. MARKS: 50

Instructions to Candidates:

- Answer **ALL** questions.
- Missing data may be suitable assumed.

1A. With the help of an example show that probability of the program failure might not change upon error removal. Explain any TWO Dynamic quality attributes of a software by giving example for each. **4M**

1B. Consider a program to multiply and divide two numbers. The inputs may be two valid integers (say 'a' and 'b') in the range of [0, 100]. Generate normal, robust and worst case BVA test cases for the program. **4M**

1C. For the program description given in Q1B. Write the output domain based equivalence classes and the corresponding test cases**.** **2M**

2A. Consider a scenario for ATM management system. The user needs to insert the valid ATM card. In case the card entered by the user is "invalid", the system has to reject the card by displaying appropriate error message. If the entered card is a valid one the system will then prompt the user to enter the PIN. In case of an incorrect PIN, the system allows the user to re-enter the PIN maximum three times. Soon after the third attempt, if the entered PIN is still invalid, the system will automatically eat the card. In the next step, the system prompts the user to enter the withdrawal amount and the entered amount is verified with the current account balance. If the balance in the account is sufficient, the system dispenses the cash to the user. Design test cases for this scenario using Decision Table Based testing. Clearly show all the steps involved. **4M**

2B. Explain the different types of Equivalence Testing techniques. Clearly show how test cases are designed using each of these techniques. **4M**

2C. Is DU path testing same as DU pair testing? Justify your answer with the help of a clear example. **2M**

3A. List the definition, c-use, p-use and DU pairs of all the variables used in the given program and also derive test cases for testing all DU pairs of the program. **4M**

```
1.  int VarTypes(int x, int y) {
2.  int i;
3.  int *iptr;
4.  i = x;
5.  iptr = malloc(sizeof(int));
6.  *iptr = i + x;
7.  if (*iptr > y)
8.  return (x);
9.  else {
10. iptr = malloc(sizeof(int));
11. *iptr = x + y;
12. return(*iptr);
13. }}
```

3B. Consider a program that checks if a given input number is prime or not. The main ( ) of the program has a call to check_prime( ). **2M**

a. Suppose the module check_prime( ) is not ready when called in main ( ). How can we check the system in such a scenario?
b. Suppose the main( )is not ready for testing of check_prime( ). How can we check the system in such a scenario?

Explain the above two questions along with a sample code.

3C. **Program P: (Assume the program is as per the requirements)** **4M**

```
1. #include<stdio.h>
2. int main()
3. {
4. int x,y;
5. printf("Enter the values of x and y:");
6. scanf("%d%d",&x,&y);
7. if(x<y+1)
8. printf("The square is %d",x*x);
9. else
10. printf("The result is %d",2*y);
11. return 0;
12. }
```

Test Set T= {(t1:x=1, y=0) (t2:x=2, y=2) (t3: x=1,y=3)}

Mutants

M1 : replace x with x+1 in line 7
M2 : replace * with + in line 8
M3 : replace < with <= in line 7

For the above data, test the adequacy of the Test Set 'T' using mutation Testing. Enhance the test cases if required.

4A. Explain Test Minimization and Test Prioritization with an example for each. **5M**

4B. What is the significance of Mutation Score and how are they calculated? Give an example for Equivalent Mutant. **2M**

4C. Explain the various data flow adequacy criteria with a common example. **3M**

5A. **4M**

```
1. Consider following code
2. Maxsum(int maxint, int value)
3. int result=0, value=0;
4. if(value<0)
5.  then value= - value;
6. while((i<=value) AND (result <=maxint))
7. DO i=i+1;
8.       result=result+i;
9. OD;
10. if (result<=maxint)
11. then output(result);
12. else output("too large");
13. end
```

Draw a CFG for above code. Write efficient test cases for statement, branch and condition testing.

5B. Explain the various static relationships of a class with the help of a sample code segment for each of them **3M**

5C. Explain the pros and cons of different integration testing strategies **3M**

****