MCA-302 - Software Engineering
Master of Computer Applications
Semester-III, Nov-2017

Time: Three Hours **Max. Marks: 70**

Note: All questions are compulsory. Attempt all parts of a question together. Write any assumptions made.

1. Answer the following questions. **(8x3=24)**
   a) What is the most important difference between generic software product development and custom software development?
   b) Giving reasons for your answer, suggest the most appropriate generic software process model that might be used as a basis for the development of an interactive travel planning system that helps users plan journeys with the lowest environmental impact.
   c) When would you recommend against the use of agile methodology for software development?
   d) Differentiate between Reverse engineering and Reengineering.
   e) What is regression testing? How the use of automated testing framework simplifies regression testing?
   f) What do you understand by Capability Maturity Model (CMM) for software process modelling?
   g) Discuss the different types of risks that could arise in software projects.
   h) How is Function Point Analysis used for software sizing?

2. Explain the COCOMO–II model in detail for project cost estimation. **(5)**

3. Explain the Jelinski-Moranda model of reliability theory. What is the relation between the parameters 't' and 'λ' of the model? **(5)**

4. The table below shows a number of tasks, their durations and their dependencies for software project activities. **(4+1+1=6)**
   a) Draw a bar chart showing the project schedule and the milestones.
   b) Giving reason, identify the tasks where people allocated are not working full time on it.
   c) What is the estimated duration of the project?

Task	Effort (person-days)	Duration (days)	Dependencies
T1	15	15 3	-
T2	8	10 2	-
T3	20	25 5	T1 (M1)
T4	5	10 2	-
T5	5	5 1	T2,T4 (M3)
T6	10	15 3	T1,T2 (M4)
T7	25	20 4	T1 (M1)
T8	75	35 7	T4 (M2)
T9	10	15 3	T3,T6 (M5)
T10	20	25 5	T7,T8 (M6)
T11	10	20 4	T9 (M7)
T12	20	10 2	T10,T11 (M8)

5. It is proposed to develop an automated teller machine (ATM) software that provides bank customers with access to financial transactions in a public space without the need of a bank staff. Customer (actor) uses bank ATM to change PIN, check balances of his/her bank accounts, deposit funds, withdraw cash, transfer funds, and print mini-statement. ATM Technician (actor) provides maintenance and repairs; Bank (actor) is also involved whether it is related to customer transactions or to the ATM servicing. To initiate a banking transaction, the software should ask for ATM card. If proper card is inserted and the customer is authenticated, the software should allow access to customer's bank account. Only one transaction can be carried at a time after which the customer is logged out.
Draw the Level-0 DFD and Use-case diagram for the case study. Also draw the Sequence diagram for withdrawal of money. **(4+4+2)**

6. Consider the fixed deposit interest calculations with the following conditions as given below:

	Deposit Amount	0 - 90 days	90 days - 1 year	1 - 3 years	3 - 5 years	Above 5 years
a	<=5000	5%	5.5%	6%	6.5%	7%
b	5001 to 50,000	5.5%	6%	6.5%	7%	7.5%
c	50,001 to 1,00,000	6.5%	7%	7.5%	8%	8.5%
d	1,00,001 to 5,00,000	7.5%	8%	8.5%	9%	9.5%

The minimum balance is Rs. 1000 and the maximum balance is Rs. 5,00,000.

a) Generate test cases by performing boundary value analysis. **(5)**

b) Generate decision table. **(5)**

7. For the code given below:

a) Draw the Control Flow Graph. **(2+3+5)**

b) Calculate cyclomatic complexity by all the three methods.

c) Perform Data-Flow analysis.

```
1.  void quadratic (float a, float b, float c) {
2.      float d, root1, root2;
3.    d = b*b – 4*a*c;
4.    if (d < 0)
5.      printf("\nImaginary roots.");
6.    else if (d = 0) {
7.      root1 = -b / (2*a);
8.      root2 = -b / (2*a);
9.      printf("\n\nEqual roots. \nRoot1 = %f \nRoot2 = %f", root1, root2);
10.   }
11.   else {
12.     root1 = (-b + sqrt(d)) / (2*a);
13.     root2 = (-b -sqrt(d)) / (2*a);
14.     printf("\n\nReal roots. \nRoot1 = %f \nRoot2 = %f", root1, root2);
15.   }
16.   return;
17. }
```