30/11/16

Indian Institute of Engineering Science and Technology, Shibpur
B.E. (CST) 7th Semester Final Examination 2016

Software Engineering (CS - 701)

Time: 3 hours **Full marks: 70**

Use same answer booklet to attempt questions from all groups.
All parts of a question are to be answered together.

Group A
Answer question no 1 from this group (20 marks)

1. **[Mandatory Question]** Answer **any** 10 questions out of 12 in brief. Each question carries 2 marks.

   a) What are the problems of *exploratory style* of software development?
   b) What is "V" model of Software development life-cycle?
   c) Mention some programming errors those can be detected during *code inspection*.
   d) What do you understand by *non-functional requirements*?
   e) Draw the *class diagram* corresponding to the *sequence diagram* of money transfer in the banking system presented in *Fig-1*.
   f) State the four values of the *Agile Manifesto*.
   g) To determine the effectiveness of the testing strategy, before the start of any kind of testing, 50 different bugs of different varieties are seeded into the 10K lines of code. On completion of all testing phases, it is observed that a total of 167 bugs are detected. Out of that, 36 are those bugs which were artificially introduced. Predict the approximate number of residual bugs still present in the code.
   h) In which types of testing testcases can be designed following *black-box* testing approach?
   i) What do you understand by the terms *Quality Assurance* (QA) and *Total Quality Measurement* (TQM)?
   j) Give examples of some *system/validation* testing.
   k) What is *spike solution*?
   l) If functions `funA` and `funB` have cyclomatic complexities of N1 and N2 respectively, then what will be the cyclomatic complexity of function `funC` stated in *Fig-2*? Why?

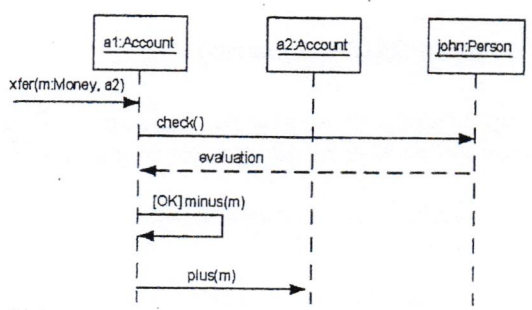

Fig-1 : Sequence diagram

```
int funA(...);
int funB(...);
int funC(...){
   int c = funA(...) + funB(...);
   return c;
}
```

Fig-2 : C code snippet

[10x2] = 20

Group B

Attempt any 4 questions from this group (4x5 = 20 marks)

2. Describe briefly the different stages in the Rapid Prototype life-cycle model of software development. Mention the applicability of this model.

[5]

3. Explain "Decision Tree" and "Decision Table" techniques to specify software requirements with a suitable example.

[5]

4. In UML, why use case diagram factoring is needed? What are the differences between `<<include >>` and `<<extend>>` relationship in UML use case factoring? Explain using an example.

[5]

5. Explain how code integration is performed? Mention the use of *structure chart* in this context.

[5]

6. A very small scale profit making IT company delivers software products from last few years. Due to very high employee satisfaction of the company, it experiences very little stuff attrition. The company has very few product lines in similar area; hence management is defensive enough to expand it in terms of human resource. The average salary of a software developer in the company is Rs 55,000 per month. The company follows "Basic COCOMO" technique to estimate the project. While estimating a software solution, the size of a software product has been estimated as 27,000 lines of source code. The company tries to maintain 30% *net profit margin* while bidding a project. Determine the (i) effort required to develop the software product, (ii) nominal development time, (iii) cost to develop the product and (iv) bidding price to develop the product.

[5]

7. Write short note on "SEI CMM Levels".

[5]

Group C

Attempt any 2 questions from this group (2x15 = 30 marks)

8. Consider the following function in C programming language, that computes the sum of all odd integers in a list (*line no mentioned in following code should NOT be altered, extend if needed*)

```
   int SumOfOddNums(int arrayElems[], unsigned char numOfElems)
 1 {
 2     unsigned char index = 0;
 3     int sum = 0;
 4     for (index = 0; index < numOfElems; index++)
 5     {
 6         int tempElem = arrayElems[index];
 7         if(1 == tempElem%2){
 8             sum += tempElem;
 9         }
10     }
11     return sum;
12 }
```

a) Draw the Control Flow Graph (CFG) for the above function.
b) Compute the McCabe's cyclomatic complexity of the above function applying all possible approaches and find all Linearly Independent Paths (LIPs) in the CFG.
c) How cyclomatic complexity of a program is useful in the context of test suite design?

[6 + 7 + 2] = 15

9. A certain project can be split into 9 distinct activities A, B, …, I. The time (in weeks) to complete each activity is mentioned in the following table along with the dependencies between the tasks.

Activity	**Order / dependency**	**Estimated time (in weeks)**
A	Must be done first	8
B	Can only start when A is completed	5
C	Can only start when A is completed	8
D	Can only start when B is completed	6
E	Can only start when B is completed	9
F	Can only start when D is completed	2
G	Can only start when E and F are completed	3
H	Can only start when C is completed	10
I	Can only start when G and H are completed	3

a) Draw the Activity Network for the project following the convention of AOA and AON.
b) For each activity, compute the following parameters so that the overall project can be completed as early as possible – (i) earliest time at which it can start, (ii) latest time at which the activity must start, (iii) slack time, (iv) critical Path and (v) minimum Time to complete the project.
c) Mention the importance of "Critical Path".

[6 + 7 + 2] = 15

10. A banking system contains data of customers and their accounts. Each account has customer information and a balance and there are 2 types of accounts: one for savings which offers an interest rate, the other for investments, used to buy stocks. Stocks are bought at a certain quantity for a certain price and the bank applies commission on stock orders.

a) Identify all Classes involved in the above mentioned fact.
b) Draw a class diagram using the UML Syntax to represent above fact, mentioning -
   i. Every class with attributes and important behaviors with access specifiers.
   ii. The relationship between Classes using standard UML notation.

[5+ (5+5)] = 15