

**NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA**  
**END - TERM EXAMINATION, 2018**  
 SESSION: 2017 – 2018 (Spring)

**Subject code: CS412**  
 No. of pages: 2

**Subject Name: Software Engineering**  
 Full Marks: **100**

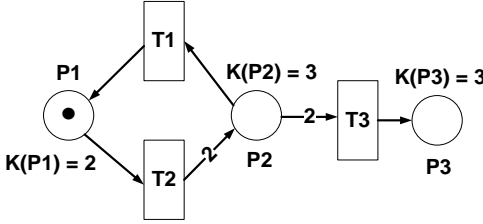
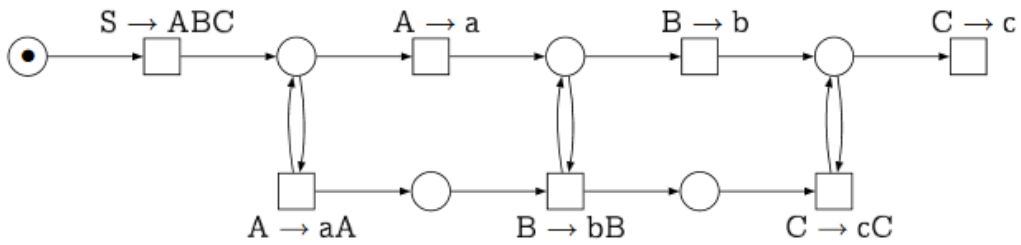
**Dept. Code: CS**  
 Duration: **3 Hrs**

**PART – A**

Q.No.	Particulars	Marks
1.	<b>ANSWER ANY TEN QUESTIONS FROM THE FOLLOWING:</b> (a) Explain why Speed of Learning and Speed of Recall are relevant with respect to GUI design. (b) In a system 12 errors were found by testing. Out of 20 errors seeded into the system, following Error Seeding strategy, 16 were caught. Find the number of latent errors in the system? (c) What is the 90-10 rule in terms of software reliability? (d) Suppose a program contains 3 decision points, each of which has 2 branches? How many test cases are necessary for branch testing? (e) What is <i>mutation testing</i> ? How is it carried out? (f) What do you mean by <i>balancing DFDs</i> ? Why is it required – explain briefly with an example? (g) Differentiate between “constructor” and “inspection” operators in Z specification. (h) State and explain the <i>Reversibility</i> property in Petrinets, with an example. (i) What is unit testing and why is it performed in the implementation phase of a module? (j) What is regression testing and in what phases of SDLC is it performed? (k) How are work products improved using Software Engineering? (l) What are the characteristics of a good software engineering process? Briefly explain them. (m) What is reverse engineering and when is it used?	2×10=20

**PART – B**  
**ANSWER ANY FOUR**

Q.No.	Particulars	Marks
2.	a) What is black-box testing? Design a black-box test suite for a program named <b>quad_solver</b> . The program accepts three integer coefficients (a, b, c) of a quadratic equation, of the form <b><math>ax^2 + bx + c = 0</math></b> , and computes the roots of the quadratic equation. b) Write a program (in any programming language) to find the <i>greatest common divisor (gcd)</i> of three input integers. Also answer the following questions. i. Draw the control flow graph of the above program and determine its cyclomatic complexity. ii. Design a test suite for the above program using path coverage based testing strategy.	10×2=20
3.	(a) Draw up to Level 2 DFD of an ATM System, assuming standard ATM operations. (b) Draw the structure chart for the above ATM system using its DFD. (c) How is software reliability different from hardware reliability? Draw and explain the implication of the “bath-tub curve” in this regard.	10+5+5=20
4.	(a) Explain how <i>Program Slicing</i> is different from <i>Backtracking</i> in program debugging. (b) Why and how is <i>Statistical Testing</i> carried out? Hence identify is the major challenge in <i>Statistical Testing</i> ? (c) Apply the Critical Path Method (CPM) to find all Slack Time availability in the following Activity Network: <div style="text-align: center; margin-top: 20px;"> <pre> graph LR     Spec[Specification 15] --&gt; Design[Design 45]     Spec --&gt; WriteManual[Write Manual 45]     Design --&gt; CL1[Code Lexer 50]     Design --&gt; CL2[Code Lexer 75]     CL1 --&gt; Finish[Finish 0]     CL2 --&gt; Finish     WriteManual --&gt; Finish                     </pre> </div>	5+5+10=20

5.	<p>(a) Draw the <i>Reachability Graph</i> of the following Petrinet, with initial marking (1,0,0), shown below:</p>  <p>(b) Transform the above Petrinet (in Q.5a) into its equivalent infinite capacity net, by <i>Complementary Place Transformation</i>.</p> <p>(c) Identify the language generated by the following Petrinet. Note here that the initial marking is (1,0,0,0,0,0).</p>  <p>[Hint: See all possible firing sequences, and what strings those generate. Final state is reached when no more transition can be fired.]</p>	10+5+5=20
6.	<p>(a) What is process based quality management in Software Engineering? Explain with suitable diagrams.</p> <p>(b) Is it possible for an engineer to be successful in a failed software project? Explain.</p> <p>(c) What are input, output and central transform modules? How do you identify. Give some examples.</p> <p>(d) Explain Capability Maturity Model.</p>	4+3+3+10=20
7.	<p>(a) When should one stop modularizing? Explain it with suitable graphs.</p> <p>(b) Maintenance is as difficult as Development. Justify this claim.</p> <p>(c) What is System testing? Explain its types in detail.</p> <p>(b) Explain Personal Software Process Model.</p>	3+3+4+10=20