SASTRA DEEMED UNIVERSITY

(A University under section 3 of the UGC Act, 1956)

B.Tech. Degree Examinations

May 2018

End Semester

Course Code: **BCSCCS 602R03 / BITCIT 602R03 / BICCIC 602R03**

Course: **SOFTWARE ENGINEERING**

Question Paper No. : **B0594** Duration: **3** hours

Max. Marks: **100**

PART – A

Answer all the questions **10 x 2 = 20 Marks**

1. Justify your answer. If software does not wear out, why does it deteriorate? Undiscovered effects, changes.
2. Describe the XP concepts of refactoring and pair programming.
3. Illustrate why is it difficult to develop estimation using use-cases?
4. How is grammatical parsing done in requirements analysis?
5. Categorize the characteristics of a well-formed design class. complexity very minimum High cohesion low coupling
6. State the roles of a traditional software component in software architecture. control, Problem domain, Infrastructure
7. How do systems interoperate with one another?
8. Discuss the benefits that can be derived for smoke testing.

9. Deduce the types of tests which are conducted for client server systems.

10. Point out the testing options available at class level.

PART – B

Answer all the questions **4 x 15 = 60 Marks**

11. (a) Suggest the most appropriate generic software process model that might be used as a basis for managing the development of "A university accounting system that replaces an existing system". (10)

(b) How do process model differ from one another? (5)

(OR)

12. (a) Identify the 12 agility principles given by Agile Alliance for those who want to achieve agility. (6)

(b) Demonstrate the model that depicts the activity followed in Rugby match. (9)

13. (a) Compute the Function point value for a project with the following information domain characteristics:
Number of user inputs = 50
Number of user outputs = 40
Number of user enquiries = 35
Number of user files = 06
Number of external interfaces = 04
Assume that all complexity adjustment factors and weighing factors are AVERAGE. (10)

(b) Narrate the basic guidelines that can help us as we do requirements analysis work. (5)

(OR)

14. (a) Develop a complete use case for an activity specified by your instructor for the software engineering subject. (5)

(b) If you have been given the responsibility to elicit requirements from a customer how would you do that? (10)

15. Analyze the important software design concepts that span both traditional and object-oriented software.

(OR)

16. Describe the various principles that can be used as a guide as each software component is developed. Design + packaging (9) (6)

17. (a) A program computers a^b where a lies in the range [1, 10], b within [1, 5]. Design test cases for this program using Boundary Value Analysis. (7)

(b) A program determine the next date in the calendar. Its input is entered in the form of <dd/mm/yyyy> with the following range:

$1 \leq mm \leq 12$

$1 \leq dd \leq 31$

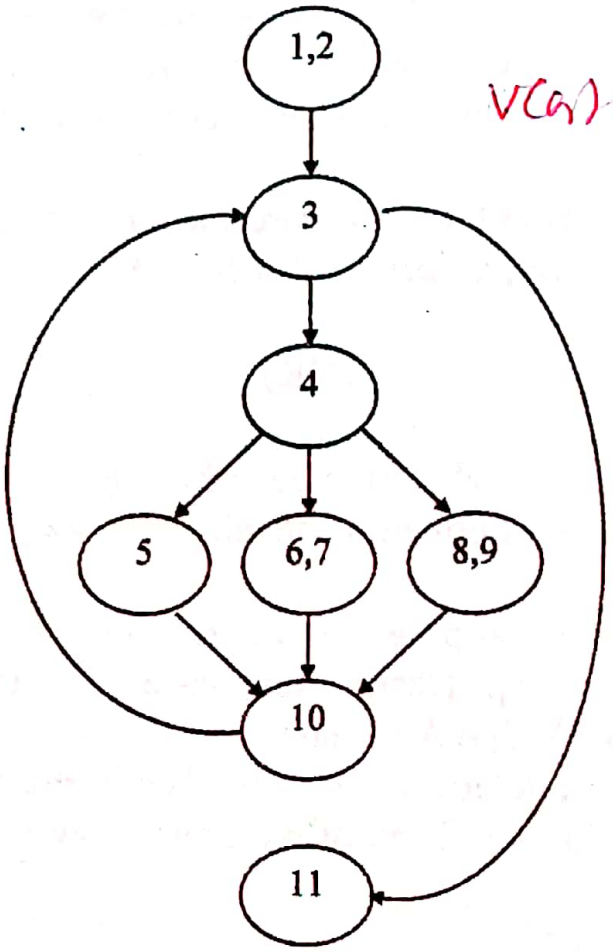
$1900 \leq yyyy \leq 2025$

Its output would be next date or it would display invalid date. Design test cases for this program using Equivalence partitioning method. (8)

(OR)

18. (a) Illustrate the unit test procedures (5) and the errors (3) that are commonly found. (8)

(b) Calculate the Cyclomatic complexity of the flow graph given below. List all independent paths also. 3+4 (7)



V(G)=4

PART – C

Answer the following **1 x 20 = 20 Marks**

19. Your company "XXX" is quite popular for its quick design and risk reduced development of software around the companies located in Chennai. You are now engaged to develop software for an online article publishing journal. The project is initiated by having frequent communications with three different types of stakeholders namely readers, author and editor to elicit the requirements of software. Based upon the stakeholders communications do the following:
Develop a complete use case for 'article submission'.

Create class diagram by specifying the entity, boundary and controller classes for the entire system and covert it as a sequence diagram.

The software is composed of three subsystem groups: user interface, control and infrastructure. Three use cased describe the user interface. Each use case is described by not more than 5 scenarios and has an average length of six pages. The control subsystem is described by 5 use cases, each of the use cases has no more than 10 scenarios and has average length of eight pages. Finally, the infrastructure subsystem is described by 4 use cases with an average of six scenarios and length of five pages. The historical data of UI software requires an average of 800 LOC per use case for the scenarios nor more than 10 scenarios and is described in less than 8 pages. Similarly for control subsystem requires an average of 1200 LOC per use case for the scenarios no more than 12 scenarios and is described in less than 10 pages. The historical data of infrastructural requires an average of 600 LOC per use case for the scenarios no more than 8 scenarios and is described in less than 7 pages. The change percentage of the traditional code is 30 %. Estimate the overall size of the software in terms of LOC.

use case dia (5)

* * * * * class dia (5)

seq dia (5)

bulletin (5)