U.S.N.										

B.M.S. College of Engineering, Bengaluru-560019

Autonomous Institute Affiliated to VTU

June 2025 Semester End Main Examinations

Programme: B.E. **Semester: V**
Branch: Information Science and Engineering **Duration: 3 hrs.**
Course Code:22IS5PESTG **Max Marks: 100**
Course: Software testing

Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

Important Note: Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. Revealing of identification, appeal to evaluator will be treated as malpractice.

			UNIT - I	CO	PO	Marks
1	a)		Identify any four test scenarios for Netflix application and write atleast three test cases for each scenario.	*CO2*	*PO1*	**10**
	b)		Elaborate on the strengths and weaknesses of automated and manual testing.	*CO2, 6*	*PO2, 5, 9*	**10**
			OR			
2	a)		Define Software Quality. Illustrate the five views of Software Quality.	*CO2*	*PO1*	**6**
	b)		Differentiate the following: ● Verification and Validation ● White box and Black box testing.	*CO2*	*PO1*	**4**
	c)		What is Regression testing and Justify Why Regression testing is considered as sub-phase of testing? Explain the different testing levels with suitable V model.	*CO2*	*PO1*	**10**
			UNIT - II			
3	a)		Explain the different levels of testing with a relevant diagram.	*CO2*	*PO1*	**10**
	b)		Explain the different steps in the code review process with a relevant diagram.	*CO2*	*PO1*	**10**
			OR			
4	a)		Illustrate the working of Test-first process in Extreme Programming (XP) with suitable diagram.	*CO2*	*PO1*	**5**
	b)		Compare and Contrast Specification based test case identification method with the code-based test case identification method.	*CO2*	*PO1*	**5**

		c)	Analyze and create 3 mutants (along with code) for the below program. Design 3 test cases to identify whether the mutants are equivalent, killable & stubborn. Update the test cases so that all mutants are killed and find the mutation score accordingly.
			`#include<stdio.h>`
			`int main()`
			`{`
			`char c;`
			`printf("Enter any character : ");`
			`scanf("%c",&c);`
			`if(c>='A' && c<='Z')`
			`printf("character is an upper case");`
			`else if(c>='a' && c<='z')`
			`printf("character is a lower case");`
			`else if(c>='0'&& c<='9')`
			`printf("it is not a character");`
			`else`
			`printf("character is a special character");`
			`return 0;`
			`}`
				CO2	***PO1***	**10**

			UNIT - III			
	5	a)	For a triangle problem, calculate the number of test cases needed and construct the normal and worst-case boundary value test cases.	***CO4***	***PO3***	**10**
		b)	Compare and contrast the single / multiple fault assumption with boundary value and equivalence class testing.	***CO3***	***PO2***	**10**
			OR			
	6	a)	What is Equivalence Class Testing? Describe Weak Normal Equivalence Class Testing and Weak Robust Equivalence Class Testing, providing relevant examples for each.	***CO1***		**10**
		b)	Consider a scenario where you are testing an online registration form for a website. The form includes the following fields: • Username: Must be between 5 to 15 characters, containing only alphanumeric characters. • Password: Must be between 8 to 20 characters, with at least one uppercase letter, one lowercase letter, and one special character (e.g., @, #, $, etc.). • Email: Must be in a valid email format (e.g., user@example.com). • Age: Must be between 18 and 100. Using Strong Robust Equivalence Class Testing, design test cases for the above fields, ensuring robust boundary testing for each field.	**CO4**	**PO3**	**10**

		UNIT - IV				
	7	a)	1. read x 2. if (x > 0) 3. a = x + 1 4. if (x <= 0) { 5. if (x < 1) 6. x = x + 1; goto (5) 7. else 8. a = x + 1 } 9. print a; For the above code, identify the All defs, All c-use, All p-use and All c-use/some p-uses associations for a variable '**a**'.	***CO4***	***PO3***	**10**
		b)	Explain the common paradigms of interfacing modules.	***CO1***		**10**
			OR			
	8	a)	Explain comparison of data flow test selection criteria with neat diagram.	***CO3***	***PO2***	**10**
		b)	Explain Incremental, Top down and Bottom up approaches to system integration as applicable to a Check-in request.	***CO1***		**10**
			UNIT - V			
	9	a)	Considering a website example, explain how a user is involved in acceptance testing.	***CO4***	***PO3***	**10**
		b)	Explain five views of software quality.	***CO1***		**10**
			OR			
	10	a)	Discuss the advantages and disadvantages of customer involvement in testing.	***CO3***	***PO2***	**10**
		b)	Illustrate ISO 9000: 2000 software quality standard.	***CO2***	***PO1***	**10**

****