WINTER – 2022 EXAMINATION

Subject Name: Software Testing **Model Answer** **Subject C** 22518

Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.
8) As per the policy decision of Maharashtra State Government, teaching in English/Marathi and Bilingual (English + Marathi) medium is introduced at first year of AICTE diploma Programme from academic year 2021-2022. Hence if the students in first year (first and second semesters) write answers in Marathi or bilingual language (English +Marathi), the Examiner shall consider the same and assess the answer based on matching of concepts with model answer.

Q. No.	Sub Q. N.	Answer	Marking Scheme
1		**Attempt any FIVE of the following:**	**10 M**
	a)	**Define the testing terminology** **i) Error** **ii) Fault** **iii) Defect** **iv) Bug**	**2 M**
	Ans	i) Error: : An error is a human action that produces the incorrect result ii) Fault: State of software caused by an error iii) Defect: A defect is an error or a bug, in the application which is created. A programmer while designing and building the software can make mistakes or error. These mistakes or errors mean that there are flaws in the software. These are called defects. iv) Bug: The presence of error at the time of execution of the software.	½ M for each definition
	b)	**List the levels of testing.**	**2 M**
	Ans	Following are the levels of testing: a) Unit test b) Integration test	½ M for each level

		c) System test d) Acceptance test	
	c)	**State any four needs to prepare a test plan.**	**2 M**
	Ans	Need of test plan: ● Test Plan Ensures all Functional and Design Requirements are implemented as specified in the documentation. ● Test plan gives detail aspects such as test scope, test estimation, strategy, etc. ● Test plan determines the time, cost, and effort. ● It helps in determining the quality of software applications. ● Provide a schedule for testing activities. ● Test Plan Document can be used for similar projects. ● It helps to understand the test details.	1/2 M for each need
	d)	**Give the defect classification and its meaning.**	**2 M**
	Ans	**Requirement/Specification Defects:** Requirement-related defects arise in a product when one fails to understand what the customer requires. These defects may be due to the customer gap, where the customer is unable to define his requirements. Producer gap, where the developing team is not able to make a product as per requirements. **Design Defects:** Design defects occur when system components, interactions between system components, interactions between the outside software/hardware, or users are incorrectly designed. Design defects generally refer to the way of design creation or its usage while creating a product. **Coding Defects:** This defect arises when variables are not initialized properly or variables are not declared correctly or database is not created properly. Coding also needs adequate commenting to make it readable and maintainable in future. **Testing Defects:** These would encompass incorrect, incomplete, missing inappropriate test cases and test procedures.	½ M for each classification and meaning

e) Compare verification and validation (any two points). **2 M**

Ans 1 M for each point (2 point)

verification	validation
It includes checking documents, design, codes, and programs.	It includes testing and validating the actual product.
Verification is the static testing.	Validation is the dynamic testing.
It does not include the execution of the code.	It includes the execution of the code.
Methods used in verification are reviews, walkthroughs, inspections, and desk checking.	Methods used in validation are Black Box Testing, White Box Testing, and non-functional testing.
It checks whether the software conforms to specifications or not.	It checks whether the software meets the requirements and expectations of a customer or not
Quality assurance team does verification.	Validation is executed on software code with the help of testing team.

f) State the need of automated testing tools. **2 M**

Ans 1/2 M for each need (any 4 should write)

- An automated testing tool can playback pre-recorded and predefined actions, compare the results to the expected behavior and report the success or failure of these to a test engineer.
- Once automated tests are created, they can easily be repeated, and they can be extended to perform tasks impossible with manual testing.
- Automated Software Testing Saves Time and Money.
- Software tests must be repeated often during development cycles to ensure quality.
- Every time source code is modified software tests should be repeated.
- For each release of the software, it may be tested on all supported operating systems and hardware configurations. Manually repeating these tests is costly and time consuming
- Once created, automated tests can be run repeatedly at no additional cost, and they are much faster than manual tests.
- Testing Improves Accuracy, Even the most conscientious tester will make mistakes during monotonous manual testing.
- Automated tests perform the same steps precisely every time they are executed and never forget to record detailed results.

		• They can even be run on multiple computers with different configurations. • Automated software testing can look inside an application and see memory contents, data tables, file contents, and internal program states to determine if the product is behaving as expected.	
	g)	**Give the objectives of software testing.**	**2 M**
	Ans	• To find any defects or bugs that may have been created when the software was being developed • To increase confidence in the quality of the software • To prevent defects in the final product • To ensure that end product meets customer requirements as well as specifications • To provide customers with a quality product and increase their confidence in the team.	1 point for each objective (any 2 points are required)
2.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**State the Entry and Exit criteria's for the software testing.**	**4 M**
	Ans	**Entry criteria** Entry criteria are the condition or the set of conditions, which should exist or be met in order to start a process. Some of the conditions or situations, which may be seen as an entry criterion for the initiation of testing activities. • Requirements should be clearly defined and approved. • Test Design and documentation plan is ready. • Availability of the test environment supporting necessary hardware, software, network configuration, settings, and tools for the purpose of test execution. • Testers are trained, and necessary resources are available. • Availability of proper and adequate test data (like test cases). • It depends upon which software development model is used. **Exit criteria** Exit Criteria is often viewed as a single document concluding the end of a life cycle phase. Some of the conditions or situations which may be seen as an exit criterion for testing activities. • Testing Deadline • Completion of test case execution. • Completion of Functional and code coverage to a certain point.	2 M for entry criteria and 2 M for exit criteria

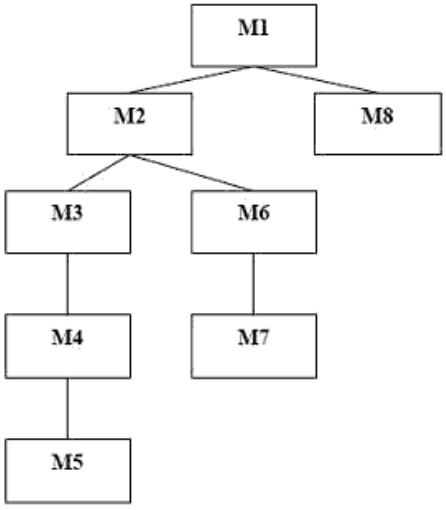
		• Bug rates fall below a certain level and no high priority bugs are identified. • Management decision.	
	b)	**State and describe top-down approach of integration testing with diagram.**	**4 M**
	Ans	**Top-down integration** ➢ Modules are integrated by moving downward through the control hierarchy, beginning with the main module. ➢ It takes help of dummy program called stub for testing. ➢ Subordinate modules are incorporated in either a depth-first or breadth-first fashion. Integration can be done in two ways: • Depth First Method: All modules on a major control path are integrated. • Breadth First method: All modules directly subordinate at each level are integrated.  **Fig-Top-down integration** **Incremental approach→ Top-down integration procedure** 1. Main control module used as a test driver and stubs are substitutes for components directly subordinate to it. 2. Subordinate stubs are replaced one at a time with real components. (Following the depth-first or breadth-first approach). 3. Tests are conducted as each component is integrated. 4. On completion of each set of tests and other stub is replaced with a real component. 5. Regression testing may be used to ensure that new errors not introduced.	1 M for diagram 3 M for explanation
	c)	**Describe the "Test Infrastructure' components with diagram.**	**4 M**
	Ans	Testing requires a robust infrastructure to be planned upfront. This infrastructure is made up of three essential elements.	Component/ Diagram-1 M Explaination-3 M

Fig: Components of Test Infrastructure

1. **A test case database (TCDB):** A test case database captures all the relevant information about the test cases in an organization. Some of the entities and the attributes are given in the following table.

Sr. No.	Test Case	Purpose	Attributes
1	Test case	Records all static information about tests.	1)Test case Id 2) Test case name (File name) 3) Test case owner 4) Associated files for test case.
2	Test case product cross reference	Provide mapping between the tests and the corresponding product features, enables identification of test cases for given feature.	Test case Id Module Id
3	Test case run history	Gives the history of when the test case was run and what was result, provided inputs on selection of test for regression runs	1) Test case Id 2) Run date 3) Time taken 4) Run status (Success/ Failure)
4	Test case defect cross reference	Gives details of test cases introduced to test certain specific defects detected in the product, provides inputs on the selection of test for regression runs.	1) Test case Id 2) Defect reference

2. **Defect Repository**: It captures relevant details of defects. It is a tool of communication. Defects matrices are derived from defect repository.
3. **Configuration management repository and tools**: They keep track of change control of all the files/entities that make up a software product. They keep track of version control of all files/entities that makeup a software product.

<table>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>d)</td><td>State the limitations of manual testing.</td><td>4 M</td></tr>
<tr><td></td><td>Ans</td><td>i. Manual testing is slow and costly.
ii. It is very labour intensive; it takes a long time to complete tests.
iii. Manual tests don’t scale well. As the complexity of the software increases the complexity of the testing problem grows exponentially. This leads to an increase in the total time devoted to testing as well as the total cost of testing.
iv. One tester may approach and perform a certain test differently from another, resulting in different results on the same test, because the tests are not being performed identically.
v. GUI objects size difference and color combinations are not easy to find in manual testing.
vi. Not suitable for large scale projects and time bound projects</td><td>Write any 4 limitations for 4 M</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td>3.</td><td></td><td>Attempt any <u>THREE</u> of the following:</td><td>12 M</td></tr>
<tr><td></td><td>a)</td><td>Differentiate between white box testing and black box testing (any four points).</td><td>4 M</td></tr>
<tr><td></td><td>Ans</td><td>
<table>
<tr><th>Sr. No.</th><th>Black Box Testing</th><th>White Box Testing</th></tr>
<tr><td>1.</td><td>It is a way of software testing in which the internal structure or the program or the code is hidden, and nothing is known about it.</td><td>It is a way of testing the software in which the tester has knowledge about the internal structure or the code or the program of the software.</td></tr>
<tr><td>2.</td><td>It can be referred to as outer or external software testing.</td><td>It is the inner or the internal software testing.</td></tr>
<tr><td>3.</td><td>It is a functional test of the software.</td><td>It is a structural test of the software.</td></tr>
<tr><td>4.</td><td>This testing can be initiated based on the requirement specifications document.</td><td>This type of testing of software is started after a detailed design document.</td></tr>
<tr><td>5.</td><td>It is the behavior testing of the software.</td><td>It is the logic testing of the software.</td></tr>
<tr><td>6.</td><td>It is also called closed testing.</td><td>It is also called clear box testing.</td></tr>
</table>
</td><td>Any 4 Points- 1 M each</td></tr>
</table>

		7.	Can be done by trial-and-error ways and methods.	Data domains along with inner or internal boundaries can be better tested.	
	b)	**State the contents of "Test Summary Reports" used in test reporting.**			**4 M**
	Ans	Test reporting is a means of achieving communication through the testing cycle. There are 3 types of test reporting. 1. Test incident report: 2. Test cycle report: 3. Test summary report: **Test summary Report**: The final step in a test cycle is to recommend the suitability of a product for release. A report that summarizes the result of a test cycle is the test summary report. **There are two types of test summary report:** 1. **Phase wise test summary**, which is produced at the end of every phase. 2. **Final test summary report**, which has all the details of testing done by all phases. A Summary report should be presented. 1. Test Summary Report Identifier 2 Description: Identify the test items being reported in this report with test id 3 Variances: Mention any deviation from test plans, test procedures, if any. 4 Summary of results: All the results are mentioned here with the resolved incidents and their solutions. 5 Comprehensive assessment and recommendation for release should include: Fit for release assessment and recommendation of release.			Types – 1 M Contents-3 M
	c)	**Prepare defect report after executing test cases for any login form.**			**4 M**
	Ans	Defect Report in Software Testing is a detailed document about bugs found in the software application Following is Defect report after executing test cases for **Email-log in form.**			Format of defect report- 2 M

ID number	#123
Name	loginform - Unable to login Email
Reporter	Person's name (xyz)
Submit Date	03/01/2023
Summary	When I put my mail id and password, I am unable to

		<table><tr><td></td><td>login while login credentials are right.</td></tr><tr><td>URL</td><td>www.gmail.com</td></tr><tr><td>Screenshot</td><td>https://accounts.google.com/signin/</td></tr><tr><td>Platform</td><td>AngularJS</td></tr><tr><td>Operating System</td><td>OS X 10.12.0</td></tr><tr><td>Browser</td><td>Chrome 53</td></tr><tr><td>Severity</td><td>Major</td></tr><tr><td>Assigned to</td><td>/</td></tr><tr><td>Priority</td><td>High</td></tr></table> **Description** When I put mail id and password, I am unable to login while login credentials are right. **Steps to reproduce** > go to the www.gmail.com > Click on login button > Put Right mail id and password and click next. > and take Screenshot. **Expected result** The mail account should logged in after putting the right mail id and password. **Actual result** The mail account is not logging in after putting the right details.	Description-2 M
	d)	**Enlist the factors considered for selecting a testing tool for test automation.**	**4 M**
	Ans	The following factors are important during tool selection: i. Assessment of the organization's maturity (e.g., readiness for change). ii. Identification of the areas within the organization where tool support will help to improve testing processes. iii. Evaluation of tools against clear requirements and objective criteria. iv. Proof-of-concept to see whether the product works as desired and meets the requirements and objectives defined for it. v. Evaluation of the vendor (training, support and other commercial aspects) or open-source network of support.	Any 4 factors- 1 M each

		vi. Identifying and planning internal implementation (including coaching and mentoring for those new to the use of the tool).	
4.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**Describe graphical user interface (GUI) testing and its important traits.**	**4 M**
	Ans	GUI Testing ● There are two types of interfaces for a computer application. ● Command Line Interface is where you type text and the computer responds to that command. ● GUI stands for Graphical User Interface where you interact with the computer using images rather than text. ● GUI testing is the process of testing the system's Graphical User Interface of the Application Under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars - toolbar, menu bar, dialog boxes and windows, etc. ● GUI is what the user sees. A user does not see the source code. The interface is visible to the user. Especially the focus is on the design structure, images that they are working properly or not. **GUI Testing Guidelines** 1. Check Screen Validations 2. Verify All Navigations 3. Check usability Conditions 4. Verify Data Integrity 5. Verify the object states 6. Verify the date Field and Numeric Field Formats **Advantages of GUI Testing:** ● Good GUI improves the feel and look of the application; it psychologically accepts the application by the user. ● GUI represents a presentation layer of an application. Good GUI helps an application due to better experience of the users. ● Consistency of the screen layouts and designs improves usability of an application.	Description - 2 M Importance -2 M

	b)	**Describe test deliverables in details.**	**4 M**
	Ans	1) Test Deliverables are the artifacts which are given to the stakeholders of a software project during the software development lifecycle. There are different test deliverables at every phase of the software development lifecycle. Some test deliverables are provided before the testing phase, some are provided during the testing phase and some after the testing cycle is over. The different types of Test deliverables are: ● Test cases Documents ● Test Plan ● Testing Strategy ● Test Scripts ● Test Data ● Test Traceability Matrix ● Test Results/reports ● Test summary report ● Install/config guides ● Defect Reports ● Release notes 2) The test plan describes the overall method to be used to verify that the software meets the product specification and the customer's needs. It includes the quality objectives, resource needs, schedules, assignments, methods, and so forth. 3) Test cases list the specific items that will be tested and describe the detailed steps that will be followed to verify the software. 4) Bug reports describe the problems found as the test cases are followed. These could be done on paper but are often tracked in a database 5) Test tools and automation are listed and described which are used to test the software. If the team is using automated methods to test software, the tools used, either purchased or written in-house, must be documented. 6) Metrics, statistics, and summaries convey the progress being made as the test work progresses. They take the form of graphs, charts, and written reports	Definition-1M Types-1 M any other 4 points - 2 M
	c)	**Describe load testing and stress testing with suitable example.**	**4 M**
	Ans	**Load Testing** ● Load Testing is a type of performance testing to check system with constantly increasing the load on the system until the time load reaches its threshold value. ● Here Increasing load means increasing number of concurrent users, transactions & check the behavior of the application under test. ● It is normally carried out underneath controlled environment to distinguish between two different systems. ● The main purpose of load testing is to monitor the response time and staying power of application when the system is performing well under heavy load. ● The successfully executed load testing is only if the specified test cases are executed without any error in allocated time. ● Load testing is testing the software under customer expected load.	Description of each- 1 M each

		● In order to perform load testing on the software you feed it all that it can handle. Operate the software with the largest possible data files. ● If the software operates on peripherals such as printer, or communication ports, connect as many as you can. ● If you are testing an internet server that can handle thousands of simultaneous connections, do it. With most software it is important for it to run over long periods. ● Some software should be able to run forever without being restarted. So, Time acts as an important variable. Load testing can be best applied with the help of automation tools. **Simple examples of load testing**: ● Testing printers by sending large jobs. ● Editing a very large document for testing of word processor ● Continuously reading and writing data into the hard disk. ● Running multiple applications simultaneously on the server. ● Testing of mail server by accessing thousands of mailboxes ● In case of zero-volume testing & system fed with zero load **Stress Testing** ● Stress Testing is performance testing type to check the stability of software when hardware resources are not sufficient like CPU, memory, disk space etc. ● It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above the expected maximum. ● Main parameters to focus during Stress testing are "Response Time" and "Throughput". ● Stress testing is Negative testing where we load the software with large number of concurrent users/processes which cannot be handled by the systems hardware resources. This testing is also known as **Fatigue testing.** Stress testing is testing the software under less-than-ideal conditions. So, subject your software with low memory, low disk space, slow CPU, slow modems and so on. Look at your software and determine what external resources and dependencies it has. Stress testing is simply limiting them to a bare minimum. With stress testing you starve the software. For e.g. Word processor software running on your computer with all available memory and disk space, it works fine. But if the system runs low on resources, you have a greater potential to expect a bug. Setting the values to zero or near zero will make the software execute a different path as it attempts to handle the tight constraint. Ideally the software would run without crashing or losing data	Example of each- 1 M each
	d)	**State the advantages and disadvantages of using tools.**	**4 M**
	Ans	**Advantages of using tools:** **Save Time /Speed:** Due to advanced computing facilities, automation test tools prevail in speed of processing the tests. Automation saves time as software can execute test cases faster than humans.	Advantages any 4 - 2 M

Reduces the tester's involvement in executing tests: It relieves the testers to do some other work.

Repeatability/Consistency: The same tests can be re-run in exactly the same manner eliminating the risk of human errors such as testers forgetting their exact actions, intentionally omitting steps from the test scripts, missing out steps from the test script, all of which can result in either defects not being identified or the reporting of invalid bugs (which can again, be time consuming for both developers and testers to reproduce)

Simulated Testing: Automated tools can create many concurrent virtual users/data and effectively test the project in the test environment before releasing the product.

Test case design: Automated tools can be used to design test cases also through automation, better coverage can be guaranteed than if done manually.

Reusable: The automated tests can be reused on different versions of the software, even if the interface changes.

Avoids human mistakes: Manually executing the test cases may incorporate errors. But this can be avoided in automation testing.

Internal Testing: Testing may require testing for memory leakage or checking the coverage of testing. Automation can do this easily.

Cost Reduction: If testing time increases, the cost of the software also increases. Due to testing tools time and therefore cost is reduced.

Disadvantages 2 M

Disadvantages of using tools:
- Unrealistic expectation from the tool
- People always make mistake by understanding time cost and effort for the initial introduction of the tool
- People frequently miscalculate the time and effort needed to achieve significant and continuing benefits from the tools
- Mostly people underestimate the effort required to maintain the test assets generated by the tool
- People depend on the tool a lot. (Over reliance on the tool)

e) Write the test cases for Notepad application. (any eight test case) **4 M**

Ans

PROJECT:	**NOTEPAD**
MODULE:	**FIND AND REPLACE**
FUNCTIONAL SPECIFICATION:	**FIND AND REPLACE**
TEST CASE NO: -	**TC-FR-1**
TEST OBJECTIVE: -	**To Check functionality of " Find and Replace" in notepad.**

ENVIRONMENT:WIN 2k, Notepad.

TC#	Test Scenario	Pre-Condition	Test Steps	Test Data	Expected Result	Actual Result	Remarks
1&2	Check the availability of the Find Option.	-	Click the Edit menu from the menu bar.		After clicking on the **find**,the window should pop up	The search box available.	Pass
3-a)	Check the navigation through Shortcut keys	-	Press Ctrl + F		After pressing CTRL + F , should produce the search box	The find box available.	Pass
3-b)	Check the navigation through Shortcut keys	-	Press Ctrl + H		After pressing CTRL + H, should produce the replace box	The Replace box available.	Pass
3-C)	Check the navigation through mouse	-	Click Edit menu from menu bar and then FIND		In the menu by clicking the edit and then find, the search box should open	The find Box is available.	Pass
4	The cursor default position.	Find box should be open.			The cursor should be present in the typing space box.	The cursor is Available	Pass
5	To Check find Button.	Find box should be open.	Press Find Button		Without typing anything, the **find** button should not be enabled and functional	It is not enabled.	Pass
6	To check the Search control in the page.	Find box should be open.	Press Find Button		The typed text in the search field should match, otherwise generate an	If it is matching, then it highlights it.	Pass

Any eight valid test cases ½ M each

							error that word does not exist.				
			7	If the user want to search a single word, more than once. Then after competition of 1st search, the search button should be enable for next also	Find box should be open.	Press Find Button, Find Next		The Search/Find button should be enabled for the next search also.	The find button is available for next search also.	Pass	
			8	To check the Replace control.	Find box should be open.	Click on Replace		The user should be asked before replacing any word.	The message is not coming.	Fail	
			9-a)	Replace the exact work	Find box should be open.	Click on Replace		By clicking "match whole word only", it should replace only the whole word.	It is replacing the whole word only.	Pass	
			9-b)	Don't replace when there is no text in replace space	Find box should be open.	Click on Replace		Don't replace when there is no text in replace space	It is replacing the find value with the blank space.	Fail	
			10	Check Cancel button functionality	Find box should be open.	Click Cancel		After pressing the cancel button the window should exit.	The window is exit.	pass	
5.		**Attempt any <u>TWO</u> of the following:**									**12 M**
	a)	**Design test cases for simple calculator application. (Black box testing.) (Any six points.)**									**6 M**

Ans

Test Case - ID	Test case Objective	Input data	Expected Result	Actual Result	Status
TC-1	To add two integer and display the result on ten-digit calculator	176 + 100	276	276	Pass
TC-2	To subtract two integer and display the result on ten-digit calculator	176 - 100	76	76	Pass
TC-3	To multiply two integer and display the result on ten-digit calculator	100 x 20	2000	2000	Pass
TC4	To divide two integer and display the result on ten-digit calculator	100/ 5	20	20	Pass
TC5	To clear the screen		Symbol "0" should appear on screen	Symbol "0" appears on screen	Pass
TC6	To delete digits one by one		One Digit should be deleted from right hand side	One Digit is deleted from right hand side	Pass

6 test cases of test cases for simple calculator application: 6 M; 1M each; any other valid test cases shall be considered

b) **Design test cases for Web pages testing of any Web site (take a suitable example).** **6 M**

Ans

Test Case ID	Test case objective	Input data	Expected result	Actual result	Status
TC1	Check cursor position at email or mobile number field	Click on email or mobile number field	Cursor should be placed on the field	Placed the cursor on the field	Pass
TC2	Check cursor position at password field	Click on password field	Cursor should be placed on the password field	Placed the cursor on the password field	Pass
TC3	Check the continue button	Click on continue button	It should redirect to password page	It redirected to the password page.	Pass
TC4	Readability of font	Try to read the contents on login page	Contents should be readable	Content s are readable	Pass
TC5	Testing of spelling of login	Check the spelling of login	Login spelling should be correct	Spelling of Login is correct	Pass
TC6	Testing of hyperlink	Hover the mouse on hyperlink	It should change the cursor and should redirect to respective page on click	Cursor changed and redirects to other page.	Pass

6 test cases of test cases for any web site: 6 M; 1M each; any other valid test cases shall be considered

c) Design test cases for MS Word application using an Automation tool. **6 M**

Ans

Test Case ID	Test case objective	Input data	Expected result	Actual result	Status
TC1	Check whether Undo in Edit main menu undoes the previous action		Previous action should be undone	Previous action was undone	Pass
TC2	Checks whether the Undo button in right click context menu undoes the previous action		Previous action should be undone	Previous action was undone	Pass
TC3	Checks whether Undo button in the Edit main menu is disabled when there is not any previous actions		Undo Button should be disabled	Undo Button was disabled	Pass
TC4	Checks whether Undo button in right context menu is disabled when there are not any previous actions		Undo Button should be disabled	Undo Button remained disabled	Pass
TC5	Checks whether hotkey (CTRL+Z) response when there is no any of previous actions		No response is expected	No response	Pass
TC6	Checks whether the Cut options in Edit main menu cuts the selected text		Selected text should be cut	Selected text was cut	Pass
TC7	Checks whether the Cut options in Edit Menu is disabled when no texts are selected		Cut Options should be disabled	Cut Option Was Disabled	Pass

Above test cases will be executed on any automation tool like autoIT, QTP etc.

6 test cases of test cases for any function of MS Word: 6 M; 1M each; any other valid test cases shall be considered

6.		**Attempt any <u>TWO</u> of the following:**	**12 M**
	a)	**Write program for calculating even numbers from 1 to 20 And design test cases for the same.**	**6 M**

	Ans	Program :
		#include <stdio.h>
		int main()
		{
		for(int i=0;i<=20;i=i+2)
		{
		printf("%d\n",i);
		}
		return 0;
		}
			Correct program: 2M; valid test cases shall be considered:4 M

Test Case ID	EV_001	**Test Case Description**	Test program prints the even number from 1 to 20		
Created By	ABC	**Reviewed By**	PQR	**Version**	2.1
Tester's Name	ABC	**Date Tested**	1-Jan-2023	**Test Case (Pass/Fail/Not Executed)**	Pass

S #	Prerequisites :
1	C program using for loop

S #	Test Data

<u>Test Scenario</u>	Verify the even number

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Check initial condition of for loop	Initial value of For loop should be 0 or 1	Initial value of For loop is 0 or 1	Pass
2	Check final condition of for loop	Final condition should be "< 20" or "<=20"	Final condition is "< 20" or "<=20"	Pass

		3	Check the increment operator	Increment operator should increment by 2	Counter is incremented by 2	Pass
		4	Check output	Even number is displayed on output screen	It is displaying even number	Pass
	b)	**Prepare test plan for "Cam Scanner' which is installed on mobile.**				**6 M**
	Ans	**Test plan for Cam Scanner:** **Test Plan Identifier** TP_10 **Introduction:** The purpose of this document is to create test plan for CamScanner application installed on mobile. The purpose of testing this program is to check the correct operation of its functionality, ease of use. **Test Items:** Working with the document (Scan document, Edit document, PDF conversion) **Features to be tested** ● Scan Document ● Edit Document ● PDF Conversion **Approach** ● On the test object: ○ functional ○ non-functional ● According to the requirements ○ positive ○ negative ● By degree of preparedness - intuitive testing (ad hoc) **Item Pass/Fail Criteria:** All test cases with high priority are closed with the result - pass. The test coverage is checked and sufficient, where the criterion of sufficiency is not less than 99% of the coverage of requirements by tests. The test report was compiled and approved by the team lead and customer. **Suspension Criteria and Resumption Requirements** Criterion for interrupting testing: ● The appearance and entering into the bug-tracking system of blocking bugs. Criterion for continuation of testing: ● Closing the blocking bug in the bug tracking system. **Test Deliverables:** Test plan, test cases, test report. **Test Tasks** ● Writing a test plan ● Writing test cases ● Development of criteria for the success of testing ● Conducting the testing and evaluation of the results ● Creating test reports				

Environmental Needs
Mobile Phone
CamScanner Installed
Responsibilities

Sr. no	Functionality and Responsibilities	Responsible
1	Scan Document	Test Engineer 1
2	Edit Document	Test Engineer 1
3	PDF Conversion of Document	Test Engineer 3

Staffing and Training Needs
To perform the tasks, you need to have the following knowledge and skills:

- knowledge and practical application of the camscaaner;
- knowledge and ability to apply in practice the basic techniques of test design
- Knowledge of various types of testing including functional and non-functional.

Schedule
The deadline for completion of all works and delivery of the project is 25/01/2023 by 5.00pm
Risks and Contingencies
Possible risks during testing:

- Insufficient human resources for testing the application in deadlines.
- Changing the requirements for the product

Approvals
Team Lead
Test engineer 1
Test engineer 2

c) Prepare defect report after executing test cases for withdrawn of amount from ATM machine. — **6 M**

Ans

ID	**R1**
Project	**Cash Simulator Cash (ATM)**
Product	http://www.motc.gov.qa/en/ditoolkit/migrantworkers/cash-machine-simulator-atm
Release Version	v1.0
Module	Home Page> Our Programs > Digital Inclusion tools
Detected Build Version	V1.1
Summary	Limited denomination options in cash withdrawal function, restricting cash withdrawal only till 3000
Description	No option of withdrawing of amount excess of 3000.
Steps to Replicate	1. Open the website 2. Select our programs 3. Proceed to Digital Inclusion tools and select cash machine simulator (ATM)

any valid defect report related with withdrawal functionality shall be considered

				4. Select language and skip to simulator 5. Enter the card 6. Select the account type 7. Go to Other functions and select cash withdrawal		
			Expected Results	It should add more options in denominations in withdrawal function or it should take amount input from the user.		
			Actual Results	It is displaying limited options of denominations in cash withdrawal option.		
			Attachments			
			Remarks	Causes inconvenience to the user in terms of limited cash withdrawal options.		
			Defect Severity	High		
			Defect Priority	High		
			Reported By	Test Engineer1		
			Assigned To	XYZ		
			Status	Assigned		