MCQ's-set-1

1. Refactoring is not one of the stages of TDD**.**

- **False**

2 TDD required a developer to write the test cases before writing the actual production code.

- **True**

3. TDD is basically unit testing.

- **False**

4. TDD can help a developer improve his initial design of a method by forcing him to come up with situations in which the test case can fail

- **True**

5. TDD does not help in detection of bugs at an early developmental stages of SDLC.

- **False**

6. There are **3** types of TDD.

7. TDD has been discovered or reinvented by **KENT BECK**

8. TDD involves automating of test cases according to the need of the user

- **True**
- False

9. Give an example of a situation where TDD will not work.(Hint: Think of situations where entire production code is required before starting the testing phase)

The development of a new social media platform with innovative features and complex user interactions. In this scenario, the project may require extensive frontend and backend development to implement the various features and functionalities. Since the platform's architecture and user experience need to be well-defined before testing, following the strict TDD approach might not be practical.

10. ATDD is also called **Acceptance Test- Driven Development.**.

MCQ's-set-2

1. What does BDD stand for
   - **Behaviour Driven Development**

2. BDD is about the collaboration between teams
   - **True**

3. What are the 3 Practices of BDD
   - **Discovery, Formulation, Automation**

4. What is the sequence of the 3 Practices of BDD
   - **Discovery, Formulation, Automation**

5. What is the outcome of the Discovery phase

The outcome of the discovery phase typically includes a thorough understanding of the project's objectives, requirements, constraints, risks, and opportunities. It often results in the creation of a project plan or roadmap to guide the development process.

6. What is the outcome of the Formulation phase
   - **Documented Examples of system behavior**

7. What is the outcome of the Automation phase
    - **Implemented Code + Automation Tests**

8. What is Cucumber
   - **All of the above**

9. Using Cucumber means you are doing BDD
   - **True**

10. If you want to implement BDD, you have to use Cucumber
    - **No**

Mcq Ques-set-3

1. The process of BDD starts with
   - **Discovery**

2. Why we call the Discovery meeting as the"**3Amigosmeeting**"
   - **To have a different perspective from atleast 3different teams to discuss and create common understanding**

3. Who are the"3Amigos"
   - **Product Owner, Developer & Tester**

4. What is a USER STORY
   - **General description of the software feature or change from an end-user perspective**

5. Who writes a USER STORY
   - **Product Owner**

6. What is the purpose of the Discovery meetings
   - **To explain the user story and explore and discuss the system behavior**

7. In Discovery Meetings User Stories can be updated based on new rules or features discovered during the meeting

- **Yes**

8. What is the sequence of the 3 Practices of BDD
   - **Discovery, Formulation, Automation**

9. If you are not doing Discovery meetings, you are not doing BDD

- **False**

10. What is the out come of the Discovery phase
    - **Common agreed understanding of the behavior of the system**

Ques- set-4

1 .

AGILE is an iterative development methodology, where bothdevelopment and testing activities are concurrent.

A)Agile

2 .

Agile Testing covers all the levels of testing and all types oftesting.

A)True

3 .

Scrum is an Agile development method that emphasizes onteam-centric approach.

A) Scrm

4 .

What are advantage's of Agile testing?

A) All of the above

5 .

FDD stands for?

A) Feature Driven Development

6 .

Which testing is based on the expected behavior of thesoftware being developed?

C)BDD

7 .

The product backlog is prioritized by_____

A)product owner

8 .

What is the Agile?

D)Both (A) and (C)

9 .

How do tools like FIT and Fitness compliment Agile?

A) Automated Acceptance Tests

10 .

In Agile Testing Practices, which of them has two teammembers work together at the same keyboard?

A)Pairing

ques- set-5

1) White box texting is also known as__**Glass box testing**_____
2) White box texting is dynamic ,static, or infeasible **Ans: Dynamic**
3) A **Negative** test is when an invalid input is put and error are received
4) **Automated** tool is involved in automation of regression test
5) **Equivalent random** test cases are when random test cases are performed and equivalence partioning is applied to those test cases

6) **Technical** is normally used to evaluate a product to determine itssuitability for intended use and to identify discrepencies

7) **Re-testing** insures the original fault has been removed

8) Static analysis cannot

- **Enforce coding**

Set-6 •

1.Not a Behavior-Driven Development tools

Concordat

2. Check if all the steps have the step definition before execute

DryRun

3. Which is not a Cucumber Report

None of the above •

4. _____is the actual code implementation of the feature mentioned in feature file.

Step definition

5. What does Cucumber do?

Cucumber tests other software

6. In which programming language was Cucumber originally written?

Ruby •

7. What language does Cucumber use?

Gherkin

8. What does Gherkin do?

Provides simple documentation of the code under test •

9. Which sign is used in Gherkin's syntax?

#

10. Which file extension do all Gherkin files have?

feature

11. What is Cucumber?

A tool

12. _____is a process of developing software based on behavioral specification of software units.

BDD

13. _____is a software development technique where automated tests are written before the code.

TDD

14. This person is most concerned with the scope of the application. This involves translating user stories into a series of features.

Product Owner

15. What's the subdivision of Gherkin's features?

Scenarios

16. How many parts does a Gherkin feature have?

2

17. Which one of these Formatter Plugins Cucumber doesn't use to provide output?

Www

18. Which languages does Cucumber not support?

C

19. What software is needed to run a Cucumber Web Test?

All of the above •

20. What are the advantages of Cucumber?

All of the above

21)Mention atleast two differences between tdd and bdd?

TDD	BDD
Software development process in which the requirements are turned into specific test cases and then the software is improved to pass the new tests	Agile software development process that encourages collaboration between developers, QA and non-technical or business participants in a software project
TDD stands for Test Driven Development	BDD stands for Behavior Driven Development
The tests are written to cover each functionality/unit	Concerns on the application as a whole
TDD involves designing high level scenarios	BDD involves designing low level scenarios

1.write a simple string calculator using TDD principles.(CALCI)
• The function should return 0 for an empty string.
• Write the function to handle an empty string and return 0.
• The function should add numbers separated by commas.

Write a failing test for an empty string: Start by writing a test that expects the function to return 0 when passed an empty string.

Implement the minimal code to pass the test: Write the simplest code possible to make the test pass. In this case, the function should return 0 if the input string is empty.

Write a failing test for single numbers: Next, write a failing test for the case where the function should return the number itself if only one number is provided in the string.

Modify the function to handle single numbers: Adjust the function to return the single number if only one number is provided in the string.

Write a failing test for multiple numbers: Now, write a failing test for the case where the function should return the sum of multiple numbers separated by commas.

Modify the function to handle multiple numbers: Update the function to split the string by commas, convert each number to an integer, and return the sum of all numbers.

Write additional tests for edge cases: Add more tests to cover different scenarios, such as numbers with spaces, mixed delimiters (e.g., commas and newlines), and any other edge cases you can think of.

Refactor and optimize the code (if necessary): Once all tests pass, refactor the code if needed to improve readability or performance.

Repeat the process: If there are more features or requirements, repeat the process by writing failing tests and implementing the code to make them pass.

By following these steps, you can gradually build and improve your string calculator while ensuring its correctness through automated tests.

TDD:(2. write a password validator function using VALIDATOR)
• The function should return "Password cannot be empty" for an empty string
• (Initial): Define the function with a placeholder message
• The function should return "Password must be at least 8 characters long" for a password less than 8 characters.
• Update the function to check password length and return the appropriate message if it's less than 8 characters

Function Definition (validate_password):
The function validate_password takes one parameter, password, which represents the password to be validated.

Checking for an Empty Password:
Inside the function, there's an if statement that checks if the password is empty. It does this by evaluating not

password. If password is empty (i.e., ""), then not password evaluates to True, and the code block inside the if statement is executed.

Returning a Message for an Empty Password:
If the password is empty, the function returns the message "Password cannot be empty". This message indicates that an empty password is not allowed.

Test Case (test_validate_password_empty):
The test case test_validate_password_empty verifies the behavior of the validate_password function when an empty password is provided.
It uses an assert statement to check if calling validate_password("") returns the expected message "Password cannot be empty".

If the condition is True, the test passes; otherwise, it fails.

Overall, this code implements the first step of the password validator function as per the requirements: returning a specific message when the password is empty. This is the initial step in Test-Driven Development (TDD), where you start by writing a failing test case and then implement the simplest solution to make the test pass.