1. Which of the following statements is a key principle of testing? 1 / 1 point

- ○ You should run as few software tests as possible.
- ○ You should prefer manual tests over automated tests.
- ◉ You can only test for the factors that you know.
- ○ You can save testing for the beta stage onward.

Correct
Correct! No matter how well developers design a product, they can only test what they know.

2. Testing has which of the following benefits for development? 1 / 1 point

- ○ Increases code readability
- ◉ Improves code quality
- ○ Reduces project spending
- ○ Produces file backups

Correct
Correct! Testing allows developers to refactor code and add new features with confidence.

3. At which level of the software testing process do developers test the entire software process? 1 / 1 point

- ○ Integration testing
- ○ Acceptance testing
- ◉ System testing
- ○ Unit testing

Correct
Correct! In system testing, developers ensure that the entire system works together.

4. What is the purpose of integration testing? 1 / 1 point

- ◉ Expose flaws in the interaction between components
- ○ Validate that each unit performs as designed
- ○ Assess the system's acceptability for delivery
- ○ Evaluate compliance with specific technical requirements

Correct
Correct! The purpose of integration testing is to expose flaws in the interaction between integrated units or components.

5. Behavior driven development (BDD) ensures which of the following for developers? 1 / 1 point

- ○ Bolstering customer service
- ○ Identifying the code's purpose
- ◉ Building correct software
- ○ Writing concise code

Correct
Correct! BDD ensures that developers are building the correct software.

6. In the basic workflow for test driven development (TDD), what happens in the third step? 1 / 1 point

- ○ Write the program code.
- ○ Create the test case.
- ○ Describe the system's behavior.
- ◉ Improve the code's quality.

Correct
Correct! In the third step in the basic TDD workflow, developers refactor the code to improve its quality.

7. In which Python testing framework can you use nearly an infinite number of setup and teardown levels? 1 / 1 point

- ○ RSpec
- ◉ Pytest
- ○ Doctest
- ○ PyUnit

Correct
Correct! With Pytest, you can use nearly infinite setup and teardown levels.

8. Assume you are using Python and have installed PyUnit, Nose, and Pinocchio. Which of the following commands can you use to perform unit tests and receive color-coded output? 1 / 1 point

- ○ nosetests --with-spec –cover-tests
- ○ unittest --with-spec –cover-tests
- ○ unittest --with-spec --spec-color
- ◉ nosetests --with-spec --spec-color

Correct
Correct! To perform unit tests and receive color-coded output, run nosetests --with-spec --spec-color.

9. Which of the following PyUnit assertions is useful for determining if an object is in a result set? 1 / 1 point

- (●) assertIn()
- ( ) assertIsInstance()
- ( ) assertRegex()
- ( ) assertLogs()

Correct
Correct! The assertion assertIn(a, b) checks if “a” is in the collection known as “b.” This assertion is useful for determining if an object is in a result set.

10. PyUnit provides six test fixtures. Assume you include all six in a test module. Which of the following sequences shows the order in which the test runner will execute the final three test fixtures? 1 point

- (●) tearDownClass, tearDownModule, tearDown
- ( ) tearDown, tearDownClass, tearDownModule
- ( ) setUpModule, tearDownModule, tearDown
- ( ) tearDownModule, setUpModule, tearDownClass

Incorrect
Incorrect. Refer to the Test Fixtures video.

11. How does a missing test coverage report help developers improve their testing? 1 / 1 point

- ( ) Announces when no bugs remain in the code
- (●) Reveals lines of code that need test cases
- ( ) Lists functions that respond to exceptions appropriately
- ( ) Shows the lines of code that have coverage

Correct
Correct! A missing test coverage report can reveal which lines were not tested so that developers can write more test cases that execute those lines.

12. When generating fakes in Python, what is one advantage of using the Faker class to generate fake data? 1 / 1 point

- ( ) Arranges the test results in a logical order
- (●) Creates a diverse set of test data
- ( ) Uses a small amount of code
- ( ) Decreases the number of attributes to test

Correct
Correct! Faker’s providers generate a large, diverse set of data for the given attribute, such as address or barcode. The larger and more diverse your test data, the more comprehensive and reliable your testing will be. You may catch many bugs for which you might not have tested otherwise.

13. Developers should use mocking for which of the following purposes? 1 / 1 point

- ○ Include internal systems in their testing
- ○ Avoid calls of untested code
- ○ Ensure that database transactions actually work
- ◉ Isolate their tests from a remote component

Correct
Correct! Developers should mock any component to which they don't have a remote connection if that component is important for testing.

14. For which of the following purposes is it useful to patch return values with data? 1 / 1 point

- ○ Testing mocks
- ◉ Controlling data returned from a function call
- ○ Testing objects that call other objects
- ○ Testing objects with method calls

Correct
Correct! Patching a function's return value with data is useful for controlling the data returned from a function call. You can pass back any data structure or object that your program expects. Using the return_value patch, you can return anything that can be returned as a return value from a function call.

15. Developers use mock objects for which of the following purposes? 1 / 1 point

- ○ Identifying a sad path
- ○ Change a function call's parameters
- ◉ Mimicking a real object
- ○ Generating a data set

Correct
Correct! Developers use mock objects to mimic the behavior of real objects in ways that the developers can control.

16. Which of the following characteristics is common to both behavior driven development (BDD) and test driven development (TDD)? 1 / 1 point

- ○ Value for testing individual units of software
- ○ Focus on the system from the inside out
- ◉ Emphasis on a test-first approach
- ○ Goal of building the application correctly

Correct
Correct! Like TDD, BDD is a test-first approach to development; the tests drive code development.

17. Using your specifications which item can you generate automatically with most behavior driven development (BDD) tools? 1 / 1 point

- (●) Technical documents
- ( ) Testing factories
- ( ) Project schedules
- ( ) Kanban boards

Correct
Correct! With most BDD tools, you can draw from specifications to automatically generate technical documentation.

18. What does the 'When' keyword specify at the beginning of a step in Gherkin syntax? 1 / 1 point

- ( ) Feature of the system and each of its related scenarios
- ( ) Conditions needed to put the system into the required state for testing
- (●) Actions taken by the user to interact with the system under test
- ( ) Outcome expected from the actions that the user performs

Correct
Correct! Steps that begin with the When keyword describe actions the user takes to interact with the system under test.

19. Which keyword displays first in a Gherkin document? 1 / 1 point

- (●) Feature
- ( ) Scenario
- ( ) Background
- ( ) Given

Correct
Correct! In a Gherkin document, the first keyword is usually Feature, followed by a colon and a title.

20. If your step files are written in Ruby, which BDD tool will you use? 1 / 1 point

- ○ Behave
- ○ Gherkin
- ◉ Cucumber
- ○ Concordion

Correct
Correct! If your step files are written in Ruby, use Cucumber.

21. Which folder is used to store all Python files that contain step functions while using Behave? 1 / 1 point

- ○ functions
- ○ python
- ○ matches
- ◉ steps

Correct
Correct! The steps subfolder is where you must include all Python files that contain step functions that match the Gherkin statements in the feature files.

22. What are the before_feature() and after_feature() test fixtures ideally used for in Behave? 1 / 1 point

- ◉ Establishing a clean environment before and after each feature
- ○ Setting up web drivers for tools like Selenium
- ○ Declaring the context values that all steps can access
- ○ Checking specific parts of your feature set

Correct
Correct! If you use multiple features, before_feature() and after_feature () test fixtures are ideal for setting up a clean environment before and after each feature.

23. What is the purpose of the background section in a feature? 1 / 1 point

- ◉ Specify the context shared by all scenarios
- ○ Execute a scenario two or more times
- ○ Represent a business rule to implement
- ○ Check parts of the feature set selectively

Correct
Correct! Background is a test fixture that you use to specify a context shared by all scenarios in the feature. By including this context in the background section, you establish the context before every scenario.

24. Which Selenium function can you use to type input into an HTML input field? 1 / 1 point

- ○ text()
- ○ clear()
- ◉ send_keys()
- ○ submit()

Correct
Correct! To type input into an HTML input field, you can use Selenium's send_keys function.

25. In a Gherkin feature, you have the following Gherkin statement: 1 / 1 point

When I visit the "FAQs Page"

What is the complete code needed in the first line of the corresponding Python step?

- ○ when('I visit the %')
- ○ When I visit the "FAQs Page."
- ◉ @when('I visit the "FAQs Page"')
- ○ @when

Correct
Correct! In the first line of a Python step, you must include a decorator corresponding to the Gherkin statement's keyword. You must also include a text string that matches the rest of the statement.

26. When Behave loads data from a Gherkin table, it converts each row into which type of Python collection? 1 / 1 point

- ○ Set
- ○ Tuple
- ○ List
- ◉ Dictionary

Correct
Correct! When Behave loads data from a Gherkin table, it converts each row into a Python dictionary.

27. When you run the Behave tool the first time against your feature file with no steps, what will the Behave tool do for you? 1 / 1 point

- ○ Give you only the missing steps
- ○ Will show a blank screen
- ○ Will only validate your steps
- ◉ Give a set of suggested steps

Correct
Correct! If you haven't written any steps, then all of the steps are missing the first time you run Behave. Behave will seed your development efforts with suggested steps if this is the case.

28. Which steps show the NotImplementedError exception? 1 point

- ○ Failed steps
- ○ Default steps
- ◉ Missing steps
- ○ Passed steps

Incorrect
Incorrect. Review the Implementing Your First Steps video.

29. How can you pass information from one step to another? 1 / 1 point

- ◉ Store information in the context variable of one step and call the variable in another step
- ○ Store the information in data and call the string function
- ○ Store the information in import and call it from another step
- ○ Store the information in the get method and call it from another step

Correct
Correct! To pass information between steps, store it in the context variable of one step and call on that variable in another step.

30. What is the benefit of variable substitution? 1 / 1 point

- ◉ To make the steps generic and reusable
- ○ To use different names for different variables
- ○ To make the steps specific and clear
- ○ To make one sentence in your feature file match a single unique step

Correct
Correct! Variable substitution is used to make steps as generic as possible for maximum reuse.