1. What is the purpose of testing? Discuss this definition in some detail, describing each of the components of the definition.

Ans: To get the evidence of value to someone(usually the client) that the software does something useful. We try our best to make the software bug free. Every single line of code shows an impact on the product it either gives the customer some thing better or worse. Value implies potential benefit. It means that it does something beneficial or detrimental.

1. Describe the concept of test driven development. Why is it useful?

Ans: Test driven development is a software programming that ensures the code quality. In this concept the programmers will develop short pieces and then test the code and clean up any new defects in code quality if they showup. After each piece of code’s testing is done it moves to the next segment and the bugs are removed. And this process goes on. TDD helps to keep our productivity high by reducing the area we should concentrate at the moment. We can keep our focus on one part of the code while we are executing which increases the quality and productivity.

1. What are the major components of behavior driven development, and why does it provide a better connection to customer requirements?

Ans: The behavior driven development team includes both the business and technical teams sitting together with the client to discuss about the project improving communication between them to create software with business value. Because that the teams sit together, the chance of miscommunication among them would be reduced to minimal levels and the chances of creation of the software what they exactly need would be increased. Some cases, the client explains some frequent real-time problems to the teams and they try to embed some suitable solutions for them within the software itself. It often includes testing the important parts of the application with Selenium or similar.

1. What is a testing partition? Why is this so important concept, and how does it relate to completeness of testing?

Ans: Testing partition is a kind of blackbox testing. We can implement that on all levels of testing. Here we divide the input data into different equal data classes. We use this method to reduce the total number of test cases to a certain number of testable test cases still satisfying the maximum requirements. Testing partition relates to completeness of testing because we use one representative from every input class to design our test cases. Test case values are selected in such a way that highest number of attributes of equivalence can be exercised.

1. What is it meant to test a specific application layer? Which layers are commonly tested?

Ans: We will test the layers to check that the application works efficiently across all platforms and on all browsers. The primary purpose of testing is to ensure that the GUI works as it is expected to work. At each and every layer if testing is done then the chance of occurrence of a problem detection would be greatly reduced at the higher layer. GUI layer, function layer, environment layer are some of the layers that are tested.

1. Give examples of three commonly tested application layers, and describe the tools that might be used to test those layers.

Ans: GUI layer: is one of the most commonly tested layer. We can use Webdriver, Test complete and Testing whiz to record testing and scripting.

Web API layer: It is used to perform checks for the Website forms. We use requests framework for testing those web forms. Let’s say if there is a website and we make a request to it then it sends back some data in return as a response. If we throw some values, then in return we get the accordingly result as response.

Functional layer is tested using QTP(Quick Test Professional by Hewlett packard), Parasoft and Testingwhiz. They actually perform direct on that specific function to check if output satisfies the requirement of that function.

1. Describe a testing scenario we are user interface testing is a good idea, and a scenario where user interface testing is a bad idea.

Ans: The user interface testing done in the perspective of product owners like stake holders, client/ customer is a good idea because he is the person who exactly knows what the final output is that he needs. The user interface testing in the perspective of a developer is not of that much importance because he is more concerned about the functionality of the product than the user interface. He concentrates on the logic than the appearance. (Of course he works on appearance as well, but priorities matter)

1. Describe at least three purposes for using testing tools that are not directly related to seeing if a piece of software works correctly or not.

Ans: The software that is developed through testing tools possess much higher quality than the one that isn’t. When we prepared our test scripts using the automated testing tools, it stores them for future reuse. If we are using automated testing tools for testing our software, then the number of people working on one project would be reduced and we can employ them to work on some other project. The testing tools are developed by people who have done a lot of manual testing. So with all their experience they have assumed all the coming situations and prepared these tools. So these tools have a long way to go.

1. Describe the difference between white box or clear box testing, and black box testing. Is it possible to verify that software has no bugs in black box testing? Why or why not?

Ans: It is also called glass box testing as the name itself says, we know what’s inside of the program and we have some knowledge about the system and the programming done there. So we would be able to solve the issues from an internal perspective. But where as for blackbox testing, we don’t know what’s inside and what’s there in the code. It’s more like we are doing some trial and error method to test the application. Its not really possible to verify that the software has no bugs through blackbox testing because if an exception raises internally and the execution is on hold, we won’t be able to detect that there’s an exception which is to be handled.

1. Why is acceptance testing different from other kinds of testing? What do these differences mean in terms of different kinds of activities used in those forms of testing?

Ans: The regular forms of testing usually look for the errors, quality code and all the factors. But the acceptance testing tries to test the customer experience. Acceptance testing is actually performed by the client/stake holders in some cases. Its not much of the technical testing, its more like a common man’s testing. You’re checking if the user is getting his final desired experience(May be we can say it’s some sort of Beta testing). For a bad example, if you just design a music software, the acceptance testing doesn’t care about any functionality or nothing. It just checks if his desired music is playing if we click play or not, if the next song in playlist plays when we click ‘next’ or not.