**JUnit Test**

1. Which of the following elements in Junit framework is a collection of one or more test cases or test suites that are executed to perform the testing of an application ?

1. Test case
2. Test runner
3. Test
4. Test Result

2. Which of the following colors signifies the code area that is completely covered by test cases ?

1. Red
2. Yellow
3. Green
4. White

3. Which of the following annotations is used to annotate the test class while creating parameterized test ?

1. @RunWith(Parameterized.class)
2. @Suite(parameterized.class)
3. @Suite.SuiteClasses(parameterized.class)
4. @RunWith()

4. An application calculates the grade of a student based on its score , for e.g. if the student has scored more between 60 to 69 its grade should be satisfactory. Which of the following boundary conditions will help youy ensure the preceding requirement ?

1. Cardinality
2. Conformance
3. Range
4. Ordering

5. Thomas has developed a web application which is being tested by him for its response time in a usual scenario as well as in alternate scenario. Which for of testing is referred in this case?

1. Performance Testing
2. Acceptance Testing
3. System Testing
4. Integration Testing

6. Jason has developed an employee management application that uses the MySQL database. Post development, he wants to test this application using DBUnit. Therefore he creates a test class by extending the DatabaseTestCase class. Now he connects to the database using the getConnection () method. Which one of the following code snippets should he use to connect to the employee database?

1. protected iDatabaseConnection getConnection() throws Exception

{

Class.forName(“com.mysql.jdbc.Driver”);

Connection jdbcConnection = DriverManager.getConnection(“jdbc:mysql”,”root”,”root”);

return new DatabaseConnection(jdbcConnection);

}

1. protected iDatabaseConnection getConnection() throws Exception

{

Class.forName(“com.mysql.jdbc.Driver”);

Connection jdbcConnection = DriverManager.getConnection(“jdbc:mysql://localhost:3306”,”root”,”root”);

return new DatabaseConnection(jdbcConnection);

}

1. protected iDatabaseConnection getConnection() throws Exception

{

Class.forName(“com.mysql.jdbc.Driver”);

Connection jdbcConnection = DriverManager.getConnection(“jdbc:mysql://localhost:3306/employee”,”root”,”root”);

return (jdbcConnection);

}

1. protected iDatabaseConnection getConnection() throws Exception

{

Class.forName(“com.mysql.jdbc.Driver”);

Connection jdbcConnection = DriverManager.getConnection(“jdbc:mysql://localhost:3306/employee”,”root”,”root”);

return new DatabaseConnection(jdbcConnection);

}

7. Susan has created an application for inserting, retrieving, updating, and deleting employee data. Now she wants her application to be integrated continuously at every five minutes in the first quarter of every hour. Which one of the following cron syntax does she need to follow to build her project ?

1. H(0-14)/5 \* \* \* \*
2. H(0-14) 5 \* \* \*
3. H(1-15)/5 \* \* \* \*
4. H(1-15) 5 \* \* \* \*

8. Jason wants to load test his website with a set of 60 users where in each user should hit the server after 10 seconds. In addition, he wants to execute this test 15 times. Which one of the following options should Jason use to meet the preceding requirements of load testing ?

1. Number of Threads 60, Ramp up period 600, Loop Count 15
2. Number of Threads 600, Ramp up period 0, Loop Count 10
3. Number of Threads 60, Ramp up period 100, Loop Count 15
4. Number of Threads 60, Ramp up period 10, Loop Count 15

9. David has developed a website for a travel agency that contains various pages, such as Home Page, Login Page, Registration Page, and Admin page. The Login Page allows users to login as an administrator or as a customer by specifying the valid credentials. Post development, David has to test LoginPage and verify whether the components , such as textboxes named username and password, radio buttons for administrator and customer options, and buttons for submit and reset functionalities , are available on the page. For this, he has written thye following code snippet:

@Test

Public void testPageComponents{

beginAt(“/HomePage.jsp”);

assertTitleEquals(“Login Page”);

assertFormElementPresent(“textboxes”);

assertRadioOptionPresent(“administrator”,”customer”);

assertButtonPresentWithText(“Submit”);

}

However, the preceding code does not meet the specified testing requirements . Which one of the following code snippets should David use to resolve this problem ?

1. @Test

Public void testPageComponents{

beginAt(“/HomePage.jsp”);

assertTitleEquals(“Login Page”);

assertFormElementPresent(“username”);

assertFormElementPresent(“password”);

assertRadioOptionPresent(“rBtn1”,”administrator”);

assertRadioOptionPresent(“rBtn1”,”customer”);

assertButtonPresentWithText(“Submit”);

}

1. @Test

Public void testPageComponents{

beginAt(“/HomePage.jsp”);

assertTitleEquals(“Login Page”);

assertFormElementPresent(“username”);

assertFormElementPresent(“password”);

assertRadioOptionPresent(“rBtn1”,”administrator”);

assertRadioOptionPresent(“rBtn2”,”customer”);

assertButtonPresentWithText(“Submit”);

}

1. @Test

Public void testPageComponents{

beginAt(“/LoginPage.jsp”);

assertTitleEquals(“Login Page”);

assertFormElementPresent(“username”);

assertFormElementPresent(“password”);

assertRadioOptionPresent(“rBtn1”,”administrator”);

assertRadioOptionPresent(“rBtn1”,”customer”);

assertButtonPresentWithText(“Submit”);

}

1. @Test

Public void testPageComponents{

beginAt(“/HomePage.jsp”);

assertTitleEquals(“Login Page”);

assertFormElementPresent(“username”);

assertFormElementPresent(“password”);

assertRadioOptionPresent(“rBtn1”,”user”);

assertRadioOptionPresent(“rBtn1”,”customer”);

assertButtonPresentWithText(“Submit”);

}

10. Tory has developed a student management application that uses the MySQL database. In this application , he has created the student database having the StudentDetails table. This table contains the StudentName, StudentMarks, and StudentId column. Post development, he wants to test this application using DbUnit. For this , he creates a test class by extending the DatabaseTestCase class. Further , he connects to the database by overriding the getConnection() method. Next, he need to create the data file , which will be used as input.For this he has created the following XML file.:

<?xml version=”1.0” encoding=”UTF-8”>

<dataset>

<!—student table-->

<studentdetails StudentID=001 StudentName=John Hodge StudentMarks=85/>

<studentdetails StudentID=002 StudentName=Sarah Williams StudentMarks=95/>

</dataset>

However when the test code is executed, it generates errors. Which one of the following options should he use to rectify the error in the XML file?

1. <?xml version=”1.0” encoding=”UTF-8”>

<dataset>

<!—student table-->

<studentdetails StudentID=”001” StudentName=”John Hodge” StudentMarks=”85”/>

<studentdetails StudentID=”002” StudentName=”Sarah Williams” StudentMarks=”95”/>

</dataset>

b. <?xml version=”1.0” encoding=”UTF-8”>

<dataset>

<StudentDetails StudentID=”001” StudentName=”John Hodge” StudentMarks=”85”/>

<StudentDetails StudentID=”002” StudentName=”Sarah Williams” StudentMarks=”95”/>

</dataset>

c. <?xml version=”1.0” encoding=”UTF-8”>

<!—student table-->

<StudentDetails StudentID=001 StudentName=John Hodge StudentMarks=85/>

<StudentDetails StudentID=002 StudentName=Sarah Williams StudentMarks=95/>

d. <?xml version=”1.0” encoding=”UTF-8”>

<StudentDetails>

<!—student table-->

<studentdetails StudentID=001 StudentName=John Hodge StudentMarks=85/>

<studentdetails StudentID=002 StudentName=Sarah Williams StudentMarks=95/>

</StudentDetails>

11. Abraham has created an EJB module that converts Dollars to Yen , as shown in the following code snippet:

Private double yenRate = 116.45;

Public double DollarToYen(double dollars){

Double result = dollars\*yenRate;

Return result;

}

The DollarToYen() method contains the code to convert the specified units of Dollar into Yen. Now, he wants to create a test case for this EJB module by performing the in-container testing. Which one of the following options will help him fulfill the desired requirements ?

1. ...............

...............

@BeforeClass

Public static void setUpClass()

{

Container.close();

}

@AfterClass

Public static void tearDownClass()

{

Container = javax.ejb.EJBContainer.createEJBContainer();

}

b. ...............

...............

@BeforeClass

Public static void setUpClass()

{

Container = javax.ejb.EJBContainer.createEJBContainer();

}

@AfterClass

Public static void tearDownClass()

{

Container.close();

}

c. ...............

...............

@Before

Public static void setUpClass()

{

Container = javax.ejb.EJBContainer.createEJBContainer();

}

@After

Public static void tearDownClass()

{

Container.close();

}

d. ...............

...............

@Before

Public static void setUpClass()

{

Container.close();

}

@After

Public static void tearDownClass()

{

Container = javax.ejb.EJBContainer.createEJBContainer();

}

12. Jennifer has developed a Student class that defines two methods writeStudentData() and readStudentData(). These methods invoke the writeData() and readData() methods of the WriteStudent class. Jennifer needs to test the Student class. However, the WriteStudent class does not implement the logic for reading and writing data. Therefore, to represent this class , she can create stubs for the writeData() and readData() methods. To ensure that the test passes successfully , the String values , Success and Student Data , needs to be returned from the writeData() and readData() methods respectively. Which of the following options will she use to create stubs for the writeData() and readData() methods?

1. Public class writeStudentStub

{

Public String writeData(Student s)

{

Return “Success”;

}

Public String readData()

{

Return “Student Data”;

}

}

1. Public class writeStudentStub

{

Public String writeData()

{

return “Success”;

}

Public String readData()

{

return “Student Data”;

}

}

c. Public class writeStudentStub

{

Public String writeData(Student s)

{

return “Student Data”;

}

Public String readData()

{

return “Success”;

}

}

d. Public class writeStudentStub

{

Public void writeData(Student s)

{

return “Student Data”;

}

Public String readData()

{

return “Success”;

}

}

13. Steve has created a Web Application to calculate the simple interest , using the following code snippet:

Public double SimpleInterest(double principal, double rate, double time)

{

If(principal >=0 && rate >=0 && time >=0)

{

Double si;

Si = (principal\*rate\*time)/100;

Return si;

}

Return 0;

}

He has written the following code snippet to test the preceding code.

@Test

Public void testSimpleInterest()

{

System.out.println(“SimpleInterest”);

Double principal = 12000;

Double rate =5;

Double time =3;

Interest instance = new Interest();

Double expResult = 1800;

Double result = instance.SimpleInterest(principal,rate,time);

assertEquals(expResult, result,0.5);

}

However he is not able to achieve 100% code coverage. Write the code to achieve the desired result.

1. @Test

Public void testSimpleInterest()

{

System.out.println(“SimpleInterest”);

Double principal = 12000;

Double rate =5;

Double time =3;

Interest instance = new Interest();

Double expResult = 1800;

Double result = instance.SimpleInterest(principal,rate,time);

assertEquals(expResult, result,0.5);

}

1. @Test

Public void testSimpleInterest()

{

System.out.println(“SimpleInterest”);

Double principal = 12000;

Double rate =5;

Double time =-3;

Interest instance = new Interest();

Double expResult = 0;

Double result = instance.SimpleInterest(principal,rate,time);

assertEquals(expResult, result,0.0);

}

1. @Test

Public void testSimpleInterest()

{

System.out.println(“SimpleInterest”);

Double principal = 10000;

Double rate =3;

Double time =3;

Interest instance = new Interest();

Double expResult = 900;

Double result = instance.SimpleInterest(principal,rate,time);

assertEquals(expResult, result,0.0);

}

1. @Test

Public void testSimpleInterest()

{

System.out.println(“SimpleInterest”);

Double principal = 18000;

Double rate =6;

Double time =4;

Interest instance = new Interest();

Double expResult = 4320;

Double result = instance.SimpleInterest(principal,rate,time);

assertEquals(expResult, result,0.0);

}

14. Sage has developed an EJB module for the conversion of distances into different units , such as kilometers, meters, and miles. Now she needs to test this module. For this, she has written the following code snippet in the pom.xml file to add the server property.

<dependencies>

<endorsed.dir> ${project.build.directory}/endorsed</endorsed.dir>

<project.build.sourceEncoding>UTF-8</ project.build.sourceEncoding>

<glassfish.embedded-static-shell.jar>C:/Program Files/glassfish-4.1/glassfish/lib/embedded/glassfish-embedded-static-shell.jar</ glassfish.embedded-static-shell.jar>

</dependencies>

However the preceding code snippet encounters certain errors. Which one of the following code snippets should Sage use to resolve the errors ?

1. <dependencies>

<endorsed.dir> ${project.build.directory}/endorsed</endorsed.dir>

<project.build.sourceEncoding>UTF-8</ project.build.sourceEncoding>

<glassfish.embedded-static-shell.jar>C:/Program Files/glassfish-4.1/glassfish/lib/embedded/glassfish-embedded-static-shell.jar</ glassfish.embedded-static-shell.jar>

</dependencies>

1. <properties>

<endorsed.dir> ${project.build.directory}/endorsed</endorsed.dir>

<project.build.sourceEncoding>UTF-8</ project.build.sourceEncoding>

<glassfish.embedded-static-shell.jar>C:/Program Files/glassfish-4.1/glassfish/lib/embedded/glassfish-embedded-static-shell.jar</ glassfish.embedded-static-shell.jar>

</properties>

1. <dependencies>

<dependency>

<endorsed.dir> ${project.build.directory}/endorsed</endorsed.dir>

<project.build.sourceEncoding>UTF-8</ project.build.sourceEncoding>

<glassfish.embedded-static-shell.jar>C:/Program Files/glassfish-4.1/glassfish/lib/embedded/glassfish-embedded-static-shell.jar</ glassfish.embedded-static-shell.jar>

</dependency>

</dependencies>

d. <properties>

<property>

<endorsed.dir> ${project.build.directory}/endorsed</endorsed.dir>

<project.build.sourceEncoding>UTF-8</ project.build.sourceEncoding>

<glassfish.embedded-static-shell.jar>C:/Program Files/glassfish-4.1/glassfish/lib/embedded/glassfish-embedded-static-shell.jar</ glassfish.embedded-static-shell.jar>

</property>

</properties>

15. Jasmine has created an application to check whether an entered character is a vowel or a consonant, using the following code snippet:

Public class VowelConsonant{

Public String check(String character)

{

String v = “vowel”;

String c=”consonant”;

If(character==”a” || character==”a” || character==”a”|| character==”a” || character==”a”)

{

Return v;

}

Else

{

Return c;

}

}

Public static void main(String[] args)

{

// To Do Code Application Logic here.

}

}

Now she has been assigned the task to test this application by creating three test cases, one that checks for a vowel, second that checks for a consonant, and third one falls and displays an appropriate human readable error message. Which one of the following options will help him fulfill the requirements?

1. @Test

Public void testCheck2()

{

System.out.println(“Testing Consonant”);

VowelConsonant vc = new VowelConsonant();

String result = vc.check(“s”);

assertThat(result.is(equalTo(“consonant”)));

}

@Test

Public void testCheck3()

{

System.out.println(“Testing Consonant”);

VowelConsonant vc = new VowelConsonant();

String result = vc.check(“e”);

assertThat(result.is(equalTo(“vowel”)));

}

16. Consider the following code snippet:

@Before

Public void setup(){

employeeService = Mockito.verify(EmployeeService);

employeeManager = new EmployeeManager();

employeeManager.setEmployeeService(employeeService);

}

In the preceding code snippet, the setup() method is defined, which initializes the components of the application required for performing unit test. Within this method, the employeeService mock object is created using the Mockito framework and passing the EmployeeService class as its parameter. Further, an instance of the EmployeeManager class is created and its employee service is set as employeeService. However, the preceding code generates an error.Analyze the code and rectify the error.

1. @Before

Public void setup()

{

employeeService = Mockito.mock(EmployeeService.class);

employeeManager = new EmployeeManager();

employeeManager.setEmployeeService(employeeService);

}

1. @Before

Public void setup()

{

employeeService = Mockito.when(EmployeeService.class);

employeeManager = new EmployeeManager();

employeeManager.setEmployeeService(employeeService);

}

17. Lychee has created an application for maintaining the details of the author of a publication. In this application, she has created a servlet that validates the login credentials and redirects the authors to the appropriate page. Now she needs to test the functionality of the servlet by using the Mockito framework. For this, she have added the dependencies in the pom.xml file, using the following code snippet:

<dependency>

<groupId>org.easymock</groupId>

<artifactId>easymock</artifactId>

<version>3.2</version>

<scope>test</scope>

<groupId>org.glassfish.extras</groupId>

<artifactId>glassfish-embedded-static-shell</artifactId>

<version>4.1</version >

<scope>test</scope>

</dependency>

Then she created the test class and executed test cases. On executing the test , she encountered certain errors. Which one of the following code snippets will she use to rectify the errors?

1. <dependency>

<groupId>org.easymock</groupId>

<artifactId>easymock</artifactId>

<version>3.2</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.glassfish.extras</groupId>

<artifactId>glassfish-embedded-static-shell</artifactId>

<version>4.1</version >

<scope>test</scope>

</dependency>

1. <dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito.all</artifactId>

<version>1.9.5</version>

<groupId>org.glassfish.extras</groupId>

<artifactId>glassfish-embedded-static-shell</artifactId>

<version>4.1</version >

<scope>test</scope>

</dependency>

1. <dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito.all</artifactId>

<version>1.9.5</version>

</dependency>

<dependency>

<groupId>org.glassfish.extras</groupId>

<artifactId>glassfish-embedded-static-shell</artifactId>

<version>4.1</version >

<scope>test</scope>

</dependency>

1. <dependencies>

<groupId>org. easymock</groupId>

<artifactId>easymock</artifactId>

<version>3.2</version>

<scope>test</scope>

</dependencies>

<dependencies>

<groupId>org.glassfish.extras</groupId>

<artifactId>glassfish-embedded-static-shell</artifactId>

<version>4.1</version >

<scope>test</scope>

</dependencies>

18. Ben wants to create a mock object using the Mockito framework. To implement the Mockito framework, he needs to add the Mockito dependency. Which one of the following code snippets will he use to add the Mockito dependency in the pom.xml file.

1. <dependencies>

<dependency>

<groupId>org.mockito</groupId>

<artifactId>Mockito-all</artifactId>

<version>1.9.5</version>

</dependency>

</dependencies>

1. <dependencies>

<groupId>org.mockito</groupId>

<artifactId>Mockito-all</artifactId>

<version>1.9.5</version>

</dependencies>

1. <dependency>

<groupId>org.mockito</groupId>

<artifactId>Mockito-all</artifactId>

<version>1.9.5</version>

</dependency>

1. <dependencies>

<dependency>

<groupId>org.easymock</groupId>

<artifactId>easymock</artifactId>

<version>1.9.5</version>

</dependency>

</dependencies>

19. Sebastian has developed an application to store the employee details, such as name, address, department, employee id, and marital status. In this application, you have created a bean class to get and set the values for these properties. In addition, you have created a DAO class that inserts, retireves, and deletes the records of employee. Therefore, to invoke the insert () method of the DAO class, you need to ensure that the bean class is instantiated first. Which one of the following boundary conditions will help you ensure the preceding requirement?

1. Cardinality
2. Conformance
3. Reference
4. Ordering

20. The test cases for a web application are being executed under the given test environment to verify the test results. Which one of the following phases of the STLC life cycle is being referred in this case?

1. Test Planning
2. Test Case development
3. Test environment Setup
4. Test Execution.

21. You have to develop an application for a college to manage the data of its faculty. Post development , you need to test this application with numerous input values and verify the expected values with the actual result. Therefore, you plan to create separate data files for input and expected values instead of creating multiple test cases for each set of values. Which advantage of data-driven testing will you be able to achieve by p[erforming the preceding task?

1. Manageable Records
2. Updating data files with the change in the test plan.
3. Reduced redundancy in test cases.
4. Independent test scripting.

22. Which of the following options in a summary report represents the amount of requests per second that the server can handle ?

1. KB/Sec
2. #Samples.
3. Throughput
4. Average

23. Which one of the following options is an advantage of CI testing?

1. Automated deployment
2. User-oriented development
3. Manageable data records
4. Platform- dependent development

24. Which one of the following colors signifies the code area that is left uncovered by the test cases ?

1. Green
2. Yellow
3. White
4. Red

25. Which one of the following options do you need to select to include a CSV Data Set Config element into your plan in JMeter ?

1. Controller
2. Configuration Element
3. Assertion
4. Listener

26. Which one of the following formulas is used to calculate the code coverage ?

1. (No. of statements exercised/Total number of statements)\*100
2. (Total Number of Statements/No. of Statements exercised)\*100
3. (No. of Conditions exercised/Total Number of Conditions)Total number of statements
4. (No. of loops exercised/Total number of loops)\* Total number of statements

27. Which mock object requires all the expected methods in the module being tested to be called arguments in any order and calls to unexpected methods do not cause the test to fail ?

1. Nice
2. Normal
3. Strict
4. Real

28. Which one of the following Hamcrest matchers is used to add a custom descriptor to the matcher?

A. is

B. allOf

c. describedAs

D. containsString

29. Which one of the following annotations causes the method it has been attached to run after all the methods in the class have been executed?

A. @BeforeClass

B. @After

c. @Before

D. @AfterClass

30. Which one of the following options will you use to set the path where conversation with the web application begins?

A. setBaseUrl(String url)

B. clickLinkWithExactText(String url)

c. beginAt(String aRelativeUrl)

D. submit(String buttonName)

31. Which one of the following options will you use to verify if a button with the SignUp text is present in the current window?

A. assertButtonPresent("SignUp");

B. assertButtonPresentWithText("SignUp");

C. assertResetButtonPresent("SignUp");

D. assertSubmitButtonPresent("SignUp");

32. Which one of the following Hamcrest matchers matches if the given object is the exact same instance as

another?

A.instanceOf

B.samelnstance

C.hasltem

D.closeTo

33. Which one of the following assert statements checks that the specified boolean condition is true?

A. assertTrue([message], boolean condition)

B. asserttrue([message])

c. assertNuli([message], object)

D. assertSame([message], expected, actual)

34. You have to develop an application for a college to manage the data of its students. The development of this

application will begin soon and until then you have been assigned the task to prepare the input and expected

data in separate data files. which advantage of data—driven testing ensures that you are able to perform the

assigned task?

A. Knowledge of scripting language

B. Reduced redundancy in test cases

C. Structuring the data file

D. Independent test scripting

35. You have developed an application to store the employee details, such as name, address, department,

employee id, and marital status. In this application, you have set the values for these properties within the

constructor. Now, you want to confirm if these values are being passed to the constructor in the same

sequence as received in the constructor because all of these properties have the String data type. Which one

of the following boundary conditions will help you ensure the preceding requirement?

A. Cardinality

B. Conformance

C. Range

D. Ordering

36. Lisa has developed an employee management application that uses the MySQL database. In this application,

she has created multiple classes and multiple tables in the database. Few of these tables are dependent on

each other. For example, when a record is inserted into EmployeeDetails table, his/her login information

should automatically be inserted into the Login table. Post development, Lisa wants to test her application

and verify if tables are properly linked with each other and maintaining the complete and accurate data. Which

one of the following benefits of database testing ensures that the preceding requirement is met?

A.Manageable records

B.Data mapping

C.Data integrity

D.Independent test scripting

37. Which one of the following code coverage metrics independently measures the coverage of all the conditions

that can affect or determine the outcome?

A.Function Coverage

B.Statement Coverage

C.Condition Coverage

D.Loop Coverage

38. A Web application is being tested to identify bugs in the code that has been modified. Which level of testing is

being referred in this case?

A.Regression testing

B.Performance testing

C.Integration testing

D.System testing

39. Dan has developed a student management application contain Web pages, classes, and database tables. Post

development, John wants to perform

data—driven testing forhis application. For this, he has created a data file and added the following code snippet

into it:

<dataset>

<!——student Table ——><studentinfo StudentID="O01" StudentName="John Hodge" StudentMarks="85"/>

<studentinfo StudentID="OO2" StudentName="Sarah WiIliams" StudentMarks="95"/>

<studentinfo StudentID="OO3" StudentName="Roger HaiIs" StudentMarks="9O"/>

</dataset>

Which one of the following formats has John used to create the preceding data file?

A. .xml

B. .xlsx

C. .csv

D. .txt

40. Lisa has developed a website for a travel agency that contains various pages, such as HomePage, LoginPage,

RegistrationPage, and AdminPage. In this application, HomePage allows users to select the login or registration

option. An existing user is redirected to LoginPage. On the other hand, a new user is redirected to

RegistrationPage. Now, you need to verify that on selecting the New User option, a user should navigate to

RegistrationPage. Which one of the following code snippets will you use to accomplish the preceding task?

Ans.

@Test

public void testNewUser() {

beginAt("/HomePage.jsp");

clickRadioOption("IoginType", "newuser");

submit();

assertTitleEquais("Registration Page");

}

41. develop a test program for the following scenario :

James has created a Web applncatlon to calculate the compound nnterest, using the following code snlppet

................

................

public double CompoundInterest(double principal, double rate, double time, double n){

if(principal >=0 && rate>=O && time>=0 && n>0){

double part1,part2,ci;

part1 = 1+(rate/n);

part2 = Math.pow(part1,time);

ci=principal\*part2;

return ci;

}

return 0;

}

......

.......

He has written the following code snnppet to test the precednng code

@Test

publlc vold testCompoundInterest() {

System.out.println( "Compoundlnterest" );

double principal = 1000;

double ra e = 0.05;

double time = 2;

Interest mstance = new Interest();

double expResult = 1050.625;

double result = mstance CompoundInterest(principal, rate, tume, n);

assertEquals(expResult, result, 0.5);

........

........

However, he is not able to achleve 100% code coverage. write a test case to help him

to achieve 100% code coverage?

42. Wilber wants to create a mock object using the EasyMock framework. To implement the EasyMock framework,

he needs to add the EasyMock dependency. Which one of the following code snippets will he use to add the

EasyMock dependency in the pom.xml file?

Ans.

<dependencies>

<dependency>

<groupId>org.easymock</groupId>

<artifactId>easymock</artifactld>

<version>3.2</version>

<scope>test</scope>

</dependency>

</dependencies>

43. You have developed the Techwrite application wherein users need to login to access its features. If the correct

login id and password, 'user' and 'password123', is provided, the welcome page will be displayed. Otherwise, the

error page will be displayed. Now, you want to test your website against four sets of data for login id and

password, including the success and failure conditions. For this, you plan to create a CSV file and add the

required input data to it. Which one of the following options will you use to create the loginData.csv file?

A. user, password@ 123

B. login id:21062,password:password@123

C. login id:123,password:pass

login id:user,password:password123

login id:usenpassword:password@123

D. 21062,password@123

123,pass

user, password123

user, password@123

44. Wilber has developed an EJB module for the conversion of distances into different units, such as kilometers,

meters, and miles. Now, he needs to test this module. For this, Wilber has written the following code snippet in

the pom.xmI file to add the server property:

<properties>

<endorsed.dir>${project.build.directory}/endorsed</endorsed.dir>

<project.build.sourceEncoding>UTF-8</project.bui|d.sourceEncoding>

<glassfish.embedded-static-shell.jar>C:/Program ﬁles/glassfish-4.1/glassfish/lib/embedded/glassfish-embedded-static-shell.jar</glassfish.embedded-static-shell.jar>

</properties>

Now, he needs to add the server dependency. Which one of the following code snippets should Wilber use to

accomplish the preceding requirement?

Ans.

<dependency>

<groupId>org.glassfish.extras</groupId>

<artifactId>glassfish-embedded-static-shell</artifactId>

<version>4.1</version>

<scope>system</scope>

<systemPath>${glassfish.embedded-static-shell.jar}</systemPath>

</dependency>

45. John has developed an application that calculates the volume of a sphere, a cube, and a cone. He has created

separate Java classes for each of the operations, such as Cone.java, Cube.java, and Sphere.java. In addition,

such as ConeTest.java, CubeTest.java, and

he has developed separate test classes to test each of the classes,

SphereTest.java. These test classes are defined within the volume package. Now, he wants to create a test

class and combine all the three test classes to test the overall functionality of the application. Which one of the

following options will fulfill the requirements?

Ans.

.......

.......

@RunWith(Suit.Class)

@Suite.SuiteClasses({volume.SphereTest.class, volume.ConeTest.class, volume.CubeTest.class})

public class TestSuite{

..............

..............

}

46. Maria has created an application for maintaining the details of the author of a publication. In this

application, she has created a servlet that validates the login credentials and redirects the authors to the

appropriate page. Now, you need to test the functionality of the servlet by using the Mockito framework. For

this, you have added the dependencies in the pom.xml file, using the following code snippet:

<dependency>

<groupId>org.easymock</groupId>

<artifactId>easymock</artifactId>

<version>3.2</version>

<scope>test</scope>

<groupId>org.glassfish.extras</groupId>

<artifactId>glassfish-embedded-static-shell</artifactId>

<version>4.1</version>

<scope>test</scope>

</dependency>

Then, you created the test class and executed your test cases. On executing your test, you encountered certain

errors. Rectify the error from the above code.

47. Which one of the following elements of a test plan in JMeter is used to configure and send the request to the

server?

a.Controller

b.Configuration element

c.Thread Group

d.Listener

48. Which one of the following assert statements checks whether the expected and actual values are same for the float and double type of values?

a.assertEquals([message],expected,actual)

b.assertSame([message],expected,actual)

c.assertEquals([message],expected,actual,tolerance)

d.assertNotSame([message],expected,actual)

49. Which one of the following options do you need to select to include a CSV Data Set Config element into your test plan in JMeter?

a. controller

b. configuration element

c. assertion

d. listener

50. The individual units of a Web application are being integrated and tested to verify the overall functionality.

Which level of testing is being referred in this case?

a. System testing

b. Performance testing

c. Integration testing

d. Acceptance testing

51. You have developed an application to print the prime numbers till n, where n is an input from the user. Now,you want to verify if the loop, which is displaying the prime numbers, is executed n times. Which one of the

following boundary conditions will help you ensure the preceding requirement?

a. Cardinality

b. Conformance

c. Range

d. Ordering

52. You have developed an application to store the employee details, such as name, address, contact number,department, employee id, and date of joining. In this application, you have created the getter and setter methods for these properties. Now, you want to confirm if the format of date returned by the getter method of the date of joining property matches the format in the database because date can be stored in multiple formats. Which one of the following boundary conditions will help you ensure that the expected date is in compliance with the expected date's format?

a. Cardinality

b. Conformance

c. Range

d. Ordering

53. You have developed an application to store the employee details, such as name, address, department,employee id, and marital status. In this application, you have set the values for these properties within the constructor. Now, you want to confirm if these values are being passed to the constructor in the same

sequence as received in the constructor because all of these properties have the String data type. Which one of the following boundary conditions will help you ensure the preceding requirement?

a. Cardinality

b. Conformance

c. Range

d. Ordering

54. You have to develop an application for a college to manage the data of its faculty. Post development, you need to test this application with numerous input values and verify the expected values with the actual result.Therefore, you plan to create separate data files for input and expected values instead of creating multiple test cases for each set of values. Which advantage of

data-driven testing will you be able to achieve by performing the preceding task?

a. Manageable records

b. Updating data files with the change in the test plan

c. Reduced redundancy in test cases

d. Independent test scripting

55. Lynda has created an email application thatcontains various Web pages, such as login, inbox, compose, draft,sent items, junk, logout, adminSuccess, and failure. In addition, she has added the username and password

having the administrator rights. For this, she has created the following servlet class:

public class LoginServlet extends HttpServlet

Override protected void processRequest(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException

{

String name = request.getParameter("username");

String password = request.getParameter("password");

if (name.equals("admin") && password.equals("response.sendRedirect("adminSuccess.jsp");

else

response.sendRedirect("failure.jsp");

Now, she needs to test this servlet for the successful login scenario. Which one of the following code snippets should she use to test the preceding servlet?

find the correct ans from book

a. @Test

public void testProcessRequest() throws IOException, ServletException

{

when(mockHttpServletRequest.getParameter("username")).thenReturn("admin");

when(mockHttpServletRequest.getParameter("password")).thenReturn("

LoginServlet loginServlet= new LoginServlet();

loginServlet.processRequest(mockHttpServletRequest, mockHttpServletResponse);

verify(mockHttpServletResponse).sendRedirect("adminSuccess.jsp”);

}

56. Harry wants to test the functionality of the Web application using JWebUnit. For this, he needs to add the JWebUnit dependency into the Maven project. Which one of the following options will he use to fulfill the requirements?

check the correct ans from book

a.<dependencies>

<dependency>

<groupId>net.sourceforge.jwebunit</groupId>

<artifactId>jwebunit-htmlunit-plugin</artifactId>

<version>3.2</version>

<scope>test</scope>

</dependency>

</dependencies>

57. Taylor has developed an application to calculate the speed of an object, using the following code snippet:

public double speed(double distance, double time)

if(distance<0 ll time<0)

throw new IllegalArgumentException();

double 5;

s=distance/time;

return 5;

He has written the following code snippet to test the preceding code:

Test

public void testSpeed()

System.out.println("Testing speed() method");

double distance = 300;

double time = 5;

Speed instance = new Speed();

double epresult = 60;

double result = instance.speed(distance, time);

assertEquals(epresult, result,0.0);

However, he is not able to achieve 100% code coverage. Which one of the following test cases will help him achieve 100% code coverage?

a. @Test

public void testSpeed2()

System.out.println("Testing speed() method");

double distance = 545.5;

double time = 7.2;

Speed instance = new Speed();

double expResult = 75.76;

double result = instance.speed(distance, time);

assertEquals(epResult, result,0.5);

b.@Test(expected=IllegalArgumentException.class)

public void testSpeed2()

System.out.println("Testing speed() method");

double distance = 545.5;

double time = 7.2;

Speed instance = new Speed();

double expResult = 75.76;

double result = instance.speed(distance, time);

assertEquals(epResult, result,0.5);

c.@Test

public void testSpeed2()

System.out.println("Testing speed() method");

double distance = 200;

double time = 5;

Speed instance = new Speed();

double expResult = 40;

double result = instance.speed(distance, time);

assertEquals(expResult, result,0.0);

d. @Test

public void testSpeed2()

System.out.println("Testing speed() method");

double distance = 769;

double time = 7.2;

Speed instance = new Speed();

double epResult = 106.80;

double result = instance.speed(distance, time);

assertEquals(epresult, result,0.5);

58.. Williams has developed an application for a shopping website hosting various types of products, such as electronics, cosmetics, clothing, and jewelry. However, only authorized users can access and purchase these products. For this, you have created a filter in the application. The filterredirectsusers from the items.jsp page to the login.jsp page, whenthey access the items.jsp page directly. Now, you need to test this filter using the Mockito framework. Which one of the following code snippets will you use to accomplish the preceding task?

a. public void testDoFilterMethod() throws Exception

{

HttpServletRequest mockHttpServletRequest = mock(HttpServletRequest.class);

HttpServletResponse mockHttpServletResponse = mock(HttpServletResponse.class);

FilterChain mockFilterChain = mock(FilterChain.class);

when(mockHttpServletRequest.getRequestURI()).thenReturn("/ShopTop/login.jsp");

MyFilter myFilter = new MyFilter();

myFiIter.doFilter(mockHttpServletRequest, mockHttpServletResponse, mockFilterChain);

verify(mockHttpServletRequest, atLeast(1)).getRequestURI();

verify(mockHttpServletResponse).sendRedirect("/ShopTop/Login.jsp");

}

b. @Test

public void testDoFilterMethodO throws Exception

{

FilterChain mockPilterChain = mock(FilterChain.class):

when(mockHttpServletRequest. getRequestURI()).thenReturn(" ShopTop/items.jsp");

MyFilter myFilter = new MyFilter():

myFilter.doFilter(mockFilterChain):

verify(mockHttpServletRequest. atLeast(1)).getRequestURI():

verify(mockHttpServletResponse).sendRedirect("/ShopTop/login.jsp"):

c. @Test

public void testDoFilterMethodO throws Exception

HttpServletRequest mockHttpServletRequest = mock(HTTPServletRequest.class);

HttpServletResponse mockHttpServletResponse

mock(HttpServletResponse.class):

FilterChain mockFilterChain = mock(FilterChain.class):

when(mockHttpServletRequest. getRequestURI()).thenReturn( ", "ShopTop/login.jsp");

MyFilter myFilter = new MyFilter():

myFilter.doFilter(mockHttpServletRequest. mockHttpServletResponse.

mockFilterChain);

verify(mockHttpServletRequest. atLeast(1)).getRequestURI();

verify(mockHttpServletResponse).sendRedirect("/ShopTop/login.jsp");

}

d. @Test

public void testDoFilterMethodO throws Exception

{

HttpServletRequest mockHttpServletRequest = mock(HttpServletRequest.class);

HttpServletResponse=mockHttpServletResponse=

mock(HttpServletResponse.class):

FilterChain mockFilterChain = mock(FilterChain.class):

when(mockHttpServletRequest. getRequestURI()).thenReturn( "ShopTop/items.jsp");

MyFilter myFilter = new MyFilterO:

myFilter.doFilter(mockHttpServletRequest. mockHttpServletResponse.

mockFilterChain);

verify(mockHttpServletRequest.atL east( 1)).getRequestURI():

verify(mockHttpServletResponse).sendRedirect( "/ShopTop/login.jsp");

59. Sam has developed an EJB module for the conversion of distances into different units, such as kilometers,meters, and miles. Now, he needs to test this module. For this, Sam has written the following code snippet in the pom.xml file to add the server property:

<properties>

<endorsed.dir>${project.build.directory}/endorsed</endorsed.dir>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<glassfish.embedded-static-shell.jar>C:/Program Files/glassfish-4.1/glassfish/lib/embedded/glassfish-

embedded-static-shell.jar</glassfish.embedded-static-shell.jar>

</properties>

Now, he needs to add the server dependency. Which one of the following code snippets should Sam use to accomplish the preceding requirement?

</dependency>

a. <dependency>

<groupId>org.glassfish.extras</groupId>

<artifactId>glassfish-embedded-static-shell</artifactId>

<version>4.1</version>

<scope>system<scope>

<systemPath>${glassfish.embedded-static-shell.jar}</systemPath>

</dependency>

60. Samantha has created an application for maintaining the details of the author of a publication. In this application, she has created a servlet that validates the login credentials and redirects the authors to the appropriate page. Now, you need to test the functionality of the servlet by using the Mockito framework. For this, you have added the dependencies in the pom.xml file, using the following code snippet:

<dependency>

<groupId>org.easymock</groupId>

<artifactId>easymock</artifactId>

<version>3.2</version>

scope>test</scope>

<groupId>org.glassfish.extras</groupId>

<artifactId>glassfish-embedded-static-shell</artifactId>

<version>4. 1</version>

scope>test</scope>

</dependency>

Then, you created the test class and executed your test cases. On executing your test, you encountered certain errors. Which one of the following code snippets will you use to rectify the error?

please search for ans in book

61. Sandra has developed a book

management application that uses

the LibraryDetails table created in the

MySQL database. Post development.

she wants to test this application

using DbUnit. Therefore. she creates

a test class by extending the

DatabaseTestCase class. Now. she

needs to write a test case for testing

the application. For this. she has

written the following code snippet:

public void testCheckDataLoaded()

throws Exception{

IDataSet databaseDataSet =

getConnection().createDataSet();

IDataSet expectedDataSet = new

FlatXmlDataSetBuilder().build(new

File("expectedData.xml"));

Assert.assertEquals(expectedDataSet.

databaseDataSet);

However,the preceding test case

generates an error on execution.

Which one of the following code

snippets should she use to rectify the

error?

a. public void testCheckDataLoaded throws Exception{

IDataSet databaseDataSet = gctConnection().createDataset():

ITable actualTable = databaseDataSet.getTable("LibraryDetails");

IDataSet expectedDataSet = new FlatXmlDataSetBuilder().build(new

File("expectedData.xml"));

ITable expectedTable = expectedDataSet.getTable("LibraryDetails");

Assert.assetEquals(expectedDataSet. actualTable);

b. public void testCheckDataLoaded() throws Exception{

IDataSet databaseDataSet = getConnection().createDataSet();

ITable actualTable = databaseDataSet.getTable("LibraryDetails");

IDataSet expectedDataSet = new FlatXmlDataSetBuilderObuildnew

File("expectedData.xml"));

ITable expectedTable = expectcdDataSet.getTable("LibraryDetails");

Assert.assertEquals(expectedDataSet. databaseDataSet);

c. public void testCheckDataLoadedO throws Exception{

IDataSet databaseDataSet = getConnection().createDataSet();

ITable actualTable = databaseDataSet.getTable("LibraryDetails");

IDataSet expectedDataSet = new FlatXmlDataSetBuilder().build(new

File("expectedData.xml"));

ITable expectedTable = expectedDataSet.getTable(LibraryDetails");

Assertion.assertEquals(expectedTable. actualTable);

d. public void testCheckDataLoaded() throws Exception{

IDataSet databaseDataSet = getConnection().createDataSetO:

ITable actualTable = databaseDataSet.getTable("LibraryDetails");

IDataSet expectedDataSet = new FlatXmlDataSetBuilder().build(new

File("expectedData.xml"));

ITable expectedTable = expectedDataSet.getTable("LibraryDetails");

Assert.assertEquals(expectedTable. actualTable);

71. Which of the following elements in Junit framework is a collection of one or more test cases or test suites that are executed to perform the testing of an application?

Test case

Test runner

Test

Test Result

72. Raj wants to test his Java Application and after testing he wanted to know about the code coverage report of his application using JaCoCo tool. He observed that the report showed the codes of his program being displayed in different colors for coverage report. Raj wants to know the codes which have not been fully covered by the tests. Which of the following colors would highlight the partially covered code to Raj?

a. Red

b. Yellow

c. Green

d. White

73. Nancy has created a Calculator Application having functions like sum, difference, product, division, power, log, exponential, factorial etc. She wants to test his application thoroughly before using it. she decides to test each functions with five different set of inputs and expected outputs. She came to know that creating parameterized class is the solution to his problem. How should nancy annotate her test class?

a. @RunWith(Parameterized.class)

b. Suite(parameterized.class)

c. @Suite.SuiteClasses(parameterized.class)

d. @RunWith()

74. An application calculates the grade of a student based on its score, for e.g. if the student has scored between 60 to 69 its grade should be satisfactory. Which of the following boundary conditions will help you ensure the preceding requirement?

a.Cardinality

b.Conformance

c.Range

d.Ordering

75. Jack is testing his website www.HotSpot.com. He finds out when the number of user trying to access the website is 1000, the time taken for the website to respond is 0.2 s, whereas if the number of user is doubled then the response time becomes 0.5s.Which form of testing is Thomas trying to perform ?

a.Performance Testing

b.Acceptance Testing

c.System Testing

d.Integration Testing

76. Jason has developed an employee management application that uses the MySQL database. Post development, he wants to test this application using DbUnit. Therefore he creates a test class by extending the DatabaseTestCase class. Now he connects to the database using the getConnection() method. Which one of the following code snippets should he use to connect to the employee database?

a. protected iDatabaseConnection getConnection() throws Exception

{

Class.forName(“com.mysql.jdbc.Driver”);

Connection jdbcConnection = DriverManager.getConnection(“jdbc:mysql”,”root”,”root”);

return new DatabaseConnection(jdbcConnection);

}

b. protected iDatabaseConnection getConnection() throws Exception

{

Class.forName(“com.mysql.jdbc.Driver”);

Connection jdbcConnection = DriverManager.getConnection(“jdbc:mysql://localhost:3306”,”root”,”root”);

return new DatabaseConnection(jdbcConnection);

}

c. protected iDatabaseConnection getConnection() throws Exception

{

Class.forName(“com.mysql.jdbc.Driver”);

Connection jdbcConnection = DriverManager.getConnection(“jdbc:mysql://localhost:3306/employee”,”root”,”root”);

return (jdbcConnection);

}

d. protected iDatabaseConnection getConnection() throws Exception

{

Class.forName(“com.mysql.jdbc.Driver”);

Connection jdbcConnection = DriverManager.getConnection(“jdbc:mysql://localhost:3306/employee”,”root”,”root”);

return new DatabaseConnection(jdbcConnection);

}

77. Jacquiline has created an application Employee Management System for manipulating the Employee data. Now she wants her application to be integrated continuously at every five minutes in the last quarter of every hour. Which one of the following cron syntax does she need to follow to build her project?

a.H(45-60)/5 \* \* \* \*

b.H(46-60) 5 \* \* \*

c.H(45-59)/5 \* \* \* \*

d.H(45-59)/ 5 \* \* \* \*

78. Jason wants to load test his website with a set of 60 users where in each user should hit the server after 10 seconds. In addition, he wants to execute this test 15 times. Which one of the following options should Jason use to meet the preceding requirements of load testing?

a.Number of Threads 60, Ramp up period 600, Loop Count 15

b.Number of Threads 600, Ramp up period 0, Loop Count 10

c.Number of Threads 60, Ramp up period 100, Loop Count 15

d.Number of Threads 60, Ramp up period 10, Loop Count 15

79. Sandra has created a website for an Ad Agency that contains various pages, such as Home Page, Login Page, Registration Page, and Admin page. The Login Page allows users to login as an administrator or as a customer by specifying the valid credentials. She wants to test the LoginPage and verify whether the components, such as textboxes named username and password, radio buttons for administrator and customer options, and buttons for submit and reset functionalities, are available on the page. For this, he has written the following code snippet:

public void testPageComponents {

beginAt(“/HomePage.jsp”);

assertTitleEquals(“Login Page”);

assertFormElementPresent(“textboxes”);

assertRadioOptionPresent(“administrator”,”customer”);

assertButtonPresentWithText(“Submit”);

}

However, the preceding code does not meet the specified testing requirements. Which one of the following code snippets should Sandra use to resolve this problem?

a.@Test

public void testPageComponents{

beginAt(“/HomePage.jsp”);

assertTitleEquals(“Login Page”);

assertFormElementPresent(“username”);

assertFormElementPresent(“password”);

assertRadioOptionPresent(“rBtn1”,”administrator”);

assertRadioOptionPresent(“rBtn1”,”customer”);

assertButtonPresentWithText(“Submit”);

}

b.@Test

Public void testPageComponents{

beginAt(“/HomePage.jsp”);

assertTitleEquals(“Login Page”);

assertFormElementPresent(“username”);

assertFormElementPresent(“password”);

assertRadioOptionPresent(“rBtn1”,”administrator”);

assertRadioOptionPresent(“rBtn2”,”customer”);

assertButtonPresentWithText(“Submit”);

}

c.@Test

Public void testPageComponents{

beginAt(“/LoginPage.jsp”);

assertTitleEquals(“Login Page”);

assertFormElementPresent(“username”);

assertFormElementPresent(“password”);

assertRadioOptionPresent(“rBtn1”,”administrator”);

assertRadioOptionPresent(“rBtn1”,”customer”);

assertButtonPresentWithText(“Submit”);

}

d.@Test

Public void testPageComponents{

beginAt(“/HomePage.jsp”);

assertTitleEquals(“Login Page”);

assertFormElementPresent(“username”);

assertFormElementPresent(“password”);

assertRadioOptionPresent(“rBtn1”,”user”);

assertRadioOptionPresent(“rBtn1”,”customer”);

assertButtonPresentWithText(“Submit”);

}

80. Henry has developed a student management application that uses the MySQL database. In this application, he has created the student database having the StudentDetails table. This table contains the StudentName, StudentMarks, and StudentID column. Now he wants to test this application using DbUnit. For this, he creates a test class by extending the DatabaseTestCase class. Further, he connects to the database by overriding the getConnection() method. Next, he needs to create the data file, which will be used as input. For this he has created the following XML file.

<? xml version=”1.0” encoding=”UTF-8”>

<dataset>

<!—student table-->

<studentdetails StudentId=001 StudentName=John Hodge StudentMarks=85/>

<studentdetails StudentId=002 StudentName=Sarah Williams StudentMarks=95/>

</dataset>

However when the test code is executed, it generates errors. Which one of the following options should he use to rectify the error in the XML file?

a.<?xml version=”1.0” encoding=”UTF-8”>

<dataset>

<!—student table-->

<studentdetails StudentID=”001” StudentName=”John Hodge” StudentMarks=”85”/>

<studentdetails StudentID=”002” StudentName=”Sarah Williams” StudentMarks=”95”/>

</dataset>

b.<?xml version=”1.0” encoding=”UTF-8”>

<dataset>

<!—student table-->

<StudentDetails StudentID=”001” StudentName=”John Hodge” StudentMarks=”85”/>

<StudentDetails StudentID=”002” StudentName=”Sarah Williams” StudentMarks=”95”/>

</dataset>

c.<?xml version=”1.0” encoding=”UTF-8”>

<!—student table-->

<StudentDetails StudentID=001 StudentName=John Hodge StudentMarks=85/>

<StudentDetails StudentID=002 StudentName=Sarah Williams StudentMarks=95/>

</details>

d.<?xml version=”1.0” encoding=”UTF-8”>

<StudentDetails>

<!—student table-->

<studentdetails StudentID=001 StudentName=John Hodge StudentMarks=85/>

<studentdetails StudentID=002 StudentName=Sarah Williams StudentMarks=95/>

</StudentDetails>

81. John has created an EJB module that converts Dollars to Yuan, as shown in the following code snippet:

private double yuanRate = 6.45;

public double DollarToYuan(double dollars)

{

double result = dollars\*yuanRate;

return result;

}

The DollarToYuan() method contains the code to convert the specified units of Dollar into Yuan. Now, he wants to create a test case for this EJB module by performing the in-container testing. Which one of the following options will help him fulfill the desired requirements?

1. a. ...............

...............

@BeforeClass

public static void setUpClass()

{

container.close();

}

@AfterClass

public static void tearDownClass()

{

container = javax.ejb.embeddable.EJBContainer.createEJBContainer();

}

b. ...............

...............

@BeforeClass

public static void setUpClass ()

{

container = javax.ejb.embeddable.EJBContainer.createEJBContainer();

}

@AfterClass

public static void tearDownClass()

{

container.close ();

}

c. . ...............

...............

@Before

public static void setUpClass()

{

container = javax.ejb.embeddable.EJBContainer.createEJBContainer();

}

@After

public static void tearDownClass()

{

container.close();

}

d. ...............

...............

@Before

public static void setUpClass()

{

container.close();

}

@After

public static void tearDownClass()

{

container = javax.ejb.embeddable.EJBContainer.createEJBContainer();

}

82. Sonia has developed a Student class that defines two methods writeStudentData () and readStudentData (). Which in turn invoke the writeData() and readData() methods of the WriteStudent class respectively. Sonia needs to test the WriteStudent class. However, the WriteStudent class does not implement the logic for reading and writing data. Therefore, to represent this class, she can create stubs for the writeData() and readData() methods. To ensure that the test passes successfully, the String values, Success and Student Data, needs to be returned from the writeData () and readData () methods respectively. Which of the following options will she use to create stubs for the writeData () and readData () methods?

1. public class writeStudentStub

{

public String writeData(Student s)

{

return “Success”;

}

public String readData()

{

return “Student Data”;

}

}

1. public class writeStudentStub

{

public String writeData()

{

return “Success”;

}

public String readData()

{

return “Student Data”;

}

}

c. class writeStudentStub

{

public String writeData(Student s)

{

return “Student Data”;

}

public String readData()

{

return “Success”;

}

}

d. class writeStudentStub

{

public void writeData(Student s)

{

return “Student Data”;

}

public String readData()

{

return “Success”;

}

}

82. Thomas has created a Web Application to calculate the simple interest, using the following code snippet:

public double calcSimpleInterest (double principal, double rate, double time)

{

if (principal >=0 && rate >=0 && time >=0)

{

double si;

si = (principal\*rate\*time)/100;

return si;

}

return 0;

}

He has written the following code snippet to test the preceding code.

@Test

public void testCalcSimpleInterest()

{

System.out.println(“Calculating SimpleInterest”);

double principal = 12000;

Double rate =5;

Double time =3;

Interest instance = new Interest();

Double expResult = 1800;

Double result = instance.SimpleInterest(principal,rate,time);

assertEquals(expResult, result,0.5);

}

However he is not able to achieve 100% code coverage. Which of the following test cases must she write additionally to achieve 100% code coverage during the test?

a. @Test

public void testCalcSimpleInterest2()

{

System.out.println(“Calculating Simple Interest”);

Double principal = 12000;

Double rate =5;

Double time =3;

Interest instance = new Interest();

Double expResult = 1800;

Double result = instance. CalcSimpleInterest (principal, rate, time);

assertEquals (expResult, result, 0.5);

}

b. @Test

public void testCalcSimpleInterest2()

{

System.out.println (“Calculating SimpleInterest”);

double principal = 12000;

double rate =5;

double time = -3;

Interest instance = new Interest();

double expResult = 0;

double result = instance.SimpleInterest (principal, rate, time);

assertEquals (expResult, result, 0.0);

c. @Test

Public void testSimpleInterest2 ()

{

System.out.println (“Calculating SimpleInterest”);

Double principal = 10000;

Double rate =3;

double time =3;

Interest instance = new Interest();

double expResult = 900;

double result = instance.SimpleInterest(principal, rate, time);

assertEquals (expResult, result, 0.0);

}

d. @Test

public void testSimpleInterest2 ()

{

System.out.println (“Calculating SimpleInterest”);

double principal = 18000;

double rate = 6;

double time = 4;

Interest instance = new Interest();

double expResult = 4320;

double result = instance.SimpleInterest(principal, rate, time);

assertEquals (expResult, result, 0.0);

}

83. Rati has developed an EJB module for the conversion of distances into different units, such as kilometers, meters, and miles. Now she needs to test this module. For this, she has written the following code snippet in the pom.xml file to add the server property.

<dependencies>

<endorsed.dir> ${project.build.directory}/endorsed</endorsed.dir>

<project.build.sourceEncoding>UTF-8</ project.build.sourceEncoding>

<glassfish.embedded-static-shell.jar>C:/Program Files/glassfish-4.1/glassfish/lib/embedded/glassfish-embedded-static-shell.jar</ glassfish.embedded-static-shell.jar>

</dependencies>

However the preceding code snippet encounters certain errors. Which one of the following code snippets should Susan use to resolve the errors?

1. <dependencies>

<endorsed.dir> ${project.build.directory}/endorsed</endorsed.dir>

<project.build.sourceEncoding>UTF-8</ project.build.sourceEncoding>

<glassfish.embedded-static-shell.jar>C:/Program Files/glassfish-4.1/glassfish/lib/embedded/glassfish-embedded-static-shell.jar</ glassfish.embedded-static-shell.jar>

</dependencies>

1. <properties>

<endorsed.dir> ${project.build.directory}/endorsed</endorsed.dir>

<project.build.sourceEncoding>UTF-8</ project.build.sourceEncoding>

<glassfish.embedded-static-shell.jar>C:/Program Files/glassfish-4.1/glassfish/lib/embedded/glassfish-embedded-static-shell.jar</ glassfish.embedded-static-shell.jar>

</properties>

1. <dependencies>

<dependency>

<endorsed.dir> ${project.build.directory}/endorsed</endorsed.dir>

<project.build.sourceEncoding>UTF-8</ project.build.sourceEncoding>

<glassfish.embedded-static-shell.jar>C:/Program Files/glassfish-4.1/glassfish/lib/embedded/glassfish-embedded-static-shell.jar</ glassfish.embedded-static-shell.jar>

</dependency>

</dependencies>

1. <properties>

<property>

<endorsed.dir> ${project.build.directory}/endorsed</endorsed.dir>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<glassfish.embedded-static-shell.jar>C:/Program Files/glassfish-4.1/glassfish/lib/embedded/glassfish-embedded-static-shell.jar</ glassfish.embedded-static-shell.jar>

</property>

</properties>

84. David has created an application to check whether an entered character is a vowel or a consonant, using the following code snippet:

public class VowelConsonant {

public String check(String character)

{

String v = “vowel”;

String c=”consonant”;

if(character==”a” || character==”e” || character==”i”|| character==”o” || character==”u”)

{

return v;

}

else

{

return c;

}

}

public static void main(String[] args)

{

// To Do Code Application Logic here.

}

}

Now he has been assigned the task to test this application by creating three test cases, one that checks for a vowel, second that checks for a consonant, and third one fails and displays an appropriate human readable error message. Which one of the following options will help him fulfill the requirements?

1. a. @Test

public void testCheck1()

{

System.out.println (“Testing non-alphabets”);

VowelConsonant vc = new VowelConsonant();

String result = vc.check(“#”);

assertThat(result.is (equalTo(“alphabet”)));

}

@Test

public void testCheck2()

{

System.out.println(“Testing Consonant”);

VowelConsonant vc = new VowelConsonant();

String result = vc.check(“s”);

assertThat(result.is(equalTo(“consonant”)));

}

@Test

public void testCheck3()

{

System.out.println(“Testing Consonant”);

VowelConsonant vc = new VowelConsonant();

String result = vc.check(“e”);

assertThat(result.is(equalTo(“vowel”)));

}

1. @Test

public void testCheck1()

{

System.out.println (“Testing non-alphabets”);

VowelConsonant vc = new VowelConsonant();

String result = vc.check(“#”);

assertThat(result.is (equalTo(“non-alphabets”)));

}

@Test

public void testCheck2()

{

System.out.println(“Testing Consonant”);

VowelConsonant vc = new VowelConsonant();

String result = vc.check(“s”);

assertThat(result.is(equalTo(“consonant”)));

}

@Test

public void testCheck3()

{

System.out.println(“Testing Consonant”);

VowelConsonant vc = new VowelConsonant();

String result = vc.check(“e”);

assertThat(result.is(equalTo(“vowel”)));

}

1. @Test

public void testCheck1()

{

System.out.println (“Testing non-alphabets”);

VowelConsonant vc = new VowelConsonant();

String result = vc.check(“#”);

assertThat(result.is (equalTo(“alphabet”)));

}

@Test

public void testCheck2()

{

System.out.println(“Testing Consonant”);

VowelConsonant vc = new VowelConsonant();

String result = vc.check(“e”);

assertThat(result.is(equalTo(“consonant”)));

}

@Test

public void testCheck3()

{

System.out.println(“Testing Consonant”);

VowelConsonant vc = new VowelConsonant();

String result = vc.check(“e”);

assertThat(result.is(equalTo(“vowel”)));

}

1. d. @Test

public void testCheck1()

{

System.out.println (“Testing non-alphabets”);

VowelConsonant vc = new VowelConsonant();

String result = vc.check(“#”);

assertThat(result.is (equalTo(“alphabet”)));

}

@Test

public void testCheck2()

{

System.out.println(“Testing Consonant”);

VowelConsonant vc = new VowelConsonant();

String result = vc.check(“s”);

assertThat(result.is(equalTo(“consonant”)));

}

@Test

public void testCheck3()

{

System.out.println(“Testing Vowels”);

VowelConsonant vc = new VowelConsonant();

String result = vc.check(“b”);

assertThat(result.is(equalTo(“vowel”)));

}

85. Consider the following code snippet:

@Before

public void setup(){

employeeService = Mockito.verify(EmployeeService);

employeeManager = new EmployeeManager();

employeeManager.setEmployeeService (employeeService);

}

In the preceding code snippet, the setup() method is defined, which initializes the components of the application required for performing unit test. Within this method, the employeeService mock object is created using the Mockito framework and passing the EmployeeService class as its parameter. Further, an instance of the EmployeeManager class is created and its employee service is set as employeeService. However, the preceding code generates an error.Analyze the code and rectifies the error.

1. @Before

public void setup()

{

employeeService = Mockito.mock (EmployeeService.class);

employeeManager = new EmployeeManager();

employeeManager.setEmployeeService (employeeService);

}

1. @Before

public void setup()

{

employeeService = Mockito.when(EmployeeService.class);

employeeManager = new EmployeeManager();

employeeManager.setEmployeeService (employeeService);

}

1. @Before

public void setup()

{

employeeService = Mockito.expect(EmployeeService.class);

employeeManager = new EmployeeManager();

employeeManager.setEmployeeService (employeeService);

}

1. @Before

public void setup()

{

employeeService = Mockito.replay(EmployeeService.class);

employeeManager = new EmployeeManager();

employeeManager.setEmployeeService (employeeService);

}

86. Dexy has created an application for maintaining the details of the author of a publication. In this application, he has created a servlet that validates the login credentials and redirects the authors to the appropriate page. Now he needs to test the functionality of the servlet by using the Mockito framework. For this, he has added the dependencies in the pom.xml file, using the following code snippet:

<dependency>

<groupId>org.easymock</groupId>

<artifactId>easymock</artifactId>

<version>3.2</version>

<scope>test</scope>

<groupId>org.glassfish.extras</groupId>

<artifactId>glassfish-embedded-static-shell</artifactId>

<version>4.1</version >

<scope>test</scope>

</dependency>

Then he created the test class and executed test cases. On executing the test, she encountered certain errors. Which one of the following code snippets will she use to rectify the errors?

a. <dependency>

<groupId>org.easymock</groupId>

<artifactId>easymock</artifactId>

<version>3.2</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.glassfish.extras</groupId>

<artifactId>glassfish-embedded-static-shell</artifactId>

<version>4.1</version >

<scope>test</scope>

</dependency>

b. <dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito.all</artifactId>

<version>1.9.5</version>

<groupId>org.glassfish.extras</groupId>

<artifactId>glassfish-embedded-static-shell</artifactId>

<version>4.1</version >

<scope>test</scope>

</dependency>

c. <dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito.all</artifactId>

<version>1.9.5</version>

</dependency>

<dependency>

<groupId>org.glassfish.extras</groupId>

<artifactId>glassfish-embedded-static-shell</artifactId>

<version>4.1</version >

<scope>test</scope>

</dependency>

d. <dependencies>

<groupId>org. easymock</groupId>

<artifactId>easymock</artifactId>

<version>3.2</version>

<scope>test</scope>

</dependencies>

<dependencies>

<groupId>org.glassfish.extras</groupId>

<artifactId>glassfish-embedded-static-shell</artifactId>

<version>4.1</version >

<scope>test</scope>

</dependencies>

87. George wants to create a mock object using the Mockito framework. To implement the Mockito framework, he needs to add the Mockito dependency. Which one of the following code snippets will he use to add the Mockito dependency in the pom.xml file?

a. <dependencies>

<dependency>

<groupId>org.mockito</groupId>

<artifactId>Mockito-all</artifactId>

<version>1.9.5</version>

</dependency>

</dependencies>

b. <dependencies>

<groupId>org.mockito</groupId>

<artifactId>Mockito-all</artifactId>

<version>1.9.5</version>

</dependencies>

c. <dependency>

<groupId>org.mockito</groupId>

<artifactId>Mockito-all</artifactId>

<version>1.9.5</version>

</dependency>

d. <dependencies>

<dependency>

<groupId>org.easymock</groupId>

<artifactId>easymock</artifactId>

<version>1.9.5</version>

</dependency>

</dependencies>

88. Henry has developed an application to store the employee details, such as name, address, department, employee id, and marital status. In this application, you have created a bean class to get and set the values for these properties. In addition, you have created a DAO class that inserts, retireves, and deletes the records of employee. Therefore, to invoke the insert () method of the DAO class, you need to ensure that the bean class is instantiated first. Which one of the following boundary conditions will help you ensure the preceding requirement?

a. Cardinality

b. Conformance

c, Reference

d. Ordering

89. The test cases for a web application are being executed under the given test environment to verify the test results. Which one of the following phases of the STLC life cycle is being referred in this case?

a. Test Planning

b. Test Case Development

c. Test environment Setup

d. Test Execution

90. You have to develop an application for a college to manage the data of its faculty. Post development, you need to test this application with numerous input values and verify the expected values with the actual result. Therefore, you plan to create separate data files for input and expected values instead of creating multiple test cases for each set of values. Which advantage of data-driven testing will you be able to achieve by performing the preceding task?

a. Manageable Records

b. Updating data files with the change in the test plan.

c. Reduced redundancy in test cases.

d. Independent test scripting.

91. Which of the following options in a summary report represents the amount of requests per second that the server can handle?

a. KB/Sec

b. #Samples

c. Throughput

d. Average

92. Which one of the following options is an advantage of CI testing?

a. Automated deployment

b. User-oriented development

c. Manageable data records

d. Platform- dependent development

93. Which one of the following colors signifies the code area that is left uncovered by the test cases?

a. Green

b. Yellow

c. White

d. Red

94. Which one of the following options do you need to select to include a CSV Data Set Config element into your plan in JMeter?

a. Controller

b. Configuration Element

c. Assertion

d. Listener

95. Which one of the following formulas is used to calculate the statement coverage?

a. No. of statements exercised/Total number of statements)\*100

b. (Total Number of Statements/No. of Statements exercised)\*100

c. No. of Conditions exercised/Total Number of Conditions)Total number of statements

d. (No. of loops exercised/Total number of loops)\* Total number of statements

96. The test cases for a web application are being executed under the given test environment to verify the test results. Which one of the following phases of the STLC life cycle is being referred in this case?

a. Nice

b. Normal

c. Strict

d. Real

97. Julia is testing her Application using Easymock framework in which easy expects the method of the mock object should be called only once during the test process. Which one of the following methods of the IExpectationSetters interface would be most appropriate for her to achieve the task?

a. anyTimes()

b. atLeastOnce()

c. Once()

d. times(int n)

98. Rachel tries to test her JSP application using the JWebUnit framework in a maven project. She has to add the dependency of the framework in the pom.xml file.Which of the following code would be able to download the framework from the repository?

a. <dependency>

<groupId>net.sourceforge.jwebunit</groupId>

<artifactId>jwebunit-htmlunit-plugin</artifactId>

<version>3.2</version>

<scope>production</scope>

</dependency>

b. <dependency>

<groupId>net.sourceforge.jwebunit</groupId>

<artifactId>jwebunit-htmlunit-plugin</artifactId>

<version>3.2</version>

<scope>deploy</scope>

</dependency>

c. <dependency>

<groupId>net.sourceforge.jwebunit</groupId>

<artifactId>jwebunit-htmlunit-plugin</artifactId>

<version>3.2</version>

<scope>test</scope>

</dependency>

d. <dependency>

<groupId>net.sourceforge.jwebunit</groupId>

<artifactId>jwebunit-htmlunit-plugin</artifactId>

<version>3.2</version>

<scope>application</scope>

</dependency>

99. Jennifer is testing a JSP application where she begins to test the HomePage.jsp with title “Home Page” and a link to Login.jsp page having title “Login Page”. The Login.jsp page has two input elements namely username and password. After putting user as “Admin” and password as “pass123” and submitting the page the user moves to the Welcome.jsp page otherwise to the Error.jsp. Identify the correct sequence of steps that Jennifer should follow?

a. beginAt(“Home.jsp”);

assertTitleEquals(“Home Page”);

clickButtonWithExactTest(“Login”);

assertTitleEquals (“Login Page”);

setTextField(“username”,”Admin”);

setTextField(“password”,”pass123”);

submit();

assertTitleEquals(“Welcome.jsp”);

b. beginAt(“Home.jsp”);

assertTitleEquals(“Home Page”);

clickLinkWithExactTest(“Login”);

assertTitleEquals(“Login Page”);

setTextField(“username”,”Admin”);

setTextField(“password”,”pass123”);

submit();

assertTitleEquals(“Error.jsp”);

c. beginAt(“Home.jsp”);

assertTitleEquals(“Home Page”);

clickLinkWithExactTest(“Login”);

assertTitleEquals(“Login Page”);

setTextField(“username”,”Admin”);

setTextField(“password”,”abc”);

submit();

assertTitleEquals(“Welcome.jsp”);

d. beginAt(“Home.jsp”);

assertTitleEquals(“Home Page”);

clickLinkWithExactTest(“Login”);

assertTitleEquals(“Login Page”);

setTextField(“username”,”Admin”);

setTextField(“password”,”pass123”);

submit();

assertTitleEquals(“welcome.jsp”);

100. Bill has created a Servlet named LoginServlet which receives the user name and password from the login page. If the username is “Admin” and the password is “admin” it redirects the response to the “welcome.jsp” page whereas in case of wrong authentication it redirects to “error.jsp” page. Now he wants to test the Servlet for successful user authentication for which he has written the following code :

1// public class LoginServletTest

{

2// HttpServletRequest mockHttpServletRequest = mock(HttpServletRequest.class);

3// HttpServletResponse mockHttpServletResponse = mock (HttpServletResponse.class);

public LoginServlet (){

}

…………………………………..

……………………………………

@Test

public void testSuccessProcessRequest() throws IOException, ServletException

{

4// when(mockHttpServletRequest.getParameter(“username”)).thenReturn(“Admin”);

5// when(mockHttpServletRequest. getParameter (“password”)).thenReturn(“admin”);

StringWriter sw = new StringWriter();

PrintWriter pw = new PrintWriter(sw);

6// when(mockHttpServletResponse.getResponse()).thenReturn(pw);

LoginServlet loginServlet = new LoginServlet ();

loginServlet.processRequest(mockHttpServletRequest, mockHttpServletResponse);

7// verify (mockHttpServletRequest, atLeast (1)).getParameter (“username”);

8// verify (mockHttpServletRequest, atLeast (1)).getParameter (“password”);

9// verify (mockHttpServletResponse, atLeast (1)).getWriter();

10// verify (mockHttpServletResponse).sendRedirect(“success.jsp”);

But he notices that the Program doesn’t compile successfully. Which of the following options should be the corrective measure for Bill to resolve the issue?

1. Line 4 should be when(mockHttpServletRequest.getParameter(“Admin”).thenReturn(“username”);
2. Line 6 should be when(mockHttpServletResponse.getWriter()).thenReturn(pw);
3. Line 7 should be verify (mockHttpServletRequest, atLeast (1)).getParameter(“admin”);
4. Line 10 should be verify(mockHttpServletResponse).send (“success.jsp”);

101. To unit test a Servlet filter which one of the following interface should not be mocked?

a. HttpServletRequest

b. HttpServletResponse

c. FilterChain

d. Filter

102. Identify the method of the EJB container class that can be used to retrieve a naming context for references to session beans executing in the embedded container.

a. getContext()

b. getNamingContext()

c. getContextName()

d. getLookupContextName()

103. Statement A: Load Testing is performed to determine the website’s performance under the normal conditions and peak conditions by varying the load.

Statement B: Load Testing simulates the usage traffic that helps you determine the reliability and functionality of your website.

1. Statement A is true B is false
2. Statement A is false B is true.
3. Both Statement A and B is True.
4. Both Statement A and B is false.

104. Which one of the following statements is not a valid feature of the JMeter Tool?

a. It is an Open Source Software.

b. It is platform independent i.e. can be used in both Windows and Linux.

c. It stores its test in the JSON format.

d. It is highly extensible as it allows you to write your own tests and support visualization plug-ins to enhance your tests.

105. Arrange the following statements according to their order of execution regarding website testing using JMeter Tool.

1. Create a sample HTTP request.

ii. Create a thread group.

iii. Specify the server name/IP of the website being tested, inside the sample HTTP request.

iv. Add a summary report to view the results of the test.

v. Create a Test Plan in JMeter.

a. V🡪ii🡪iii🡪i🡪iv

b. V🡪ii🡪i🡪iii🡪iv

c. V🡪ii🡪iii🡪i🡪iv

d. V🡪ii🡪i🡪iv🡪iii

106. Which of the following options represents the amount of time the server takes to receive and process a request?

a. Sample Time

b. Throughput

c. Status

d. Latency

107. Which of the following assertions in JMeter Tool verifies whether the response length is equal, not equal, greater, or less that the expected size in bytes ?

a. Response assertion

b. Size assertion

c. HTML assertion

d. XML assertion

108. Which of the following constants of the DatabaseOperation class updates the existing values in the database that are specified in the data file and inserts the non-existing values into the table from the data file?

a. UPDATE

b. INSERT

c. REFRESH

d. CLEAN\_INSERT

109. Which of the following constants of the DatabaseOperation class updates the existing values in the database that are specified in the data file and inserts the non-existing values into the table from the data file?

A. Red

B.Blue

C.YELLOW

D.GREY