

Which of the following is correct to implement interservlet communication?

Single Choice - Select one correct answer from the options list.

RequestDispatcher rd = RequestDispatcher.getRequestDispatcher("path"); rd.include(request, response);

RequestDispatcher rd = getServletContext().getRequestDispatcher("path"); rd.include(request, response);

RequestDispatcher rd = RequestDispatcher.getRequestDispatcher(path);

RequestDispatcher rd = getServletContext().getRequestDispatcher("path");
rd.dispatch(request, response);

Q2

```
public class TestClass {  
    @Test(timeout=300)  
    public void testDisp()  
    {  
        assertEquals(11,Integer.parseInt("11"));  
    }  
}
```

Consider the given java code snippet & predict the output.

Single Choice - Select one correct answer from the options list.

Shows compilation error as @Test doesn't have timeout parameter.

The test has to be done with in 300 milli seconds. If not, the test should fail.

The test has to be done with in 300 seconds. If not, the test should fail.

The test has to be done with in 300 minutes. If not, the test should fail.

Q3

```
let productCode number //line1  
let productName string //line 2  
productCode = 101 //line 3  
productName = "Sony Xperia" //line 4  
console.log(productCode); //line 5
```

What will be the output of above code?

Single Choice - Select one correct answer from the options list.

Compilation

error at line 1

Compilation

error at line 2

compilation error at line

1 and line 2

compilation error at line

3 and line 4

Q4

Angular framework provides event binding using in-built event as well as custom event.

Custom events are the EventEmitter instances. To create a custom event we need to create an instance of EventEmitter annotated by _____.

Single Choice - Select one correct answer from the options list.

@Input()

@Get()

@Output()

@Set()

Q5

_____ shortcut method to perform put request.

Single Choice - Select one correct answer from the options list.

http.put

http.post

http.get

http.delete

Q6

Which of the following statement is incorrect about anonymous inner class?

Single Choice - Select one correct answer from the options list.

Anonymous inner classes have no name, and their type must be either a subclass of the named type or an implementer of the named interface.

Because of polymorphism, the only methods you can call on an anonymous inner class reference are those defined in the reference variable class (or interface), even though the anonymous class is really a subclass or implementer of the reference variable type

An anonymous inner class can extend more than one subclass or implement more than one interface.

An anonymous inner class is always created as part of a statement; don't forget to close the statement after the class definition with a curly brace. This is a rare case in Java, a curly brace followed by a semicolon.

Q7

```
import java.util.function.BiFunction;
public class TestQuestion {
    public static void main(String[] args)
    {
        BiFunction<Integer, Integer, Boolean> isMax = (x, y) -> {
            return (x > y ? true : false);
        };
        System.out.println(isMax.apply(45, 50));
    }
}
```

What will be the output of above code?

Single Choice - Select one correct answer from the options list.

True

False

Compilation error as a BiFunction requires only 2 arguments
Exception is thrown

Q8

```
import java.util.TreeSet;
public class LambdaDemo {
    public static void main(String[] args) {
        TreeSet<Integer> treeSet = new TreeSet<Integer>((o1, o2) -> (o1 > o2) ? -1
        : (o1 < o2) ? 1 : 0);
        treeSet.add(850);
        treeSet.add(235);
        treeSet.add(1080);
        treeSet.add(15);
        treeSet.add(5);
        System.out.println(treeSet);
    }
}
```

What will be the output?

Single Choice - Select one correct answer from the options list.

[1080, 850, 235, 15, 5]

[1080, 850, 235, 5, 15]

Compilation error

[]

Q9

Refer to the below code.

```
import java.util.Arrays;
import java.util.List;
import java.util.TreeSet;

public class MyArray{

    public static void main(String a[])
    {
        String[] strArr = {"one", "two", "three", "four", "four", "five"};
        //convert string array to list
        List<String> tmpList = Arrays.asList(strArr);
        //create a treeset with the list, which eliminates duplicates
        TreeSet<String> unique = new TreeSet<String>(tmpList);
        System.out.println(unique);
    }
}
```

What will be the output?

Single Choice - Select one correct answer from the options list.

[five, four, one, three, two]

[five, four, four, one, three, two]

compilation fails as list object cannot be passed to constructor of TreeSet

[one,two,three,four,five]

Q10

```
public class StreamDemo {  
  
    public static void main(String[] args)  
    {  
        BiFunction<String,String,String>fun = (str1,str2)->str1+str2;  
        String result=null;  
        result = fun.apply("10", "20");  
        System.out.println(result);  
    }  
}
```

What will be the output of above code?

Single Choice - Select one correct answer from the options list.

1020

30

compilation error

Exception

Q11

Consider the code given:

```
List<String> names = Arrays.asList("Jack","Sam","Abhishek","Mohan");
```

```
names.stream().filter(str -> str.length() > 5).forEach(str->System.out.println(str));
```

What will be the output of given stream operation?

Single Choice - Select one correct answer from the options list.

Jack

Sam

Abhishek

Mohan

Q12

What will be the output of the program?

```
import java.util.*;  
class Test  
{  
    public static void main (String[] args)  
    {  
        Set hashSet = new HashSet();  
        hashSet.add(1);  
        hashSet.add(null);  
        hashSet.add(null);  
        hashSet.add("1");  
    }  
}
```

```

        System.out.println(hashSet);
    }
}

```

Single Choice - Select one correct answer from the options list.

[null,1,1]
 [1,1]
 [1]
 [1,null,null]

Q13

```

import java.util.ArrayList;
class LambdaDemo {
    public static void main(String args[]) {
        ArrayList<Integer> arrL = new ArrayList<Integer>();
        arrL.add(1);
        arrL.add(2);
        arrL.add(3);
        arrL.forEach(n -> System.out.println(n));
        arrL.forEach(n -> {
            if (n % 2 == 0)
                System.out.println(n);
        });
    }
}

```

What will be the output?

Single Choice - Select one correct answer from the options list.

1 2 3 2
 1 3 3 2
 1 1 2 2
 1 3 2 2

Q14

```

public class Test {
    public static void main(String[] args)
    {   String name=new String("aravind");

        int position=name.replace('a', '*').substring(2,4).indexOf('i');
        System.out.println(position);
    }
}

```

//end of class

What will be the output?

Single Choice - Select one correct answer from the options list.

2
 4
 0
 -1

Q15

```

package com.capgemini.corejava;
public class Test {
    String name; // Line1
    public Test()
    {
        name = "Sara";
    }
    public static void main(String[] args)
    {

        System.out.println(name);
    }
}

```

What is the output of the above Java code?

Single Choice - Select one correct answer from the options list.

Sara

Compilation error, as String name "in Line 1" is not declared with static modifier.

Throws NullPointerException

The code runs with no output

Q16

Given:

```

2. public class Person {
3.     private String name;
4.     private String city;
5.     String getName()
6.     { return name; }
7.     void setName(String n) { name = n; }
8.     void setCity(String c) {
9.         if(c == null) throw new NullPointerException();
10.        city = c;
11.    }
12.    String getCity()
13.    { return city; }
14. }

```

Which are true?

Single Choice - Select one correct answer from the options list.

Compilation fails

The class is well encapsulated.

The setCity() method is an example of loose coupling.

The setCity() method has better encapsulation than setName().

Q17

Given:

```

1. public class LoopTesting {

```

```

2. public static void main(String[] args) {
3. int i = 0;
4. short s = 0;
5. for(int j = 0, k = 0; j < 3; j++) ;
6. for(int j = 0; j < 3; counter(j)) ;
7. for(int j = 0, int k = 0; j < 3; j++) ;
8. for(; i < 5; counter(5), i++) ;
9. for(i = 0; i < 3; i++, System.out.print("howdy ")) ;
10. }
11. static int counter(int y) { return y + 1; }
12. }

```

What is the result? (Choose all that apply.)

Single Choice - Select one correct answer from the options list.

howdy howdy howdy

Compilation fails due to an error on line 5.

Compilation fails due to an error on line 6.

Compilation fails due to an error on line 7.

Q18

```

List<String> list = new ArrayList<>(); //Line 1
list.add("Hello");
list.add("World");
Consumer<List<String>> consumer = Collections ::reverse; //Line 2
consumer.accept(list);
System.out.println(list);

```

Refer to the above code snippet; what will be the output in console?

Single Choice - Select one correct answer from the options list.

Display [World, Hello] in console

Display [Hello, World] in console

Compilation error at Line number 1, as generics not specified

Compilation error at Line number 2, as reverse method returns does not return void

Q19

```

public class JDBCdemo {

    public static void main(String[] args) {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","sys");
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM Emp");
            rs.next();
            while(rs.next())
            {
                System.out.println(rs.getInt(1));
                System.out.println(rs.getString(2));
            }
        }
    }
}

```

```

    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

Single Choice - Select one correct answer from the options list.

It will print records, from second row onwards

No output

It will throw SQLException as rs.next() is called twice

it will print all records

Q20

```

public class JunitTestDemo
{
    @Ignore @Test(expected = ClassCastException.class)
    public void testGetLastName()
    {}
    @Ignore @Test
    public void getFirstName()
    {}
}

```

Consider the above java code snippet. How many test cases will execute?

Single Choice - Select one correct answer from the options list.

to test for expected exceptions

to test the exception that should be thrown

1

0

Q21

Which of the following are correct guidelines for creating exception handlers?

A) No need to analyze the consequences of failures but need to find out correct probability.

B) No need to document the reason for the exception.

C) Catch the exception in the appropriate parent module.

D) Raise the exception whenever the condition arises in the sub module.

Single Choice - Select one correct answer from the options list.

only A & C

only B & C

only C & D

only D & A

Q22

Assuming that there are 5 rows in the ProductDetails table

(COLUMNS: product_id number(10), product_title varchar(30)),

What will be the output of the following Java code?

```
DriverManager.registerDriver (new oracle.jdbc.driver.OracleDriver());
Connection conn =
    DriverManager.getConnection (
        "jdbc:oracle:thin:@10.212.74.18:1521:orcl", "user", "password");
Statement stmt1=conn.createStatement();
ResultSet rs=stmt1.executeQuery("SELECT * FROM ProductDetails");
System.out.println(rs.getInt(1)+" "+rs.getString(2));-----Line 1
```

Single Choice - Select one correct answer from the options list.

Compilation error in the Line 1 as rs.next() is not given before this

SQLException is raised. Because rs.next() is not given in the code

Prints the contents of the first row from the table

SQLException is raised. Because while loop is not given in the code

Q23

```
public static void main(String[] args) {
5. try {
6. if(args.length == 0) throw new Exception();
7. }
8. catch (Exception e) {
9. System.out.print("done ");
10. doStuff(); // assume this method compiles
11. }
12. finally {
13. System.out.println("finally ");
14. }
15. }
```

if we run this code without commandline arguments,

Which is the output of this code?

Single Choice - Select one correct answer from the options list.

done

finally

done finally

compilation error

no output

Q24

Select correct query to find 2nd max salary:

Single Choice - Select one correct answer from the options list.

SELECT MAX(salary) AS salary

FROM emp

WHERE salary IN

(SELECT salary FROM emp MINUS SELECT MAX(salary)

FROM emp);

SELECT MAX(salary) AS salary

FROM emp

WHERE salary IN

```
(SELECT salary FROM emp PLUS SELECT MAX(salary)
FROM emp);
```

```
SELECT MAX(salary) AS salary
FROM emp
WHERE salary IN
(SELECT salary FROM emp PLUS SELECT MIN(salary)
FROM emp);
```

```
SELECT MAX(salary) AS salary
FROM emp
WHERE salary IN
(SELECT salary FROM emp MINUS SELECT MNI(salary)
FROM emp);
```

Q25

```
abstract public class Rectangle {
    abstract void addRectangle();
}
```

```
class ConsumableRectangle extends Rectangle{
    public void addRectangle()
    {
        // add Rectangles
    }
}
```

Pickup the valid statement about the above Java code.

Single Choice - Select one correct answer from the options list.

code will compile without any errors.

Gives compilation errors because addRectangle method is declared as public in child class

Gives compilation errors because ConsumableRectangle must also be declared as abstract.

None of the above

Q26

Which is/are true from the given statements?

1. Static variable live basically as long as their class lives.
2. Instance Variables live as long as their object lives
3. Local variables live as long as their method is on stack ,however ,if their method invokes another method, ,they are temporarily unavailable.
4. Block variables (in a for loop or if stmt) live until the block completes.

Single Choice - Select one correct answer from the options list.

only 1

only 1 and 2

All are correct

1,2,3

Q27

Examine the content of EMPLOYEE table:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	5
7521	WARD	SALESMAN	7698	22-FEB-81	1250	7
7566	JONES	MANAGER	7839	02-APR-81	2975	
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1
7698	BLAKE	MANAGER	7839	01-MAY-81	4000	
7782	CLARK	MANAGER	7839	09-JUN-81	2450	
7788	SCOTT	ANALYST	7566	09-DEC-82	3000	
7839	KING	PRESIDENT		17-NOV-81	5000	
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	3
7876	ADAMS	CLERK	7788	12-JAN-83	2100	
7900	JAMES	CLERK	7698	03-DEC-81	950	1
7902	FORD	ANALYST	7566	03-DEC-81	3000	
7934	MILLER	CLERK	7782	23-JAN-82	3000	
7369	SMITH	CLERK	7902	17-DEC-80	800	1

Identify the query to display
hiredate and number of employees joined in that date in
a given department number. Also order the details based on hiredate.

Single Choice - Select one correct answer from the options list.

SELECT hiredate,COUNT(*) FROM employee GROUP BY hiredate ORDER BY 1

SELECT hiredate,COUNT(*) FROM employee WHERE deptno=20 GROUP BY hiredate ORDER BY 1;

SELECT hiredate,COUNT(*) FROM employee WHERE deptno=20 GROUP BY hiredate ORDER BY 0;

SELECT hiredate,COUNT(*) FROM employee GROUP BY hiredate ORDER BY 1 WHERE deptno=20;

Q28

Examine the structure of dept table:

SQL> desc dept;		
Name	Null?	Type
-----	-----	-----
DEPTNO	NOT NULL	NUMBER
DNAME		VARCHAR2
LOC		VARCHAR2

A developer had created department_master table with the similar structure of dept table using the below query:

```
CREATE TABLE department_master AS SELECT * FROM dept WHERE 1=2;
```

Identify the query to copy all records from dept table and insert the same into department_master table.

Single Choice - Select one correct answer from the options list.

INSERT INTO department_master SELECT * FROM dept;

INSERT INTO department_master VALUES SELECT * FROM dept;

INSERT INTO department_master VALUES AS SELECT * FROM dept;

Copying of records from one table and inserting to another table is impossible.

Q29

What is the correct way to add background color in HTML?

Single Choice - Select one correct answer from the options list.

<body color = "yellow">

<body bg-color = "yellow">

<body style = "background-color=yellow">

<body style = "background-color: yellow;">

Q30

Will the below code snippet displays the heading 'BootStrap' at Medium Mode in bootstrap:

```
<div class="visible-md"><h1>BootStrap</h1></div>
```

Single Choice - Select one correct answer from the options list.

TRUE

FALSE

Q31

Choose the correct style example of class selector?

Single Choice - Select one correct answer from the options list.

```
p {  
text-align: center;  
color: red;  
}
```

```
#para1 {  
text-align: center;  
color: red;  
}
```

```
.center {  
text-align: center;  
color: red;  
}
```

```
h1, h2, p {  
text-align: center;  
color: red;  
}
```

Q32

_____directive is used to iterate an array of items.

Single Choice - Select one correct answer from the options list.

ngFor

ng-For

ngSwitch

ngIf

Q33

Which of the following is correct about Angular 5 Routing?

Single Choice - Select one correct answer from the options list.

Routing helps in directing users to different pages based on the route specified.

Based on the option they choose, the required Angular Component will be rendered to the user.

Both of the above.

None of the above

Q34

Which of the following is correct about Angular 5 Components?

Single Choice - Select one correct answer from the options list.

AngularJS had a focus of Controllers but Angular 5 has changed the focus to having components over controllers.

Components help to build the applications into many modules.

This helps in better maintaining the application over a period of time.

All of the above.

Q35

Which of the following statement is appropriate to be implemented in line No3 .

Assume that id property is unique in Product class.

```
String qStr = "SELECT product FROM Product product WHERE product.id='BPRO13455' "; // Line
```

1

```
TypedQuery<Product> query = entityManager.createQuery(qStr , Product.class); // Line 2  
----- // Line 3
```

Single Choice - Select one correct answer from the options list.

Product product = query.getSingleResult();

List<Product> productList = query.getResultList();

List<Product> productList = query.executeUpdate();

List<Product> productList = query.executeQuery();

Q36

Select the correct way of specifying Single Table per Class hierarchy entity mapping strategy:

Single Choice - Select one correct answer from the options list.

InheritanceType.JOINED

InheritanceType.SINGLE_TABLE

InheritanceType.TABLE_PER_CLASS

None of these

Q37

Select the correct choice based on given statements:

1. DataSource in Spring can be acquired using either DriverManagerDataSource or by using JNDI lookup.

2. Entity manager could be either application managed or container managed.

Single Choice - Select one correct answer from the options list.

Statement 1 is true where as 2 is false

Statement 1 is false where as 2 is true

Both statements are true

Both statements are false

Q38

In Spring ReSTful controller _____ annotation is used to get the data from the URI path.

Single Choice - Select one correct answer from the options list.

@PathVariable

@RequestParam

@RequestMapping

@RestController

Q39

Select the correct entity class code for empty and length validation of name, emailId and mobileNumber of an User, which given option is correct?

Single Choice - Select one correct answer from the options list.

```
class User{
```

```
@Empty @Length
```

```
String name;
```

```
@Empty @Length
```

```
String email;
```

```
@Empty @Length  
String mobile;}
```

```
class User{  
@NotEmpty("name is required") @Length(10)  
String name;  
@NotEmpty("email is required")  
String email;  
@NotEmpty("mobile is required") @Length(10)  
String mobile;}
```

```
class User{  
@NotEmpty(message="name is required") @Size(min=4, max=10)  
String name;  
@NotEmpty(message="email is required")  
String email;  
@NotEmpty(message="mobile is required") @Size(min=10, max=10)  
String mobile;}
```

None of the above

Q40

Select the correct statement about spring annotations?

Single Choice - Select one correct answer from the options list.

@Param annotation is class level annotation used to map requested URL

@ModelAttribute annotation binds a method parameter to a named attribute, exposed to web view

@RequestMapping can only be applied to methods of repository class

Usage of @Controller annotation at class level represents that class is used for Database connection

Q41

In a Spring application, Consider InternalResourceViewResolver is used and handleRequest method returns ModelAndView ("home", "msg", message) then which JSP will be invoked?

Single Choice - Select one correct answer from the options list.

home.jsp

msg.jsp

message.jsp

model.jsp

Q42

In Spring, if Classes are annotated with @Repository then _____.

Single Choice - Select one correct answer from the options list.

we have to explicitly declare a Dao bean

we won't have to explicitly declare a Dao bean.

we won't have to explicitly declare a EntityManagerFactory bean

we won't have to explicitly declare a Entities

Q43

Which annotation can be used to map the JSON object coming through request with Java Class Object?

Single Choice - Select one correct answer from the options list.

@RequestParam

@RequestBody

@PathVariable

@ResponseBody

Q44

Which of the below annotation can be used to read request inputs?

Single Choice - Select one correct answer from the options list.

@PathParam

@PathVariable

@RequestParam

@RequestData

Q45

Match Spring layers given below :

Layer list:

- 1) Spring IO foundation layer
- 2) Spring IO Coordination layer
- 3) Spring IO execution layer

Need list:

- a) Provides API to connect to cloud services
- b) provides domain specific runtime for application built using IO foundation
- c) A cohesive set of APIs and embeddable runtime components that enable to build applications.

Single Choice - Select one correct answer from the options list.

1-c

2-a

3-b

1-c

2-b

3-a

1-a

2-c

3-b

1-b

2-c

3-a

Q46

Which Spring modules usage will be more suitable for designing spring application in the below given scenarios?

Scenario 1: Developer wants to use new data access technologies, such as non-relational databases in his application.

Scenario 2: Developer wants to use an embedded server in a development environment.

Single Choice - Select one correct answer from the options list.

Scenario 1. Spring Data

Scenario 2. Spring Boot

Scenario 1. Spring Data

Scenario 2. Spring Cloud

Scenario 1. Spring XD

Scenario 2. Spring Framework

Scenario 1. Spring XD

Scenario 2. Spring Web

Q47

Pick up the valid option for the scope attribute in Spring 3.0:

```
<bean id="app " class="com.cg.payment.dto " scope=" " />
```

Single Choice - Select one correct answer from the options list.

context

prototype

httpSession

page

Q48

What is the use of @PersistenceUnit annotation?

Single Choice - Select one correct answer from the options list.

Used to persist Entity Objects

Use to fetch data from Entity Objects

Used to obtain an instance of EntityManager object and inject it into DAO specific class

Used to create an instance of EntityManagerFactory

Q49

Choose the correct statement about the given Spring code:

```
@Component  
public class Order {  
    @Autowired
```

```

private Product product;
private Customer customer;
@Autowired
public Order(Customer customer)
{
    this.customer = customer;
}

// ...
}

```

Single Choice - Select one correct answer from the options list.

We can do @AutoWired by constructor ,but not by name

We can do @AutoWired by constructor & by name

We can do @AutoWired by name & but not by constructor

We can do @AutoWired not by name & by constructor

Q50

State whether true or false:

Statement A: A handler mapping is a bean configured in the web application context that implements the HandlerMapping interface. It is responsible for returning an appropriate handler for a request.

Statement B: Handler mappings usually map a request to a handler according to the request's URL. It is an arbitrary Java object that can handle web requests.

Statement C: The most typical handler used in Spring MVC for handling web requests is a controller

Single Choice - Select one correct answer from the options list.

Statement A is false and Statement B and Statement C is true

All statements are true

Statement A is true and Statement B and Statement C is false

All statements are false

Q51

In traditional automation more knowledge is required for designing as compared to BDD.

Single Choice - Select one correct answer from the options list.

This statement is correct

This statement is wrong

This statement is irrelevant to both approach.

None of these

Q52

Which of the below is the most appropriate example of a scenario?

Single Choice - Select one correct answer from the options list.

Then some testable outcome is achieved

And something else we can check happens too

When some action by the actor

Then some testable outcome is achieved

And something else we can check happens too

k

Scenario: Some determinable business situation
Given some precondition
And some other precondition
When some action by the actor
Then some testable outcome is achieved
And something else we can check happens too

Scenario: Some determinable business situation
Then some testable outcome is achieved
And something else we can check happens too

Q53

Regarding the BDD which of the below statement is correct?

Single Choice - Select one correct answer from the options list.

Test written are more complicated

Test are written in plain descriptive english

BDD is same as TDD

None of these

Q54

Feature: Is it Friday yet?

Everybody wants to know when it's Friday

Scenario: Sunday isn't Friday

Given today is Sunday

When I ask whether it's Friday yet

Then I should be told "Nope"

Which of the below generated step file matching with the feature mentioned above?

Single Choice - Select one correct answer from the options list.

@Given("^today is Sunday\$")

public void today_is_Sunday()

{

throw new PendingException();

}

@When("^I ask whether it's Friday yet\$")

public void i_ask_whether_is_s_Friday_yet()

{

throw new PendingException();

}

@Then("^I should be told \"([^\"]*)\"\$")

public void i_should_be_told(String arg1)

{

throw new PendingException();

}

```

@Given("^today is Sunday$")
public void today_is_Sunday()
{
throw new PendingException();
}

@Then("^I should be told \"([^\"]*)\"$")
public void i_should_be_told(String arg1)
{
throw new PendingException();
}

@Given("^today is Sunday$")
public void today_is_Sunday()
{
throw new PendingException();
}

@When("^I ask whether it's Friday yet$")
public void i_ask_whether_is_s_Friday_yet()
{
throw new PendingException();
}

@When("^I ask whether it's Friday yet$")
public void i_ask_whether_is_s_Friday_yet()
{
throw new PendingException();
}

@Then("^I should be told \"([^\"]*)\"$")
public void i_should_be_told(String arg1)
{
throw new PendingException();
}

```

Q55

Which of the following Selenium Web Driver code is correct for "Reset Button" on application web page?

Single Choice - Select one correct answer from the options list.

```

driver.findElement(By.name("btnReset")).sendKeys("You pressed Ok Button");
driver.findElement(By.name("btnReset")).click();
driver.findElement(By.name("btnReset")).submit();
driver.findElement(By.name("btnReset")).reset();

```

Q56

Background in Cucumber is:

Single Choice - Select one correct answer from the options list.
used to define a condition in the feature file.

used to define a step or series of steps which are common to all the tests in the feature file.
used to define a action in the feature file.
used to define a test.

Q57

Scenario outline is:

Single Choice - Select one correct answer from the options list.

used to replace different keyword

basically replaces variable/keywords with the value from the table.

Each row in the table is considered to be a scenario.

used to add negative type comments.

used to add conditions to your steps.

Q58

Which of the following feature is NOT supported by Selenium IDE ?

Single Choice - Select one correct answer from the options list.

Test in multiple browsers

Test against Firefox only

Export recorded tests to scripts

Record and Play

Q59

In WebDriver code what is the purpose of implicit wait?

Single Choice - Select one correct answer from the options list.

There is no feature for wait.

The implicit wait will tell to the web driver to wait for certain amount of time before it throws a No Such Element Exception

The implicit wait is used to tell the Web Driver to wait for till the page gets loaded before throwing an ElementNotVisibleExceptio.

The implicit wait is used to tell the Web Driver to wait for expected condition or the maximum time exceeded before throwing an ElementNotVisibleException

Q60

Consider the following scenario:

Test Engg. wants to automate below scenario using Selenium IDE:

Open URL: <http://www.google.com/> and search for "Sachin Tendulkar"

After that he/she needs to ensure link "Sachin Tendulkar - Wikipedia, the free encyclopedia" is present on the results page

Which Selenium IDE commands would you suggest to Test Engg.?

Single Choice - Select one correct answer from the options list.

Test Engg. has to use Open,Sendkeys,waitForElementPresent,assertTextPresent commands to automate this given scenario

Test Engg. has to use get,type,VerifyText commands to automate this given scenario

Test Engg. has to use Open,type,waitForTextPresent,VerifyLink commands to automate this given scenario

Test Engg. has to use Open,type,waitForElementPresent,assertText commands to automate this given scenario

Servlets

1

Developer Tools

2

Typescript

3

Angular 2

45

Core Java

67891011121314151617181920212223242526

Oracle

2728

Web Basics

293031

Angular 6

323334

JPA with Hibernate

3536

Spring

3738394041424344454647484950

BDD

51525354555657585960