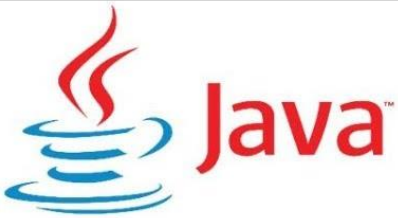
What is the difference between Assert and Verify statements in Selenium?

Assert command	Verify command
When an "assert" command fails then test execution will be aborted.	When a "verify" command fails then test will continue executing and logging the failure.
Assert is best used when the check value has to pass for the test to be able to continue to run. Like a log in.	Verify is best used to check non critical things. Like the presence of a headline element.

What is the difference between findElement() and findElements()?

	findElement	findElements
When multiple elements are present on webpage	Returns the first element	Returns all the elements
When single element is present on webpage	Returns a single element	Returns a single element
When no element is present on webpage	Throws an Exception i.e., NoSuchElementException	Doesn't throw any Exception, return zero elements
Return type	WebElement	List<WebElement>
How to access	Can be accessed directly	Iterate the list and access each item using for/foreach

What is the difference between List and Set



List Vs. Set | Java Collection Framework

List	Set
List is an ordered collection it maintains the insertion order, which means upon displaying the list content it will display the elements in the same order in which they got inserted into the list.	Set is an unordered collection, it doesn't maintain any order. There are few implementations of Set which maintains the order such as LinkedHashSet (It maintains the elements in insertion order).
Allow duplicate values.	Does not allow duplicate values. All the elements of a Set should be unique if you try to insert the duplicate element in Set it would replace the existing value.
List allows any number of null values	Set can have only a single null value at most
ListIterator can be used to traverse a List in both the directions(forward and backward)	ListIterator can not be used to traverse a Set. We can use Iterator (It works with List too) to traverse a Set.
List interface has one legacy class called Vector	Set interface does not have any legacy class.

Difference Between Method Over Loading and method Over ridding

Method Overloading	MethodOverriding
1. It occurs with in the same class. 2. Inheritance is not involved. 3.One method does not hide another. 4.Parameters must be different. 5.return type may or may not be same. 6. Access modifier & Non access modifier can also be changed.	1. It occurs between two classes i.e., Super class and a subclass. 2. Inheritance is involved. 3. child method hides that of the parent class method. 4. Parameters must be same. 5. return type must be same. 6. Access modifier should be same or increases the scope of the access modifier. Non access modifier – • **final** : if a method can contain final keyword in a parent class we cannot override. • **static**: if a method can contain static keyword child cannot override parent class methods but hide (child).

Difference between Abstraction and Interface ?

Interface	Abstract
Java interface are implicitly abstract and cannot have implementations	A Java abstract class can have instance methods that implements a default behavior
Variables declared in a Java interface is by default final	An abstract class may contain non-final variables.
Members of a Java interface are public by default	A Java abstract class can have the usual flavors of class members like private, protected, etc
Java interface should be implemented using keyword "implements"	A Java abstract class should be extended using keyword "extends"
An interface can extend another Java interface only	an abstract class can extend another Java class and implement multiple Java interfaces.
Interface is absolutely abstract and cannot be instantiated	A Java abstract class also cannot be instantiated, but can be invoked if a main() exists.
java interfaces are slow as it requires extra indirection	Comparatively fast

Difference Between Method and Constructor ?

Difference between constructor and method

Method	constructor
• Method can be executed when we explicitly call it.	• Constructor gets executed only when object is created
• Method name will not have same name as class name	• Constructor name will be same as class name
• Method should have return type	• Constructor should not have return type
• A method can be executed n number of times on a object	• Constructor will get executed only once per object

Difference between HashMap And Hash Table

HashMap	Hashtable
1) HashMap is **non synchronized**. It is not-thread safe and can't be shared between many threads without proper synchronization code.	Hashtable is **synchronized**. It is thread-safe and can be shared with many threads.
2) HashMap **allows one null key and multiple null values**.	Hashtable **doesn't allow any null key or value**.
3) HashMap is a **new class introduced in JDK 1.2**.	Hashtable is a **legacy class**.
4) HashMap is **fast**.	Hashtable is **slow**.
5) We can make the HashMap as synchronized by calling this code Map m = Collections.synchronizedMap(hashMap);	Hashtable is internally synchronized and can't be unsynchronized.
6) HashMap is **traversed by Iterator**.	Hashtable is **traversed by Enumerator and Iterator**.
7) Iterator in HashMap is **fail-fast**.	Enumerator in Hashtable is **not fail-fast**.
8) HashMap inherits **AbstractMap** class.	Hashtable inherits **Dictionary** class.

Difference Between Abstraction and Encapsulation

Abstraction	Encapsulation
Abstraction is a general concept formed by extracting common features from specific examples or the act of withdrawing or removing something **unnecessary**.	Encapsulation is the mechanism that binds together code and the data it manipulates, and keeps **both safe from outside interference** and **misuse**.
You can use abstraction using **Interface** and **Abstract** Class	You can implement encapsulation using **Access Modifiers** (Public, Protected, Private & etc.)
Abstraction solves the problem in **Design** level.	Encapsulation solves the problem in **Implementation** level.
For simplicity, abstraction means hiding implementation using Abstract class and Interface.	For simplicity, encapsulation means hiding data using gettrs and setters.

Difference Between Array List and Array

Difference between array vs ArrayList in Java

1. An array is static, you cannot change it's length once created, but ArrayList is dynamic, it can grow to accommodate more elements.
2. The array doesn't support generics, hence they are not type-safe but ArrayList support Generics, hence they provide compile time type-safety.
3. Array takes less memory than ArrayList for storing same number of elements or objects.
4. ArrayList allows you to remove element, but array doesn't provide such methods.
5. Array can accommodate both primitive and objects, but ArrayList can only accommodate objects.
6. Array can be multi-dimensional but ArrayList is always one dimensional.
7. Array provides length attribute and ArrayList provides size() but both are different, length is capacity, while size() return number of elements.

Difference Between this and Super

super	this
it represents current instance of parent class.	it represents current instance of the class.
it is used to call constructor of parent class.	it is used to call constructor of same class.
non static keyword	non static keyword.

Difference Between driver.get and driver.navigate

driver.navigate.to()	*driver.get()*
It can go forward and backwards in web browser history.	It doesn't have this feature.
It doesn't refresh while navigating to the mentioned URL.	It refreshes the webpage in process of switching to the mentioned URL.
It maintains web browser history.	It doesn't maintain web browser history.

Difference Between driver.close and driver quit

- `driver.quit()`: The `quit()` method quits the driver, closing every associated window.
- `driver.close()`: The `close()` method closes the currently focused window, quitting the driver if the current window is the only open window. If there are no windows open, it will error out.

Difference Between string , string buffer and string builder

Index	*String*	*String Buffer*	*String Builder*
Storage Area	Constant String Pool	Heap	Heap
Modifiable	No (immutable	Yes(mutable)	Yes(mutable)
Thread Safe	Yes	Yes	No
Thread Safe	Fast	Very slow	Fast

Difference between Throw and Throws

Throw vs Throws

Throw	Throws
Java throw keyword is used to explicitly throw an exception.	Java throws keyword is used to declare an exception.
Checked exception cannot be propagated using throw only.	Checked exception can be propagated with throws.
Throw is followed by an instance.	Throws is followed by class.
Throw is used within the method.	Throws is used with the method signature.
You cannot throw multiple exceptions.	You can declare multiple exceptions e.g. public void method()throws IOException,SQLException.

Differences between final , finally and Finalized

final	**Finally**	**finalize**
final is a keyword in java final in java can be used with : • **variable** - If you mark a variable as final then once you initialize value to that variable, it can never be changed. • **method** - If you mark a method as *final* in your class then that method cannot be overridden by the sub-classes method. • **class** - If you mark a class as final in java then that class cannot be extended (or sub-classsed)	finally is a block in java finally block is used with try block in exception handling. This block will surely get executed irrespective of whether exception is handled or not.	finalize is a method in java finalize() method is called at least once, before the object is garbage collected, to perform any cleanup activity like releasing any system resources if held or closing the open connection.

final vs finally vs finalize

What is the difference between List, Set and Map ?

Ans. List - Members are stored in sequence in memory and can be accessed through index.
Set - There is no relevance of sequence and index. Sets doesn't contain duplicates whereas multiset can have duplicates. Map - Contains Key , Value pairs.

Difference between Public, Private, Default and Protected ?

Ans. Private - Not accessible outside object scope.
Public - Accessible from anywhere.
Default - Accessible from anywhere within same package.
Protected - Accessible from object and the sub class objects.

Difference between == and .equals() ?

Attribute	"=="	. equals ()
Type	Java binary Operator	Java Method of Object class
Compare logic	Compare variable and object references – memory location in Java heap	Compare the state or Content of objects – as per the logic in equals method
Override behaviour	Behaviour cannot be overridden	.equals() method can be overridden as it is java method. (String class cannot be extended as it is immutable class)
String Object Comparison	True, if objects are referring to same address	True- if objects hold the same String value.
Primitives	use "==" operator of equality check	Cannot use
Objects	True, if Objects pointed to same memory location.	Preferred to use equals() method . Ability to change the compare logic for equality
Objects created by new key word	Always return false though content are same.	True -if contents are same False if contents are not same

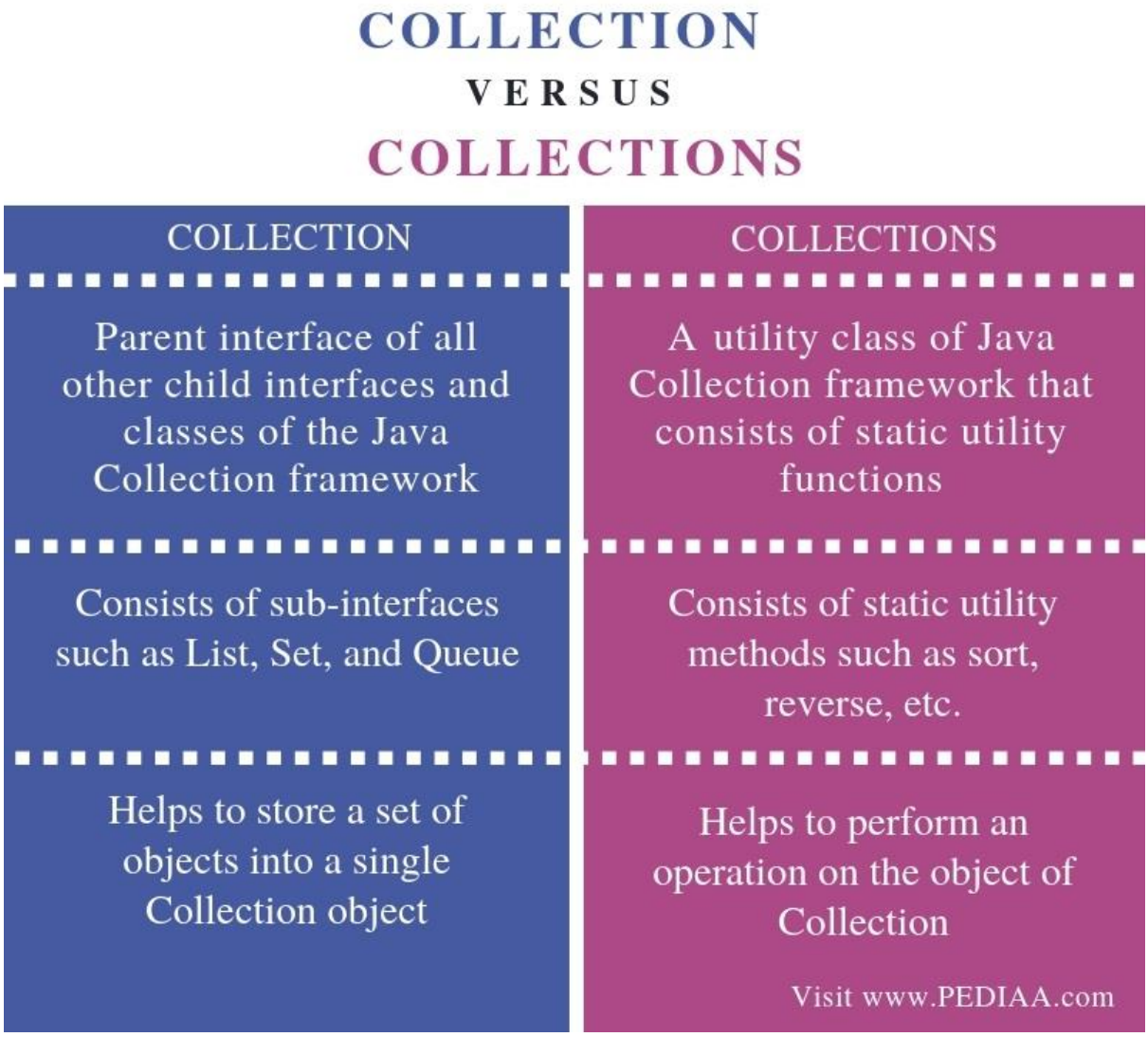
Difference between Checked and Unchecked exceptions ?

Unchecked Exception	Checked Exception
1) All the subclasses of RuntimeException are called unchecked exception.	All subclasses of Throwable class except RuntimeException are called as checked exceptions
2) Unchecked exceptions need not be handled at compile time	Checked Exceptions need to be handled at compile time.
3) These exceptions arise mostly due to coding mistakes in our program.	
4) ArrayIndexOutOfBoundsException, ClassCastException, IndexOutOfBoundException	SqlException, FileNotFoundException,ClassNotFoundException

What is the difference between yield() and sleep()?

Ans. When a object invokes yield() it returns to ready state. But when an object invokes sleep() method enters to not ready state.

what is the difference between collections class vs collections interface ?



What is the difference between comparable and comparator in java.util pkg?

Comparable interface	Comparator interface
Comparable interface present in java.lang package.	Comparator interface present in java.util package.
Sort the elements according to natural sorting order.	Sort the elements according to customized sorting order.
It contains only one method i.e. compareTo()	It contains two methods compare() and equals()
CompareTo() method is responsible to sort the elements.	compare() method is responsible to sort the elements

Difference between Implicit wait and explicit wait?

Implicit Wait	Explicit Wait
Implicit Wait time is applied to all the elements in the script	Explicit Wait time is applied only to those elements based on the conditions
In Implicit Wait, we need not specify "ExpectedConditions" on the element to be located	In Explicit Wait, we need to specify "ExpectedConditions" on the element to be located
It is recommended to use when the elements are located with the time frame specified in implicit wait. E.g. For Small Application	It is recommended to use when the elements are taking long time to load and also for verifying the property of the element like(visibilityOfElementLocated, elementToBeClickable,elementToBeSelected)

Difference between Ambulate and Relative Xpath

Absolute XPath	Relative XPath
It contains the complete path from the Root Element to the desired element.	It contains the path from a reference element to the desired element.
It starts from /	It starts from //
Absolute XPaths are prone to more regression as slight change in DOM makes them invalid or refer to a wrong element	Relative XPath is not altered if root element is changed/updated

elementToBeClickable(By locator)

An expectation for checking an element is visible and enabled such that you can click it.

presenceOfElementLocated(By locator)

An expectation for checking that an element is present on the DOM of a page.

visibilityOf(WebElement element)

An expectation for checking that an element, known to be present on the DOM of a page, is visible.

Difference between X path and CSS Selector

Xpath traverse in both direction , forward and Reverse Direction

But CSS Selector : Traverse in forward direction only