**Java Interview Questions For Selenium WebDriver - Part 1**

**PART 1**

**1** : **What is object in java?**

**Answer** : Object Is an Instance of class and it has its own state and behavior. In real world we can say, Dog is object of Animal class which have different state like breed, color, name, hungry, etc and behavior  like wagging tail, fetching, barking etc.

**2** : **What is class?**

**Answer** : A class is the blueprint or we can say template from which individual objects are created.

**3** : **What is constructor?**

**Answer** : Constructor is a code block just like a method which is used to initialize the state of an object. It will be invoked at the time of object creation to construct the value for object. [**VIEW DETAIL**](http://www.software-testing-tutorials-automation.com/2014/04/selenium-webdriver-java-tutorials_28.html)

**4** : **What is default constructor?**

**Answer** : If there is not any constructor in class then java compiler creates default constructor. It is no argument constructor which initializes any uninitialized fields to their default values.

**5** : **Constructor returns any value?**

**Answer** : Yes. It will return instance of current class (However we can not use return type with constructor).

**Interview Questions Of Core Java For Selenium WebDriver - Part 2**

**Part 2**

**6** : **What is the difference between static and not static variable?**

**Answer** : Main differences are as bellow.

* Static variables are preceded by static keyword. For non-static variable, there is not any preceding keyword.
* Memory is allocated for static variables at the time of class loading. Memory is allocated to non- static variables whenever an object is created.
* Memory is allocated only once to static variables on class loading. Memory is allocated multiple time whenever a new object is created to non-static variables.
* Static variable example : Collage name of students, Company name of employees..

[**READ MORE**](http://www.software-testing-tutorials-automation.com/2014/04/java-tutorials-for-selenium-webdriver.html) about static and non-static stuff.

**7** : **What is the difference between static and not static(Instance) method?**

**Answer** : Difference between static and non static method is as bellow.

* Method declared with static keyword is static method. If Method declared without static keyword then it is instance method.
* No need of object to call static methods. Object needed to call instance method.
* Can not access non static stuff inside static methods directly. Opposite to it, We can access static and non static stuff directly inside instance method.

[**READ MORE**](http://www.software-testing-tutorials-automation.com/2014/04/java-tutorials-for-selenium-webdriver.html) detail on static and non-static stuff.

**8** : **What is inheritance in java?**

**Answer** : In Java, Inheritance provides mechanism using which one object of child class can acquire  all the properties and behaviors of parent object. It will crate IS-A relationship. Main usage of inheritance in java is for code re-usability and  method overriding to achieve run-time polymorphism. [**VIEW MORE**](http://www.software-testing-tutorials-automation.com/2014/04/inheritance-in-java-tutorials-for.html) detail on inheritance.

**9** : **Multiple inheritance is supported in java on class level? If No.. Why?**

**Answer** : No.. Multiple inheritance is not supported in java in case of class to simplify the language and reduce the complexity.

**10** : **What is method overriding in java?**

**Answer** : Method overriding is a feature which allows a child class or sub class to provide a specific implementation of a method which is already provided by one of its parent classes or super class. It is used for runtime polymorphism. You can read more about method overriding on [**THIS POST**](http://www.software-testing-tutorials-automation.com/2014/04/inheritance-in-java-tutorials-for.html).

**Core Java Interview Questions For Selenium WebDriver - Part 3**

**Part 3**

**11** : **Why main method is static?**

**Answer** : As we know, We can access static stuff without creating object of class. Because of static keyword with main method, Java virtual machine can directly call it without creating object of class. This way it will provide kind of root to start execution of program.

**12** : **What is method overloading?**

**Answer** : Method overloading is ability to create multiple methods with same in same class but with different signatures (different input parameters and types). Method names will be same but parameters will be different for all overloaded methods.

**13** : **What is constructor overloading?**

**Answer** : Same as method overloading, Single class can have multiple constructors with same name as class name but all have different signatures (different input parameters and types) is called constructor overloading. [**READ MORE**](http://www.software-testing-tutorials-automation.com/2014/04/selenium-webdriver-java-tutorials_28.html) about constructor overloading.

**14** : **Can we override static methods?**

**Answer** : We can declare static method with same signature in subclass but it will not behave as overridden method. So answer is No.. We can not override static methods as they are part of  class not object.. You can override static methods but output will be different than the expected.

**15** : **How to reverse string in java?**

**Answer** : StringBuffer class has a method called reverse(). We can use it to reverse the string.  
Example :

public static void main(String[] args) {

**// buffer string using StringBuffer class.**

StringBuffer a = new StringBuffer("I like java very much.");

**// use reverse() method to reverse string**

System.out.println(a.reverse());

}

**Frequently Asked Java Interview Questions Part - 4**

**Part 4**

**16** : **Can we overload static methods?**

**Answer** : Yes.. There is not any restriction to overload static methods. We can overload static and non static methods in java. [**VIEW MORE**](http://www.software-testing-tutorials-automation.com/2015/08/what-is-method-overloading-in-java.html) on overloading in java.  
  
**17** : **Can we use private member of parent class in sub class?**

**Answer** : No.. It will not allow to use private members like private method, variable of parent class in child class. Private members are accessible only inside same class. [**VIEW MORE**](http://www.software-testing-tutorials-automation.com/2014/04/access-modifiers-in-java-java-tutorials.html) about class modifiers.

**18** : **What is an interface in java?**

**Answer** : An interface is a blue print of a class which can hold abstract methods (Methods without implementation) only. It creates Rules To Follow structure for class where It Is Implemented. We can achieve 100% abstraction using interface in java. [**READ MORE**](http://www.software-testing-tutorials-automation.com/2014/04/interface-in-java-tutorials-for.html) about interface in java.

**19**: **Can we access protected method of parent class in sub class?**

**Answer** : Yes.. We can access protected members of parent class in all it's sub classes and classes within the same package. [**VIEW EXAMPLE**](http://www.software-testing-tutorials-automation.com/2014/04/inheritance-in-java-tutorials-for.html) on how to access protected method in sub class.

**20** : **What is an array in java?**

**Answer** : An array is container object in java which can hold fixed number of values of same type. [**VIEW ARTICLE**](http://www.software-testing-tutorials-automation.com/2014/04/arrays-basic-java-tutorials-for.html) on array.

**Java Questions For Selenium WebDriver Interview**

**Part 5**

**21** : **Explain System.out.println();**

**Answer** :

* **System** : is a final class in  java.lang package.
* **out**: is a static member of system class. It is an instance of java.io.PrintStream. This stream is already open and ready to accept output data.
* **println**: is a method of java.io.PrintStream .It is an overloaded method.

**22** : **Write program to print fibonacci series 0, 1, 1, 2, 3, 5, 8, 13, 21,...**

**Answer** :

public class fibonaccci {

public static void main(String[] args) {

int f1 = 0;

int f2 = 1;

int sum = 0;

for(int i=0; i<=21;){

System.out.println(f1);

sum = i+f2;

f2=i;

f1=sum;

i=f1;

}

}

}

**23** : **Write program to get result of 52+42-32+22-12**

**Answer** :

public class SquareSum {

public static void main(String[] args) {

int sum = 0;

for (int i=5;i>=1;i--){

if(i%2!=0){

if(sum<(i\*i)){

sum = (i\*i)-sum;

}else{

sum = sum-(i\*i);

}

}else{

sum = sum+(i\*i);

}

}

System.out.println(sum);

}

}

**24** : **What is return type of testng @DataProvider annotation method?**

**Answer** : It will return double object array “Object[][]”.

**25** : **int x=10 and y=20. Swap both variable values without using any temp variable.**

**Answer** :

public class swapNumbers {

public static void main(String[] args) {

int x = 10;

int y = 20;

System.out.println("Before swapping x = " + x + " and y = " + y);

x = x + y;

y = x - y;

x = x - y;

System.out.println("After swapping x = " + x + " and y = " + y);

}

}

**Interview Questions on Java For Selenium**

**Part 6**

**26** : What is local variable in java?  
**Answer** : Local variable is declared inside method or constructor and it is limited for that method or constructor only. View [**more detail**](http://www.software-testing-tutorials-automation.com/2014/04/variable-types-in-java-webdriver.html) on local variable in java.  
  
**Local Variable Example**:

public class JavaVariable {

public void Calc() {

**// Local Variables.**

int sum;

int item1 = 5;

int item2 = 7;

sum = item1 + item2;

System.out.println("Sum is : " + sum);

}

public static void main(String args[]) {

JavaVariable j = new JavaVariable();

j.Calc();

}

}

**27** : What is a Instance Variable in java?  
**Answer**: Instance Variable is declared parallel to method or constructor in class. It is visible for all methods and constructors of that class. View [**more detail**](http://www.software-testing-tutorials-automation.com/2014/04/variable-types-in-java-webdriver.html) on instance variable in java.  
  
**Instance Variable Example**:

public class JavaVariable {

**//Instance Variable.**

int sum;

public void Calc() {

**// Local Variables.**

int item1 = 5;

int item2 = 7;

sum = item1 + item2;

System.out.println("Sum is : " + sum);

}

public static void main(String args[]) {

JavaVariable j = new JavaVariable();

j.Calc();

}

**28** : What is a Class Variable in java?  
**Answer**: Class variable is declared with static keyword in class parallel to methods and constructor. Class variable is initialized only once at the start of execution and destroyed on end of program. Class variable is also known as static variable. View [**more detail**](http://www.software-testing-tutorials-automation.com/2014/04/variable-types-in-java-webdriver.html) on class variable in java.  
  
**Class Variable Example**:

public class JavaVariable {

**//Class Variable**

static String sumType = "Basket";

**//Instance Variable.**

int sum;

public void Calc() {

**// Local Variables.**

int item1 = 5;

int item2 = 7;

sum = item1 + item2;

System.out.println(sumType + " Sum is : " + sum);

}

public static void main(String args[]) {

JavaVariable j = new JavaVariable();

j.Calc();

}

**29** : What is the difference between instance variable and class variable?  
**Answer**:  
  
**Instance Variable** :

* It is unique to each instance of the class.
* Declared without static modifier.
* Memory allocation, loading and initialization is done at run time.

**Static Variable** :

* It is shared by all instances of the class.
* Declared with static modifier.
* Memory allocation is done at compile time, loaded at load time and they are initialized at class initialization time.

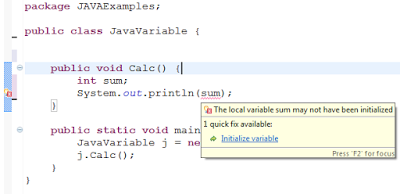
**30**. What is access modifier in java?  
**Answer**: Access modifiers allows us to set access levels for variables, methods, classes and constructors in java. We can control access levels using access modifiers in java. View [**more details**](http://www.software-testing-tutorials-automation.com/2014/04/access-modifiers-in-java-java-tutorials.html) on different access modifiers in java.

**Java Programming Interview Questions**

**Part 7**

**31** : What is default value of local variable.

**Answer** : There is not any default value of local variable. You must have to initialize it. View **more details** on local variables in java.

[](http://1.bp.blogspot.com/-JdGIxtw1p1Q/V29euRpCctI/AAAAAAAADzE/_QhlaJrjieIrTd1rS3zT_hcBVN2M7_ILgCK4B/s1600/default%2Bvalue%2Bof%2Blocal%2Bvariable.png)

**32** : Java support constructor inheritance?

**Answer** : No, Constructor inheritance is not supported in java. View **more details** on constructor in java.

**33** : Which is super class of all other classes in java?

**Answer** : java.lang.Object is  super class of all other classes in java.

**34** : What is Encapsulation?

**Answer** : Encapsulation is process of packing code and data together In to a single Unit. View [**more details**](http://www.software-testing-tutorials-automation.com/2015/07/what-is-encapsulation-in-java.html) on Encapsulation in java.

**35** : Write a program to reverse a string without using reverse function.

**Answer** : Program to reverse a string without using reverse function in java is as bellow.

package JAVAExamples;

public class StringReverse {

public static void main(String[] args) {

**//String to reverse.**

String str = "This Is String.";

String revstring = "";

for (int i = str.length() - 1; i >= 0; --i) {

**//Start getting characters from end of the string.**

revstring += str.charAt(i);

}

System.out.println(revstring);

}

}

Output : .gnirtS sI sihT

**Java Questions For Selenium Interview**

**Part 8**

**36** : What is the difference between the Constructor and Method?  
**Answer**: Main difference between the Constructor and Method is as bellow.  
  
**Constructor**:

1. Name of the constructor must be same as class name.
2. Constructor must not have any return type.
3. It is used to initialize the state of an object.
4. It is not possible to call constructor directly. Constructors called implicitly when the new keyword creates an object.

**Method**:

1. Method name can be any.
2. Method must have return type.
3. It is used to expose behavior of an object.
4. Methods can be called directly.

Read more on [**Constructor**](http://www.software-testing-tutorials-automation.com/2014/04/selenium-webdriver-java-tutorials_28.html) and [**Method**](http://www.software-testing-tutorials-automation.com/2014/04/methods-in-java-tutorials-for-selenium.html).

**37**: What are the different types of types of constructors in java?  
**Answer**: Mainly there are two types of constructors available in java.

* Default Constructor : Constructor without parameter is called default constructor.

package JAVAExamples;

public class City {

**//Default Constructor**

City()

{

System.out.println("City is created");

}

public static void main(String args[]){

City c=new City();

}

}

* Parameterized constructor : Constructor with parameter is called Parameterized constructor.

package JAVAExamples;

public class City {

int id;

String name;

**// parameterized Constructor**

City(int i, String n) {

id = i;

name = n;

}

void display() {

System.out.println(id + " " + name);

}

public static void main(String args[]) {

City c1 = new City(1, "New York");

City c2 = new City(2, "London");

c1.display();

c2.display();

}

}

Output :  
1 New York  
2 London  
  
**38** : Write a program for Fibonacci series in Java ?  
**Answer**: Program for Fibonacci series is as bellow.

package JAVAExamples;

public class FibonacciSeries {

public static void main(String args[]) {

int x1 = 0, x2 = 1, x3, i, cnt = 15;

**// To print 0 and 1**

System.out.print(x1 + " " + x2);

**// loop starts from 2 as 0 and 1 are already printed.**

for (i = 2; i < cnt; ++i) {

x3 = x1 + x2;

System.out.print(" " + x3);

x1 = x2;

x2 = x3;

}

}

}

**39**: Write a program to print below given pattern.

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

**Answer**: Program to print above pattern is as bellow.

package JAVAExamples;

public class Pattern {

public static void main(String[] args) {

for (int a = 1; a <= 5; a++) {

for (int x = 1; x <= a; x++) {

System.out.print(x+" ");

}

**// To print new line.**

System.out.println();

}

}

}

**40**. What is the difference between “this” and “super” keywords in Java?  
**Answer**: Difference is as bellow.

1. “this” keyword is used to store current object reference while “super” keyword is used to store super class object in sub class..
2. “this” is used to access methods of the current class while “super” is used to access methods of the base class.
3. this() used to call constructors in the same class whereas super() is used to call super class constructor.

**Java Questions And Answers For Selenium**

**Part 9**

**41** : In java, What is return type of main method?

**Answer**: Main method doesn't have any return type. It is void.

**42** : Can We Overload main method in java?

**Answer**: Yes, Java class can have any number of main methods so it is possible to overload main method. But when you run program, It will not execute overloaded main method. Always It will execute only public static void main(String[] args) method.

**43** : Can we declare class as protected?

**Answer**: No, You can not declare class as protected.

**44** : Write a program to remove given character from string.

**Answer**: Program to remove given character from string is as bellow.

package JAVAExamples;

public class RemoveChar {

public static String removeChar(String str, char c) {

if (str == null)

return null;

return str.replaceAll(Character.toString(c), "");

}

public static void main(String[] args) {

System.out.println(removeChar("chicago", 'c'));

}

}

**45** : How to convert string from upper to lower and lower to upper case?

**Answer** : You can use toUpperCase and toLowerCase methods to convert string from lower to upper and upper to lower case.

**Interview Questions In Java For Selenium**

**Part 10**

**46**: What is Polymorphism?  
**Answer**: Polymorphism is ability using which we can create reference variables or methods which behaves differently in different programmatic context. Best example of polymorphism is human. We behaves differently with different people in different environment. Our behavior will be different when we meet to boss and meet to friend. [**Read more**](http://www.software-testing-tutorials-automation.com/2015/08/what-is-polymorphism-in-java-oop.html) on Polymorphism in java.  
  
**47** : What is the advantages of Polymorphism?  
**Answer**: Main advantage of polymorphism is code reusabilty. You can dynamically supply different implementations through polymorphism. So it will reduce your work volume in terms of handling and distinguishing various objects.  
  
**48**: What is a package?  
**Answer**: A package is a namespace which allows developer to organizes a group of related classes and interfaces. Conceptually it is just like folder which contains different types of files. It is easy to keep things organised by keeping related classes and interfaces into packages.  
  
**49**: What is string in java?  
**Answer**: In Java programming, String is object which is prepared by sequence of characters.java.lang package has String class to create and manipulate strings. [**Read more**](http://www.software-testing-tutorials-automation.com/2014/05/string-in-java-tutorials-for-webdriver.html) on string in java.  
  
**50**: What is StringBuffer in java?  
**Answer**: StringBuffer help us to create mutable(modifiable) string in java. That means we can modify the string if we use StringBuffer.

**Q: What is JRE and why is it required?**

JRE stands for "Java Runtime Environment" which you usually download as a Java software. The JRE comprises of the Java Virtual Machine, Java platform classes, and supporting libraries. The JRE is the runtime component of Java software and is all you need to run any Java application.

**Q: What is JDK and why is it required?**

The JDK is a superset of the JRE and includes everything that the JRE contains. Additionally, it comes with the compilers and debuggers tools required for developing Java applications.

**Q: What is JVM and why is it required?**

JVM stands for The Java Virtual machine. It translates and executes the Java bytecode. It's the entity which transform Java to become a "portable language" (i.e. write once, run anywhere). Though, each platform has its implementation of JVM like the Windows, Linux, MacOS, etc. have a distinct version of JVM to run bytecode.

**Q: Distinguish between the Path and Classpath?**

The <Path> and <Classpath> are OS level environment variables. Path defines the location where the system can look up for the executables (.exe) files, and classpath specifies the location of the Java class files.

**Q: Distinguish between a constructor and method?**

A constructor gets automatically invoked to create an object whereas the method gets called explicitly.

**Q: Is it permissible for a constructor to have a different name than its class name in Java?**

No, constructors in Java should have the same name as their classes. If the name is different, then it would behave like a standard method.

**Q: Is there any difference between an argument and a parameter?**

While defining methods, you pass variables which you refer as parameters. And when you call these methods and pass values for the variables then they are phrased as arguments.

**Q: How your program would behave if you declare the main method as private?**

It would get compiled correctly but will throw the error "Main method not public." at runtime.

**Q: What if an application get multiple classes having main() methods?**

It's certainly possible to have multiple main methods in different classes. When you start the application, you've to provide the startup class name for execution. The JVM then looks up for the main method only in the class whose name you've supplied. Hence, you won't observe any conflict with the multiple classes having the <main()> definition.

**Q: What difference you see between pass by reference and pass by value in Java?**

Pass by reference indicates, passing the address itself rather than passing the value. Pass by value means is giving a copy of the value.

**Q: What do you understand by Byte Code?**

Java compiler generates bytecode for all the Java code and converts into class files. The bytecode is platform independent and needs the platform-specific JVM for the execution.

**Q: What do you make of each keyword in public static void main(String args[])?**

* Public- <main(..)> is the entry point method which the JVM calls when a program is run. So it's mandatory for it to get accessible from the Java environment. Hence, the access specifier has to be public.
* Static- JVM must be capable of calling this method w/o creating an instance of the class. So the method has to be declared as static.
* Void- <main()> doesn't return anything so it's return type must be void.
* The argument string represents the argument type passed from the console, and the <args> is an array of strings specified at the command line.

**Q: How do compare the final, finally and finalize keywords?**

* final– It's used to declare a constant.
  + Variables defined in an interface are implicitly final.
  + You can't extend a final class.
* finally– It makes you handle exceptions.
  + It's a keyword used for exception handling. The code under the <finally> block gets executed apparently.
* finalize– It helps in garbage collection.
  + The <finalize()> method is used just before an object is destroyed and garbage collected.

**Q: Can you compile a Java class successfully without having the main method?**

Yes, we can compile, but it won't run. The <main> method works as the startup function for a Java class, and the JVM calls it for the program execution.

**Q: What do you make of System, out and <println> in the function System.out.println()?**

* System -> A predefined final class,
* out -> PrintStream object and,
* The <println> -> built-in overloaded method of the out object.

**Q: What do you understand by the explicit casting?**

It's a process which instructs the complier about transforming the object into a different type.

**e.g.** *long no = 99999;*  
*int new\_no = (int) no; // Explicit casting*

**Q: Would a Java program compile/run if we use <static public void> instead of <public static void>?**

Yes, the program will compile and run as usual.

**Q: How would you prove that an array is not null but is empty?**

Call the <Print array.length>. It will print 0. That suggests that the array is empty. If it would've been null then, it would've thrown a <*NullPointerException*> on calling the <Print array.length>.

**Q: What do you understand of Garbage Collection and how to call it explicitly?**

If the object is no longer belong to any variable, Java automatically reclaims the memory. This process is known as garbage collection. You can use the <System.gc()> method to call it explicitly.

**Q: How comes an unreachable object become reachable again, is it at all possible?**

Yes, an unreachable object may get to reachable state. It can happen if the object <finalize> method gets called during the garbage collection, and there you have set an object making a reference to it. This situation would cause the garbage collection to skip and make the object reachable again.