<https://www.tutorialspoint.com/questions_and_answers.htm---> : for all questions

Class – A template/model/blueprint that describes the kinds of states and behaviour that objects of its type support.

**Constructor**: that is invoked always whenever a new object for the class is created.

Every class has atleast one constructot by default i.e, no args constructor.

It is similar to method syntax but it do not have a return type.

Always the constructor name should be same as your class name.

Constructors can be overloaded just like methods.

Constructor can have any access modifiers-public, private, protected, default.

Within constructor if your are using this() or super() – They should be first statements(the very first line) in constructor.

Constructors can be used to initialise the instance variables at the time of object creation itself.

this keyword in java – this is used to refer the member(instance variables,methods) of current object from method or constructor.

It can be used inside a method or a constructor of the class.

Primitive data types casting:

byte , short , int , long , float, double

left🡪right **upcasting** or **implicit** casting

right->left **downcasting** or **explicit** casting

**static** : key word in java

It is class level but not any object specific

Static can be used for variables, methods and blocks.

Every object has its own copy of instance variables but static variables are not specific to any object, they are available at class level rather than instance level.

Static variables will be shared across your objects, instance variables are specific to object.

**final:**

final is applicable to variables, methods and classes

class is final- no other class can extend this final class

variables as final – we cannot reassign/change the values of variable – constant values

methods as final – nobody can override the method

**OOPS Concepts:**

**Encapsulation** : hide implementation behind an interface.

Encapsulated code will have 2 things:

Instance variables are protected(we can use private modifier and protect it)

We provide getters and setters to access the instance variables.

**Inheritance**( IS A relation) : Inheritance allows a class to be subclass(child class)of a superclass(parentclass) – therefore subclasses inherit public and protected variables and methods of the superclass.

Car- Benz is a car, BMW is a car

**Polymorphism** – exisiting in many forms….

A singe refrence variables can take its own type of object and also its subtype object – through inheritance

A single method can be given different behaviours(different logics) by method overriding – through inheritance

Strictly speaking – polymorphism is applicable to overriding but not overloading.

**Method overriding** – redefining the superclass method logic in subclass

Overriding is only applicable w.r.to inheritance….

Exceptions- checked(compile time)- IOException – FNF exception, unchecked(run time)- nullpointer exception, numberformat exception

|  |  |
| --- | --- |
| Overloading | Overriding |
| Method arguments **must** change-either by number of args or by datatype | Method arguments **must not** change |
| Return type can be changed | Cannot change the return type except the covariant return types |
| Exceptions declared in method signature can change | Cannot change exceptions of super class in subclass method overriding-u can still add any unchecked (runtime)exceptions,narrower checked exxcpetions, but u cannot throw new or broader checked exceptions |
| Access modifiers can change | Cannot make **more restrictive** acces modifier or we cannot reduce the visibility… |
| It decides which method to call at compile time- it exhibits virtual method invocation at run time | At run time based on which object is getting(subclass or superclass) created it calls that particular method, but at compile time it just decides on object reference – so we need to make sure that the refrenec objet has the method u r calling at compile time. |

**AbstractClass**:

**When I need abstract class** - If we have some common functionality to be provided to sub classes and we want to declare some abstract methods which my child class should definitely implement(I am creating rules that child class should definitely follow)…

Abstract class – is a normal java class with abstract methods along with concrete methods. So this class acts as parent class.

Abstract methods – do not have method body, they only have method declarations.

Concrete methods – have method body/implementation.

Abstract class can use any of the access modifiers including private(concrete methods)…but still abstract methods cannot be private.

The first child class which is not abstract should defintely implement all my abstract methods.

Abstract class can have constructor.. but we cannot create the object for abstract class, only way the constructor is called is when the child class object is created.

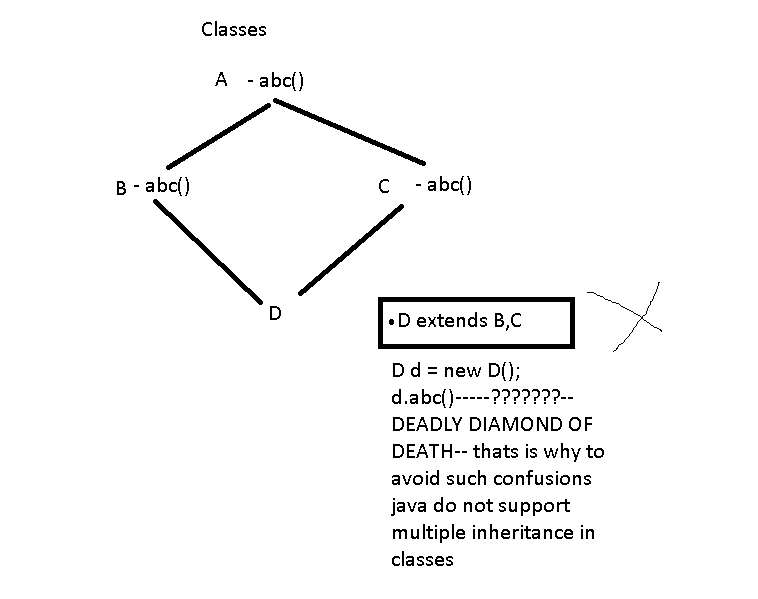
**Interface**: Interface is a contract which defines what a class should do.

It is 100% abstract class – bcz no concrete methods …

By default all the interface methods are public and abstract, we cannot have any concrete methods in interface like abstract class

By default all the variables in interface are constants- public, static and final

Interface do not have constructors.



**When to choose abstract class and when to choose interface:**

**When we have some common functionality to share to all child classes and we want some abstract methods also which child class must implement – we choose abstract classes.**

**Interface- When we have to define only methods to implement(contract) and no need of any concrete methods, also if we need multiple inheritance – we choose interface.**

**Also we choose interface for defining Constants.**

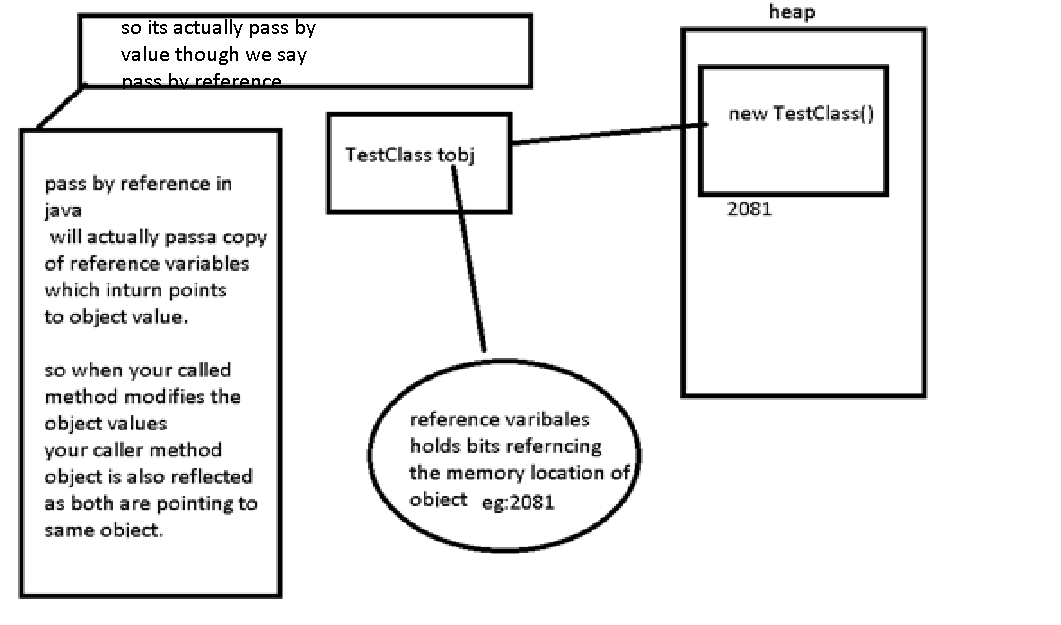
**Interface reference variable can take any child class or implementation class object, this is same with abstract class also – its just like parent-child relation ship.**

**super: is a keyword by which we can access parent class members(methds,variables,constructors) in child class.**

**Pass by value/pass by reference:**

**Primitives refer to actual value in memory location**

**ObjectsRefernce refer to memory adress of actual object.**

****

**String** – String is a class in java.lang package , it is under Object hierarchy like all other classes in java.

String is immutable in java.

String class is declared final in java.

String is set of 16 bit unicode characters….

String is set of char array data

String can be created using literals or using new Object

String can be compared using == and equals but it is always good to use equals method bcz == works only for your string literals as == compares references but not the actual value.

equals compares actual value but not the reference unlike ==

String literals are created in string constant pool – if there is exisiting same value then new reference points to same value instead of creating new object again unlike string objects creation

String objects are create in heap memory

Keeping efficient memory managment in mind- java people created string constant pool mechanism.

**Why string immuatble?**

Keeping stringconstant pool in mind they made string immutable.

Other reasons are for security, cache etc…

How u can make any class immutable in java?

By making class final and there are some other rules…

■ charAt() Returns the character located at the specified index

■ concat() Appends one String to the end of another ( "+" also works)

■ equalsIgnoreCase() Determines the equality of two Strings, ignoring case

■ length() Returns the number of characters in a String

■ replace() Replaces occurrences of a character with a new character

■ substring() Returns a part of a String

■ toLowerCase() Returns a String with uppercase characters converted

■ toString() Returns the value of a String

■ toUpperCase() Returns a String with lowercase characters converted

■ trim() Removes whitespace from the ends of a String

**StringBuffer and StringBuilder:**

These classes are mutable unlike String class.

**StringBuffer** methods are synchronized , they are thread safe.

**StringBuilder** methods are not synchronized, they are faster.

StringBuffer st = new StringBuffer(“hello”);

st.append(“world”);

StringBuilder st = new StringBuilder(“hello”);

st.append(“world”);

eg: when dealing with file data appending and whenever we have to modify string frequently we prefer stringbuffer and stringbuilder classes.

append

delete

insert

reverse

**JAVA MOCK:**

1.Benefits of OOPS?

1.Reusability

2.Extensibility

3.ReImplementation

4.Modularity

5.Security purpose

2. What is an Exception?

Ans.Run time error is called as an Exception.

Error : which comes at the time of development

Bug : which comes at the time of testing / pre release

Defect : which comes at the time of production / post release

3. Why Exception Handling Mechanism?

to handle the run time weeor...to avoid abnormal termination.

4. What code we will write with in try,Catch and Finally blocks ?

try block :while developing the prog write one statement which may throw an error

catch block : we have to write the statements to display user friendly messages

finally block : error free or cleaning up code , code that is irrespective of the cleanup code

5.what is thread?

is an independent execution path ...able to run with other execution paths.

6.what is the baseclass library for the threading ?

system.threading is the base class library

7.what are oops principles ?

Encapsulation

data abstraction

polymorphism

inheritance

8.What is Encalsulation ?

Wrapping states and behaviours...by implementing the class we can achieve encapsulation.

We do this getters and setters private variables can be accesed other classes using

9.what is polymorphism? types of polymorphism ?

implementing mutiple functinalities with the same name : two types

1.static polymorphism

2.dynamic polymorphism

10.what are static and dynamic polymorphisms.

static : method which bind at complie time and which will execute at runtime or early binding time binding , compile time polymorphisms.

Dynamic : method which will bind at compile time will not execute instead of that it will be binded andexecuted at runtime is called dynamic polymorphism, late binding,or runtime polymorphism.

Java Mock:2

1.What is Encapsulation?

2.Explain This keyword ?

**this** keyword in java – this is used to refer the member(instance variables,methods) of current object from method or constructor.

It can be used inside a method or a constructor of the class.

this is a reference to current object of execution.

3.diff between == equals

In Java, when the “==” operator is used to compare 2 objects, it checks to see if the objects refer to the same place in memory. In other words, it checks to see if the 2 object names are basically references to the same memory location.

 compare 2 String objects, then as long as the actual sequence of characters is equal, both objects are considered equa

equals method is actually meant to compare the contents of 2 objects, and not their location in memory.

4.Diff between overloading and overriding

Overloading – having same method name with different input parameter

Within the same class , diff no. of arguments or diff type of arguments.

Overloading is little fast compare to overriding.overloading it decides at compile time.

Overriding : with inheritance …parent class method can be overrided in child class.

5.What is inheritance?

6.write a program to reverse the string

7.swap two variables without using temp variables ?

8.how to find duplicate numbers?

9.

MOCK 3 :

1.Define static and final in java?

2.diff between abstract and Interface?

3.what are multi level inheritance and multiple inheritance ?

4.write for multiplication table

5.sorting for string?

6.Armsstrong 153 = 1\*\*3+5\*\*3+3\*\*3

1. What is the difference between Arrays and Collections?
2. What is an Object in java and what are thw different object methods in java?
3. What is an Iterator and what is an listIterator?
4. What are fail-safe and fail-fast iterators?Fail-Fast iterators immediately throw ConcurrentModificationException if a collection is modified while iterating over it. Where as Fail-Safe iterators don’t throw any exceptions if a collection is modified while iterating over it. Because, they operate on the clone of the collection, not on the actual collection.
5. What is the queue interface and methods in queue interface?
6. What is linkedlist and its different methods?

### What is a legacy classe? what are different legacy classes? Legacy Classes

Early version of java did not include the Collections framework. It only defined several classes and interfaces that provide methods for storing objects. When Collections framework were added in J2SE 1.2, the original classes were reengineered to support the collection interface. These classes are also known as Legacy classes. All legacy classes and interface were redesign by JDK 5 to support Generics. In general, the legacy classes are supported because there is still some code that uses them.

The following are the legacy classes defined by **java.util** package

1. Dictionary
2. HashTable
3. Properties
4. Stack
5. Vector

There is only one legacy interface called **Enumeration**

**NOTE:** All the legacy classes are synchronized

**Enumeration interface**

1. **Enumeration** interface defines method to enumerate(obtain one at a time) through collection of objects.
2. This interface is superseded(replaced) by **Iterator** interface.
3. However, some legacy classes such as **Vector** and **Properties** defines several method in which **Enumeration** interface is used.
4. It specifies the following two methods
5. What are different collection class methods?collections.sort

**JUNIT and TestNg Mocks :**

1. What is the Unit Testing ?What is a Unit in Unit testing?
2. What is a TestCase?
3. Why you use Junit to test your code ? : we can test all code together when some changes made , we want to ensure everything is working fine without any errors.we use this to automate the test cases.open source framework which can be used to for resuablity.somethimes these unit test cases will be useful for the realtime testcases.
4. What is parameterized test? Parameterized test enables developer to perform the same test over and again using different values.
5. Significance of testing ?
6. What are Annotations @Test,@Before,@BeforeClass
7. How you set the Priority ?
8. Manual Testing Vs Automation Testing?
9. Set of execution condition with steps.you will be giving the function

**TrstNg and Junit**

1. What are the advantages of testing over Junit?

TestNG framework has following benefits over JUnit.

1. **1-** TestNG annotations are more logical and easier to understand.
2. **2-** Unlike JUnit, TestNG does not require to declare @BeforeClass and @AfterClass.
3. **3-** There is no method name constraint in Selenium TestNG framework.
4. **4-** TestNG supports three additional setups:
5. **4.1-** @Before/AfterSuite,
6. **4.2-** @Before/AfterTest, and
7. **4.3-** @Before/AfterGroup.
8. **5-** In Selenium TestNG projects, there is no need to extend any class.
9. **7-** In TestNG, it is possible to run Selenium test cases in parallel.
10. **8-** TestNG supports grouping of test cases which is not possible in JUnit.
11. **9-** Based on the group, TestNG allows you to execute the test cases.
12. **10-** TestNG permits you to determine the dependent test cases. Every test case is autonomous to other test cases.
13. **What is the significance of <testng.xml> file?**

**Answer:** In a Selenium TestNG project, we use <testng.xml> file to configure the complete test suite into a single file. This file makes it easy to group all the test suites and their parameters in one file. It also gives the ability to pull out subsets of your tests or split several runtime configurations. Few of the tasks which we can group in the <testng.xml> file are as follows.

**1-** Can configure test suite comprising of multiple test cases to run from a single place.  
**2-** Can include or exclude test methods test execution.  
**3-** Can mark a group to include or exclude.  
**4-** Can pass parameters in test cases.  
**5-** Can add group dependencies.  
**6-** Can configure parallel test execution.  
**7-** Can add listeners.

1. **Can you specify any 6 assertions of TestNG to be used in a Selenium WebDriver software testing tool.**
2. **Answer:**There are multiple assertions available In TestNG but generally we use the following assertions in out test cases.
3. **1-** assertEquals  
   **2-** assertNotEquals  
   **3-** assertTrue  
   **4-** assertFalse  
   **5-** assertNull  
   **6-** assertNotNull
5. **Why soft assertion is used in Selenium WebDriver and TestNG automation project?**
6. **Answer:** TestNG soft assertion allows to continue the test execution even if the assertion is failed. That means once the soft assertion fails, remaining part of the <@Test> method is executed and the assertion failure is reported at the end of the <@Test> method.

**Atomation Framework :**

1. What is framework? advantages of framework ?
2. When to automate and when not to automate?

Tight DeadLines – we cannot automate under tight dead lines

When UI changes /going to change in future then do not automate that particular module.When application is not going to change at all and its already in maintenance- meaning no enhancements or no new user stories – then also no need to automate

1. What is object repository in automation framework ? where all the elements are placed in one file.

Types of frameworks? some of the popular frameworks which are avaible :

1.Modular Based Testing Framework

2.Library Architecture Framework

3.Data driven Framework

4.Keyword Driven Framework

5.Hybrid Driven Testing Framework

1. Explain about your automation framework ?

**SQL Mock :**

1. Explain Joins?
2. How you define the delete statement?
3. Commit, primary key and foreign key?
4. Views and advantages views?
5. Check constraint?
6. Drop, truncate and delete ?
7. Distinct?
8. Rownum <=3 order by top 3 records
9. Fetch alternate records from the table?
10. Select the unique firnames whose dob between somedates?
11. Select maximum salary from all the departments?
12. Diff between local and temporary variables?
13. Identify in sql?
14. Sql query to find duplicate rows, and delete if you find any duplicate rows in table?

**Mobile Automation Mock :**

**1). What is the difference between Mobile device testing and mobile application testing?**

**Ans.** Mobile device testing means testing the mobile device and mobile application testing means testing of mobile application on a mobile device.

**2).What are the types of mobile applications?**

**Ans.** Mobile applications are of three types:

**Native** **Application**– Native app installed from application store like Android’s google play and apple’ app store. The application which can be installed into your devices and run are known as native application for E.G. whats App, Angry birds etc.

**Web** **Application**– Web applications runs from mobile web browsers like Chrome, Firefox, Opera, Safari etc using mobile network or WIFI. E.G. of web browser applications are m.facebook.com, m.gmail.com, m.yahoo.com, m.rediffmail.com etc. need the internet connection.

**Hybrid Application-** Hybrid apps are combinations of native app and web app. They can run on devices or offline and are written using web technologies like HTML5 and CSS. For E.G. ebay, flipkart etc

**3). What are the defects tracking tools used for mobile testing?**

**Ans.** You can use same testing tool which you use for web application testing like QC, Jira, Rally and Bugzilla etc.

**4. Which things to consider testing a mobile application**[**through black box technique**](http://www.softwaretestinghelp.com/black-box-testing/)**?**

**Ans.**

* By testing your application on multiple devices.
* By changing the port and IP addresses to make sure the device is getting connected and disconnected properly.
* By making calls and sending messages to other devices.
* By testing your web application on different mobile browsers like chrome, Firefox, opera, dolphin etc.

**5).** **What is the basic difference between Emulator and Simulator?**

**Ans.** Emulator is based on hardware and software where in simulator is based on software. Simulation is system that behaves *similar*to something else while emulation is system that *exactly*behave like something else.

**6).** **What are the tools based on cloud based mobile testing?**

**Ans.** Seetest, Perfecto Mobile, BlazeMeter, AppThwack, Manymo, DeviceAnywhere etc.

**7).** **web services are used by mobile app?**

**Ans.** They are many depend upon the application. [SOAP and REST](http://www.softwaretestinghelp.com/soapui-tutorial-13-soap-vs-rest-services/) web services are used but RESRful is more common now.

**8.) What does a test plan for Mobile App contain?**

**Ans.** Test plan for mobile app is very similar to software app

1. Objective
2. Automation tools required
3. required features to be tested:
   1. network
   2. security
   3. performance
   4. size
   5. battery
   6. memory
4. features not to be tested
   1. display size
   2. resolution
5. Test cases
6. Test Strategy
7. Tested by
8. Time required
9. No. of resources required

9. **What is the strategy used to test new mobile app?**

**Ans.**

* System integration testing
* Functional testing
* Installation and uninstallation of the app
* Test HTML control
* Performance
* Check in multiple mobile OS
* Cross browser and cross device testing
* Gateway testing
* Network and Battery testing

10. **Full form of the various extensions**

**Ans.**

* apk – Android Application Package File
* exe – Executable Files
* iPA –iOS App Store Package
* prc – Palm Resource Compiler
* jad – Java Application Descriptor
* adb – Android Debug Bridge
* Aapt**–**Android Asset Packing Tool

<http://www.softwaretestinghelp.com/mobile-testing-interview-questions-answers/>

I

1. What are the main challenges in mobile testing?
2. Why do you want to go for mobile testing?
3. How many years of experience you have in mobile testing?
   1. Depending upon resume
4. Do you have more experience in mobile testing or web?
5. What are the key focus areas in mobile testing as QA
   1. Ui layouts
   2. Screen orientations
   3. Screen resolution
   4. Performance
   5. Background events
6. What is the difference between emulator and simulator?
7. What are the different automation FW? Which one you have used in your last project?
   1. Robotissun
   2. Uiautomator
   3. Expresso
   4. Calabash
   5. Selendroid
   6. Appium
8. What is uiautomator?
9. How do you setup appium?
10. Describe appium architecture?
11. What are desired Capabilities?
12. What is ADB?
13. What are the real time challenges with appium?

**JAVA COLLECTIONS MOCK :** Arrays are fixed in size. supports

SDLC Mock:

1. What is SDLC?
2. What is waterfall model?
3. What is agile?
4. What are product backlogs and sprint backlogs?
5. User stories,sprint?

Mock Questions:

1.whats are the expected results and actual result? before developement / after the developemnet

2.what is test data why we need it ?

Soapui Mocks :

1. What kind of web services you have tested using SoapUI?

2. What are the difference between Soap and Rest API ?

3. In SoapUI tool how do we automate the test cases ? test steps to automate web services?

4. What are the different types of Assertions in SoapUI tool ?

5. What is Mock Service in SoapUI ?

6. Rest API test some data is updated / created in DB …how do we make sure of data is inserted into the database or not?

7. What is Parametarisation ?

8. How do we do data driven testing in soapui ?

Data source loop in soapui pro…we can also do with property transfers ..parameterisation

9. What is the purpose of groovy scripting in soapUI ?

10. How do you get the reports of your tests after excuted ?

Soupui logs in bin forlder ..but in soapui pro has the feature for reports where we can create the reports..

11. how do you set request data for post in RestAPI using SoapUI?

12. What is the difference between Web service and API ?

2/7/2017

1. What is test case andtest Suite?
2. Write test cases for login page?
3. What is smoke testing? : it is mandatory.
4. Diff between Soup and REST?
5. What is WSDL?

UNIX MOCK :

1.What is Unix?

2.What are types of Operating Systems?

Single user –single task

Single user-multi tasks

Multi user –multi task

3. What is a Directory?

4. Differentiate multi user from multi task?

5. What is Kernel?

6. What is Shell?

7. Describe file system in Unix?

8. What is inode ?

9. What is piping ?

10. Explain about the flavor of Unix and Linux ?

Unix:

1. What is unix?
2. What are system calls?
3. Tell about ls command?
4. How to copy a fille from home directory to current directory?
5. Diff between cp and mv?
6. Sorting file?
7. Cat >, cat>> diff ?
8. Displaying first 20 lines from file ?
9. Grep command?
10. Using vi print count no of lines with bird in a file? File=50 bird=20
11. Who?

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

1. Why do we need testing?
2. Whats verification and validation ?
3. When to automate ? when not to automate why ?
4. Types of testing ?