

Summary 0001500: Task 1.2 Introduction - Yourself
Description Prepare relevant information about you

I am Navin Jindal,

=====

Summary 0001497: Task 1.0 Introduction - Education
Description Prepare education background to speak while giving introduction in Interview

Example

I did my post-graduation MCA from Rajeev Gandhi University Bhopal in 2012. I got 1st class degree with 70% aggregate.

I did my graduation BSc, Computer Science from Devi Ahilya University, Indore in 2009. I scored 75% aggregate.

I did my Higher Secondary from MP Board/ CBSC in 2003 and got 60 %

I did my High School from MP Board/ CBSC in 2003 and got 65 %

I got SCJP/CCNA/MSCP/OCJP/..... Certificate with 70% in 2003

I am pursuing MBA from Punjab Technical University that will be complete in 2015.

=====

Summary 0001804: Task 1.10 Tell Me About Your Project
Description Q: Tel me about your project

A:

Project Title _____

For Organization/Client _____

Purpose of this _____

This has multiple modules like _____

I have worked on module _____

Project Architecture _____

I have followed many design patterns like _____

We have used following technologies _____

Steps To Reproduce A: My current project is FDTs, Feature and Defect Tracking systems. This is web based defect tracing system for a large project development team. It facilitates tester to record defects during functional, integration (SIT) and user acceptance (UAT) testing. These defects are assigned to project lead or project manager. A project manager can reassign these defects to his/her developers. Defect status is notified to the assigned user by email. After login a dashboard is displayed to the user that lists users' assigned defects.

FDTs follows MVC design pattern and has layered architecture. MVC pattern is implemented with help of Struts 2.0. Data layer is implemented with help of Hibernate framework. And application layers are integrated with help of Spring.

During development we have used so many design patterns.

=====

Summary 0001498: Task 1.1 Introduction - Work Experience
Description Prepare relevant experience summary to speak while introducing yourself.

Example

I have been working in IT industry since last 2 years on J2EE technology. I have been working with SUNRAYS Technologies as Software Engineer since last 2 years.

I have been playing a role of developer in my team and responsible for understanding the design model and developing the code for applications.

I have expertise on Java, JSP/Servlet, Struts and Hibernate.

I have good understanding of UML diagrams.

I have worked on Layered Architecture and got chance to use multiple design patterns in my applications.

=====

Summary 0001501: Task 1.5 Introduction
Description Q: Can you tell me about yourself? Or brief me about yourself.

I am Tarun Adiwal. I have been working in IT industry since last 2 years on J2ee technology. I did my MCA from GSITS in 2006 with 1st division. I did my graduation BSc Comp Sc. from DA University, Indore in 2003. I have been working with sunRays Technologies as a (trainee) software engineer since last 2 years.

I have been playing a role of developer in my team and responsible for understanding the design model and developing the code for applications.

I have expertise on Java, JSP/Servlet, Struts and Hibernate. I also have a good understanding of UML diagrams.

I also have worked on design patterns in my applications.

*Good interpersonal & communication skills for understanding customer requirements.

*self-motivated and goal-oriented professional

Expected questions

1. Q - What is design pattern?
2. A- Design patterns are standard practices, derived over the period of time and followed by industry to implement certain tasks.
 - a. Q – What design patterns you have used in your application?
 - b. A – DTO pattern, Business Delegate pattern, Service Locator pattern, Factory pattern, Singleton pattern, Decorator pattern, Bridge pattern etc.

=====

Summary 0001515: Task 1.6 Introduction Telephonic Interview
Description Q: Can you tell me about yourself? Or brief me about yourself.

I am Tarun Adiwal. I have been working in IT industry since last 2 years on J2ee technology. I did my MCA from GSITS in 2006 with 1st division. I did my graduation BSc Comp Sc. from DA University, Indore in 2003. I have been working with sunRays Technologies as a (trainee) software engineer since last 2 years.

I have been playing a role of developer in my team and responsible for understanding the design model and developing the code for applications.

I have expertise on Java, JSP/Servlet, Struts and Hibernate. I also have a good understanding of UML diagrams.

I also have worked on design patterns in my applications.

Expected questions

1. Q - What is design pattern?
2. A- Design patterns are standard practices, derived over the period of time and followed by industry to implement certain tasks.
 - a. Q – What design patterns you have used in your application?
 - b. A – DTO pattern, Business Delegate pattern, Service Locator pattern, Factory pattern, Singleton pattern, Decorator pattern, Bridge pattern etc.

Primitive Language Questions

1. What are differences between Interface and Abstract classes?
 - a. Abstract classes have partial implementation whereas interfaces have only methods declarations/signature.
 - b. Interfaces
 - i. Interfaces are used in design phase when multiple behaviors from a subsystem are expected. Or I use interfaces when I see that any subsystem of my design will change frequently for example if my application need to support multiple databases then I will define interfaces and write implementation classes for each database.
 - ii. JDBC is the best example of interface that supports multiple databases
 - iii. Interface another example - For example, the Strategy pattern lets you swap new algorithms and processes into your program without altering the objects that use them. A media player might know how to play CDs, MP3s, and wav files. Of course, you don't want to hardcode those playback algorithms into the player; that will make it difficult to add a new format like AVI. Furthermore, your code will be littered with useless case statements. And to add insult to injury, you will need to update those case statements each time you add a new algorithm. All in all, this is not a very object-oriented way to program.
 - c. Abstract Classes
 - i. Abstract classes are used when parent classes need to provide some default behavior (methods) and child classes need to provide some specialized behavior (methods). Application framework is an excellent candidate for abstract classes.
 - ii. For example, if you have an application framework, an abstract class may provide default services such as event and message handling. Those services allow your application to plug in to your application framework. However, there is some application-specific functionality that only your application can perform.
 2. What are differences between String and StringBuffer classes?
 3. What is difference between StringBuffer and StringBuilder?
 - a. StringBuffer is synchronous and thread safe
 - b. StringBuilder is asynchronous and not thread safe.
 4. Write a program on bubble sort.
 5. Write a program for prime number generation.
 6. Write a program to replace a word in a String without using a library function.
 7. What is Java and J2EE.
 8. What is JVM?
 9. What is JRE (Java Runtime Environment)
 10. What is J2SE?
 - a. Java Standard Edition
 11. What is J2ME?
 - a. Java 2 Micro Edition
 12. What is J2EE?
 - a. Java Enterprise Edition
 13. Does Serializable interface have any method?
 - a. No, it is a marker interfaces
 14. What is transient variable
 - a. Attributes defined as transient will not be serialized.
 15. What type of variables are declared as transient
- String :- immutable
StringBuffer - mutable
StringBuilder - immutable*

- Value Objects
- SessionFactory - HB
- Wrappers Classes,

} these are immutable classes.

www.sunrays.co.in

THINK IT. THINK US.

- a. Calculated filed, memory point and database pointer objects like ResultSet.
 - 16. Does Cleanable interface have any method?
 - a. No, it does only first level cloning. It follows flyweight core design pattern.
 - 17. What are wrapper classes and why?
 - a. Ans : Integer, Float, Double, Boolean are the wrapper classes those are used to convert primitive data types (int, float, double, float, boolean) into objects.
 - 18. What are value objects?
 - a. Values objects are java beans those have private attributes and accessor (getter and setter) methods. These objects are used hold and carry data across multiple components.
 - b. Sometimes value objects are called DTOs (Data Transfer Objects).
 - 19. What is immutable object?
 - a. When objects value can be initialized once and can never be changed, It is called immutable object. All wrapper classes are immutable objects.
 - b. When value object or DTO has only getter methods then they are called immutable objects.
 - 20. What is JVM architecture?
 - 21. What is Cloneable Interface? How many methods contain by this interface?
 - 22. What is deep cloning?
 - 23. What is the difference between static and dynamic polymorphism?
 - 24. What are differences between final, finally and finalize keywords?
 - 25. What are the new features of JDK 1.5 ? - vectors, enhanced for, generics, annotations, scanners,
 - 26. What are the new features of JDK 1.6 ? - javascript support, enhanced swing facilities.

Object Oriented Programming Concepts

Expected Interview questions

1. What is encapsulation? Can you write a sample code?
 - a. The process of gathering all related attributes and methods in a class. In other words the result of an encapsulation will be a class.
 2. What is polymorphism?
 - a. Can be achieved through interfaces, abstract class, method overriding and method overloading.
 3. How you can achieve polymorphism with help of method overriding and overloading?
 4. What is data hiding?
 - a. To hide class attributes and methods from out side world with help access modifier private.
 5. What is data abstraction?
 - a. To make available class attributes and methods to out side world with help access modifier public and protected.
 6. What is difference between **overloading** and **overriding** ?
 7. What are differences in data abstraction and data **hiding**?

Static polymorphism is associated with overloaded methods because it gives the impression that a single named method will accept a number of different argument types. The `System.out.println()` method is an example that may take `String` or `Object` references, `boolean` and other primitive types as an argument. In fact, each overloaded method is separate and the compiler can see the difference between them. In this case, the argument types are fixed at compile time and are considered static. This has nothing to do with the Java keyword `static`.

Dynamic polymorphism is where a class overrides a superclass method or implements an interface. For example, any class may override the `Object.toString()` method and

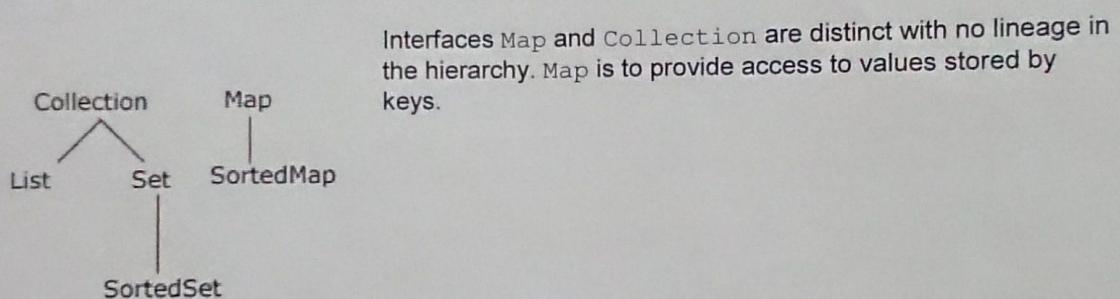
provide its own implementation, and this is known at compile time. However, for a simple Java program that instantiates a series of objects and calls their `toString()` method, the compiler does not consider the object references differently. Any differences in the objects' `toString()` implementations are only seen at runtime, so they are considered dynamic.

method overloading is an example of static polymorphism whereas overriding is an example of dynamic polymorphism.

Collection Framework

Packages – `java.util`

The Collection Framework provides a well-designed set of interface and classes for sorting and manipulating groups of data as a single unit, a collection. The Collection Framework is made up of a set of interfaces for working with the groups of objects.



Interface	Set	List		Map	
Implementation	HashSet			HashMap	
		ArrayList			
	TreeSet			TreeMap	
		LinkedList			
Historical		Vector	Stack	Hashtable	Properties

Key Classes & Interfaces

Interfaces	Classes
1. List 2. Set 3. Iterator 4. Enumeration	1. HashMap 2. Hashtable 3. ArrayList 4. Vector 5. TreeSet

Expected Interview questions

- What is collection framework?

- a. It contains 3 basic interfaces List, Set and Map. Set can not contain duplicate values whereas List can contain duplicate values. Map contains key-value pairs but Value and Key can not be null.
- 2. What is an Iterator?
 - b. Iterator is used to read data sequentially from collection classes (Vector, ArrayList, TreeSet, etc.). Collection interface contains iterator() method that returns Iterator object.
- 3. What is Enumeration?
 - c. Just like iteration, it reads data sequentially from collection objects.
- 4. What is java.util package
 - d. It is a package that contains all utility classes and interfaces like Vector, ArrayList, HashMap, HashTable, HashSet, List, Set, Map etc.
- 5. What are differences between Hashtable and HashMap?
- 6. What are differences between Vector and ArrayList?
- 7. What are differences between Enumeration and Iterator interface?
 - a. An iterator over a collection. Iterator takes the place of Enumeration in the Java collections framework. Iterators differ from enumerations in two ways:
 - i. Iterators allow the caller to remove elements from the underlying collection during the iteration with well-defined semantics.
 - ii. Method names have been improved.
- 8. What is the importance of equals and hashCode methods in Collection classes.
- 9. What is difference between Comparable and Comparable classes.
- 10. Which design pattern is followed by Iterator interface?
- 11. Which type of objects are compared by Comparable and Comparator objects?
 - a. POJO
 - b. DTO
 - c. VO

Comparator
- equals()
comparator()
java.util
- customized
sorting order

I/O

Packages

`java.io` - The `java.io` package contains many classes that your programs can use to read and write data. Most of the classes implement sequential access streams. The sequential access streams can be divided into two groups: those that read and write bytes and those that read and write Unicode characters. Each sequential access stream has a specialty, such as reading from or writing to a file, filtering data as its read or written, or serializing an object.

One class, `RandomAccessFile`, implements random input/output access to a file. An object of this type maintains a file pointer, which indicates the current location from which data will be read or to which data will be written.

Key Classes

1. File
2. Read Text Data
 - a. Reader
 - b. FileReader
 - c. InputStreamReader
 - d. BufferedReader
 - e. Scanner
3. Write Text Data
 - f. Writer
 - g. FileWriter
 - h. BufferedWriter
 - i. PrintWriter

Comparable
- compareTo()
- java.lang.
- natural sorting
order



4. Read Binary Data
 - j. InputStream
 - k. FileInputStream
 - l. BufferedInputStream
 - m. ObjectInputStream
5. Write Binary Data
 - n. OutputStream
 - o. FileOutputStream
 - p. BufferedOutputStream
 - q. ObjectOutputStream
6. Read/Write primitive data
 - r. DataInputStream
 - s. DataOutputStream
7. Primitive File Handling like C/C++
 - t. RandomAccessFile

Key Interfaces

1. Serializable

Key Exception Classes

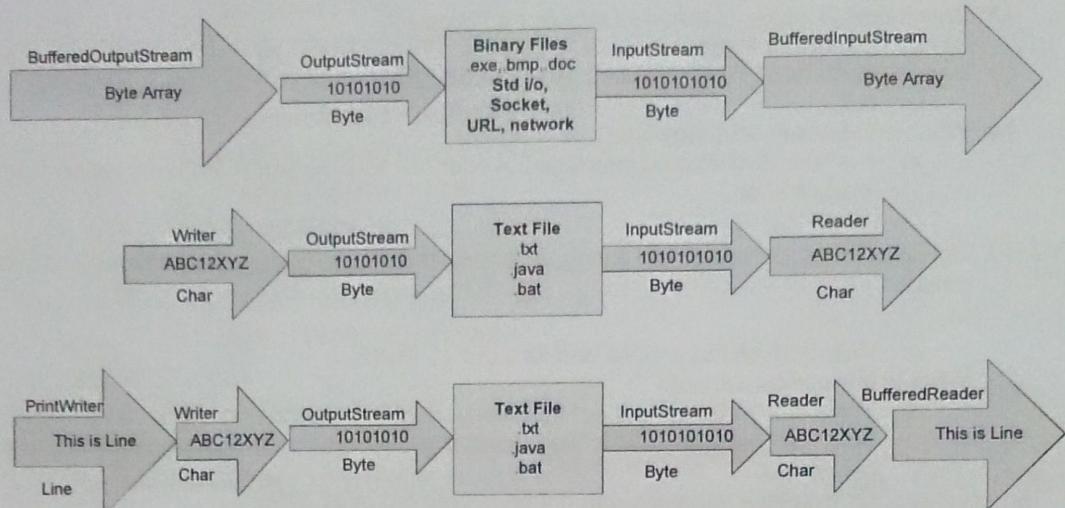
1. EOFException
2. FileNotFoundException
3. IOException

Key Methods

1. flush()
2. close()
3. print/println method of PrintWriter class

Assignment

1. Create a copycon, dir, copy, delete, and rename windows command. Usage should be
 - a. URAGE : java <command> filename
 - b. Dir – Use File class
 - c. CopyCon – Use BufferedReader/Scanner and PrintWriter
 - d. Copy – Use BufferedInputStream/FileInputStream and
BufferedOutputStream/FileOutputStream classes
 - e. Delete – Use File class
 - f. Rename – Use File class
2. Create a word counter – Use StringTokenizer, BufferedReader
3. Create a char counter – Use FileReader
4. Create a find and replace utility – Use Scanner/BufferedReader and
String.replace()
5. Create an employee data base that will add/delete/modify and list data from a text file. –
Use DataInputStream and DataOutputStream
6. How to write an objects to a file – ObjectInputStream and ObjectOutputStream
7. How to append data in a file?



Design Patterns

1. Decorator pattern followed by
 - a. Buffered I/O classes
 - b. Object I/O classes
 - c. Data I/O Classes

Expected Interview questions

1. How can I read from, and write to, files in Java?
 - a. Use `FileReader` and `FileWriter` for text files and `FileInputStream` and `FileOutputStream` for binary files
2. When and why reader/writer and InputStream/OutputStream classes are used
 - b. Reader/Writer are used for text data and InputStream/OutputStream are used for binary data.
3. Which class is used to convert bytes into text? OR How do I connect a Reader class to an InputStream?
 - c. Ans : `InputStreamReader`
4. How can I append an existing file in Java?
5. Why `flush()` method call is required before closing to OutputStream/Writer?
 - d. To push data from RAM(temporary memory) to target
6. What is the function of filter classes?
 - e. Decorate the input/output streams
7. Which class support primitive language (C/Pascal) file handling
 - f. Ans : - `RandomAccessFile`
8. Which classes are used to read/write java Objects
 - g. `ObjectInputStream`/ `ObjectOutputStream`
9. What is mandatory to read/write an object
 - h. Class must be serialized
10. How to serialize a class



- i. Class must implement Serializable interface
- 11. How many methods are contained by Serializable interface?
 - j. Ans = 0, Why?
- 12. Which design pattern is followed by Filter IO classes?
 - k. Decorator Pattern
- 13. Does Serializable interface have any method?
 - l. No
- 14. What is Externalizable interface?
 - m. Just like Serializable interface but user has to write logics for serialization and de-serialization.
- 15. How can I provide a directory listing, and allow the user to navigate directories and select a file?
 - n. With help of File class and its list() method.
- 16. What is a transient variable?
 - o. Transient variables will not be persisted during sterilization. When a serialized object stored/persist in a file or send over the network then transient variable are discarded.
 - p. Or transient variables can not be moved out of its JVM.
- 17. How to read/write primitive data values from/to file
 - q. DataInputStream and DataOutputStream
- 18. How can I create directories in Java?
 - r. With the help of File.mkdir() method
- 19. Differentiate between System.out and System.err.
 - s. System.err: The "standard" error output stream. This output stream is used to display error messages
 - t. System.out: The "standard" output stream. Typically this stream corresponds to display output
- 20. How you can break a string?
 - u. With Help of StringTokenizer
- 21. How you will read line by line from key board?
- 22. What is difference between flush and close method?

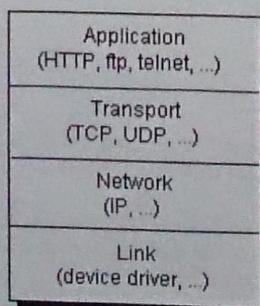
IO Hierarchy

```
java.io
+-- InputStream
|   +-- FilterInputStream
|   |   +-- BufferedInputStream
|   |   +-- DataInputStream
|   |   +-- LineNumberInputStream
|   |   +-- PushbackInputStream
|   +-- ByteArrayInputStream
|   +-- FileInputStream
|   +-- PipedInputStream
|   +-- SequenceInputStream
|   +-- StringBufferInputStream
+-- OutputStream
|   +-- FilterOutputStream
|   |   +-- BufferedOutputStream
|   |   +-- DataOutputStream
|   |   +-- PrintStream
|   +-- ByteArrayOutputStream
|   +-- FileOutputStream
|   +-- PipedOutputStream
+-- File
+-- RandomAccessFile
+-- FileDescriptor
+-- StreamTokenizer
```

Networking

Packages

Java.net - Computers running on the Internet communicate to each other using either the **Transmission Control Protocol (TCP)** or the **User Datagram Protocol (UDP)**, as this diagram illustrates:



When you write Java programs that communicate over the network, you are programming at the application layer. Typically, you don't need to concern yourself with the TCP and UDP layers. Instead, you can use the classes in the `java.net` package. These classes provide system-independent network communication. However, to decide which Java classes your programs should use, you do need to understand how TCP and UDP differ.

TCP provides a point-to-point channel for applications that require reliable communications. The Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), and Telnet are all examples of applications that require a reliable communication channel. The order in which the data is sent and received over the network is critical to the success of these applications.

UDP (User Datagram Protocol) is a protocol that sends independent packets of data, called datagrams, from one computer to another with no guarantees about arrival. UDP is not connection-based like TCP.

The UDP protocol provides for communication that is not guaranteed between two applications on the network. UDP is not connection-based like TCP. Rather, it sends independent packets of data, called *datagrams*, from one application to another. Sending datagrams is much like sending a letter through the postal service. The order of delivery is not important and is not guaranteed, and each message is independent of any other.

Key Classes

1. URL
2. URLConnection
3. URLDecoder
4. URLEncoder
5. ServerSocket
6. Socket
7. DatagramPacket
8. DatagramSocket
9. InetAddress

Key Interfaces

None

Key Exception Classes

1. BindException
2. MalformedURLException
3. SocketException

Assignment

1. TCP Multithreaded Chat Server/ Clint – Like Yahoo messenger, user should be able to communicate with each others. Use `Socket`, `ServerSocket`, `AWT` and `Thread` classes.
2. UDP Multithreaded Chat Server/Clint – Use `DatagramSocket`.
3. Download Utility – Develop a utility that will accept a URL as a command line parameter and download site contents up to 3 levels down.

Thread

Packages

`Java.lang` - A thread is a thread of execution in a program. The Java Virtual Machine allows an application to have multiple threads of execution running concurrently.

Key Classes

1. `Thread`
2. `ThreadGroup`
3. `Runtime`

Key Interfaces

1. `Runnable`

Key Methods

1. `start()`
2. `stop()`
3. `yield()`
4. `getPriority()`
5. `setPriority()`
6. `join()`
7. `sleep()`
8. `isDaemon()`
 - a. Supporting Threads like Garbage Collector

Assignment

1. Develop a multithreaded chat server.

Expected Interview questions

1. How two threads can communicate to each other?
 - a. With the help of shared memory area
2. How two processes can communicate to each other?
 - a. With help of Environment variables.
3. What is Daemon thread?
4. What is green thread?
 - a. Controlled by JVM
5. What is native thread?
 - a. Controlled by Operating Systems (OS)
6. What is a deadlock?
 - a.
7. Why threads are called light weight processes?
 - a. Since they share resources so consume fewer resources than processes
8. What are differences between process and thread?
 - a. Process contains thread.
9. What is starvation or live-lock and when it is occurred?
 - a. It is occurred in case of non-preemptive scheduling
10. What is block/method synchronization?

11. What is a monitor key?
12. What is Synchronization?
13. How many Monitor keys are assigned to a class?
 - cc. Ans 2, one for static methods and one for instance methods.
14. What is a racing condition?
 - a. When multiple threads try to access same resources (objects) concurrently
15. How do you solve racing problems?
 - dd. Ans- With help of synchronization
16. How many types of synchronizations are there?
 - ee. 2 – One for Block second is Method

References

17. Java Tutorial - C:\javatutorial\index.html->Essential Java Classes->Concurrency->Processes and Threads
- 17.18. [Java Docs](#)

JDBC Version 2.0

A JDBC driver is a bridge between Database and Application. JDBC specification has only interfaces Connection, Statement, ResuleSet etc., those are implemented by database vendors Oracle, MySQL, Sybase etc. At runtime, DriverManager identifies a driver from supplied URL and connect with database. Multiple database drivers can be loaded with help of Class.forName() method. Each database has a separate database driver.

There are four type of database driver

1. Type 1 : JDBC-ODBC drivers - MS Access User DSN
2. Type 2 : Native Driver – Oracle OCI driver
3. Type 3 : Native Driver is installed at middleware
4. Type 4 : Pure java drivers – ex. "Class12" for Oracle and "mysql-connector-java-3.1.13" for MySql etc.