**Interview Questions in Siemens**

**First round**

**Core Java**

1. What is exception?

Ans:==>Exception is an object that represents a logical mistake occurred in

the program due to wrong input given by end user during

program execution.

2. What is checked and unchecked exception?

Ans:==>Checked Exception :=>The exception which is checked by compiler for smooth execution of the program at runtime are called Checked Exception.

eg:=>FileNotFoundException

SQLException

HallTicketMissingException(CustomeException)

PenNotWorkingException(CustomException)

In the case of Checked Exception compiler will check whether we are handling exception if the programmer not handling then we get compile time error.

Unchecked Excpetion:=>The exceptions which are not checked by compiler are called Checked Exceptions.

eg:=>ArithmeticException

NullPointerException

BombBlastException(CustomeException)

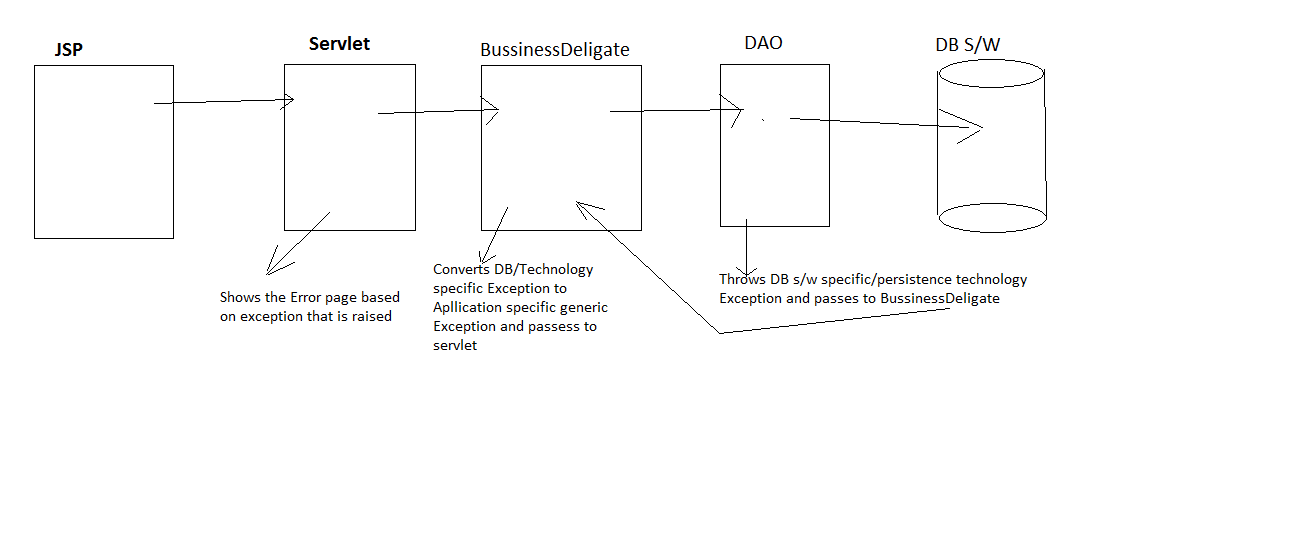
In the case of UncheckedException compiler won't check whether programmer handling exception or not.

3. Which one u prefer to handle exception try-catch or throws?

Ans==>try-catch

4. In your project how u handle the exception in which layer and how u map ?

Ans:=>In our project we did not eat or suppress the exceptions in any layer of business tier always delegate or forward the exception to controller component by converting into application specific exception. So that the controller servlet can display error page showing the exception.



5. Which collection u used in your project write some pseudo code?

Ans:==> We are mainly List and Map collection used in our project.

A collection is a java container object it is used for storing multiple homogenous and heterogeneous object ,unique and duplicate object without size limitation further it is used for carrying all objects at a time from one class to another class as method arguments and return value.

We are used collection object(List, Map) in DAO classes for storing all customer, students (as per your project required) information without size limitation as many record as available in database and further we want to transfer(return) all these customer, student(as per your project) information at a time from DAO classes to service classes and from service class to controller classes for displaying on browser.

**Pseudo code:**

Customer.java

public class Customer{

private int customerId;

private String name;

private Date date;

//use setter and getter method

}

In DAO class we can write this method

Public ArrayList<CustomerDTO> getAllCustomers(){

//Create ArrayList object

ArrayList<CustomerDTO> customers=new ArrayList<CustomerDTO>();

Connection conn=null;

Statement stmt=null;

ResultSet rs=null;

try{

if(conn!=null){

conn=openConnection();

}

If(stmt!=null){

Stmt=conn.createStatement();

}

String query=”SELECT \* FROM Customer”;

rs=stmt.executeQuery(query);

while(rs.next()){

CustomerDTO customer=new CustomerDTO();

//here we can write code to fill up your DTO

customers.add(customer);

}//while

}catch(Exception e){

e.printStackTrace();

}

finally{

//here we can write code to release resource

}

}

6. Difference between HashMap and HashTable ?

Ans:==>HashMap:

1. In HashMap no method is synchronized and its object are not thread-safe.

2. Performance is high.

3. HashMap contains one null key and multiple null values.

HashTable:

1. In Hashtable all methods are synchronized and hence it is thread safe.

2. Performance is low compare to HashMap

3. Null insertion is not possible for both key and value voilation lead

NullPointerException.

7. If 2 thread try to access my collection Object at same time then is there any problem ? If yes how can I handle this?

Ans:==> we will get ConcurrentModificationException  
  
 To avoid it we have to use ConcurrentHashMap

8. What is synchronization, and what are the various places where we can use it

Ans:==>Synchronization is a modifier applicable only for methods and blocks but not for classes and variables. The process of allowing multiple threads to modify an object in sequence is called synchronization. We can allow multiple threads modifying the object sequentially only by executing that object's mutator methods logic in sequence from multiple threads. This possible by using object locking concept.

Yes, the problem is data inconsistency.

To overcome these problem we should go for synchronized keyword. If a method or blocks declared as synchronized then at a time only one thread is allow to execute that method or block on the given object. so that data inconsistency problem is resolved.

9. What is static keyword, where we have to prefer that keyword and why?

Ans:==>The class level members which have static keyword in their definition are called static member.

Java supports 4 types of static members

1. static variable

2. static block

3. static method

4. static main method

All static members are identified and get memory location at the time of class loading by default by JVM in method area.

10. What is the difference between static block and static method?

Ans:==> **Static Block:**

=>Java static block is the group of statements that gets executed when the class is loaded into memory by [Java ClassLoader](http://www.journaldev.com/349/java-classloader).

=>Static block is used to initialize static variables of the class. Mostly it’s used to create static resources when class is loaded.

=>We can’t access non-static variables in static block. We can have multiple static blocks in a class, although it doesn’t make much sense. Static block code is executed only once when class is loaded into memory.

**Example:**

static{

//Here we can write JDBC register code

}

**Static Method**:

=>Same as static variable, static method belong to class and not to class instances. A static method can access only static variables of class and invoke only static methods of the class.

=>Usually static methods are utility methods that we want to expose to be used by other classes without the need of creating an instance. For example [Collections class](http://www.journaldev.com/1260/collections-in-java-tutorial#collections-class).

=>[Java Wrapper classes](http://www.journaldev.com/1002/java-wrapper-classes-tutorial-with-examples) and utility classes contains a lot of static methods. The main() method that is the entry point of a java program itself is a static method.

11. Can I access an instance field from static context?

Ans:==> No

12. Can I call a static method through the object reference?

Ans:==>Yes

13. Can I declare a class as private?

Ans:==>No

14. HashMap internal works?

Ans:==>

15. Programme: I have one string 9845321067 I have to sort the string in reverse order

o/p:-9876543210 without using any predefined method

Solution :

import java.util.\*;

class SortDescendArray  
{  
public void getFormatData(String input)  
{  
char[] ch=input.toCharArray();  
List<Character> data=sort(ch);  
arrangeDescending(data);  
}  
//logic for sort algorithim  
public List<Character> sort(char[] ch)  
{  
int temp=0;  
List<Character> characters=new ArrayList<>();  
for (int i=0;i <= ch.length - 1;i++)  
{  
for (int j=i + 1;j <= ch.length - 1;j++)  
{  
if (ch[i] >= ch[j])  
{  
temp = ch[i];  
ch[i] = ch[j];  
ch[j] = (char)temp;  
}  
}  
characters.add(ch[i]);  
}  
return characters;  
}

//Logic for reverse(descending)  
public void arrangeDescending(List<Character> input)  
{  
int length=input.size() - 1;  
for (int q=length;q >= 0;q--)  
{  
System.out.println(input.get(q));  
}  
}

public static void main(String[] argd)  
{

new SortDescendArray().getFormatData("197532468");  
}  
}

Ans:=>

16. What is polymorphism give one example?

Ans:==>It is one thing but multiple behavior.

=> Polymorphism means "mean forms".

=>Single form behaving differently in different situation.

=>A single function or single characters in different places.

eg:->President in home acts as husband and parent.

and office acts good president.

Polymorphism refers to the ability of two or more objects belonging to difference classes to respond exactly to the same message in class specific ways.

**Hibernate:->**

1. If JDBC is there why hibernate?

Ans:->**Limitation of Hibernate**

1=>JDBC code is based on SQL queries and these queries are DB s/w dependent

SQL queries. So JDBC code is DB s/w dependent persistence logic.

2=>Changing DB s/w in the middle of project development or in the middle

of the production is very complex.

3=>JDBC code is having boiler plate code problem.

4=>Exception handling is mandatory because it raise checked Exception is

called SQLException.

5=>In select SQL query execution gives ResultSet object which is not serializable object to send over the network.

6=>There is no direct supports for services like caching, transaction management and etc.

Then after we can tell about Hibernate Features

**Hibernate Features:**

1=>Support POJO/POJI model programming.

2=>Hibernate is light weight.

3=>Hibernate persistence logic is DB s/w independent i.e. it is portable across the multiple DB software.

4=>Gives DB s/w independent Query language called HQL(Hibernate Query language)

5=>Gives built in jdbc connection pool and also allows to use 3rd party JDBC con pool like C3p0, proxool and server managed JDBC con pool.

6=>providing caching/buffering support , which holds DB table records across the multiple same request and reduces network round trips between java App

and DB s/w.

7. Support lazy loading.

8. Support versioning

9. Support Timestamp

10. Support criteria API

11. Support Association mapping and etc.

1. Can I reuse the session? If yes how if no then what is the problem?

Ans:--> Yes we can reuse the session if it is not closed, else after closing we try to access the existing session then it will throw Exception LazyInitializationException.

1. What is the difference between level-1 and level-2 cache?

Ans**:=> Level 1:->**

=> It is built in cache, associated with session object.

=>It is 1 per session object.

=>Two benefits of Level 1 cache

1=>keep Loaded object from DB s/w and uses it across the multiple loadings.

2=>Keeps changes happened on the object in the course of Transaction

Management and generate single SQL query on DB s/w reflecting all

the changes when transaction is committed.

**Level-2=>**

=>It is associated with Session Factory object i.e.

=>It is 1 per Session Factory object.

=>It is configurable cache. There are multiple second level cache provides s/w.

1.

we need to configure 1 of them to enable second level cache in the application.

1. Difference between get and load?

Ans:= GET method:->

1. Perform eager loading of object.
2. Does not generate proxy object.
3. Involves total 1 object of domain class in the loading of records.
4. Suitable to check whether records is available or not.
5. Returns null value when records is not available.
6. Useful in single layered environment.

USECASES=> To show offer of the day without user permission.

To show job posting.

**Load Method:->**

1. Performs lazy loading of object.
2. Generate proxy object.
3. Involves total 2 objects of domain class(1 real object + 1 proxy object) in the loading of records.
4. We should use this method by assuming records is available
5. Throws ObjectNotFoundException when records is not available.
6. Useful in multiple layer environment where the records marked in layer will be utilized in another layer.

USECASES:-> To show personalized offer based on user interest.

To show notification related details in which u r interested.

5. Difference between update and merge?

Ans:=>Both methods are useful to make Transient, Detached state objects as persistent state object.

Update:=>

When we use session.update(-) method to make Detached state object as persistent state object and if the L1 cache session object is already having one or more object with same identity value then application throws org.hibernate.NonUniqueObjectException.

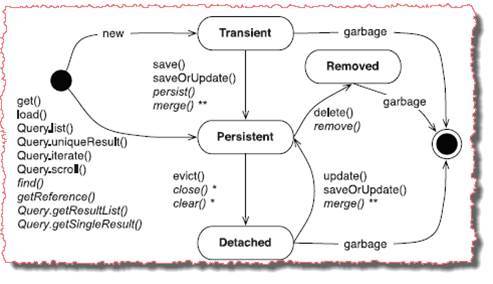
Merge=>

If we are using session.merge(-) methods in the same situation then Hibernate merges changes between both objects and create new persistence state object without throwing any Exception. This indicates session.merge(-) is powerful than Session.update(-).

6. What is the state of object?

Ans:-> There are mainly 3 types of state of object in Hibernate.

1. Transient(Not Associated with session)
2. Persistent(Associated with session)
3. Detached (previously persistent, But currently not associated with session)



7. What is the lazy attribute and how it works?

Ans:-> In Association mapping parent object will be loaded normally and associated child object will be loaded only one demand basis. This is called lazy Loading in Association Mapping.

We can enable or disable this lazy loading in one to many association by using lazy attribute of <set> , <map> ,<list> and etc. This tag allows 3 possible values and the default value is “true” .

Eg. <set name=”phones” cascade=”all” lazy=”true/false/extra”>

* - - -
* - - -

</set>

8. How u configure data Source and name of the data Source?

Ans:-> Normally as we are not manually configure the data source that configuration part was done by our DB Admin and it placed in JNDI Registry we need to look-up it ,and as we are using spring-JDBC in our project so we are using DriverManagerDataSource.

**Spring:**

1. What is dependency injection?

Ans:==>If the underlying server or container dynamically assign the

dependent value to our resources (objects/classes) program then

it is called Dependency Injection.

There are 4 types of dependenecy Injection

1. Setter Injection (We can inject through setter method call)

2. Constructor Injection(We can inject through parameterized constructor)

3. Aware Injection or Interface Injection

4. Lookup Method Injection

2. What is the spring MVC annotation you used in your project?

Ans:==>There are many sprng mvc annotations used in our project.

1.@RequestParam

2.@ModelAttribute

3.@RequestMapping

4.@Controller

5.@Service

6.@Resource

7.@Autowire

8.@Bean

9.@InitBinder

10.@Inject

11.@Qualifier and etc..

3. Explain the Spring MVC flow relate with your project architecture?

Ans:==>Explain Spring MVC flow with respect to your project..

4. For every request I have to trace the IP address of user note :

( before control will come to the my controller class I want to trace)

Ans:==> Simply we can use spring MVC interceptor so before request will come to our controller it will handle the request cause it work flow seems to same as Filter.

5. What is AOP?

Ans=>AOP is the methodology of programming that makes the programmer primary logics of application from the secondary logics or helper logics. AOP is not replacement for OOP more ever it complements for OOP.

In AOP we develop business methods in classes by mixing up both primary, secondary logics. Due to this business method becomes heavy and reusability of secondary logics will be killed. We cannot enable or disable secondary logics without touching source code of business method.

To overcome this method use AOP that makes the programmer to write primary logics in separate class and secondary logics in another class and allow to mix and bind both logics dynamically at runtime by generating new class.

6. What is the difference between join point and point cut?

Ans==>JoinPoint:=>The possible points in classes on which aspect logics can be applied is called Join point. They are like fields, Constructor, methods and etc..

Spring supports only methods at joint point.

PointCut:=> It is a collections of joint point on which aspects are applied

(i.e. aspects are configured)

7. What is the difference between interceptor and around advice in spring AOP?

ANs:==> Interceptor means simply before request will hit to my web component it will handle by interceptor like J2EE filter but around advice means simple

we need to specify one point cut to apply cross cutting logic in target class joint point so here after request hit to my web component then control will goes to AOP So interviewer will try to confuse u by asking unrelated question.

8. Can we access stored procedure through spring?

Ans:==>Yes, by using SpringJdbc

9. Can I write 2 times init() method in bean?

Ans:==>Yes

10. In my application unused bean is there I don't want to instantiate that bean how can I handle?

Ans:==> Suppose I don’t want to instantiate all bean then we have to use one attribute in Spring bean configuration file Lazy-Init=’true’ otherwise spring IOC will instantiate all beans .