**Java day2 answers**

29.How do you handle exceptions?

The **Exception Handling in Java** is one of the powerful mechanism to handle the runtime errors so that the normal flow of the application can be maintained. By using try catch block

30.is it mandatory to have catch block with try block?

Yes you can write [try](http://download.oracle.com/javase/tutorial/essential/exceptions/try.html) without catch. In that case you require finally block. Try requires either catch or finally or both that is at least one catch or finally is compulsory.

Eg”:

try{

// throw exception

} finally{

// do something.

}

Or

try{

// throw exception

} catch(exception ) {

// do something.

}

31.can we have multiple catch block in a single try block

Yes we can have

32.difference between string buffer and string builder?

Java provides multiple classes through which strings can be used. Two such classes are StringBuffer and StringBuilder.

String buffer:

Java StringBuffer class is used to create mutable (modifiable) String objects. The StringBuffer class in Java is the same as String class except it is mutable i.e. it can be changed. And it synchronized

String builder:

Java StringBuilder class is used to create mutable (modifiable) String. The Java StringBuilder class is same as StringBuffer class except that it is non-synchronized. It is available since JDK 1.5.

| **StringBuffer Class** | **StringBuilder Class** |
| --- | --- |
| StringBuffer is present in Java. | StringBuilder was introduced in Java 5. |
| StringBuffer is synchronized. This means that multiple threads cannot call the methods of StringBuffer simultaneously. | StringBuilder is asynchronized. This means that multiple threads can call the methods of StringBuilder simultaneously. |
| Due to synchronization, StringBuffer is called a thread safe class. | Due to its asynchronous nature, StringBuilder is not a thread safe class. |
| Due to synchronization, StringBuffer is lot slower than StringBuilder. | Since there is no preliminary check for multiple threads, StringBuilder is a lot faster than StringBuffer. |

33.what is string class in java and how do you compare Strings?

In [Java](https://www.javatpoint.com/java-tutorial), string is basically an object that represents sequence of char values. An [array](https://www.javatpoint.com/array-in-java) of characters works same as Java string.

**Java String** class provides a lot of methods to perform operations on strings such as compare(), concat(), equals(), split(), length(), replace(), compareTo(), intern(), substring() etc.

There are three ways to compare String in Java:

1. By Using equals() Method
2. By Using == Operator
3. By compareTo() Method

By Using equals() Method:

The String class equals() method compares the original content of the string. It compares values of string for equality.

Eg:

String s1="Sachin";

   String s2="Sachin";

   String s3=**new** String("Sachin");

   String s4="Saurav";

   System.out.println(s1.equals(s2));//true

   System.out.println(s1.equals(s3));//true

   System.out.println(s1.equals(s4));//false

By Using == Operator:

The == operator compares references not values.

Eg:

**class** Teststringcomparison3{

**public** **static** **void** main(String args[]){

   String s1="Sachin";

   String s2="Sachin";

   String s3=**new** String("Sachin");

   System.out.println(s1==s2);//true (because both refer to same instance)

   System.out.println(s1==s3);//false(because s3 refers to instance created in nonpool)

 }

}

By compareTo() Method

The String class compareTo() method compares values lexicographically and returns an integer value that describes if first string is less than, equal to or greater than second string.

Suppose s1 and s2 are two String objects. If:

* **s1 == s2** : The method returns 0.
* **s1 > s2** : The method returns a positive value.
* **s1 < s2** : The method returns a negative value.

Eg:

**class** Teststringcomparison4{

**public** **static** **void** main(String args[]){

   String s1="Sachin";

   String s2="Sachin";

   String s3="Ratan";

   System.out.println(s1.compareTo(s2));//0

   System.out.println(s1.compareTo(s3));//1(because s1>s3)

   System.out.println(s3.compareTo(s1));//-1(because s3 < s1 )

 }

}

34.how to create a immutable class?

There are many immutable classes like String, Boolean, Byte, Short, Integer, Long, Float, Double etc. In short, all the wrapper classes and String class is immutable. We can also create immutable class by creating final class that have final data members as the example given below

Mutable means can modified, immutable means cannot modified

Eg:

**public** **final** **class** Employee

{

**final** String pancardNumber;

**public** Employee(String pancardNumber)

{

**this**.pancardNumber=pancardNumber;

}

**public** String getPancardNumber(){

**return** pancardNumber;

}

}

**public** **class** ImmutableDemo

{

**public** **static** **void** main(String ar[])

{

Employee e = **new** Employee("ABC123");

String s1 = e.getPancardNumber();

System.out.println("Pancard Number: " + s1);

}

}

Output:

Pancard Number: ABC123

* The instance variable of the class is final i.e. we cannot change the value of it after creating an object.
* The class is final so we cannot create the subclass.
* There is no setter methods i.e. we have no option to change the value of the instance variable.

35.what is serialization?

**serialization in Java** is a mechanism of writing the state of an object into a byte-stream. It is mainly used in Hibernate, RMI, JPA, EJB and JMS technologies.

The reverse operation of serialization is called deserialization where byte-stream is converted into an object. The serialization and deserialization process is platform-independent, it means you can serialize an object on one platform and deserialize it on a different platform.

For serializing the object, we call the **writeObject()** method of ObjectOutputStream class, and for deserialization we call the **readObject()** method of ObjectInputStream class.

We must have to implement the Serializable interface for serializing the object.

Eg: [Serialization in Java - javatpoint](https://www.javatpoint.com/serialization-in-java)

36.if I have class in that class I have four to five variables and I don’t want to convert all the variables to bytes what is the way to avoid that?

Create a custom method in serializable class and use variable you need to convert object into bytes

Eg:

import java.io.\*;

class YourClass implements Serializable {

// Your variables

private int variable1;

private String variable2;

private double variable3;

// ...

// Constructor and other methods

// Serialization method

private void writeObject(ObjectOutputStream out) throws IOException {

out.writeInt(variable1);

out.writeObject(variable2);

out.writeDouble(variable3);

// Write other variables if needed

}

// Deserialization method

private void readObject(ObjectInputStream in) throws IOException, ClassNotFoundException {

variable1 = in.readInt();

variable2 = (String) in.readObject();

variable3 = in.readDouble();

// Read other variables if needed

}

}

37.I have a string let us consider I have a string madam (m a d a m) so if I traverse it from right to left or left to right It remains same can you tell me how to check whether the string is palindrome or not?

Three methods are to find palindrome:

[Java program to check whether a string is a Palindrome - GeeksforGeeks](https://www.geeksforgeeks.org/java-program-to-check-whether-a-string-is-a-palindrome/)

38.why should we use spring?

[10 Reasons to Use Spring Framework in Projects - GeeksforGeeks](https://www.geeksforgeeks.org/10-reasons-to-use-spring-framework-in-projects/)

39.what is spring bean?

In Spring, the objects that form the backbone of your application and that are managed by the Spring IoC container are called beans. A bean is an object that is instantiated, assembled, and otherwise managed by a Spring IoC container.

Inversion of control:

[Inversion of Control](https://www.baeldung.com/inversion-control-and-dependency-injection-in-spring) (IoC) is **a process in which an object defines its dependencies without creating them.** This object delegates the job of constructing such dependencies to an IoC container.

40. Life cycle of a bean?

[Bean life cycle in Java Spring - GeeksforGeeks](https://www.geeksforgeeks.org/bean-life-cycle-in-java-spring/)

41. what are the starter dependency?

[Starter dependencies are **jar dependencies that provide a quick way to configure our application, without manually including a lot of related dependencies**](https://www.bing.com/ck/a?!&&p=204b3391903d9482JmltdHM9MTcwODczMjgwMCZpZ3VpZD0wMDM3ZDY3Ny0wMTc1LTY3YTktMDdmZi1jNGM5MDBjNzY2ZjQmaW5zaWQ9NTc5Ng&ptn=3&ver=2&hsh=3&fclid=0037d677-0175-67a9-07ff-c4c900c766f4&psq=what+are+the+starter+dependency+&u=a1aHR0cHM6Ly93d3cubG9naWNiaWcuY29tL3R1dG9yaWFscy9zcHJpbmctZnJhbWV3b3JrL3NwcmluZy1ib290L2Jvb3Qtc3RhcnRlcnMuaHRtbA&ntb=1).

Refers this page: [Spring Boot Starters - javatpoint](https://www.javatpoint.com/spring-boot-starters).

42. can you change the port of your contact server

[Spring Boot Change Port - javatpoint](https://www.javatpoint.com/spring-boot-change-port#:~:text=We%20can%20change%20the%20port%20in%20Spring%20Boot,4%20Using%20WebServerFactoryCustomizer%20Interface%205%20Using%20Command-Line%20Parameter)

43. what is swagger?

44.ideas about swagger

45.what is primary key

46.what is foreign key