Java OOP interview questions

what is the difference between String and StringBuilder?

String	StringBuilder
1.String objects are immutable	**1.StringBuilder objects are mutable**
2.new strings are created if we try to modify strings. hence memory wastage will be more.	**2.new strings will not be created when we modify strings. Hence no memory wastage.**

what is the difference between stringbuilder and stringbuffer?

StringBuilder	StringBuffer
string builder objects are **not synchronized** and not thread safe.	string buffer objects are **synchronized** and thread safe.

what is a class

1. **class** is a virtual entity.
2. class **acts as blueprint** for objects.
3. class is used for storing related variables & methods

eg:

```
public  class  Student
{
     public  int  sno = 1;
     public  String  sname = “ramesh”;
     public  String  subject = “java”;
 }
```

what is an object

1. object is a real world entity.

2. we use **new keyword** for creating object

eg:

```
Student  s1  =  new  Student();
```

what is constructor

1. by using **constructor** we **can initialize objects.**
2. constructor **name** should be **same as class name**
3. constructor **should not have return type**

eg :

```
public class Bank
{
        public String name;
        public long acno;

        public Bank(String name, long acno)
        {
                this.name = name;
                this.acno = acno;
        }
}
```

what is encapsulation?

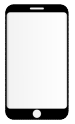
binding logically related data and functions in a common related place, is known as encapsulation.

```
public class Doctor
{
    public string name;
    public int exp;
    public Doctor(string name, int exp)
    {
      this.name=name; this.exp= exp;
    }
    public string suggestMedicine(String disease)
    {
      //return medicine name based on disease.
    }
}
```

In the above example, we are storing all doctor related data and doctor related methods in one class. So we can say doctor class is following encapsulation.

what is abstraction?

hiding implementation details and exposing the required details to the users, is called as abstraction

eg : 

mobile phone is an example for abstraction.
because we don't know the internal working mechanism of mobile phone.

what is default constructor?

if programmer **doesn't** supply any constructor for a class,
then **java compiler** supplies **a default constructor** for that class

what is inheritance?

1. inheritance allows us to access parent class variables and functions, in child class.
2. Inheritance reduces code duplication

eg:

```
public class A
{
}
public class B extends A
{
}
```

what is polymorphism?

when an entity is appearing with the **same name** in **different forms,** then that entity is said to exhibit **polymorphism.**

java supports 2 types of polymorphisms.

1. compile time polymorphism / static polymorphism
2. runtime polymorphism / dynamic polymorphism

method overloading is example for compile time polymorphism

method overriding is example for runtime polymorphism

what is method overloading?

when 2 or more methods having same method name and different parameters in a given class,
then we can say the method is overloaded.

eg:

```
public class C
{
   public void m1(int x)
   {
    System.out.println("hi");
   }
   public void m1(int x, int y)
   {
    System.out.println("hello");
   }
}
```

what is method overriding ?

1.overriding is used for changing the behavior of parent class method in the child class.

2.method overriding requires same access specifier, same return type, same method name and same parameter in base class and derived class.

```
public class D {
        public void m2()        {
            s.o.pln("this is parent class m2 function");
        }
}
public class E extends D {
        public void m2()        {
            s.o.pln("this is child class m2 function");
        }
}
```

what is the use of **this** keyword?

this keyword is used to refer current instance or current object.

eg:

```
public class Doctor
{
    public string name;
    public int exp;

    public Doctor(String name, int exp)
    {
      this.name=name;
      this.exp= exp;
    }

}
```

what is super keyword?

by using super keyword we can access parent class members from child class.

```
public class F
{
        public int x = 10;
}
public class G extends F
{
        public int x = 20;
        public void m3()
        {
                System.out.println(this.x);
                System.out.println(super.x);
        }
}
```

what is the difference between instance variables and static variables?

Instance variable	Static variable
Instance variables are object specific.	static variables are class specific.
Instance variables must be accessed by using object name	static variables can be accessed by using class name

```
public class H
{
        public int x = 10;
        public static int y = 20;
}
```

```
H  h1 = new H();
System.out.println( h1.x );
System.out.println( H.y );
```

what is the difference between instance methods and static methods?

Instance method (non static method)	Static method
1. Instance methods must be called by using object name	1. Static methods can be called by using class name
1. we can use this and super keywords in instance methods.	1. we can't use this or super keyword in static methods.

```
public class J
{
      public void m4() {
            System.out.println ("hi") ;
       }
       public static void m5() {
            System.out.println("hello");
       }
}
```

```
J  j1 = new J();

j1.m4();

J.m5();
```

what is an abstract method

a method **without** body, is called as **abstract method**

eg : public abstract void m6();

what is an abstract class?

1. abstract class is a class which contains zero or more abstract methods.
2. we can't create object for abstract class.

```
public abstract class K
{
        public abstract void m7();
}
```

what is an interface?

1. interface can contain only final variables and methods without bodies.
2. we can't create object for interface.
3. by default all methods of interface are abstract methods.

```
public interface I
{
        public final int x = 10;
        public void m8();
}
```

```
public class L implements I
{
        public void m8()
        {
            System.out.println("hi");
        }
}
```

what is the difference between abstract class and interface ?

abstract class	Interface
1. abstract class can contain abstract methods, and concrete methods (i.e methods with bodies).	1. Interfaces can only contain abstract methods.
2. abstract class members can be **private / protected / default / public**	2. interface members should be only **public.**
3. for using abstract class we need to use extends keyword	3. for using interface we need to use implements keyword

what is access specifier?

java supports 4 types of access specifiers.
public, private, protected, and default.

1. public members are accessible from anywhere in the application.
2. private members are accessible only within the declared class
3. protected members are accessible in the declared class, derived class and also any where within the same package
4. default members are accessible in the declared class and also anywhere within the same package

what is an exception?

Exception is a run time error which terminates the program abruptly.

how do you handle exceptions?

exception handling is done by using try-catch blocks

```
try
{
        // code which may give error
}
catch( exception ex  )
{
        // error storing code
}
```

can we have multiple catch blocks for a try?

1. yes, a try block can be followed by multiple catch blocks.
2. exceptions of child class should be handled first, then parent class exceptions should be handled.

what is the difference between throw and throws?

throw	throws
throw is used to re-throw the exception to the previous function in call stack, after catching the exception	if we are not catching an exception in current method, then we use throws keyword to pass exception to method caller

what are the types of exceptions in java?

java supports 2 types of exceptions

1. checked exceptions
2. unchecked exceptions.

what is the difference between checked exceptions and unchecked exceptions?

checked exceptions	**unchecked exceptions**
1.it is mandatory to handle checked exceptions, else compiler will throw error.	1.it is not mandatory to handle unchecked exceptions. compiler will not throw error.
2.Checked exceptions inherits from Exception class. **eg : IOException, FileNotFoundException**	2.Un checked exceptions inherits from Error class or RunTimeException class. **eg : ArithmeticException**

what is finally block?

1. finally block code will be executed in all the scenarios, no matter whether exception occur or does not occur.
2. finally block is used to clean important systems resources, like closing files, closing database connections.

one try block can have how many finally blocks?

one try block can have maximum one finally block.

Is it possible to have a try block without catch block?

yes we can have try with finally block, without catch blocks.

what is the parent class for all exceptions?

Throwable class is the parent most class for all exceptions in java.

write one example for exception handling?

```
try
{
        int[] arr = {10,20,30};
        System.out.println(arr[3]);
}
catch(ArrayIndexOutOfBoundsException ex)
{
        System.out.println("trying to access invalid
                                index");
}
```

what is collection [or] collection framework?

collection framework is set of predefined interfaces and classes, used to store and process the data efficiently.

What are the types of collections in java?

java supports 3 types of collections

1. sets
2. lists
3. maps

What is map?

1. map is used to store pairs of elements.
2. Keys should be unique, and values can be duplicate, in map.
3. there are 3 types of maps.

   HashMap, TreeMap, LinkedHashMap

eg:

```
HashMap<String, String> m =
                    new HashMap<String, Sting> ();

m.put( “user1@gmail.com”, “abcd”);
m.put( “user2@gmail.com”, “xyz”);
System.out.println(m);
```

What is list?

1. list is an ordered collection of elements.
2. we can access list elements with index.
3. list allows duplicate elements.
4. there are 2 types of lists.

   ArrayList, and LinkedList.

eg:

```
ArrayList<Integer> x = new ArrayList<Integer> ();
x.add(10);
x.add(20);
System.out.println(x);
```

What is the difference between ArrayList and LinkedList?

Array List	LinkedList
In array list, elements will be stored in the sequential order.	In LinkedList, elements will be stored in random order.
Arraylist is faster compared to LinkedList	LinkedList is slow

What is the difference between ArrayList and Array?

Array List	Array
ArrayList is part of collection framework	Array is not part of collection framework
Array list is dynamically growable	Array is fixed sized entity

What is set?

1. set does not allow duplicate elements.
2. set stores elements in random order.
3. there are 3 types of sets.

   HashSet, TreeSet, LinkedHashSet

eg:

```
HashSet<String> h = new HashSet<String> ();
m.add( “java”);
m.add( “j2ee”);
System.out.println(h);
```

What is the difference between HashMap and TreeMap?

HashMap	TreeMap
HashMap internally uses hashing algorithm to store elements.	TreeMap internally uses binary trees to store elements.
HashMap stores elements in random order.	TreeMap stores elements in key sorted order.

What is the difference between HashSet and TreeSet?

HashSet	TreeSet
HashSet internally uses hashing algorithm to store elements.	TreeSet internally uses binary trees to store elements.
HashSet stores elements in random order.	TreeSet stores elements in sorted order.

What is the difference between List and Set

List	Set
List stores elements in insertion order (first come first store order)	Set stores elements in random order
List allows duplicate elements.	Sets doesn't allow duplicate elements.
Lists are slow compared to sets.	Sets are faster than lists.

What is the difference between Collections and generics?

Collections	Generics
Collections are not safe to use.	Generics are more safe.
While using collections we don't have to specify the data type of the elements.	While using generics we need to specify the date type of elements.

what is thread?

thread is an independent path of execution.

threads are used for multitasking in our projects.

how will you create a thread in java?

there are 2 ways to create threads in java.

1. by extending Thread class.
2. by implementing Runnable interface.

```
public class M extends Thread
{
        public void run()
        {
                S.o.pln("thread A");
        }
}

M   m1 = new M();
m1.start();
```

```
public  class N implements Runnable
{
        public void run()
        {
                S.o.pln("thread N");
        }
}

N  n1 = new N();
Thread  t = new  Thread(n1);
t.start();
```

what happens if we call run() in place of start()?

if we call run() directly, then it will bypass the thread creation. (which means JVM will not create a thread)

what is the difference between Thread class and Runnable interface?

Thread	Runnable
Thread is predefined class of JDK, which internally implements Runnable interface	Runnable is a predefined interface of JDK, which contains run() method.

what is the use of sleep() method?

sleep() method is used to halt a thread execution, for few milliseconds.

what is the best way to create a thread, by extending Thread class or by implementing Runnable interface?

creating a thread by implementing Runnable interface is better, because

1. if we create a thread by extending Thread class, then in future if we want to inherit any other class into our class then it is not possible, as java doesn't support multiple inheritance.

2. if we create a thread by implementing Runnable interface, then in future if we want to inherit any other class into our class then it is possible to inherit.

what is join() method?

join() method allows one thread to wait until other thread completes its execution.

what is notify() method?

notify() is used to wake up a thread, which is in waiting state.

what is a daemon thread?

1. daemon thread is a thread which will not die, even after program execution stops.
2. garbage collector is an example of daemon thread.

what is final method?

final method is a method which we can't override in child class

what is final class?

we can't inherit or extend final class

what is package?

package is a logical container used for grouping related classes and interfaces.

What is garbage collector?

Garbage collector cleans all the un-referenced objects from the heap memory.

When to use abstract method?

If you know only method name, but you don’t know the logic, then use abstract method.

When to use abstract class?

If a class contains some concrete methods and some abstract methods, then declare the class as an abstract class.

Is it possible to make a class as abstract class, if it contains 0 abstract methods?

yes

How do you use an abstract class?

By inheriting abstract class into another class and by overriding all abstract methods of parent class in the child class.

how to use an interface?

by using implements keyword.

can we implement multiple interfaces into a class?

yes

can we inherit multiple classes into a class?

no

what is the parent most class for all classes in java?

Object class

can we override static method?

no

can we overload static method?

yes

explain public static void main()?

public means jvm can access this method from anywhere

static keyword allows JVM to call this method without creating object for this class

void means it won't return any value

main is predefined keyword which marks this method as entry point for the project.

explain System.out.println()?

System is a predefined class of java.

out is a static object present in System class.

println() is a method of out.