[1] What are the principal concepts of OOPS?

1. Abstraction
2. Encapsulation
3. Inheritance
4. Polymorphism

[2] What is a Class?

A class is simply a templet that defines the attributes (data members) and methods (member functions) of an object.

[3] What is an Object?

An object is an instance of a class. When we call the constructor of a class with "new" operator, memory is allocated in heap area and an object is created.

[4] What is Abstraction?

Abstraction means providing the services by hiding the internal functionalities or implementation of the services.

[5] What is Encapsulation?

Encapsulation means binding the data to a single unit as well as providing access or hiding (restricting access) to the data as per requirement.

In simple language;

Encapsulation = Abstraction + Data Hiding

[6] What is the difference between Abstraction and Encapsulation?

Abstraction means providing the functions with hiding the internal implementations of the functions.

Encapsulation means binding the data in a single unit as well as restricting access to the data and functions. (Encapsulation = Abstraction + Data Hiding)

[7] What is Inheritance?

Inheritance means acquiring the features of one class in another class. The class whose features has been acquired is called as Parent class or Super class or Base class.
The class which acquires the features of parent class is called as Child class or Sub class or Derived class.
Inheritance facilitates code reusability. The child class has all the features of parent class as well as some extra features added as per requirement.

[8] What is Polymorphism?

Polymorphism means two or more methods having same name those perform same operation but with different internal implementation/logic with same parameter(s) or different parameter(s).
In JAVA polymorphism can be achieved by

(a) Method overloading
(b) Method overriding

[9] What is runtime polymorphism or dynamic method dispatch?

In inheritance when a child class has method with same name and same parameters as of parent class but, with different implementation then it is called as Method overriding.
Method overriding is also called as runtime polymorphism or dynamic method dispatch because, which implementation (parent or child) of the overridden method will be called depends upon the type of object created and object is created at runtime.

[10] What is method overriding?

In inheritance when a child class has method with same name and same parameters as of parent class but, with different implementation then it is called as Method overriding.

[11] What is method overloading?

Two or more methods having same name but, with different number or different types of parameters performing similar operation is called as method overloading.

[12] What is Super keyword?

It is a non-static reference i.e. used to refer to the super class data members and member functions within child class and super class constructor within child class constructor.

[13] How do you prevent a method from being overridden?

Method overriding can be prevented by specifying a “private” access modifier or “final” modifier to the implemented method in base class.

[14] What is an Interface?

Interface is a blueprint of a class or abstract type in which all the variables are “public”, “static” and “final” by default i.e. variables are constant and all the methods are public and abstract i.e. methods are declared only but not implemented.

[15] What do you mean by Constructor?

A constructor is a special method when called through “new” operator, creates/instantiates an object of its respective class type and returns the object/instance.

[16] Explain String class?

A string is a predefined class datatype in java which represents a sequence of characters.

Strings are immutable.

[17] How System.out.println() works?

The println() method belongs to PrintStream class which is present in java.io package. This method takes a string as input and prints the specified string in a new line on the console. But we cannot create the object of PrintStream class directly.

System is a predefined final class in java present in java.lang package and out is a static field of class type PrintStream present in System class that already holds the instance of PrintStream class.

Thus, System.out refers to the PrintStream object and when println() is called through System.out, indirectly we call the println() through PrintStream object and it prints the specified string on the console.

[18] What is Java Collections Framework? List out some benefits of Collections framework?

A collection is a container that contains a group of objects.

A collection framework is a unified architecture for representing and manipulating collections. It contains interfaces, implementation of interfaces and algorithms to perform useful operations such as searching, sorting and shuffling etc.

Benefits of collection framework:

(1) Reduces programming effort.
(2) Increases program speed and quality.
(3) Allows interoperability among unrelated APIs.
(4) Reduces effort to learn and to use new APIs.
(5) Reduces effort to design new APIs.
(6) Fosters Software reuse.

[19] What is the benefit of Generics in Collections Framework?

Generics is used to create classes, interfaces and methods that will work for various types of data i.e. codes that will work independent of data in a type safe manner. All type casts are done implicitly.

3 Benefits of generics in collection framework:

(1) Strong type casting forces the user to add objects of declared type only in the container thus avoiding ClassCastException at runtime.
(2) Type casting not required since we do not need casting and instanceOf operator.
(3) Helps to write generic algorithms.

[20] What are the basic interfaces of Java Collections Framework?

java.util.Iterable interface is the parent of the whole collection framework. It has only one child interface called Collection. Collection interface has 3 basic interfaces. They are –

1. List
2. Queue
3. Set

Again these interfaces have many child classes, abstract classes and interfaces.

[21] What is an Iterator?

Iterator is a generic interface present in java.util package. It has been implemented in collection framework to traverse or retrieve objects through any collections.

[22] How HashMap works in Java?

Hashing is the process of converting an object into integer value for indexing and faster searches. HashMap is a part of collection framework that implements Map interface and stores the data in a key and value pair where the key is always unique. It uses an array and LinkedList data structure internally for storing key and value.

[23] What are different Collection views provided by Map interface?

There are 3 collection view provided in Map interface. They are –

(1) Key set view

(2) Value set view

(3) Entry set view

[24] How to decide between HashMap and TreeMap?

HashMap should be used when we do not need key-value pair in sorted order and TreeMap should be used when we need key-value pair in sorted (ascending) order.

[25] What are similarities and difference between ArrayList and Vector?

Similarities between ArrayList and Vector:

---

(1) Both implement java.util.List<E> interface.
(2) Both are dynamic.
(3) Both contain duplicates.

Difference between ArrayList and Vector:

---

(1) ArrayList is not synchronized and thus faster. Vector is synchronized and thus slower.
(2) When the number of elements exceeds the capacity, ArrayList increases it's size by 50% of its current size where as Vectors increase it's size by 100% of it's current size.
(3) ArrayList is not a legacy class. But, vector is a legacy class.
(4) ArrayList uses the Iterator interface to traverse through the elements where as Vector uses both Iterator and Enumeration interface to traverse through the elements.

[26] What is difference between Array and ArrayList? When will you use Array over ArrayList?

(1) Array size is fixed but ArrayList size increases automatically if more elements were added.

(2) Array is static while ArrayList is dynamic.

(3) It is mandatory to specify the size of the Array for array initialization while in case of ArrayList, it is not.

(4) Array is faster than ArrayList.

(5) We can store both primitive types and objects in Arrays where as we can only store objects in ArrayList.

(6)Array is multi-dimensional where as ArrayList is single-dimensional.

So, if you need a fixed size array, then use Array otherwise use ArrayList.

[27] What is difference between ArrayList and LinkedList?

(1) ArrayList internally uses dynamic array.

LinkedList internally uses doubly linked list.

(2) ArrayList manipulation is slower.

LinkedList manipulation is faster.

(3) ArrayList is better for sorting and accessing data.

LinkedList is better for manipulating data.

(4) ArrayList implements List interface.

LinkedList implements List interface as well as Deque interface.

(5) The default size of ArrayList is 10.

There is no such default size for LinkedList.

[28] What is Collections Class?

Collections is a utility class in Collections framework present in java.util package. Collections class is basically used with the static methods that operate on collections or return the collection.

[29] What is Comparable and Comparator interface?

(1) Both Comparable and Comparator interface is used to order the objects of user-defined class.

(2) Comparable uses a method i.e. compareTo(Object).

Comparator uses 2 methods i.e. compare(Object o1, Object o2) and equals(Object o).

(3) Comparable provides single sorting sequence only.

Comparator provides multiple sorting sequences.

[30] How can we sort a list of Objects?

A list of objects can be sorted either by implementing Comparator interface and Collections.sort() method together or by implementing Comparable interface.

[31] What Is an Exception?

An exception is an abnormal condition that occurs during program execution and disrupts the normal flow of instructions.

[32] How Can You Handle an Exception?

In JAVA we use try, catch and finally block to handle exceptions.

The try block contains code that may produce exception.

The catch block catches the exception.

The finally block executes some codes that must be executed after the try block codes.

[33] How Can You Catch Multiple Exceptions?

Just after try block we can write multiple catch blocks to catch multiple exceptions.

[34] What Is the Purpose of the Throw and Throws Keywords?

(1) “throw” keyword can throw one instance of exception where as “throws” keyword can throw multiple exceptions.

(2) “throws” keyword is used in method signatures only where as “throw” Keyword is used inside a method or static block.

(3) “throws” keyword throws the exception object instead of handling it. “throw” keyword transfers the control to the caller by throwing an exception object.

[35] What Is the Difference Between a Checked and an Unchecked Exception?

Checked Exception:

(1) Compiler forces the user to handle it and cannot be ignored.

(2) It is typically an user error.

Unchecked Exception:

(1) Compiler does not force the user to handle it and can be ignored.

(2) It is not an user error.

[36] What Are Some Advantages of Exceptions?

Advantage 1: Separation of error-handling code from regular code.

Advantage 2: Propagating errors up the call stack.

Advantage 3: Grouping and differentiating error types.