Important Keywords:

Constructors-
Constructor because it constructs the values at the time of object creation. It is not necessary to write a constructor for a class.

This
Used to refer current class instance variable.
To invoke current class method (implicitly)
This() used to invoke current class constructor
This can be passed as an argument o method call and constructor call.
Static- Can be Variable, Method, Block and Nested Class.
Static Variable makes program memory efficient.
Static Method can access Static Data Member and change value of it.

Final –
Can be a Variable [Value cannot be changed], Can be a Method [Cannot Override it] and can be a Class [Cannot be extended, stops Inheritance]

Super
Used to refer immediate parent class object.
Super can be used to invoke immediate parent class method.
Super()-can be used to invoke immediate parent class constructor.

Inheritance- You can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class.
3 Types of Inheritance–
- **Single**,
- **Multilevel** *(Not Supported in Java)*
- **Hierarchical**

Polymorphism –
Method-Overloading
(*Compile-Time Polymorphism*)- If a class has multiple methods having same name but different in parameters. Method overloading increases the readability of the program.
In Java, Method Overloading is **not** possible by changing the return type of the method only.
Method-Overriding-
(*Runtime-Polymorphism*)- If subclass (child class) has the same method as declared in the parent class.

Encapsulation
Class's variables are hidden from other classes and can only be accessed by the methods of the class in which they are found. E.g., Using Getter-Setter.
It provide- Better Control, Security, Flexibility and Data Hiding using Access Modifiers.

Abstraction

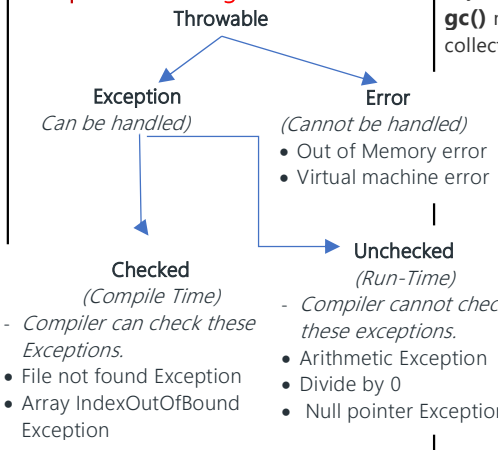
INTERFACE
A Java interface contains static constants and abstract methods and provide Total Abstraction.

To declare an interface, use the interface keyword.
All the methods in an Interface are declared with an empty body and are public and all fields are public, static, and final by default.

Functional Interface- Interface that contains exactly one abstract method. It can have any number of default, static methods but can contain only one abstract method. It is used to implement Lambda Expressions.

Access Modifiers –
Default- The Default access modifier if no access modifier mentioned.
Public- can be accessed from anywhere within the program
Private – data members, and the data methods limit to the class itself.
Protected-access is limited to the entire package

Exception Handling



Throw	Throws
-used to through exception explicitly. -used with try-catch -single exception ca be thrown	-used to through exception explicitly. -used in method signature -multiple exception can be thrown

```
try      {risky code in which exception may occur
         }
Catch {  handling code,
         mandatory to use catch or finally after try block
         }
Finally  { cleanup of opened resource (optional)
         Finally is always executed
         }
```

Printstacktrace() – used to print details of occurred Exception.

Multithreading in Java - executing multiple threads simultaneously.
a thread always exists in any one of the following states-

New: the code has not been run yet and thus has not begun its execution.
Active: When a thread invokes the start() method, it moves from the new state to the active state.
The active state contains **two** states within it: one is **runnable**, and the other is **running**.
Runnable: A thread, that is ready to run is then moved to the runnable state.
Running: When the thread gets the CPU, it moves from the runnable to the running state.
Blocked or Waiting: Whenever a thread is inactive for a span of time
Timed Waiting: After the time runs out, the thread wakes up and start its execution.
Terminated: When a thread has finished its job or when an unhandled exception occurs

There are **two** ways to create a thread:
-By extending **Thread class**
-By implementing **Runnable interface**.

Garbage Collection - a way to destroy the unused objects created using new keyword, it removes the unreferenced objects from heap memory.
gc() method is used to invoke the garbage collector to perform cleanup processing.
finalize() method is invoked each time before the object is garbage collected. It destroys objects not created through new keyword.

Synchronization in Java - capability to control the access of multiple threads to any shared resource.
When we want to allow only one thread to access the shared resource.

Optional Class: It is a public final class and used to deal with NullPointerException. It provides methods which are used to check the presence of value for particular variable.

public static <T> Optional<T> empty()
- It returns an empty Optional object. No value is present for this Optional.

public static <T> Optional<T> of(T value)
- It returns an Optional with the specified present non-null value

Java Lambda Expression Syntax
(argument-list) -> {body}
e.g. **name -> return "Hello " + name;**

Java 8 features:
Lambda expressions, Stream API, Optional class, Functional interfaces, ForEach() method, Collectors class

Generics -Introduced in J2SE 5
Generics force the programmer to store a specific type of objects. So, No type casting required, Checked at Compile time and provides type safety i.e. single type object can be stored.

```
List<String> list = new ArrayList<String>();
```