JAVA

What is Java?

Java is a programming language and a platform.Java is a high level, robust, object-oriented and secure programming language.

```
class Simple{
    public static void main(String args[]){
     System.out.println("Hello Java");
    }
}
```

Types of Java Applications

```
 There are mainly 4 types of applications that can be created using Java programming:
```

1. **Standalone Application:**

   Standalone applications are also known as desktop applications or window-based applications. These are traditional software that we need to install on every machine. Examples of standalone application are Media player, antivirus, etc. AWT and Swing are used in Java for creating standalone applications.

2. **Web Application**

   An application that runs on the server side and creates a dynamic page is called a web application. Currently, Servlet, JSP, Struts, Spring, Hibernate, JSF, etc. technologies are used for creating web applications in Java.

3. **Enterprise Application**

   An application that is distributed in nature, such as banking applications, etc. is called enterprise application. It has advantages of the high-level security, load balancing, and clustering. In Java, EJB is used for creating enterprise applications.

4. **Mobile Application**

   An application which is created for mobile devices is called a mobile application. Currently, Android and Java ME are used for creating mobile applications.

Java Platforms/Editions

There are 4 platforms or editions of Java:

1. Java SE (Java Standard Edition)
   It is a Java programming platform. It includes Java programming APIs such as java.lang, java.io, java.net, java.util, java.sql, java.math etc. It includes core topics like OOPs, String, Regex, Exception, Inner classes, Multithreading, I/O Stream, Networking, AWT, Swing, Reflection, Collection, etc.

2) Java EE (Java Enterprise Edition):
It is an enterprise platform which is mainly used to develop web and enterprise applications. It is built on the top of the Java SE platform. It includes topics like Servlet, JSP, Web Services, EJB, JPA, etc.

3) Java ME (Java Micro Edition):
It is a micro platform which is mainly used to develop mobile applications.

4) JavaFX
It is used to develop rich internet applications. It uses a light-weight user interface API.

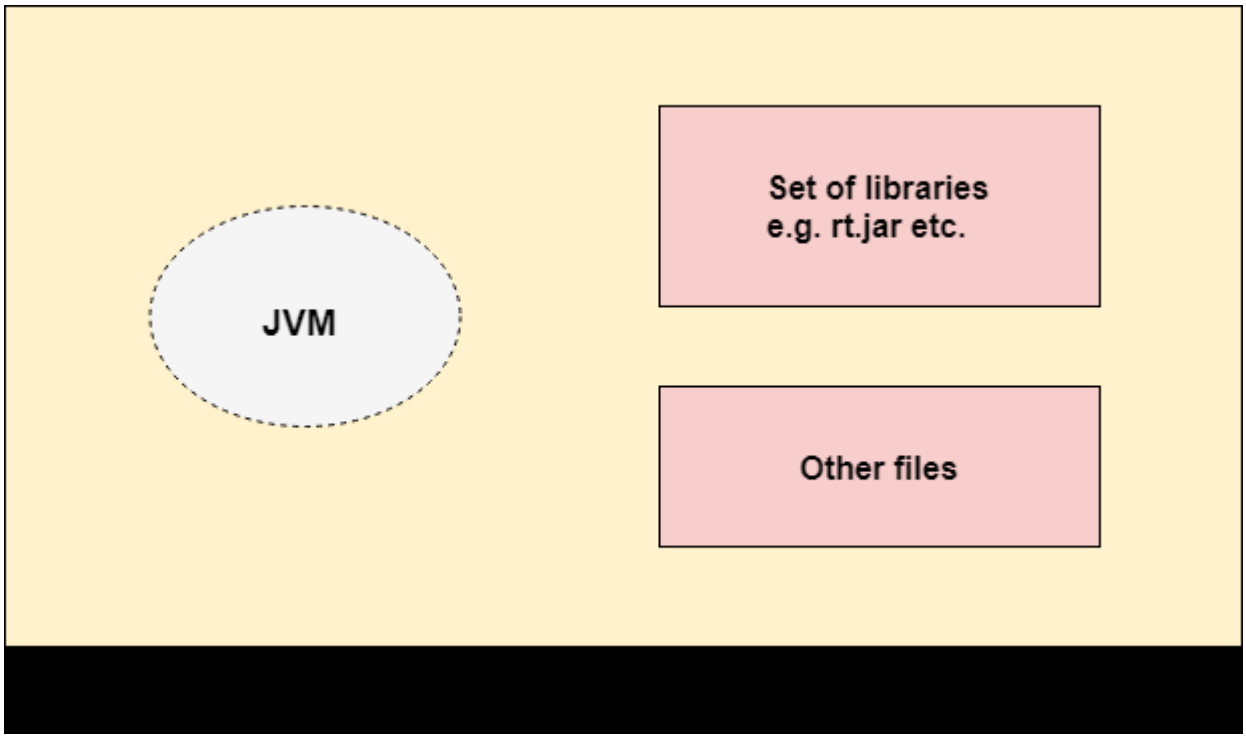
JRE JDK JVM

JVM (Java Virtual Machine) is an abstract machine. It is called a virtual machine because it doesn't physically exist. It is a specification that provides a runtime environment in which Java bytecode can be executed. It can also run those programs which are written in other languages and compiled to Java bytecode.
The JVM performs the following main tasks:

```
Loads code
Verifies code
Executes code
Provides runtime environment
```

JRE is an acronym for Java Runtime Environment. It is also written as Java RTE. The Java Runtime Environment is a set of software tools which are used for developing Java applications. It is used to provide the runtime environment. It is the implementation of JVM. It physically exists. It contains a set of libraries + other files that JVM uses at runtime.

The implementation of JVM is also actively released by other companies besides Sun Micro

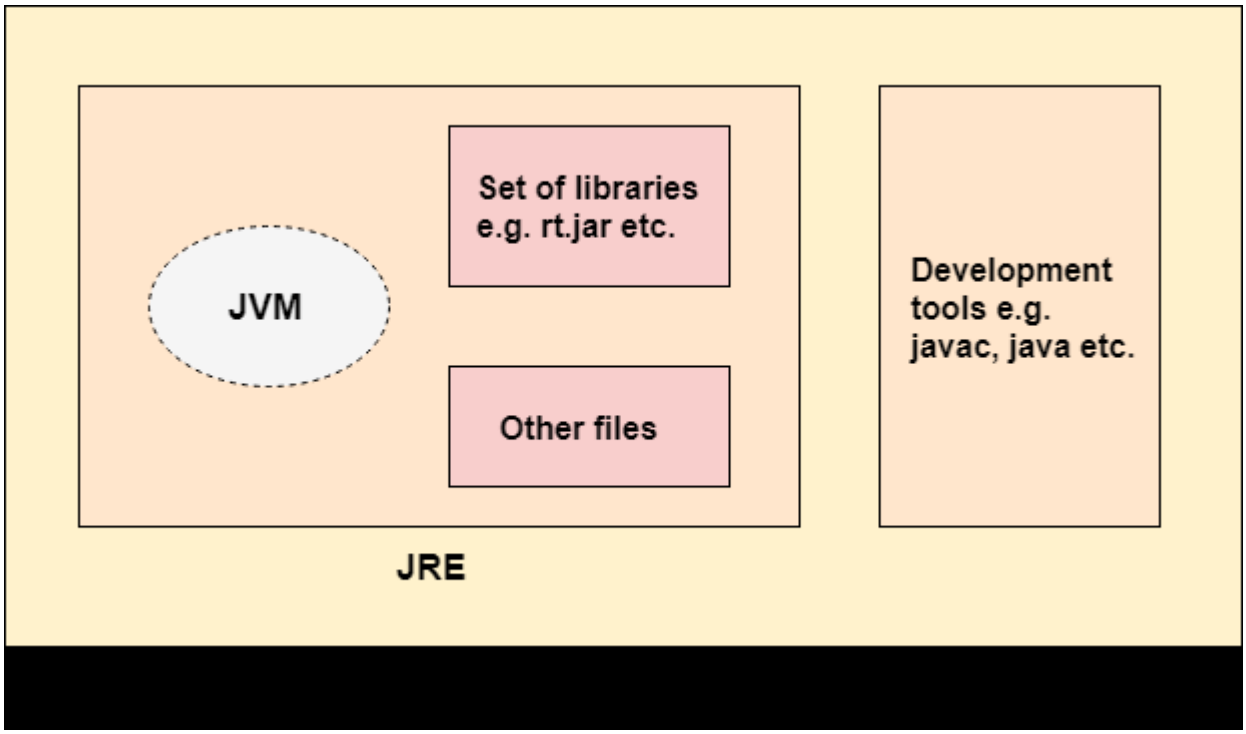
Systems. 

JDK is an acronym for Java Development Kit. The Java Development Kit (JDK) is a software development environment which is used to develop Java applications and applets. It physically exists. It contains JRE + development tools.

JDK is an implementation of any one of the below given Java Platforms released by Oracle Corporation:

```
Standard Edition Java Platform
Enterprise Edition Java Platform
Micro Edition Java Platform
```

The JDK contains a private Java Virtual Machine (JVM) and a few other resources such as an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc), etc. to complete the development of a Java Application.



Variables

Variable is name of reserved area allocated in memory. In other words, it is a name of memory location. It is a combination of "vary + able" that means its value can be changed.

1)Local Variable

A variable declared inside the body of the method is called local variable. You can use this variable only within that method and the other methods in the class aren't even aware that the variable exists. A local variable cannot be defined with "static" keyword.
2)Instance Variable

A variable declared inside the class but outside the body of the method, is called instance variable. It is not declared as static. It is called instance variable because its value is instance specific and is not shared among instances.
3) Static variable

A variable which is declared as static is called static variable. It cannot be local. You can create a single copy of static variable and share among all the instances of the class. Memory allocation for static variable happens only once when the class is loaded in the memory.

Data Type

Data types specify the different sizes and values that can be stored in the variable. There are two types of data types in Java:

```
Primitive data types: The primitive data types include boolean, char, byte, short, ir
Non-primitive data types: The non-primitive data types include Classes, Interfaces, a
```

OOP'S

Object-Oriented Programming is a methodology or paradigm to design a program using classes and objects. It simplifies the software development and maintenance by providing some concepts:

```
Object
Class
Inheritance
Polymorphism
Abstraction
Encapsulation
```

Object

Any entity that has state and behavior is known as an object.An Object can be defined as an instance of a class. An object contains an address and takes up some space in memory. Objects can communicate without knowing the details of each other's data or code. The only necessary thing is the type of message accepted and the type of response returned by the objects.

Class

Collection of objects is called class. It is a logical entity. A class can also be defined as a blueprint from which you can create an individual object. Class doesn't consume any space.

Inheritance(IS_A Relation)

When one object acquires all the properties and behaviors of a parent object, it is known as inheritance. It provides code reusability. It is used to achieve runtime polymorphism. 1) Single 2) Multilevel 3) Hierarchical

Aggregation

If a class have an entity reference, it is known as Aggregation. Aggregation represents HAS-A relationship.

Polymorphism

If one task is performed by different ways, it is known as polymorphism.In Java, we use method overloading and method overriding to achieve polymorphism.

Abstraction

Hiding internal details and showing functionality is known as abstraction.In Java, we use abstract class and interface to achieve abstraction.

Encapsulation

Binding (or wrapping) code and data together into a single unit are known as encapsulation. A java

class is the example of encapsulation. Java bean is the fully encapsulated class because all the data members are private here. Encapsulation in Java is a process of wrapping code and data together into a single unit, for example, a capsule which is mixed of several medicines. encapsulation in java

We can create a fully encapsulated class in Java by making all the data members of the class private. Now we can use setter and getter methods to set and get the data in it.

The Java Bean class is the example of a fully encapsulated class.

Objects and Classes

An object has three characteristics:

```
State: represents the data (value) of an object.
Behavior: represents the behavior (functionality) of an object such as deposit, withd
Identity: An object identity is typically implemented via a unique ID. The value of t
```

A class in Java can contain:

```
Fields
Methods
Constructors
Blocks
Nested class and interface
```

Constructors

constructor is a block of codes similar to the method. It is called when an instance of the object is created, and memory is allocated for the object.

It is a special type of method which is used to initialize the object. When is a constructor called

Every time an object is created using new() keyword, at least one constructor is called. It calls a default constructor
There are two types of constructors in Java:

```
Default constructor (no-arg constructor)
Parameterized constructor
```

Static keyword

The static keyword in Java is used for memory management mainly. We can apply java static keyword with variables, methods, blocks and nested class. The static keyword belongs to the class than an instance of the class. The static can be:

```
Variable (also known as a class variable)
Method (also known as a class method)
Block
Nested class
```

---

this keyword

There can be a lot of usage of java this keyword. In java, this is a reference variable that refers to the current object.
Here is given the 6 usage of java this keyword.

```
this can be used to refer current class instance variable.
this can be used to invoke current class method (implicitly)
this() can be used to invoke current class constructor.
this can be passed as an argument in the method call.
this can be passed as argument in the constructor call.
this can be used to return the current class instance from the method.
```

---

method overloading and overriding

If a class has multiple methods having same name but different in parameters, it is known as Method Overloading.
Advantage of method overloading

Method overloading increases the readability of the program.

Different ways to overload the method. There are two ways to overload the method in java

```
By changing number of arguments
By changing the data type
```

Can we overload java main() method?

```
Yes, by method overloading. You can have any number of main methods in a class by met
```

If subclass (child class) has the same method as declared in the parent class, it is known as method overriding in Java.
Usage of Java Method Overriding

```
Method overriding is used to provide the specific implementation of a method which is
Method overriding is used for runtime polymorphism
```

Rules for Java Method Overriding

```
The method must have the same name as in the parent class
The method must have the same parameter as in the parent class.
There must be an IS-A relationship (inheritance).
```

Can we override static method?

```
No, a static method cannot be overridden. It can be proved by runtime polymorphism, s
```

Why can we not override static method?

```
It is because the static method is bound with class whereas instance method is bound
```

Can we override java main method?

```
No, because the main is a static method.
```

---

final keyword

The final keyword in java is used to restrict the user. The java final keyword can be used in many context. Final can be:

```
variable
method
class
```

The final keyword can be applied with the variables, a final variable that have no value it is called blank final variable or uninitialized final variable. It can be initialized in the constructor only. The blank final variable can be static also which will be initialized in the static block only. We will have detailed learning of these.

Upcasting

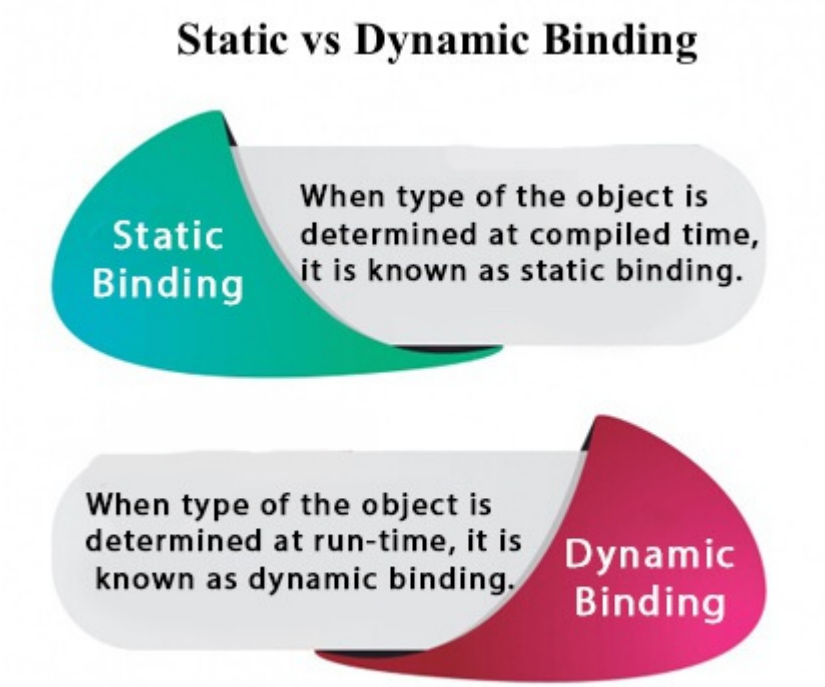
If the reference variable of Parent class refers to the object of Child class, it is known as upcasting.

Static Binding and Dynamic Binding static binding and dynamic binding in java

Connecting a method call to the method body is known as binding.

There are two types of binding

```
Static Binding (also known as Early Binding).
Dynamic Binding (also known as Late Binding).
```



Abstract class

A class which is declared with the abstract keyword is known as an abstract class in Java. It can have abstract and non-abstract methods (method with the body).

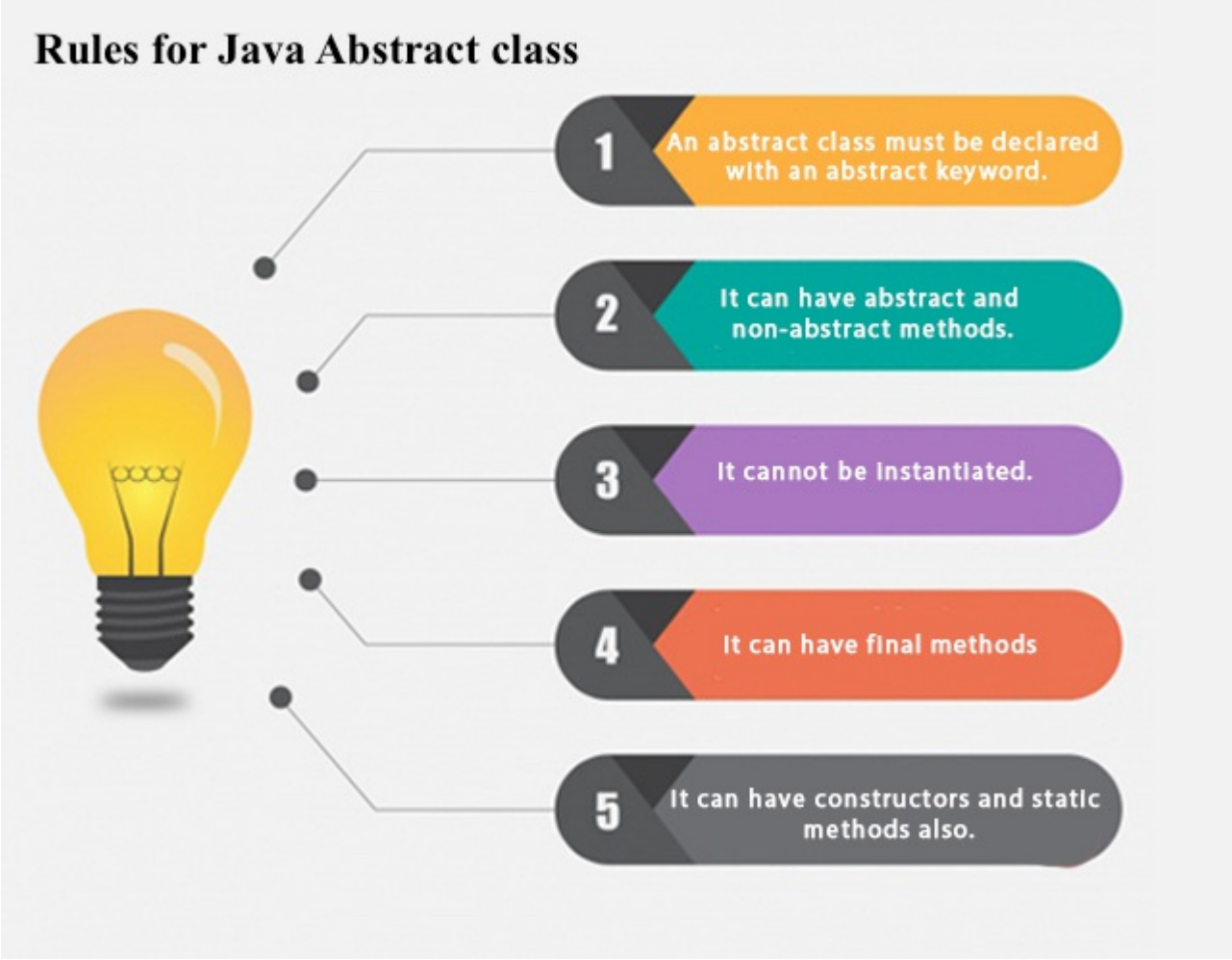
Abstraction is a process of hiding the implementation details and showing only functionality to the user.

Another way, it shows only essential things to the user and hides the internal details, for example, sending SMS where you type the text and send the message. You don't know the internal processing about the message delivery.

Abstraction lets you focus on what the object does instead of how it does it. Ways to achieve Abstraction

There are two ways to achieve abstraction in java

```
Abstract class (0 to 100%)
Interface (100%)
```



Interface

An interface in java is a blueprint of a class. It has static constants and abstract methods.

The interface in Java is a mechanism to achieve abstraction. There can be only abstract methods in the Java interface, not method body. It is used to achieve abstraction and multiple inheritance in Java. Why use Java interface?

There are mainly three reasons to use interface. They are given below.

```
It is used to achieve abstraction.
By interface, we can support the functionality of multiple inheritance.
It can be used to achieve loose coupling.
```

Difference between abstract class and interface

Abstract class and interface both are used to achieve abstraction where we can declare the abstract methods. Abstract class and interface both can't be instantiated.

But there are many differences between abstract class and interface that are given below. Abstract class Interface

1. Abstract class can have abstract and non-abstract methods. Interface can have only

2. Abstract class doesn't support multiple inheritance. Interface supports multiple i

3. Abstract class can have final, non-final, static and non-static variables. Interfa

4. Abstract class can provide the implementation of interface. Interface can't provi

5. The abstract keyword is used to declare abstract class. The interface keyword is u

6. An abstract classcan extend another Java class and implement multiple Java interfac

7. An abstract classcan be extended using keyword "extends". An interface classcan be

8. A Javaabstract classcan have class members like private, protected, etc. Members c

9. ```
   Example:
   public abstract class Shape{
   public abstract void draw();
   }  Example:
   public interface Drawable{
   void draw();
   }
   ```

Simply, abstract class achieves partial abstraction (0 to 100%) whereas interface achieves fully abstraction (100%).

Java Package

A java package is a group of similar types of classes, interfaces and sub-packages.

Package in java can be categorized in two form, built-in package and user-defined package.

There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

Here, we will have the detailed learning of creating and using user-defined packages.

   ```

Access modifiers

There are two types of modifiers in java: access modifiers and non-access modifiers.

The access modifiers in java specifies accessibility (scope) of a data member, method, constructor or class.

There are 4 types of java access modifiers:

```
private:The private access modifier is accessible only within class.
default:If you don't use any modifier, it is treated as default bydefault. The defaul
protected:he protected access modifier is accessible within package and outside the p
public:The public access modifier is accessible everywhere. It has the widest scope a
```

There are many non-access modifiers such as static, abstract, synchronized, native, volatile, transient etc. Here, we will learn access modifiers.

Object class

The Object class is the parent class of all the classes in java by default. In other words, it is the topmost class of java.

The Object class is beneficial if you want to refer any object whose type you don't know. Notice that parent class reference variable can refer the child class object, know as upcasting.

Difference between object and class

There are many differences between object and class. A list of differences between object and class are given below: No. Object Class

1. `Object is an instance of a class.  Class is a blueprint or template from which obje`
2. `Object is a real world entity such as pen, laptop, mobile, bed, keyboard, mouse, ch`
3. `Object is a physical entity.  Class is a logical entity.`
4. `Object is created through new keyword mainly e.g.Student s1=new Student();  Class :`
5. `Object is created many times as per requirement.  Class is declared once.`

6. `Object allocates memory when it is created.  Class doesn't allocated memory when it`

7. `There are many ways to create object in java such as new keyword, newInstance() met`

Let's see some real life example of class and object in java to understand the difference well:

Class: Human Object: Man, Woman

Class: Fruit Object: Apple, Banana, Mango, Guava wtc.

Class: Mobile phone Object: iPhone, Samsung, Moto

Class: Food Object: Pizza, Burger, Samosa

Difference between method overloading and method overriding in java

There are many differences between method overloading and method overriding in java. A list of differer
overloading and method overriding are given below:

No.	Method Overloading	Method Overriding
1)	Method overloading is used *to increase the readability* of the program.	Method overriding is us *specific implementatio* already provided by its
2)	Method overloading is performed *within class*.	Method overriding occu have IS-A (inheritance
3)	In case of method overloading, *parameter must be different*.	In case of method ove *be same*.
4)	Method overloading is the example of *compile time polymorphism*.	Method overriding is th *polymorphism*.
5)	In java, method overloading can't be performed by changing return type of the method only. *Return type can be same or different* in method overloading. But you must have to change the parameter.	*Return type must be s* method overriding.

Exception Handling

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

Difference between Checked and Unchecked Exceptions

1) Checked Exception

The classes which directly inherit Throwable class except RuntimeException and Error are known as che
IOException, SQLException etc. Checked exceptions are checked at compile-time.

2) Unchecked Exception

The classes which inherit RuntimeException are known as unchecked exceptions e.g. ArithmeticExceptio
ArrayIndexOutOfBoundsException etc. Unchecked exceptions are not checked at compile-time, but the

3) Error

Error is irrecoverable e.g. OutOfMemoryError, VirtualMachineError, AssertionError etc.

Java Exception Keywords

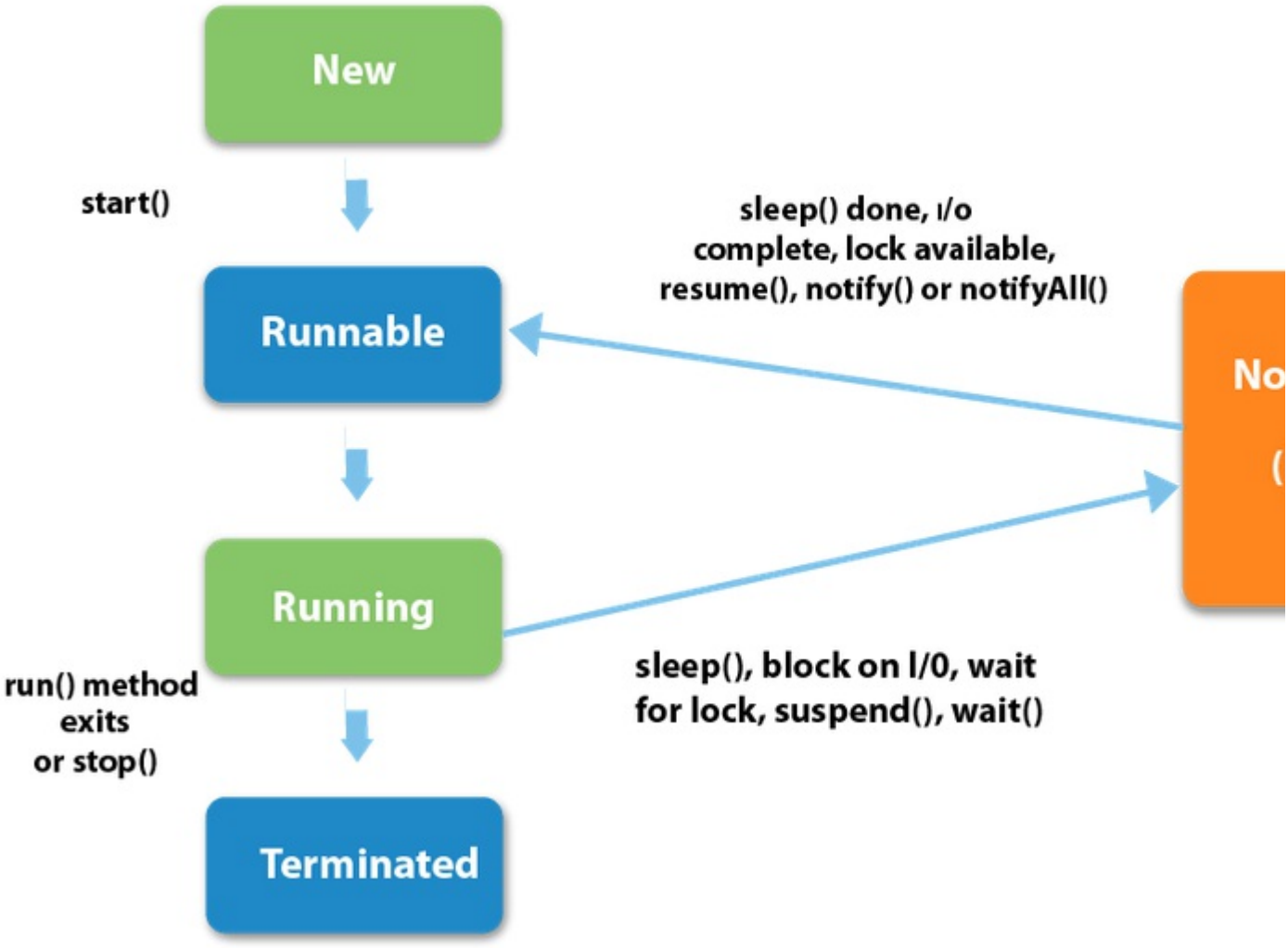
There are 5 keywords which are used in handling exceptions in Java.

Keyword	Description
try	The "try" keyword is used to specify a block where we should place exception code. The followed by either catch or finally. It means, we can't use try block alone.
catch	The "catch" block is used to handle the exception. It must be preceded by try block whi catch block alone. It can be followed by finally block later.
finally	The "finally" block is used to execute the important code of the program. It is executed handled or not.
throw	The "throw" keyword is used to throw an exception.
throws	The "throws" keyword is used to declare exceptions. It doesn't throw an exception. It s occur an exception in the method. It is always used with method signature.

There are many differences between final, finally and finalize. A list of differences between final, finally below:

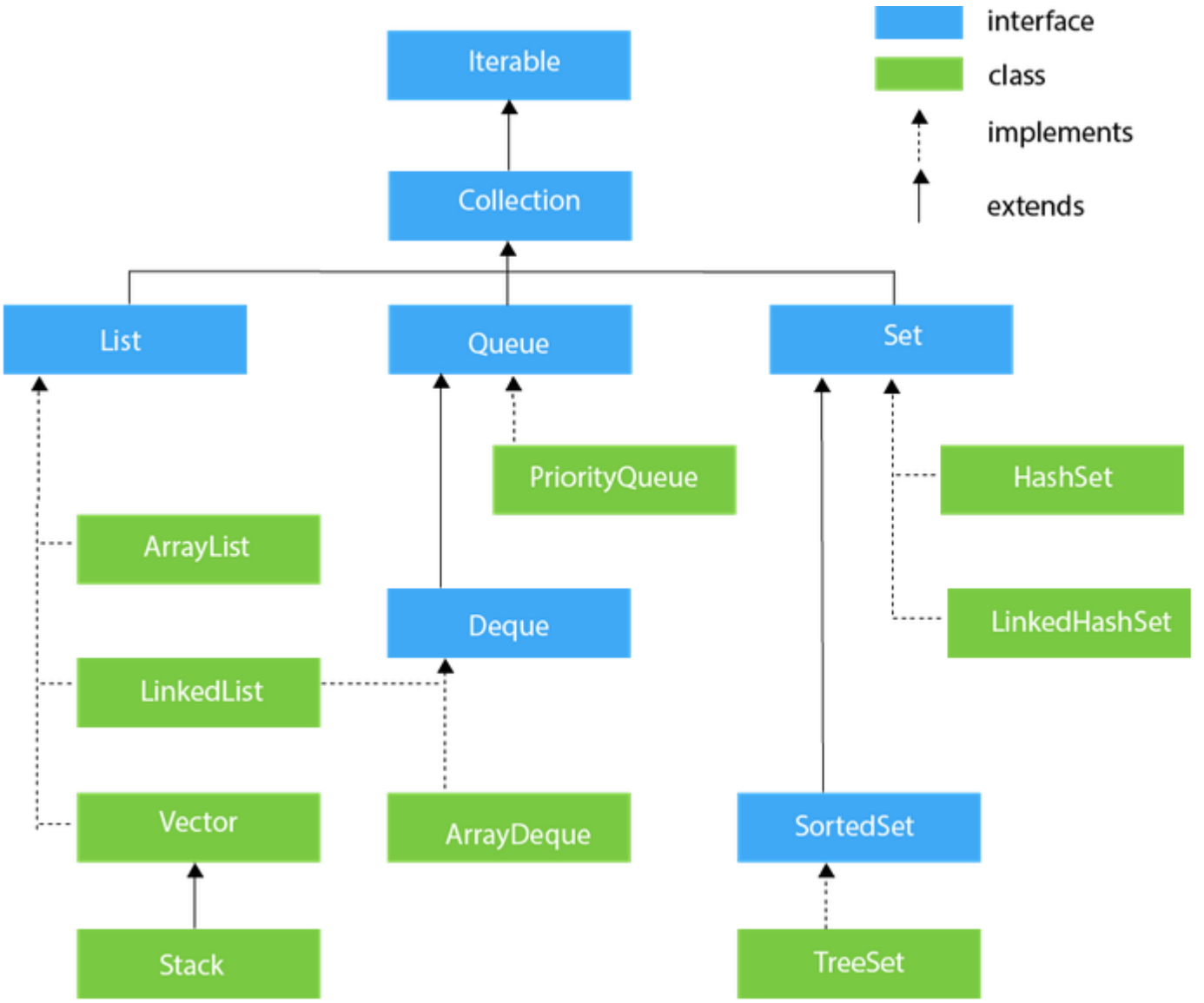
No.	final	finally	fina
1)	Final is used to apply restrictions on class, method and variable. Final class can't be inherited, final method can't be overridden and final variable value can't be changed.	Finally is used to place important code, it will be executed whether exception is handled or not.	Finali clean befor collec
2)	Final is a keyword.	Finally is a block.	Finali

Threads



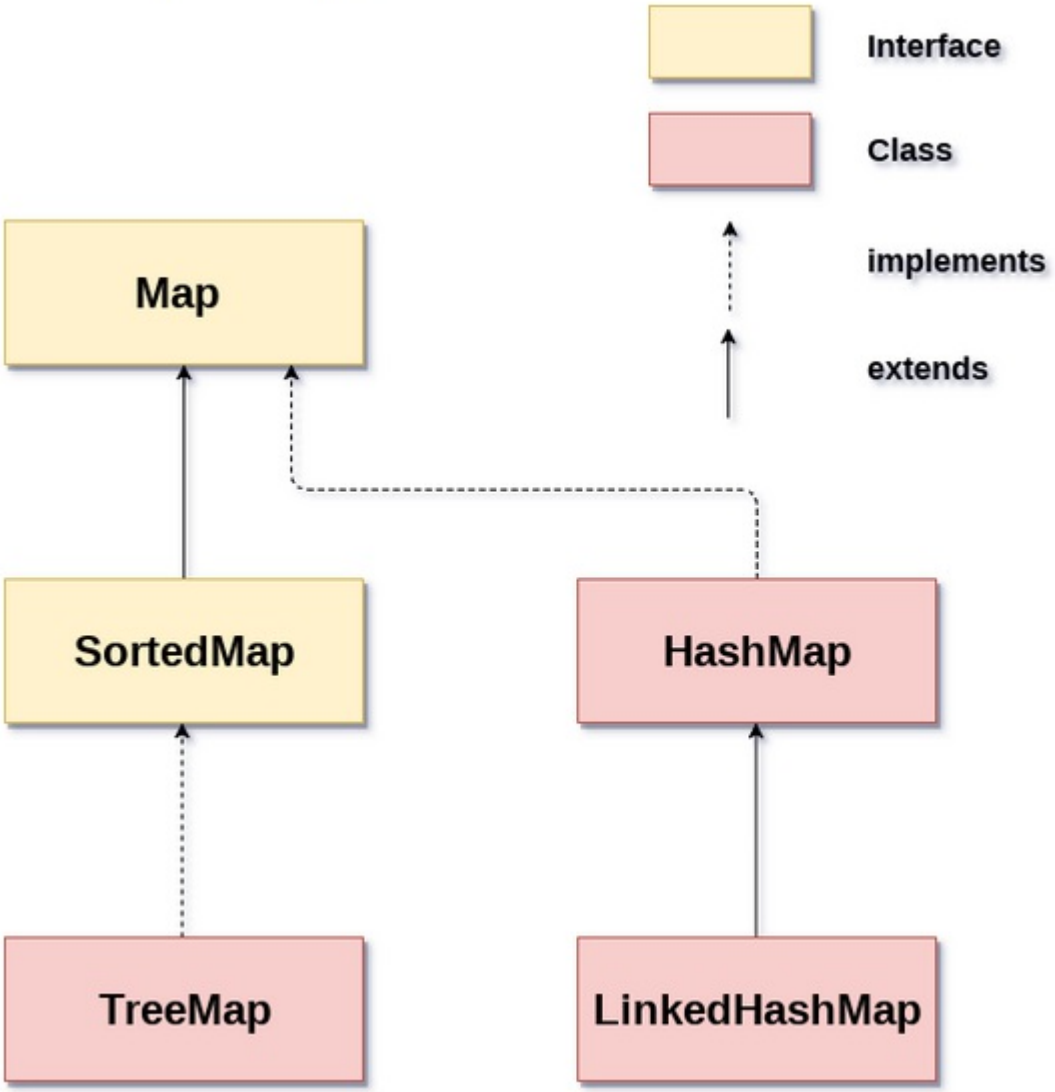
Collections

The Collection in Java is a framework that provides an architecture to store and manipulate the group of objects.


interface
class
implements
extends
Iterable
Collection
List
Queue
Set
PriorityQueue
HashSet
ArrayList
Deque
LinkedHashSet
LinkedList
Vector
ArrayDeque
SortedSet
Stack
TreeSet

There are many methods declared in the Collection interface. They are as follows:

No.	Method	Description
1	public boolean add(Object element)	is used to insert an element in this collection.
2	public boolean addAll(Collection c)	is used to insert the specified collection elements in the invoki
3	public boolean remove(Object element)	is used to delete an element from this collection.
4	public boolean removeAll(Collection c)	is used to delete all the elements of specified collection from t
5	public boolean retainAll(Collection c)	is used to delete all the elements of invoking collection except
6	public int size()	return the total number of elements in the collection.
7	public void clear()	removes the total no. of elements from the collection.
8	public boolean contains(Object element)	is used to search an element.
9	public boolean containsAll(Collection c)	is used to search the specified collection in this collection.
10	public Iterator iterator()	returns an iterator.
11	public Object[] toArray()	converts collection into array.
12	public boolean isEmpty()	checks if collection is empty.
13	public boolean equals(Object element)	matches two collections.
14	public int hashCode()	returns the hash code number of the collection.


Interface
Class
implements
extends
Map
SortedMap
HashMap
TreeMap
LinkedHashMap

Map can't be traversed so you need to convert it into Set using *keySet()* or *entrySet()* method.

Class	Description
HashMap	HashMap is the implementation of Map but it doesn't maintain any order.
LinkedHashMap	LinkedHashMap is the implementation of Map, it inherits HashMap class. It maintains in
TreeMap	TreeMap is the implementation of Map and SortedMap, it maintains ascending order.

Useful methods of Map interface

Method	Description
Object put(Object key, Object value)	It is used to insert an entry in this map.
void putAll(Map map)	It is used to insert the specified map in this map.
Object remove(Object key)	It is used to delete an entry for the specified key.
Object get(Object key)	It is used to return the value for the specified key.
boolean containsKey(Object key)	It is used to search the specified key from this map.
Set keySet()	It is used to return the Set view containing all the keys.
Set entrySet()	It is used to return the Set view containing all the keys an

---

Sql

SQL JOIN

As the name shows, JOIN means to combine something. In case of SQL, JOIN means "to combine two or more tables".

The SQL JOIN clause takes records from two or more tables in a database and combines it together.

ANSI standard SQL defines five types of JOIN :

```
inner join,
left outer join,
right outer join,
full outer join, and
cross join.
```

Why SQL JOIN is used?

```
If you want to access more than one table through a select statement.

If you want to combine two or more table then SQL JOIN statement is used .it combines

The joining of two or more tables is based on common field between them.

SQL INNER JOIN also known as simple join is the most common type of join.
```

SQL OUTER JOIN

In the SQL outer JOIN all the content of the both tables are integrated together either they are matched or not.

If you take an example of employee table

Outer join of two types:

1. ```
   Left outer join (also known as left join): this join returns all the rows from left
   ```
2. ```
   Right outer join (also known as right join): this join returns all the rows from ri
   ```

left join

The SQL left join returns all the values from the left table and it also includes matching values from right table, if there are no matching join value it returns NULL. BASIC SYNTAX FOR LEFT JOIN:

```
SELECT table1.column1, table2.column2....
FROM table1
LEFTJOIN table2
ON table1.column_field = table2.column_field;
```

right join

The SQL right join returns all the values from the rows of right table. It also includes the matched values from left table but if there is no matching in both tables, it returns NULL. Basic syntax for right join:

```
SELECT table1.column1, table2.column2.....
FROM table1
RIGHT JOIN table2
ON table1.column_field = table2.column_field;
```

full join

The SQL full join is the result of combination of both left and right outer join and the join tables have all the records from both tables. It puts NULL on the place of matches not found.

SQL full outer join and SQL join are same. generally it is known as SQL FULL JOIN. What is SQL full outer join?

SQL full outer join is used to combine the result of both left and right outer join and returns all rows (don?t care its matched or unmatched) from the both participating tables. Syntax for full outer join:

```
SELECT *
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name;
```

Cross join

When each row of first table is combined with each row from the second table, known as Cartesian join or cross join. In general words we can say that SQL CROSS JOIN returns the Cartesian product of the sets of rows from the joined table.

We can specify a CROSS JOIN in two ways:

```
Using the JOIN syntax.
the table in the FROM clause without using a WHERE clause.
```

SYNTAX of SQL Cross Join

```
SELECT * FROM [TABLE1] CROSS JOIN [TABLE2]
OR
SELECT * FROM [ TABLE1] , [TABLE2]
```

Keys

Primary

A column or columns is called primary key (PK) that uniquely identifies each row in the table.

If you want to create a primary key, you should define a PRIMARY KEY constraint when you create or modify a table.

When multiple columns are used as a primary key, it is known as composite primary key.

In designing the composite primary key, you should use as few columns as possible. It is good for storage and performance both, the more columns you use for primary key the more storage space you require.

Inn terms of performance, less data means the database can process faster. Points to remember for primary key:

```
Primary key enforces the entity integrity of the table.
Primary key always has unique data.
A primary key length cannot be exceeded than 900 bytes.
A primary key cannot have null value.
There can be no duplicate value for a primary key.
A table can contain only one primary key constraint.
```

Foriegn

In the relational databases, a foreign key is a field or a column that is used to establish a link between two tables.

In simple words you can say that, a foreign key in one table used to point primary key in another table.

Let us take an example to explain it:

Here are two tables first one is students table and second is orders table.

Here orders are given by students.

Composiite

A composite key is a combination of two or more columns in a table that can be used to uniquely identify each row in the table when the columns are combined uniqueness is guaranteed, but when it taken individually it does not guarantee uniqueness.

Sometimes more than one attributes are needed to uniquely identify an entity. A primary key that is made by the combination of more than one attribute is known as a composite key.

In other words we can say that:

Composite key is a key which is the combination of more than one field or column of a given table. It may be a candidate key or primary key.

Columns that make up the composite key can be of different data types.

Unique

A unique key is a set of one or more than one fields/columns of a table that uniquely identify a record in a database table.

You can say that it is little like primary key but it can accept only one null value and it cannot have

duplicate values.

The unique key and primary key both provide a guarantee for uniqueness for a column or a set of columns.

There is an automatically defined unique key constraint within a primary key constraint.

There may be many unique key constraints for one table, but only one PRIMARY KEY constraint for one table.

SQL UNIQUE KEY constraint on CREATE TABLE:

Alternate

Alternate key is a secondary key it can be simple to understand by an example:

Let's take an example of student it can contain NAME, ROLL NO., ID and CLASS.

Here ROLL NO. is primary key and rest of all columns like NAME, ID and CLASS are alternate keys.

If a table has more than one candidate key, one of them will become the primary key and rest of all are called alternate keys.

In simple words, you can say that any of the candidate key which is not part of primary key is called an alternate key. So when we talk about alternate key, the column may not be primary key but still it is a unique key in the column.

JavaScript

Our JavaScript Tutorial is designed for beginners and professionals both. JavaScript is used to create client-side dynamic pages.

JavaScript is an object-based scripting language which is lightweight and cross-platform.

JavaScript is not a compiled language, but it is a translated language. The JavaScript Translator (embedded in the browser) is responsible for translating the JavaScript code for the web browser. Application of JavaScript

JavaScript is used to create interactive websites. It is mainly used for:

```
Client-side validation,
Dynamic drop-down menus,
Displaying date and time,
Displaying pop-up windows and dialog boxes (like an alert dialog box, confirm dialog
Displaying clocks etc.
```

BOM

The Browser Object Model (BOM) is used to interact with the browser.

The default object of browser is window means you can call all the functions of window by specifying window or directly.

DOM

The document object represents the whole html document.

When html document is loaded in the browser, it becomes a document object. It is the root element that represents the html document. It has properties and methods. By the help of document object, we can add dynamic content to our web page.