**Java OOP**

**Classes**

1. **Access Modifiers**Public, private, default, protected
2. **State & Behaviour**

**Objects**

1. Constructor
2. This keyword
3. Static keyword
4. Object Class
5. toString method
6. Hashcode method

**4 Pillars of OOP**

**Encapsulation**

It is a *process of wrapping code and data together into a single unit*,

**Abstraction**

**Inheritance**

1. **Single Level Inheritance**
2. **Multilevel inheritance**
3. **Hierarchical Inheritance**
4. **Multiple Inheritance**
5. **Hybrid Inheritance**

**Polymorphism**

1. **Compile Time Polymorphism  
   -** Changing the no of arguments  
   - Changing the data type  
   - Type Promotion
2. **Runtime Polymorphism**It can’t be achieved with data members, only achieved with methods

**Different Relationship between the Child class and parent class (Imp)**

**Difference Between Method Overloading and Overriding**

| **1)** | **Method overloading is used *to increase the readability* of the program.** | **Method overriding is used *to provide the specific implementation* of the method that is already provided by its super class.** |
| --- | --- | --- |
| **2)** | **Method overloading is performed *within class*.** | **Method overriding occurs *in two classes* that have IS-A (inheritance) relationship.** |
| **3)** | **In case of method overloading, *parameter must be different*.** | **In case of method overriding, *parameter must be same*.** |
| **4)** | **Method overloading is the example of *compile time polymorphism*.** | **Method overriding is the example of *run time 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 same*** |

**Exception Handling**

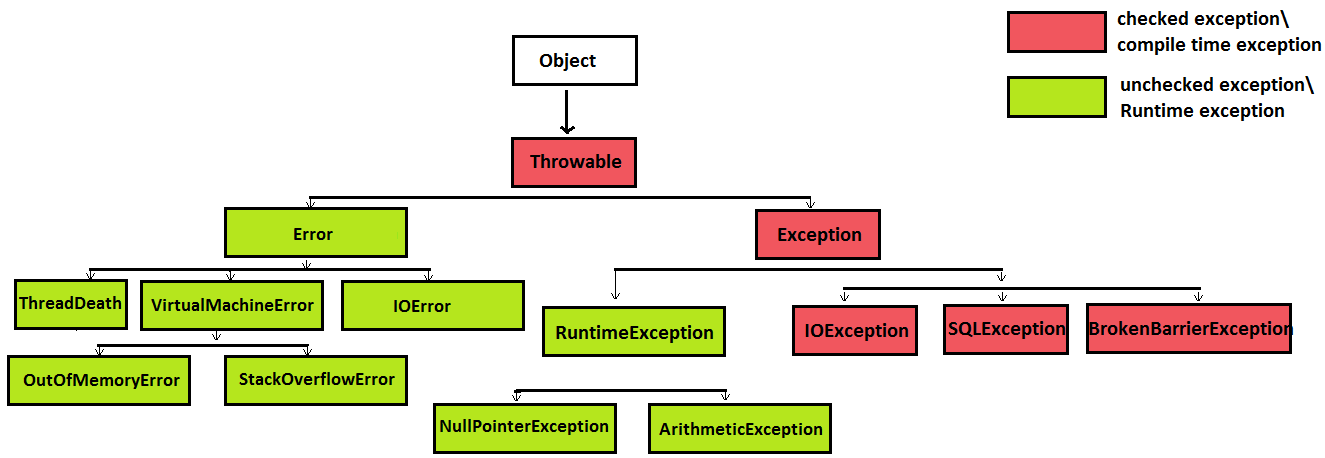
**Exception:**It is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime.

**Exception Handling**Exception Handling is a mechanism to handle errors

**Types of Exception Handling**

1. **Compile Time Exception**
2. **Runtime Exception**

**Exception Hierarchy**

****

1. Try Catch Block
2. Multiple Catch Box
3. Nested try catch
4. Finally block
5. Throw keyword  
   Only UncheckedException can be thrown without declaring it

For checked exception, it should be declare

1. Throws keyword  
   Used to declare only checked exception
2. Exception propagation: By default, Checked Exceptions are not forwarded in calling chain (propagated).

**Exception Handling with Overridden Method**

1 ) If the superclass method does not declare an exception

2 ) If the superclass method declares an exception

**Custom Exceptions**