**12. Exception Handling**

• **Theory:**

1. **Types of Exceptions** **Checked and Unchecked:-**

* **Checked exceptions:-** Checked exceptions are exceptions that are checked at compile time. The Java compiler ensures that these exceptions are either handled with a try-catch block or declared in the method signature using the throws keyword.
* **Unchecked Exceptions:-** Unchecked exceptions are exceptions that are not checked at compile timwwe. These exceptions occur during runtime, and the Java compiler does not require them to be handled or declared.

**2.try, catch, finally, throw, throws.**

* **1.try:-** The try block is used to enclose the code that might throw an exception. If an exception occurs within the try block, control is transferred to the corresponding catch block.
* **2. catch:-** The catch block is used to handle exceptions that are thrown in the try block. You can have multiple catch blocks to handle different types of exceptions.
* **3. finally:-** The finally block is used to execute code that must run regardless of whether an exception was thrown or caught. This is useful for releasing resources, such as closing files or network connections.
* **4. throw:-** The throw keyword is used to explicitly throw an exception. You can throw a built-in exception or create your own custom exception class.
* **5. throws**:- The throws keyword is used in a method signature to declare that a method can throw one or more exceptions. This informs callers of the method that they should handle those exceptions.

**3.Custom Exception Classes:- efinition of Custom Exception Classes:-** in Java are user-defined exceptions that extend the existing exception hierarchy, allowing developers to create specific error types tailored to their application's needs. They enable more granular error handling by providing meaningful and contextual information about exceptional conditions that may occur during program execution.