JA111

Day-12

Nested try block

- try inside another try block
- The nested try block has its own catch block to handle the exception; if the catch block of nested try is not able to handle the exception then the catch block of outer try will handle it.
- If a method with try-catch in its body is called within try block then it is also equivalent to nested-try block.

The finally block

- It is executed always no matter exception takes place or not
- It will be executed even when control return from try block.
- If JVM halts then only it will not be executed.
- It is used for resource recovery.

The try-with resource statement

- The try-with-resource statement is a try block with one or more resource, resource refers to an object that must be closed after program finishes. The try-with-resource statement ensures that resources must be closed after try-with-resource finishes.
- Semi-colon(;) serve as a separator for multiple resources within try-with-resource statement.

The throw and throws statement

It is used to throw the exception; JVM uses throw keyword to throw exception to our program when something abnormal takes place during the program execution.

We can have exception object in two ways

a. Grab it in catch block as it is created by JVM itself due to error in try block code
b. We can create object using new keyword also because every exception type is a class

The throws is used to inform caller about the exceptions that a method can throw

If a method is throwing checked exception then it is mandatory to write the exception-type in the throws clause; no such restriction exists for unchecked type.

Calling of method that throws checked exception must be either inside the try-catch block in caller or the caller should re-throw the exception again; no such restriction for unchecked exception.

Creating Custom Exceptions

Create subclass of Exception class

Provide a public constructor with String arguments

Provide definition of public String toString() method

Some important Points

- **If a method in super class throws no exception/unchecked exception then while overriding in sub-class method cannot throw a checked exception. Yet it can throw unchecked exception or no exception**
- **If a method in super class throws checked exception then while overriding in sub-class method can throw checked exception of same of sub-type but not super-type.**