Try Catch

Throw

Rethrow

Global Exception Handler

Exception Types: System, Business, Application, Null reference, Selector not found, …

Q: How many catch blocks we can have in an Try catch? – should have atleast one catch block

Q: Selector not found exception, System exception, Business rule exception, Null reference, IO exception

Q: Order execution in Try Catch

1. Any exception occurred in try block then catch block executed.
2. Irrespective any exception occurred in try block finally will block execute
3. Any exception occurred in catch block, finally block will not execute.

Q: Difference between Throw & Re Throw?

Throw – If we want throw an exception with custom exception type, and message we will throw activity (e.g: new <ExceptionType>(<Exception Message>)

Re-Throw – To Throw any cached exception we will use rethrow and this doesn’t have input

Rethrow can be used on in the catch block whereas throw can be used anywhere Seq, Try, catch, finally

Q: Global Exception Handler: Any unhandled exception will catch by global exception handler, we can use only one global exception handler for a project.

Ignore, Retry, Continue

Debugging

Errors – bugs

To identify the bugs / errors / mis functioning we use the methodology of debigging

Debugging - step by step execution

We can the values of variables / arguments during the execution, if it is not as expected – we will mark the bug, we will re assign a new expected value to continue.

We can evaluate expressions to know the result

Options

Break point – Workflow execution will be paused at breakpoint.

Break – Manually pause the execution at some activity.

Continue – Continue it’s execution till the next breakpoint / end of execution

Step Into – Executes each an every activity step step and pause for each activity and this will go inside of the invoked workflow

Step Over – Skip the step by step execution and perform normal execution for an invoked workflow

Step out – comes to parental seq and executes all the subsequent activities in the given sequence.