Exception Handling Clean Code Practices

# @Author: Madhusudan Rathore

# Design classes so that exceptions can be avoided

A class can provide methods or properties that enable you to avoid making a call that would trigger an exception.

Another way to avoid exceptions is to return null (or default) for extremely common error cases instead of throwing an exception. An extremely common error case can be considered normal flow of control. By returning null (or default) in these cases, you minimize the performance impact to an app.

# Use multiple catch block instead of if conditions.

If you need to act according to type of the exception, you better use multiple catch block for exception handling.

**Bad:**

1. **try**
2. {
3. // Do something..
4. }
5. **catch** (Exception ex)
6. {
8. **if** (ex **is** TaskCanceledException)
9. {
10. // Take action for TaskCanceledException
11. }
12. **else** **if** (ex **is** TaskSchedulerException)
13. {
14. // Take action for TaskSchedulerException
15. }
16. }

**Good:**

1. **try**
2. {
3. // Do something..
4. }
5. **catch** (TaskCanceledException ex)
6. {
7. // Take action for TaskCanceledException
8. }
9. **catch** (TaskSchedulerException ex)
10. {
11. // Take action for TaskSchedulerException
12. }

# Use exception builder methods

It is common for a class to throw the same exception from different places in its implementation. To avoid excessive code, use helper methods that create the exception and return it. For example:

1. **class** FileReader
2. {
3. **private** **string** fileName;
5. **public** FileReader(**string** path)
6. {
7. fileName = path;
8. }
10. **public** **byte**[] Read(**int** bytes)
11. {
12. **byte**[] results = FileUtils.ReadFromFile(fileName, bytes);
13. **if** (results == **null**)
14. {
15. **throw** NewFileIOException();
16. }
17. **return** results;
18. }
20. FileReaderException NewFileIOException()
21. {
22. **string** description = "My NewFileIOException Description";
24. **return** **new** FileReaderException(description);
25. }
26. }