DISPLAY ALL THE SPORTS ACTIVITIES AVAILABLE IN YOUR CITY FOR COMING WEEKEND

Table of Contents

1. Introduction:

To build good applications, which are reliable, scalable and maintainable, it is important for QA teams to adopt proven design techniques and good coding standards. The adoption of coding standards results in code consistency, which makes it easier to understand, develop and maintain the application.

The best practices are primarily targeted towards improvement in the readability and maintainability of code with keen attention to performance enhancements. By doing such practices the application tester team can demonstrate them proficiency, profound knowledge and professionalism.

This Guidelines document is written for QA to help them:

- Write Java code that is easy to maintain and enhance
- Increase their maintainability

2. Source File Basic

2.1 Beginning Comments

The header should be followed by the package and import statements and then the documentation comments exactly in the following sequence and indented to the same level.

2.2 Filename

The source file name consists of the case-sensitive name of the top-level class it Contains, plus the .java extension.

2.3 Source File Structure

A source file consists of, in order:

- Package Name.
- Imports.
- Class definition

2.4 No Wild Card Imports

2.5 Name Convention

Wildcard imports, static or otherwise, are not used.

2.5 Indentation

Four spaces should be used as the unit of indentation. The indentation pattern should be consistently followed throughout.

2.6 Name Convention

Naming conventions make programs more understandable by making them easier to read. Following conventions should be followed while naming a class or a member:

- Use full English descriptors that accurately describe the variable, method or class. For example, use of names like total Sales, current Date instead of names like x1, y1.
- Abbreviations should not be used as far as possible, but if used, should be documented and consistently used.

3- Exception handling

You must follow the below Java coding guidelines for implementing effective exception handling.

1- Always write a catch block for handling exceptions.

2- Make sure to add a logging message or the stack trace in the catch block.

3- Avoid catching the general exception and have a specific exception.

4- The clean-up code should be added in the <finally> block.

5- This provides a single location for the clean-up, and it's guaranteed to run.

4- Tools & Plugins

To implement Java coding guidelines, must use the Eclipse IDE And it's easy to integrate these with Eclipse IDE.

1- Selenium for testing web application

2- TestNG for unit testing.

3- Apache POI for PageObjectModel

4. Maven for building the project.

5- Jenkins for Continuous integrations

6. MS Excel for data driven concept.

4.1 Project Management

- **Trello** for collaboration tool that organizes projects.

4.2 Backup

- GitHub for Source Code Management & Backup.