Audit Online Coding Guideline

* Naming Conventions
  + “c” = camelCase
  + “P” = PascalCase
  + “\_” = Prefix with \_Underscore
  + “x” = Not Applicable.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | **Public** | **Protected** | **Internal** | **Private** | **Notes** |
| Project File | **P** | x | x | x | Match Assembly & Namespace. |
| Source File | **P** | x | x | x | Match contained class. |
| Other Files | **P** | x | x | x | Apply where possible. |
| Namespace | **P** | x | x | x | Partial Project/Assembly match. |
| Class or Struct | **P** | **P** | **P** | **P** |  |
| Interface | **P** | **P** | **P** | **P** | Prefix with a capital I. |
| Generic Class | **P** | **P** | **P** | **P** | Use T or K as Type identifier. |
| Method | **P** | **P** | **P** | **P** | Use a Verb or Verb-Object pair. |
| Property | **P** | **P** | **P** | **P** | Do not prefix with Get or Set. |
| Field | **P** | **P** | **P** | **\_c** | Only use Private fields. |
| Constant | **P** | **P** | **P** | **\_c** |  |
| Static Field | **P** | **P** | **P** | **\_c** | Only use Private fields. |
| Enum | **P** | **P** | **P** | **P** | Options are also PascalCase. |
| Delegate | **P** | **P** | **P** | **P** |  |
| Event | **P** | **P** | **P** | **P** | Might not be applicable to our API project but whenever creating any other project |
| Inline Variable | x | x | x | **c** | Avoid single-character and enumerated names. Text1, Text2… |
| Parameter | x | x | x | **c** |  |

* + Guidelines around naming
    - Always use Camel Case or Pascal Case names.
    - Always choose meaningful and specific names
    - Variables and Properties should describe an entity not the type or size.
      * strProcedureName
    - Avoid short names and abbreviation unless the full name is too long
      * Use abbreviation commonly known
    - Try to prefix Boolean variables and properties with “Can”, “Is” or “Has”.
* Coding Guideline

|  |  |
| --- | --- |
| **Code** |  |
| **Variables** | * Only declare member variables as private. * Use properties to provide access to them with public, protected, or internal access modifiers. * Use built-in C# native data types (Use int NOT Int32) * Avoid using inline numeric literals (magic numbers). Instead, use a Constant or Enum. * Always cecorate DTO string property with “AuditOnlineText” attribute   + This for Veracode scan |
| **Flow Control** | * Use the **ternary** conditional operator only for trivial conditions   + Var result = procedure == null ? x : y * Avoid explicit Boolean test in conditional   //Do Not  if(IsValid == true)  {…};  //DO  if(IsValid)  {…}   * Avoid Complex expression   //Do Not  if (((hr > \_minHr) && (hr != \_minHr)) && (hr < \_maxHr))  {…}  //DO  isHighHr = (hr >= \_minHr);  isLowHr = (hr == \_minHr);  isApprovedHr = (hr < \_maxHr);  if ((isHighHr && ! isApprovedHr) && isApprovedHr)  {…} |
| **String Comparison** | * Prefer String.Format() or StringBuilder over string concatenation. * Never concatenate strings inside a loop. * Use String.Compare() for string comparision   **Example**: (*ToLower() creates a temp string*) |
| **Exceptions** | * Do not use exceptions for flow control. * Use throw; not throw e; when re-throwing. Only catch what you can handle. * Use validation to avoid exceptions. * Derive from Execption not ApplicationException. * Never declare an empty catch block. |