2022

# Programming 271

Ofentse Mapheto

Assignment 1

7/27/2022

# Contents

# CODING STANDARDS

It is important to establish and follow good and best coding standards. The intent of creating a of coding standards is to have an acceptable list of guidelines that should be followed by all members of the team.

## Naming Standards

Of the relative multitude of parts that make up a coding standard, naming guidelines are the most noticeable and ostensibly the most significant. Having a predictable norm for naming the different items in your program will save you a tremendous measure of time both during the advancement cycle itself and furthermore during any later upkeep work. The following Three conventions are to be used for capitalizing identifiers:

## Pascal Case

The first letter in the identifier and the first letter of each subsequent concatenated word are capitalized. You can use Pascal case for identifiers of three or more characters.

For example:

BackColor

## Camel Case

The first letter of an identifier is lowercase, and the first letter of each subsequent concatenated word is capitalized.

For example:

backColor

## Uppercase

All letters in the identifier are capitalized. Use this convention only for identifiers that consist of two or fewer letters.

For example:

System.IO

System.Web.IO

# Class Naming Standards

A class is the most widely recognized sort of type. A class can be dynamic or fixed. A theoretical class requires a determined class to give an execution. A fixed class doesn't permit an inferred class. It is suggested that you use classes over different kinds. Base classes are a valuable method for gathering objects that share a typical arrangement of usefulness. Base classes can give a default set of usefulness, while permitting customization however expansion.

For example:

```
public class ClientActivity
{
 public void ClearStatistics()
   {
      //...
   }
   public void CalculateStatistics()
   {
      //...
   }
}
```

# Variable standards

A variable is only a name given to a capacity region that our projects can control. Every variable in C# has a particular sort, which decides the size and format of the variable's memory the scope of values that can be put away inside that memory and the arrangement of tasks that can be applied to the variable.

For example:

short NOT System.Int16

int NOT System.Int32

long NOT System.Int64

string NOT System.String

# Flow Control

The flow of execution refers to the order in which the code of a program is executed or evaluated. In C#, there are a variety of constructs for writing loops and for deciding which code to execute based on the value of inputs. For a great flow control avoid invoking methods within

a conditional expression. Avoid creating recursive methods. Use loops or nested loops instead. Avoid evaluating Boolean conditions against true or false.

For example:

```
// Bad!

if (isValid == true)

{…}

// Good!

if (isValid)

{…}.
```

# Exceptions

Special cases permit an application to move control starting with one piece of the code then onto the next. At the point when a special case is tossed, the ongoing progression of the code is intruded on and given back to a parent attempt get block. C# special case taking care of is finished with the follow watchwords: attempt, get, at long last, and toss. If re-throwing an exception, preserve the original call stack by omitting the exception argument from the throw statement.

For example:

```
// Bad!

catch(Exception ex)

{

 Log(ex);

throw ex;

}

// Good!

catch(Exception)

{

 Log(ex);

throw;

}
```

# Reference

Jiang, Z., Zhao, Y., Qin, Z. and Lin, X., 2013, June. Coding Standard Based Object Oriented Programming Course Teaching Reform. In 2013 International Conference on Computational and Information Sciences (pp. 1890-1892). IEEE.

Hartog, V. and Doomen, D., 2005. Coding Standard: C. Philips Medical Systems-Software/SPI http://www. tiobe. com/content/paperinfo/gemrcsharpcs. pdf.

Sartain, S., 2006. C#/VB .NET Coding Guidelines.

Hunt, L., 2007. Coding Standards for .NET. Document Version, 1.

Lowy, J., 2008. C# Coding Standard. Prírucka správnej syntaxe pre jazyk C# [online]. Dostupné z: http://ie. archive. ubuntu. com/disk1/disk1/download. sourceforge. net/pub/sourceforge/c/project/cy/cymrucrm/Standards% 20Documents/IDesign% 20C, 23.