# Software Engineering (CS301)
# Coding Conventions(Version 1)
# Travel Diaries

Group 5

November 14, 2016

**Project Members**

| ID | Name |
|---|---|
| 201452004 | Nilesh Chaturvedi |
| 201452005 | Jitendra Singh |
| 201452012 | Durga Vijaya Lakshmi |
| 201452036 | Pedapalli Akhil |
| 20152040 | B. Indu |
| 201452044 | Dileep Krishna |
| 201452050 | Shreya Singh |
| 201452056 | Ravi Patel |
| 201452057 | G. Raju Koushik |

| | |
|---|---|
| **Authored By** | **K.Durga Vijaya Lakshmi** |
| **Reviewed By** | **Raju Koushik** |

# Contents

# 1 Introduction

## 1.1 Purpose

The goal of these guidelines is to create uniform coding habits among software personnel in the engineering department so that reading, checking, and maintaining code written by different persons becomes easier. The intent of these standards is to define a natural style and consistency, yet leave to the authors of the engineering department source code, the freedom to practice their craft without unnecessary burden.

When a project adheres to common standards many good things happen:
Programmers can go into any code and figure out whats going on, so maintainability, readability, and reusability are increased. Code walk throughs become less painful.New people can get up to speed quickly.
People new to a language are spared the need to develop a personal style and defend it to death.People new to a language are spared making the same mistakes over and over again, so reliability is increased.

People make fewer mistakes in consistent environments.

## 1.2 Scope

This document describes general software coding standards for code written in any text based programming language (including high-level languages like java, python languages). Many of the guidelines described in this document can be directly applied to programming practices in android app development.This will be used as the base document for several language specific coding standard documents.Each language specific coding standard will be written to expand on these concepts with specific examples, and define additional guidelines unique to that language.

For each project, this document will be used in conjunction with language and project specific coding standards that, in total define a complete set of coding standards. A description of the general, language specific, and project specific coding standards is provided below:

## 1.3 Coding Standard Documents:

Each project shall adopt a set of coding standards consisting of three parts:

**General Coding Standard**,described in this document.
**Language specific coding** standards for each language used, described in sep-

arate appendices to this document. These language standards shall supplement, rather than override, the General Coding standards as much as possible.
**Project Coding Standards**. These standards shall be based on the coding standards in this document and on the coding standards for the given language(s). The project coding standards should supplement, rather than override, the General Coding standards and the language coding standards. Where conflicts between documents exist, the project standard shall be considered correct. Sweeping per-project customizations of the standards are discouraged, so that code can be reused from one project to another with minimal change.

## 1.4 Terms Used In This Document

- A "def" is a program unit whose primary purpose is to return a value.
- A "class" is the single, definitive source of information about your data. It contains the essential fields and behaviors of the data you're storing.
- An "identifier" is the generic term referring to a name for any constant, variable, or program unit.
- An "activity" is a building block of the User Interface.

## 1.5 Our Limited Lifetime Warranty

If these standards - when used as directed - fail to perform as expected, they can be edited and adapted to changing environments, applications, business emphasis, and an ever-evolving industry.The spirit of this document, not its rules, should dictate the place of standards and consistency within and across projects.

## 1.6 Coding Standards

Each project shall adopt a set of coding standards consisting of two parts:

- General Coding Standard, described in this document.
- Language specific coding standards for each language used. These language standards shall supplement, rather than override, the General Coding standards as much as possible.

# 2 Code Layouts

## 2.1 Indentation

Indentation is a compulsory tool in python.Without indentation there will be errors.The Python files use 4 spaces for indentation.
Continuation lines should align wrapped elements either vertically using Python's implicit line joining inside parentheses, brackets and braces, or using a hanging indent. When using a hanging indent the following should be considered; there should be no arguments on the first line and further indentation should be used to clearly distinguish itself as a continuation line.
Use underscores, not camelCase, for variable, function and method names.
Use InitialCaps for class names (or for factory functions that return classes)

## 2.2 Line Spaces

There will be a gap of 2 lines between two functions and two classes.

## 2.3 Error Handling

- Functions that can fail should always return a success or error as a return code parameter.
- Any time a subroutine calls a function that returns an error condition, the error condition should be tested for and acted on in accordance with the error handling conventions. Error recovery should e handled in the routine that is responsible for the domain in which the error occurs.

# 3 Naming conventions

## 3.1 Selecting names

The most important consideration in naming a variable, constant, or subroutine is that the name fully and accurately describe the entity or action that the structure represents. Clear, complete, and meaningful names makes the code more readable and minimizes the need for comments.

## 3.2 Naming Functions

All the functions should be in lower case and there would be a underscore if there are multiple words in a function.
Examples :post_dairy,get_dairy posts , get_followers etc.

### 3.3 Naming Classes

All the classes are written in camelCase the starting letter of all the words are written capital.
Examples :UserInfo, Diary

### 3.4 Naming Variables

Names of constants, variables, and functions shall be nouns, with or without modifiers, with the exception of Boolean identifiers as noted below.