**Good Programming Practices**

In this document I am going to discuss certain practices that will make your code more readable and understandable to you and your team and even to some outsider.

Let us start

1. **Use Descriptive Names**

Always try to use descriptive names that describe the functionality or use of Identifier. Avoid use of names such as var, var1, a. For e.g. If a function will take name as input from user we should name it getName().

Other examples are customerName,studentMarks.

1. **Use Naming Conventions**

In java we use certain conventions in naming variables that are: -

* The name of public methods and instance variables is stared with small letter and if another word is used then the other word is started with a Capital letter.

For example, getMarks, name, studentName, id, customerId.

* All private methods and local variables should be named using only small letters and subsequent words are separated by underscore.

For example, marks, length, triangle\_area, square\_area.

* All classes and interfaces should be named using labels that have first letter capital and the subsequent words must start with capital letters.

For Example, HelloWorld, Producer, BankingSystem.

* Variables that represent constant values use only capital values and underscore between words.

For Example, SAVINGS\_INTEREST\_RATE, PI.

1. **Give Each Class/Function a Specific Purpose**

Each class or function should have specific purpose i.e, we should not place the whole code inside a single function , and only one class should not have all the functions the functions should we distributed amongst several classes according to its functionality. Large chunks of code should we divided into small functions that perform a particular function.

1. **Keep the code simple**

The code that a programmer writes should be simple. Complicated logic for achieving a simple thing should be kept to a minimum since the code might be modified by another programmer in the future. The logic one programmer implemented may not make perfect sense to another. So, always keep the code as simple as possible. Simpler the code less is the time needed to debug it.

1. **Write Good Comments**

 Comments will help you and others remember what a section of code does. It will also increase overall readability. We should understand that commenting is good but over commenting is not good in the comments we should specify WHY a certain code exists rather than WHAT a certain code does.

Comments can be good for warnings (i.e. “removing this will break A, B, and C”) but for the most part should uncover things that can’t be immediately understood from the code.

1. **Use Proper Architecture**

There should be proper indentation in the code as proper indentation increases the readability of the code and the code should not contain large vertical lines of code. Suitable use of braces to separate blocks of code inside a single function.

For example

class Test{

int marks;

int noOfQuestions;

public void getResult()

{System.out.println("print the student result");

}

public static void main(String args[])

{Test t = new Test();

System.out.print(t.marks+" "+t.noOfQuestions);

t.getResult();

}

}

Without proper indentation

class Test

{

int marks;

int noOfQuestions;

public void getResult()

{

System.out.println("print the student result");

}

public static void main(String args[])

{

Test t = new Test();

System.out.print(t.marks+" "+t.noOfQuestions);

t.getResult();

}

}

With proper indentation

Now as we can see the code with proper indentation is more readable .