CODING STANDARDS AND DOCUMENTED CODE

Library Management System
Group 8

Coding Standards incorporated in the project:

1. Naming variables:

   Assigning names to all variables, functions, and methods makes the code easier to read, understand, and maintain. In using a meaningful name, the code will explain itself to anyone who reads it. This will cut down on maintenance costs and confusion when making changes. It is important to be as specific as possible with these names. The following has been incorporated in our project code as a part of naming variables:

   - Use of lower case letters to name the variables.
   - Used commonly spoken language for the names
   - Use of '_' to separate the words in a variable.
   - Static variable names have been started with a letter 's'.
   - Too much additional text can result in confusion and long, complex code hence shorter and clearer variable and function names have been used
   - Use of upper-case letters to define global constants with '_' as a separator.

Formatting (way of arranging a program in order to enhance readability) consists of indentation, alignment, and use of white spaces in the program. Consistency plays an important role while formatting a program in an organized way. A program with consistent formatting makes the code easier to read and understand. The commonly used formatting conventions are listed below.

2. Block Alignment and Indentation**:**

   This refers to one or more spaces left at the beginning of statements in the program. Indentation is useful in making the code easily readable. However, the spaces used for indentation should be followed in the entire program.
   White spaces improve readability by minimizing the compactness of the code.

The following principles have been taken into consideration regarding the indentation:

a. Indentation should be used to highlight a nested block. Some nested blocks can be made with the help of 'if-else' and 'do-while' loops.
b. Indentation is required if the statement is large enough to fit in a single line.
c. Indentation should be consistent at the beginning and at the end of the braces in the program.

- There should be a space after placing a comma between two function arguments.
- There should be no space between a function name and parenthesis.
- There should be spaces to align the operators vertically to emphasize program structure and semantics.

3. Commenting **:**

Trailing comments are used to provide an explanation of a single line of code. These comments are used to clarify the complex code. These also specify the function of the abbreviated variable names that are not clear. In some languages, trailing comments are used with the help of a double slash (//). The commenting conventions that are commonly followed in our software code:

- Comments should not be used to include information that is clearly understandable from the software.
- Comments should be used with important segments of code and code segments that are difficult to understand.
- Comments should be separated from the code to enhance the readability of the software code.

```
<?php include('includes/footer.php');?>
    <!-- FOOTER SECTION END-->
    <!-- JAVASCRIPT FILES PLACED AT THE BOTTOM TO REDUCE THE LOADING TIME  -->
    <!-- CORE JQUERY  -->
    <script src="assets/js/jquery-1.10.2.js"></script>
    <!-- BOOTSTRAP SCRIPTS  -->
    <script src="assets/js/bootstrap.js"></script>
    <!-- CUSTOM SCRIPTS  -->
    <script src="assets/js/custom.js"></script>
</body>
</html>
<?php } ?>
```

4.Function calls **:**While writing a function call statement, there must be no space between the function name and the opening parenthesis. Also, the name of the functions must be simple and easy to understand.
Example: session_start()

```php
<?php
session_start();
$_SESSION = array();
if (ini_get("session.use_cookies")) {
    $params = session_get_cookie_params();
    setcookie(session_name(), '', time() - 60*60,
        $params["path"], $params["domain"],
        $params["secure"], $params["httponly"]
    );
}
unset($_SESSION['login']);
session_destroy(); // destroy session
header("location:../adminlogin.php");
?>
```

5. Structuring the control flow statements**:** The control flow or conditional statements must be written in such a way so that it could be differentiated from function call statements. While writing if, for, while, switch and other control flow statements there must be one space between the keyword and the opening parenthesis.

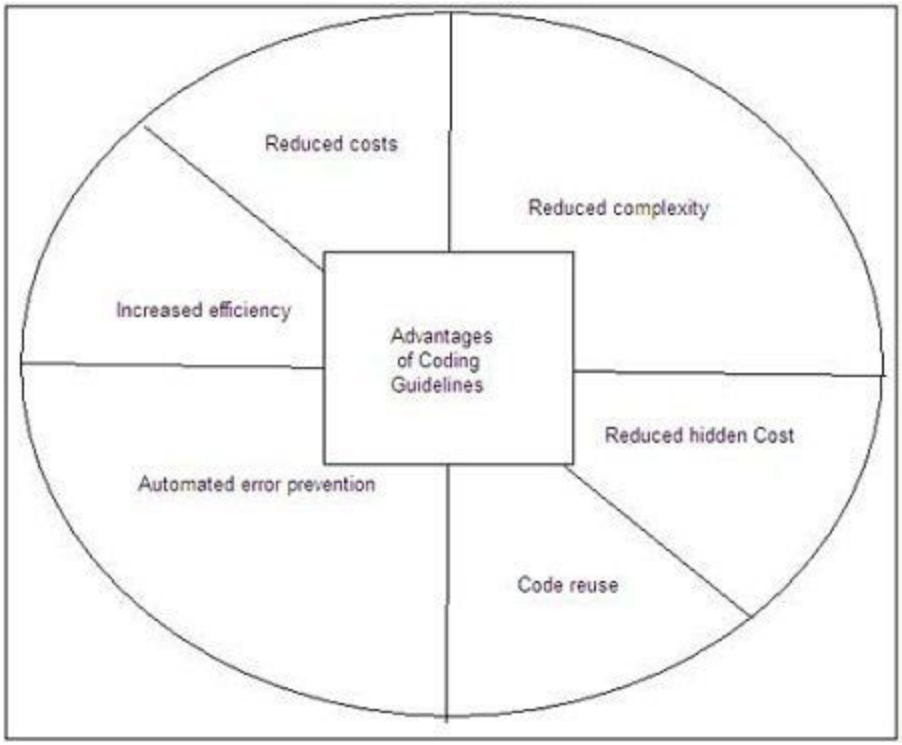
6.PHP tags : One must use the PHP standard tags(), rather than the shorthand tags() to delimit the PHP code.

7. Coding for Efficiency vs. Coding for Readability:There are many aspects to programming. These include writing software that runs efficiently and writing software that is easy to maintain. These two goals often collide with each other. Creating code that runs as efficiently as possible often means writing code that uses tricky logic and complex algorithms, code that can be hard to follow and maintain even with ample inline comments. The programmer needs to carefully weigh efficiency gains versus program complexity and readability. If a more complicated algorithm offers only small

gains in the speed of a program, the programmer should consider using a simpler algorithm. Although slower, the simpler algorithm will be easier for other programmers to understand.

By incorporating the proper coding standards one would be able to achieve many advantages which are as follows:

Coding guidelines supplement the language standard by defining acceptable and unacceptable usage of the programming language used. Acceptable usage avoids troublesome situations while unacceptable usage is conducive to errors or leads to a misunderstanding of the written code. Properly implemented coding guidelines help the developer to limit program complexity, establish the basis for code review, and guard against compiler and common programming errors. Other advantages associated with coding guidelines are listed below and depicted.



- Increased efficiency: Coding guidelines can be used effectively to save time spent on gathering unnecessary details. These guidelines increase the efficiency of the software team while the software development phase is carried out. Efficient software code is fast and economical. Software coding guidelines are used to increase efficiency by making the team productive, thus, ensuring that the software is delivered to the user on time.
- Reduced costs: Coding guidelines are beneficial in reducing the cost incurred on the software project. This is possible since coding guidelines help in detecting errors in the early stages of software development. Note that if errors are discovered after the software is delivered to the user, the process of rectifying them becomes expensive as additional costs are incurred on late detection, rework, and retesting of the entire software code.
- Reduced complexity: The written software code can be either simple or complex. Generally, it is observed that a complex segment of software code is more susceptible to errors than a segment containing a simple software code. This is because a complex software code reduces readability as well as understandability. In addition, the complex software code may be inefficient in functioning and use of resources. However, if the code is written using the given coding guidelines, the problem of complexity can be significantly avoided as the probability of error occurrence reduces substantially.
- Reduced hidden costs: Coding guidelines, if adhered to in a proper manner, help to achieve a high-quality software code. The software quality determines the efficiency of the software. Software quality is the degree to which user requirements are accomplished in the software along with conformity to standards. Note that if the quality is not considered while developing the software, the cost for activities such as fixing errors, redesigning the software, and providing technical support increases considerably.
- Code reuse: Using coding guidelines, software developers are able to write a code that is more robust and create individual modules of the software code. The reason for making a separate code segment is to enable reusability of the modules used in the software. A reusable module can be used a number of times in different modules in one or more software.
- Automated error prevention: The coding guidelines enable Automated Error Prevention (AEP).This assures that each time error occurs in software, the software development activity is improved to prevent similar errors in the future. AEP begins with detecting errors in the software, isolating its cause, and then searching the cause of error generation. Coding guidelines are useful in preventing errors as they allow the implementation of requirements that prevent the most common and damaging errors in the software code.

In addition to the above-mentioned advantages, coding guidelines define appropriate metric thresholds. These thresholds help in reducing complexity, thus, minimizing the occurrence of errors. Software developers face increasing demands to demonstrate that development practices meet the accepted coding guidelines. This is essential for companies developing safety-critical software as well as those seeking CMM and ISO certification.