Heaven's Light Is Our Guide

Rajshahi University of Engineering and Technology, Rajshahi.

Course code: ECE 3117

Course title: Software Engineering and Information System Design

Lab Report 2

Submitted to:	Submitted by:
Oishi Jyoti	Md.Ashikul Islam
Assistant Professor,	Roll:2010019
Department of ECE,RUET	

Task: Best Coding Practices.

Objectives:

- To learn techniques that make code clear and understandable for others and future reference.
- To follow standardized coding conventions and style guides for maintaining uniformity in code structure.
- To understand the importance of breaking down code into smaller so that it can be managed easily.
- practice writing meaningful comments for better explanation of the code's purpose and functionality**.**

Theory:

Best coding practices is essential for producing high-quality, reliable, and scalable software. These practices help improve collaboration, minimize bugs, and make code easier to maintain. Here are some best coding practices :

1. **Choosing meaningful variable and function names:** We should choose name for variables and fuctions accoding to their related purpose.

Bad practice:

x = 100

def f(y): return y * 3

Good Practice:

max_users = 100

def calculate_triple(number): return number * 3

2. **Camel case vs snake case:** There are two generally accepted conventions for creating variable or function names: camel case and snake case. Camel case uses capital letters to separate words in a variable name. Snake case uses underscores to separate words in variables. For example, we

would have the variable name “accountNumber” in camel case and “account_number” in snake case.

whichever case we choose, it is important to stick with it throughout our entire project. Switching between different naming conventions can be visually confusing.

3. Comment and Document Code: To to make code more readable we should add descriptive comments throughout. Good commenting will ensure code is comprehensible by someone else. Comments shoul be added to explain what each section of code is doing, especially any complex equations or functions.

```
Good Pratice:
# Calculate the total price after adding the tax
total_price = base_price + tax
```

4. Using Docstring: A docstring can be useful for someone who is using any code for the first time. This is a string literal written into the code that provides information about the code. In Python, the command line is used to find documentation on a class, method, or function, the text that is displayed is the docstring within that code**.**

```
def calculate_area(radius):
    """Calculate the area of a circle given the radius."""
    return 3.14 * radius ** 2
```

5. Error Handling: Handling edge cases and unexpected inputs make the code more efficient.

```
try:
    result = int(input_value)
except ValueError:
    print("Invalid input. Please enter a number.")
```

6. **Avoiding unnecessary loops and iterations**: Loops are often very processor-heavy tasks. One or two loops may be unavoidable, but too many loops can quickly bog down an otherwise efficient program. By limiting the number of loops and iterations in code it can boost your code’s performance.

7. Using Version Control: Using Git to manage code changes and Commit frequently with clear messages. It wil help to track down the development process in coding which is also helpful coding practices.

Conclusion: Adopting best coding practices is essential for writing clean, efficient, and maintainable code. These practices ensure that the code is easy to understand, reduces the likelihood of errors, and simplifies collaboration within teams. By focusing on aspects such as meaningful variable and function names, consistent style conventions, modularization, and error handling, developers can create software that is both reliable and scalable. Furthermore, practices like thorough documentation, testing, and version control enhance code quality and promote long-term maintainability.