3. Student Activity

Student Activity: Exploring Advanced Features of AI Coding Assistants

Welcome to this hands-on activity session where you'll get to explore the advanced features of AI coding assistants. This activity is designed for beginners and will guide you through practical examples to help you understand and apply these features effectively. Let's dive in!

---

1. Multi-language Coding Support: Translating Code Across Programming Languages

In this section, you'll practice translating code between different programming languages using AI coding assistants.

Example 1: Python to JavaScript

- **Python Code:**

```python
def greet(name):
    return f"Hello, {name}!"

print(greet("World"))
```

- **Translate to JavaScript:**

```javascript
function greet(name) {
    return `Hello, ${name}!`;
}

console.log(greet("World"));
```

Example 2: Java to C++

- **Java Code:**

```java
public class Main {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

- **Translate to C++:**

```cpp
#include <iostream>

int main() {
    std::cout << "Hello, World!" << std::endl;
    return 0;
}
```

Example 3: JavaScript to Python

- **JavaScript Code:**

```javascript
function add(a, b) {
    return a + b;
}

console.log(add(5, 3));
```

- **Translate to Python:**

```python
def add(a, b):
    return a + b

print(add(5, 3))
```

Activity: Use an AI coding assistant to translate these examples and verify the translations by running the code in the respective languages.

---

2. Automatic Documentation Generation Using AI: Code Comments and External Documentation

In this section, you'll learn how AI can help generate documentation for your code.

Example 1: Inline Comments

- **Python Code:**

```python
def calculate_area(radius):
    # Calculate the area of a circle
    return 3.14 * radius * radius
```

- **AI-Generated Comment:**

```python
def calculate_area(radius):
    # Calculate the area of a circle using the formula: π * r^2
    return 3.14 * radius * radius
```

Example 2: External Documentation

- **JavaScript Code:**

```javascript
function multiply(a, b) {
    return a * b;
}
```

- **AI-Generated Documentation:**

```
Function: multiply
Description: Multiplies two numbers and returns the result.
Parameters:
  - a: The first number.
  - b: The second number.
Returns: The product of the two numbers.
```

Example 3: Python Function with Docstring

- **Python Code:**

```
def greet_user(username):
    return f"Welcome, {username}!"
```

- **AI-Generated Docstring:**

```
def greet_user(username):
    """
    Greets the user with a welcome message.

    Parameters:
    username (str): The name of the user.

    Returns:
    str: A welcome message for the user.
    """
    return f"Welcome, {username}!"
```

Activity: Use an AI coding assistant to generate documentation for these examples and compare it with the provided documentation.

---

3. Integrating AI Assistants into CI/CD Pipelines

In this section, you'll explore how AI can be integrated into CI/CD pipelines to automate code reviews and testing.

Example 1: Code Review

- **Code Snippet:**

```
def divide(a, b):
    return a / b
```

- **AI Review Suggestion:**
  - Check for division by zero to prevent runtime errors.

Example 2: Automated Testing

- **Code Snippet:**

```
function isEven(num) {
    return num % 2 === 0;
}
```

- **AI-Generated Test Case:**

```
console.assert(isEven(4) === true, "Test Case 1 Failed");
console.assert(isEven(5) === false, "Test Case 2 Failed");
```

Example 3: Code Optimization

- **Code Snippet:**

```
public int sum(int[] numbers) {
    int total = 0;
    for (int i = 0; i < numbers.length; i++) {
        total += numbers[i];
    }
    return total;
}
```

- **AI Optimization Suggestion:**
  - Use enhanced for-loop for better readability: `for (int number : numbers)`

Activity: Use an AI coding assistant to review and test these examples, and implement any suggested improvements.

Conclusion

By completing these activities, you have gained hands-on experience with the advanced features of AI coding assistants. These tools can help you translate code, generate documentation, and automate parts of the CI/CD process, making you a more efficient and

effective developer. Keep practicing and exploring these features to enhance your coding skills further!