**Chapter 2: Communication Skills for Informatics Engineering Educators**

**Introduction**

Welcome back to our second meeting of English for Professional Educators in Informatics Engineering! Today, we're excited to dive into some important communication skills that will make teaching, professional interactions, and academic conversations even more effective. We'll explore the differences between formal and informal communication, learn some great classroom language, and practice engaging in professional discussions together. Looking forward to a fun and productive session.

**Lesson Objectives**

By the end of this lesson, you will:

1) Understand the difference between formal and informal communication in academic and professional settings.
2) Use appropriate phrases and expressions for teaching Informatics Engineering in English.
3) Practice delivering clear and structured instructions to students.
4) Engage in role-playing exercises for effective teacher-student communication.

**1. Understanding Formal vs. Informal Communication**

In an academic and professional environment, different levels of formality are required. Below are the key differences between formal and informal communication:

**Table 1. Key Differences Between Formal and Informal Communication**

| Formal Communication | Informal Communication |
|---|---|
| Used in professional, academic, and official settings | Used in casual conversations with friends or close colleagues |
| Polite and structured language | Relaxed and conversational tone |
| No slang or contractions | May include slang, idioms, and contractions |
| Example: “Could you clarify your question?” | Example: “Can you repeat that?” |
| Example: “Please submit your assignments before the deadline.” | Example: “Get your tasks done on time.” |

**Exercise 1: Identify the Communication Type**

Read the following sentences and decide whether they are formal or informal:

1) “Hey, what’s up? Can you help me with this code?”
2) “Good afternoon, students. Today, we will discuss database security.”
3) “Please ensure that your assignments are submitted by Friday.”
4) “Can you send me that document ASAP?”
5) “I appreciate your efforts in completing this project.”

**2. Using Effective Classroom Language**

When teaching Informatics Engineering, transparent and structured instructions are essential. Below are some examples of effective classroom language:

Giving Instructions:

- “Open your programming software and create a new file.”
- “Write a function that calculates the sum of two numbers.”
- “Let’s review the code line by line.”

Encouraging Student Participation:

- “Who can explain this concept?”
- “What do you think will happen if we change this variable?”
- “Can anyone suggest a different way to solve this problem?”

Providing Feedback:

- “That’s a great approach. However, consider optimizing your algorithm.”
- “Your explanation is clear, but try using more technical terms.”
- “You did well on the logic, but you need to improve your syntax.”

Handling Questions from Students:

- “That’s a good question. Let me explain it step by step.”
- “Could you clarify what part you didn’t understand?”
- “I appreciate your curiosity! Let’s explore that further.”

**Exercise 2: Role-Playing Classroom Scenarios**

In pairs, practice delivering classroom instructions using clear and structured English. One person acts as the teacher, while the other plays the student. Switch roles after each round.

Scenario 1: Explain how to write a simple Python function to beginners.

Scenario 2: Guide students on how to debug a syntax error in Java.

Scenario 3: Introduce the concept of object-oriented programming to a class.

**3. Professional Communication in Meetings and Discussions**

As future educators, you may need to engage in academic discussions, faculty meetings, or conference presentations. Below are some useful phrases:

Starting a Discussion:

- "Today, I'd like to discuss how we can improve coding assignments."
- "Let's go over the key challenges in teaching algorithms."

Expressing Opinions:

- "In my experience, students struggle with recursion because it's abstract."
- "I believe that hands-on projects are the best way to teach programming."

Agreeing and Disagreeing Politely:

- "I completely agree with your point."
- "I see your perspective, but I think there's another way to look at it."
- "That's an interesting idea, but have you considered this approach?"

Summarizing and Concluding:

- "To summarize, we need to make coding exercises more interactive."
- "In conclusion, project-based learning enhances students' problem-solving skills."

**Exercise 3: Academic Discussion Practice**

In groups, discuss the following topics using professional language:

1. How can we make programming more engaging for students?
2. What are the biggest challenges in teaching Informatics Engineering in English?
3. Should coding assignments focus more on real-world applications? Why or why not?

**Homework Assignment:**

- Write a one-page reflection on the importance of English in teaching Informatics Engineering.
- Record a 2-minute video Students practice giving classroom instruction.