Panel 1:
Are We Providing the Right Education for Computer Science/Engineering Students?

Lin-Shan Lee
Institute of Information Science
Academia Sinica
NanKang, Taipei, ROC
ls1@iis.sinica.edu.tw

Computer science and engineering is a fast changing field. What we have learned and taught in the past may be obsolete today due to the rapid advances of new hardware and software technologies. It is difficult to bring these new advances, such as Motif programming, C++, superscalar architecture, object-oriented design, and ATM networks, to classrooms in time because faculty have to catch up these new materials, the material may not be readily available in the form of textbooks, or there are insufficient laboratory resources to offer some courses. Even if we may be able to offer these new courses, what other courses should be eliminated or consolidated (we cannot expect students staying longer than four years in school)?

Six distinguished panelists from both academia and industry are invited to share their views on this issue.

- Jesse Fang, Hewlett Packard Research Lab., USA
- Yarsun Hsu, IBM Watson Research Center, USA
- Shing-Tsann Huang, National Tsing Hua University, Taiwan, ROC
- Jie-Yong Juang, National Taiwan University, Taipei, ROC
- W.H. Tsai, National Chiao Tung University, Taiwan, ROC
- Horst F. Wedde, University of Dortmund, Germany

They are expected to answer the following questions:

1. What are the core or fundamental computer courses that should be required by all students?

2. We have heard many complaints from industry that the students produced cannot contribute to industry right away. Should we offer some market-demanded technology courses to graduating students?

3. How important is the laboratory component of the computer science/engineering curriculum?

4. How critical is it to maintain state-of-the-art computing facilities for students, and where can we secure funding for upgrading and maintaining facilities?

5. How can we train students so that they are able to learn by themselves after leaving school?

6. How should we balance between training students for graduate school and for industry?

7. How can we encourage faculty to keep up with new advances in computer science and engineering?