LabVIEW is a programming environment targeted at scientific researchers and engineers that need to collect, process, and store their experimental data. The laboratory experimenters are typically non-programmers that can really benefit from using a computer to automate their work. An iconic dataflow language seemed like the ideal way for the experimenters to describe their laboratory setup and would allow the personal computer to control and synchronize the experimental events. The LabVIEW language augments the iconic dataflow with graphical control-flow structures. These structures allow for easy looping, conditional code, and sequencing. To allow the diagrams to scale for larger experiments, an abstraction mechanism was added that allows a diagram to be represented by an icon. Each of the diagrams also incorporates an interactive graphical interface that provides each diagram with graphic watchpoints and means for interaction during debugging. The LabVIEW environment contains a graphical editor for GUI building and diagramming. It also contains a compiler to generate machine code from the diagram, and an execution system that schedules the dataflow diagrams and provides the runtime libraries. The environment provides a seamless link between the editor, the help system, the debugging tools, and the execution systems.

A copy of the LabVIEW Student Edition will be given to each person taking the tutorial class.

Contents
- History of the LabVIEW language and programming environment
- Introduction to graphical dataflow diagrams
- Graphical structures that enhance dataflow diagrams
- Items that are still textual
- Scaling the diagram using hierarchy
- Overview of the context sensitive help system
- Overview of debugging features
- Leveraging traditional programming systems
- Market and customer background
- Future

Speakers’ Biographies
Greg McKaskle received his B.S. degree in Computer Science from Texas A&M University in 1990. During 1988-1990 he worked for the Nuclear Engineering department on a visualization package for reactor simulation. He joined National Instruments in 1990 and currently serves as Sr. Group Manager for LabVIEW R&D-current projects.

Omid Sojoodi received B.S. degrees in Computer Science and Psychology from Duke University in 1990, and the M.S. degree in Computer Science from the University of Texas at Austin in 1992. During 1990-1992 he worked as a research assistant at the Southwest Research Institute on the design and implementation of visualization software for satellite data collected during NASA’s Mission to Planet Earth. He joined the Research and Development staff of National Instruments in 1992.

Rahman Jamal is Applications Engineering Manager at National Instruments Germany GmbH. He joined National Instruments Austin/Texas in 1990 after graduating from the University of Paderborn, Germany, with an Dipl.-Ing. degree in Electrical Engineering and worked as an Applications Engineer before moving to his current position. In his role at National Instruments Germany, Jamal manages the applications engineering department and is also responsible for customer training and researching test and measurement related applications. He has published a number of technical papers and is the author of a book on LabWindows.

Greg McKaskle, Rahman Jamal, Omid Sojoodi, National Instruments, München, Germany,