## Final Program

Fifteen 1-day tutorials organized in 3 tracks over 5 days
December 10-14, 1979

**HOTEL DEL CORONADO • SAN DIEGO, CALIFORNIA**

<table>
<thead>
<tr>
<th>MONDAY, 12/10</th>
<th>TUESDAY, 12/11</th>
<th>WEDNESDAY, 12/12</th>
<th>THURSDAY, 12/13</th>
<th>FRIDAY, 12/14</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Software Requirements Analysis &amp; Definition</td>
<td>2</td>
<td>Software Design Techniques</td>
<td>3</td>
</tr>
<tr>
<td>Anthony I. Wasserman</td>
<td></td>
<td>Peter Freeman</td>
<td></td>
<td>Donald J. Reifer</td>
</tr>
<tr>
<td>5</td>
<td>Computer Networks &amp; Distributed Processing</td>
<td>7</td>
<td>Computer Communication Protocols</td>
<td>8</td>
</tr>
<tr>
<td>Burt H. Liebowitz</td>
<td></td>
<td>John M. McQuillan</td>
<td></td>
<td>Peter J. Sevcik &amp; Anthony F. Culmone</td>
</tr>
<tr>
<td>11</td>
<td>Microprocessors</td>
<td>12</td>
<td>Microcomputer Programming &amp; Software Support</td>
<td>13</td>
</tr>
<tr>
<td>John H. Carson</td>
<td></td>
<td>Imsong Lee</td>
<td></td>
<td>Warren G. Fee</td>
</tr>
<tr>
<td>4</td>
<td>Software Cost Estimating</td>
<td>9</td>
<td>Computer Security &amp; Integrity</td>
<td>10</td>
</tr>
<tr>
<td>Lawrence H. Putnam</td>
<td></td>
<td>Dennis K. Branstad &amp; Peter S. Browne</td>
<td></td>
<td>Kenneth Bowles</td>
</tr>
<tr>
<td>14</td>
<td>Image Processing</td>
<td>15</td>
<td>Computer Graphics</td>
<td></td>
</tr>
<tr>
<td>(including evening hands-on hardware demonstration session)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Harry C. Andrews</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**IEEE COMPUTER SOCIETY**
## Tutorial No. 1: Requirements Analysis and Definition

**Instructor:** Anthony I. Wasserman  
**Audience:** Intended for programmers, software designers, and systems analysts with 2-5 years experience; also of interest to managers, system engineers, and others concerned with defining software development projects. Not intended for those experienced in the use of modern requirements definition techniques.

### Course Description
Introduces the basic concepts underlying a systematic definition of software system requirements, and then surveys several specific techniques that employ these concepts, including SADT, structured systems analysis, RSL, and various specification techniques. Also treated is the relationship between requirements and specifications, some of the human factors involved in requirements definition, and management considerations.

**Anthony I. Wasserman** is an associate professor of medical information science at the University of California, San Francisco, and a lecturer in the Computer Sciences Division at the University of California, Berkeley. His research interests include programming language design, programming methodology, data base management, computer science education, and applications to health care.

### Course Outline
- **Basic Concepts of Requirements Analysis and Design (R&D):** historical background and goals  
  - role in the life cycle  
  - work products of R&D  
  - basic processes
- **Techniques Used for R&D:** SADT, structured systems analysis, RSL, and other techniques will be surveyed, presenting basic concepts, examples, and current usage.
- **Relationship Between Requirements and Specifications:** requirements relate to needs, specifications describe something to be built, requirements analysis results in a specification.
- **Human Factors in R&D:** need for communication, communication paths, deterrents to communication.
- **Management of R&D:** planning the analysis activity, keeping track of progress, teamwork, assuring quality.
- **Developing a Methodology for R&D:** typical problems in changing methods, a strategy for overcoming them.
- **Overview of Current Research in R&D**

---

## Tutorial No. 2: Software Design Techniques

**Instructor:** Peter Freeman  
**Audience:** Programmers, analysts, and designers of software needing an introduction to and survey of current software design techniques.

### Course Description
Provides a balanced introduction to the basic modern software design techniques. Provides new and current practice in the areas of design specification, architectural and detailed design, design quality, and management of the design activity. Exercises are used throughout to reinforce the presentation.

Peter Freeman is an associate professor of computer science at the University of California, Irvine. He has been involved in the analysis, design, and construction of advanced computer applications and the training of software engineers since 1961.

Freeman’s research activities have been concentrated in software design techniques and their application to the software engineering process. He serves on several professional organizations and is well-known for his seminars on software design (given with Prof. A. I. Wasserman) and software engineering. He has served as a consultant to the United Nations, the U.S. government, and numerous industrial organizations.

Freeman has published more than 20 technical papers and is the author of Software Systems Principles (SRA, 1975). In addition, he has co-edited Software Engineering Education: Needs and Objectives, and is revising Concepts of Programming Languages written by the late Mark Elson.

### Course Outline:
- **Introduction:** software engineering and the role of design, the interplay of system and data design, payoffs from improved design.
- **Design Theory:** fundamental operations of design, the role of representation, experience, and knowledge, desired characteristics of the design product and methods.
- **Specifications:** goals for specifications, formal specification techniques.
- **Architectural Design:** importance of structure, abstraction, modularity, and structural models.
- **Survey of Structuring Methods:** SADT, Parnas, Petri nets, FSM, structured design.
- **Detailed Design Program Design Languages Survey:** LCP, step-wise refinement, Jackson.
- **Design Quality:** establishing quality expectations, design measures, complexity measures and their use.
- **Design Management:** allocating and controlling design resources, design organizations and reviews.

---

## Tutorial No. 3: Software Management

**Instructor:** Donald J. Reifer  
**Audience:** Intended for project managers, group supervisors, line managers, and senior staff interested in learning how to manage software projects and personnel. Assumes no extensive training or experience in software.

### Course Description
Provides a basic understanding of the skills of software management. Focuses on the basic functions involved in managing a small- to large-scale programming project in data processing operations: planning, organizing, staffing, directing, and controlling. Introduces each of the functions and provides insight via the case method into the practical use of the key theories, methods, tools, and techniques available. Both commercial and military applications are discussed, as are the tradeoffs between alternative approaches. Examples illustrating approaches for both large and small systems are presented.

Donald J. Reifer is president of Software Management Consultants, a consulting firm specializing in system management-related services. He works with clients to identify technical and programmatic risks and recommend action plans to reduce their consequences. While affiliated with the Aerospace Corporation, he managed all software activities conducted in support of the Space Shuttle. These involved managing multiple projects producing over $100 million in software deliverables. Earlier, at Hughes Aircraft, he was responsible for factory automation, simulation, and software research projects. He has been an advisor to the General Accounting Office and the National Science Foundation, and has been asked to testify before Congress.

Reifer received a BSEE from New Carroll College of Engineering, an MS in operations research from USC, and the Certificate in Business Management from Technical Personnel from UCLA.

### Course Outline:
- **Introduction:** software management and the system life cycle: basic management concepts and principles, the manager’s job, what to do, and why.  
- **Planning:** program definition phase and the project plan, planning tools and techniques, work breakdown structuring and scheduling.
- **Organizing:** organizational analysis, effective use of management, team approaches, evaluation.
- **Staffing:** organizing and recruiting the project team, management by objectives, computer science/engineering problem.
- **Directing:** delegation and passing the buck, motivating professional personnel, thoughts on leadership.
- **Controlling:** measuring project performance, configuration management and quality assurance, effective review techniques.
- **Technology Forecast:** risk reduction.
### Tutorial No. 4  
**Software Cost Estimating and Life Cycle Control**

**Instructor:** Lawrence H. Putnam  
**Audience:** Intended for engineers at the MS level or business management analysts at the MBA level with quantitative background. Should have five or more years of professional experience. Desirable education: mathematics through second calculus, college undergraduate physics, probability and statistics, simple linear programming theory, managerial economics, managerial accounting, managerial finance, capital budgeting, and engineering economy.

**Course Description:** All software projects exhibit the elements of a life cycle behavior. The nature of this cycle will be reviewed, its characteristics developed and explained, and the dominant influence of time as the independent variable emphasized. The managerial practices necessary to effectively plan and control the software life cycle are clearly brought out in the context of answering the management questions: Can we do it? How long? How much will it cost? How many people? What's the risk? What are the tradeoffs?

Lawrence H. Putnam is president of Quantitative Software Management, Inc., a firm specializing in software cost estimating and life cycle management. He has had extensive experience in planning the quantitative aspects of software life cycle management including cost, schedule, and manpower determination for development and control of the process during operations and maintenance. Putnam recently worked for General Electric Company as manager of system technologies. Prior to that, he was special assistant to the commander, U.S. Army Computer Systems Command; special assistant to the Assistant Secretary of the Army, Financial Management; and special assistant to the Director of Army automation. Putnam holds an MS in physics from the Naval Postgraduate School and a BS from the U.S. Military Academy.

**Course Outline:**
- Evidence of the Software Life Cycle: the estimating and control problem  
  - management needs  
  - The Rayleigh Shape  
  - Norden's model at IBM  
  - evidence of time-varying pattern  
  - data scatter (noise)  
  - parametric data (scatter and noise)

- Phenomenology: black box model  
  - analogs with gas behavior  
  - diffusion  
  - spread of an innovation  
  - acceleration in chaos  
  - feedback processes  
  - narrow band filters  
  - Rayleigh scattering  
  - productivity  
  - difficulty of software equation  
  - tradeoff law  
  - time-sensitivity  
  - proof of the algorithm — the Remea data

- Case History: how to get the management numbers  
- manual solution  
- the automated solution

- Sizing the System: PERT estimating to reduce uncertainty in determining how many source statements have to be written

- Linear Programming: practical applications

- Risk Analysis to Hedge Our Bets

- Adapting to the Real System Behavior

---

### Tutorial No. 5  
**Software Testing and Validation**

**Instructor:** Edward F. Miller, Jr.  
**Audience:** Intended for those familiar with basic computer science topics, including elementary knowledge of programming languages and programming/debugging processes and outcomes. The main thrust of the seminar is technological — defining what is known and what is not known about software testing and validation technology. Thus, the audience is expected to be reasonably technically mature.

**Course Description:** Describes current methods of software testing in a framework that permits an attendee to select methods that work. Focuses on the philosophy of testing, the theoretical foundations of testing, tools and techniques including static and dynamic testing, management and control issues, measurement and planning questions, and research and development directions.

Besides presenting detailed investigation into the motivation and methods of a variety of software testing techniques, the seminar provides in-depth access to the technical literature pertaining to testing.

Attendees should acquire an increased understanding of the techniques and results of software testing, how software testing can be employed in a software assurance activity, and how software testing fits into the entire software life cycle.

Edward F. Miller, Jr. is an independent consultant and lecturer with Software Research Associates, San Francisco, California, a firm devoted to advanced technology and software application. His interests include software engineering management, software testing technology, automated tool technology, hierarchical design and implementation methods, and computer architecture. He was previously director of the Software Technology Center at Science Applications Inc., in San Francisco, and director of the Program Validation Project at General Research Corporation in Santa Barbara, California.

Miller received a BSEE at Iowa State University in 1962, an MS in applied mathematics at the University of Colorado in 1964, and a PhD at the University of Maryland in 1968. He is currently editor of ACM's Computer Architecture News and associate technical editor of Computer Magazine.

**Course Outline:**

- *Philosophy of Testing:* motivating forces  
  - testing principles  
  - overview of methodology  
  - organizing for testing

- *Theoretical Foundations:* general principles  
  - graph theory  
  - program modeling techniques  
  - elements of reliable test theory  
  - combinatorics of testing

- *Tools and Techniques:* general principles  
  - static testing methods  
  - dynamic testing methods  
  - automated tools and software testing  
  - systematic methodologies  
  - test data generation

- *Planning and Measurement:* general principles  
  - test coverage measures  
  - test structuring

- *Management and Control:* economics of testing  
  - test status monitoring

- Research and Development

---

### Tutorial No. 6  
**An Overview of Distributed Processing**

**Instructor:** Burt H. Liebowitz  
**Audience:** Intended for managers, engineers, programmers, systems analysts and other technical personnel who desire an overview and basic understanding of distributed processing.

**Course Description:** Presents an overview of distributed processing, including a general introduction to computer networks. Defines three major areas of distributed processing: point-of-use systems, resource-sharing networks, and multiple-processor systems. Characteristics, examples, benefits, and tradeoffs are presented for each area.

Technological issues are presented involving processors, communications, intercomputer coupling, executive software structures, system architectures, component selection, and allocation of functions and data files to multiple processors. Several case histories are discussed to provide insight into design issues, cost effectiveness, and management problems.

Burt H. Liebowitz is executive vice president of the International Computing Company in Bethesda, Maryland. He has 20 years' experience in the computer field, the last five of which have been involved with distributed systems. He was responsible for the design and software implementation of the distributed minicomputer system used for off-track betting in New York City. He was previously employed by Bellcomm, Inc., and All. Inc., and has taught at government installations and universities, including George Washington University, where he developed the course on distributed processing.

An author of several articles published in leading technical journals, Liebowitz holds MS degrees in mathematics and electrical engineering.

**Course Outline:**

- *Introduction:* Computer Technology in Distributed Systems: minis * micros * mainframes * connecting multiple computers

- *Telecommunications for Geographically Distributed Systems:* networks * protocols * commercial offerings

- *Resource Sharing Networks:* computer communication networks * computer networks

- *Intelligent Terminals:* technology * applications

- *Point-of-Use Systems*

- *Multiple-Processor Systems:* architectures * software * comparisons with large systems * design issues

- *Multiprocessors*

- *Distributed Data Bases:* file-splitting * directory-splitting * design factors * problem areas

- *Case Histories*

- *Management Issues*

- *Design Issues*

- *Future Trends*
### Tutorial No. 7

**Computer Communication Protocols**

**Instructor:** J. M. McQuillan

**Audience:** Intended for system analysts and programmers, as well as technical designers and managers in the communications field. Assumes several years experience in computing and some exposure to data communications.

**Course Description:** The growing sophistication and complexity of distributed computing and data systems place increasing demands on the communication links between system components. The design and selection of appropriate computer communication protocols is essential to meet these demands and provide reliable interconnections. A protocol is a logical abstraction of the physical process of communication. Objectives of this tutorial are (1) to examine the fundamental design choices in computer communication systems, (2) to investigate fundamental protocol choices within computer communication systems, and (3) to discuss existing offerings.

J. M. McQuillan is manager of the Systems Analysis Department at Bolt Beranek and Newman, Inc., Cambridge, Massachusetts, where he is responsible for conducting studies of computer systems, performing network analysis and design, and related consulting activities. He has been with BBN since 1971, during which period he was a major contributor to the design and implementation of the Arpanet. Currently he is investigating several types of advanced computer communication systems, and has written extensively in this field. McQuillan received his BS, MS, and PhD in applied mathematics from Harvard.

**Course Outline:**

- Function of a Computer Communication Protocol
- Choices in the Design of Computer Communication Systems: design objectives • transmission facilities • switching method • topological layout • communications equipment • subnet protocols • network interfaces
- Host-Level Protocols: basic principles • reliable, sequenced subnet with arbitrary message size • reliable, nonsequencing subnet with arbitrary message size • reliable nonsequencing subnet with maximum message size • unreliable subnets with arbitrary and maximum message sizes • internetworking
- Network Examples: point-to-point virtual circuits (Tymnet, EPSS, Telnet) • point-to-point datagrams (packet radio network, Autodin II, Cigale/Cyclades) • mixed datagram virtual circuits (Arpanet) • multipoint datagrams (Alohanet, packet satellite net, Ethernet, Ringnet, DCS)

---

### Tutorial No. 8

**Network Performance Analysis and Measurement**

**Instructors:** Peter J. Sevcik and Anthony F. Culmine

**Audience:** Same as that intended for Tutorial No. 7. Assumes several years experience in computing and some exposure to data communications.

**Course Description:** Discusses user requirements vs. performance goals. Performance prediction is treated systematically, beginning with an introduction to queuing theory and following with a discussion of simulation techniques, including "operational simulation." Basic results in queuing theory are presented and their applications discussed. Models appropriate to specific applications are described and analyzed, along with their roles in network architectural tradeoffs and their shortcomings in practical situations. General measurement issues and typical problems of meaningful performance definition, prediction, and verification — are covered at length. All phases of network testing are described, with special emphasis on performance measurement tools and techniques, including automated "test engines." Many of the models, measurement, and testing concepts are explained through case histories.

Peter J. Sevcik is a computer communications consultant at Bolt Beranek and Newman, Inc. His primary areas of research are communications network protocol design, performance analysis, and system testing. At Western Union he led the design team of the DOD's Autodin II packet switching network. At Analytics and Page, he participated in the design, implementation, and testing of several military, national, and multinational corporate networks. Sevcik received the BEE from Villanova University.

Anthony F. Culmine is a communications scientist at Bolt Beranek and Newman, Inc., where he consults with government and commercial clients in the areas of network planning, testing, and protocol design. Before joining BBN, he was involved in advanced satellite communications projects at MIT Lincoln Laboratory. Culmine received his BEE from Cooper Union, his MS from Polytechnic Institute of Brooklyn, and his PhD from Syracuse University.

**Course Outline:**

- Introduction: role of performance in network design and development • translation of user requirements into performance goals
- Performance Prediction: basic queuing theory • simulation techniques • models applied to specific systems • models used for architectural tradeoffs • shortcomings of models
- Performance Measurement: network testing techniques and tools • specification and measurement issues • verification of performance predictions • case histories

---

### Tutorial No. 9

**Computer Security and Integrity**

**Instructors:** Dennis K. Branstad and Peter S. Browne

**Audience:** Intended for computer security administrators, system analysts, and Fred P. Apple's Operations director, computer network designers and architects, and terminal equipment engineers and designers. A background in computer science, electrical engineering, or system administration is desirable.

**Course Description:** Provides an introduction and overview to current thinking and the state-of-the-art. Covers current methods for EFTS security, including methods for secret personal ID numbers used in bank card systems, as well as tradeoffs between different methods. Also presents approaches to computer security, as well as standards and guidelines in the technical areas of computer security. He was chairman of the Federal Information Processing Standards Task Group in Computer System Security.

Branstad received a BS in mathematics and an MS and PhD in computer science — all from Iowa State University.

Peter S. Browne is president of Computer Resource Controls, Inc. He has broad experience in all phases of security to include physical security, system security, and auditing. He has managed successful security organizations in two large corporations and has 15 years of experience in this field. Browne has given over 100 lectures and public presentations on security, and he is widely quoted in the literature. He holds a BA from Syracuse and an MBA from the University of Nebraska.

**Course Outline:**

- Planning for Computer Security: definitions • elements of protection • organizing for computer security and privacy
- Operating System Security: definitions • integrity flaws • penetration testing • requirements for secure systems • security architectures • enhancements
- Data Base/Applications Security: threats and vulnerabilities • security controls • programming practices • program development process • program design controls
- Physical and Administrative Controls: access controls • hazard protection • environmental hazards • personnel practices • disaster planning
- Data Communications Security: vulnerabilities • data encryption • key management • communication protocols
- Computer System Threat and Risk Assessment: major tasks • initiation of risk assessment • asset analysis • determining present security • identification of controls • cost/benefit analysis
- Audit: required skills • approaches • methodology • plan • follow-up • types of audit • audit trails

---

**All tutorials start at 9 a.m.**
**Concurrent Pascal**

**Instructor:** Kenneth L. Bowles

**Audience:** System programmers with at least two years of experience, and at least a passing familiarity with Pascal, Algol, PL/1, or a similar block structured language. We will not have time for more than a fast review of sequential Pascal, and thus will not be able to cope with a programmer who has worked only with assembly language, Fortran, Basic, or COBOL. Programming applications to be covered include operating systems, communication handlers, word processing, interactive data capture, process control, instrumentation, simulation, animation, some data base applications, and peripheral device control.

**Course Description:** Though originally defined as a sequential programming language intended mainly for teaching, Pascal has attracted widespread interest as the basis for languages designed for implementing system software. To make this possible, extensions/changes to Pascal for programming concurrent processes are almost always needed. This course reviews Per Brinch-Hansen's language "Concurrent Pascal" as well as a variety of later Pascal-like languages, designed for concurrent processing. The tradeoffs and different approaches will be discussed.

Kenneth L. Bowles is professor of computer science and electrical engineering at the University of California, San Diego. Beginning in late 1974 he organized and has, since then, directed the UCSD Pascal Project, which developed a widely used Pascal-based software system for microcomputers, with strong emphasis on machine-independence of applications programs. His current interests include microcomputer-based mass education and software for distributed networks of small computers. From 1967 until 1974, he was director of the UCSD Computer Center and involved in software enhancements for the Burroughs B6700 line of computers. Prior to joining UCSD, he worked with the Central Radio Propagation Laboratory of the National Bureau of Standards, where he established and directed the Jicamarca Radar Observatory, near Lima, Peru. He was recognized as a Fellow of the IEEE for the NBS/CRPL work.

**Course Outline:**

**Overview:** review of standard (sequential) Pascal • concurrency requirements

**Concurrency Features of Selected Pascal-Like Languages:** concurrent Pascal • Modula • Euclid • Ada • Pascal-Plus

**Implementation Issues:** primitive operators • memory management • modularity • low-level interfaces • constraints imposed by high-level syntax • machine/software dependencies • compatibility with sequential Pascal

---

**Design of Microprocessor Systems**

**Instructor:** John H. Carson

**Audience:** Intended for those who are or will be involved in designing microprocessor-based systems. Prospective students are assumed to have some knowledge of computer hardware technology and programming. Attendees should obtain an understanding of the steps in microprocessor-based system design, some knowledge of design tradeoffs, and a feel for available products and trends.

**Course Description:** Reviews and evaluates the wide range of available microprocessor products and development tools. General-purpose microprocessors, single-chip microcomputers, and single-board microprocessor-based minicomputers are categorized and reviewed. The entire design effort with emphasis on system configuration, software development, and system testing will be analyzed. Tradeoffs encountered during design efforts, discrete logic vs. microprocessor, and hardware vs. software are discussed with examples used to illustrate typical design decisions. Microprocessor software development will be discussed for the range of micros from the single-chip microprocessor to micro-based minis. The history of the microprocessor and current technological trends will also be discussed as they relate to design issues. The attendee should obtain an understanding of the steps in microprocessor-based system design, some knowledge of design tradeoffs, and a feel for available products and trends.

John H. Carson is a member of the senior technical staff of RLG Associates, Inc., as well as a member of the faculty of George Washington University. His work at RLG Associates involves the design of specialized min- and microcomputer systems. He has written and presented IEEE tutorials on mini- and microcomputer system design, and has lectured throughout the country for both the IEEE and George Washington University. He received his PhD in information science from Lehigh University.

**Course Outline**

**Microprocessor Product Review:** single-chip computers • general-purpose microprocessors • 16-bit microprocessors • single-board computers • peripheral chips • memories

**Design Process:** establishment of project goals • architecture selection • component selection • hardware/software tradeoffs • software design

**Personnel Selection:** backgrounds • attitudes • project organization

**Testing and Development Tools:** logic analyzers • microprocessor analyzers and emulators • prototyping kits • software development systems

**Design Alternatives:** build or buy? • hardware vs. software • discrete logic vs. programmed logic • mini or micro?

**Current Trends:** minicomputer processor activities • component consolidation or explosion? • peripheral super chips
**Tutorial No. 13**

**LSI Testing**

**Instructor:** Warren G. Fee

**Audience:** This tutorial is designed to give an overview of the LSI testing problem to those who do not have extensive LSI test engineering experience. A BSEE or equivalent and a knowledge of the function of microprocessors and memories are desirable for the student to assimilate the information presented.

**Course Description:** Presents an overview of the problems and techniques involved with LSI testing. Emphasis is placed on test generation based upon a knowledge of device operation through characterization and a study of the device architecture. The architectures of memory devices and microprocessors are analyzed. The difference between reliability and testing problems is stressed, with a section dedicated to reliability theory.

Warren G. Fee is currently western regional manager for Adar Associates, Inc., where he is responsible for sales, applications engineering, and service in the Western states.

He has spent the past four years specifically involved with LSI testing. During this time, he has been a frequent contributor to various technical conferences dealing with LSI testing and ATE equipment.

Fee holds a BS in engineering from California State College at San Jose. He has designed both core and semiconductor memory systems and has held various engineering and marketing management positions.

**Course Outline:**

- **Introduction** (speaker: Phillip Burlison, Farchild): test equipment + semiconductor manufacturers' testing
- **Microprocessor Testing** (speaker: Wayne E. Sohl, Macrotek): microprocessor architecture + microprocessor test development + peripheral devices
- **Memory Testing** (speaker: Warren Fee, Adar): patterns + Shmoo plots and characterization + test generation
- **Reliability** (speaker: Kenneth R. Anderson, RCA Labs): definition of reliability + life distribution + accelerated testing + activation energy

**Tutorial No. 14**

**Image Processing**

**Instructor:** Harry C. Andrews

**Audience:** Intended for students and practitioners in the fields of medical imaging, landsat remote sensing, image photo interpretation and exploitation, as well as photogrammetric scientists, industrial radiographers, and scientists and engineers responsible for processing two-dimensional data arrays. Background should include BS or equivalent in one of the hard sciences.

**Course Description:** Covers an emerging discipline which has come to be known by such names as "image understanding" and "image science", and which is receiving increasing attention in such dissimilar areas as space exploration, law enforcement, medicine, earth resource investigations, and defense. Though it began only a few short years ago with minimal funding in a few widely scattered research laboratories, the field has now grown to a multi-million dollar activity that still seems far from its peak. Attendees will have an opportunity to engage in real-time interactive enhancement and exploitation using a COMTAL Vision One/20 Image Processing System. If they so desire, they may bring digital images on 9-track 1600-bpi magnetic tape in 512 x 512 pixel format of 8 bits of brightness per pixel, for use during the hands-on demonstration.

Harry C. Andrews is vice president, director of marketing at COMTAL Corporation. He is the former director of the USC Image Processing Institute and an author of many publications on the subject, including three standard texts — Computer Techniques in Image Processing (Academic Press, 1970), Introduction to Mathematical Techniques in Pattern Recognition (Wiley, 1972), and Digital Image Restoration (with B. R. Hunt, Prentice Hall, 1977). He has lectured and consulted widely on the subject of image processing for both industry and government, and has conducted frequent tutorial sessions for the IEEE Computer Society and the Society of Photo-Optical Instrumentation Engineers. Andrews received the BSEE from Stanford University and the MS and PhD from USC.

**Course Outline:**

- Image transforms + image coding + image enhancement + image restoration + feature extraction + image understanding + hybrid optical/digital image processing

**Tutorial No. 15**

**Computer Graphics**

**Instructor:** Kellogg S. Booth

**Audience:** Some familiarity with computing is necessary. Intended for graphics users and managers of graphics users, as well as analysts and program managers. Aimed in particular at those who are new to the field or who must select their first graphics system.

**Course Description:** Provides overview and history of computer graphics. Addresses in depth the fundamental issues in hardware and software, with an emphasis on decision-making in the acquisition, implementation, and use of graphics systems. Includes a review of available hardware and software.

Kellogg S. Booth is an assistant professor of computer science at the University of Waterloo, Canada. A former assistant professor at the Davis and Berkeley campuses of the University of California, he also worked in the Computer Graphics Group at the Lawrence Livermore Laboratory. He has previously chaired sessions and tutorials both for the ACM Special Interest Group on Graphics and for the IEEE Computer Society's COMPCON Spring. An active consultant for both government and industry, he is currently doing research in computer graphics, phototypesetting, on-line text editing, and computer-controlled documentation systems. He received a BS in mathematics from Caltech, an MA in computer science from UC Berkeley, and a PhD in electrical engineering and computer science from UC Berkeley.

**Course Outline:**

- Introduction: survey and overview
- Hardware: plotters + line-drawing CRT's + raster videos + input devices
- Software: communications + display primitives + transformations + clipping + perspectives + data structures + hidden-line/hidden surface removal
- Applications: engineering + research + computer animation + architecture + urban and environmental systems + education + art
- Review of Commercial and Turn-Key Systems
- Standards

---

**Sponsored by the IEEE Computer Society**

Chairman: Joseph P. Fernandez, IBM Corp.

Cochairman: David Stolberg, CSC

Arrangements: Roberta J. Kerr, CSC

IEEE-CS Professional Development Committee

Chairman: Stanley Winkler, IBM Corp.
**Tutorial Week 79**

**Advance Registration**

Attendance limited to 100 per tutorial. There will be no registration at the door. Advance registration only: deadline — November 15, 1979. Select up to 1 tutorial per individual per day.

<table>
<thead>
<tr>
<th>Tutorial No.</th>
<th>Per Tutorial</th>
<th>All Week</th>
</tr>
</thead>
<tbody>
<tr>
<td>Members</td>
<td>$85</td>
<td>$400</td>
</tr>
<tr>
<td>Non Members</td>
<td>$100</td>
<td>$475</td>
</tr>
</tbody>
</table>

Includes bound tutorial notes and luncheon. There will be a no-host reception Monday night.

**MONDAY, 12/10**

- Tutorial No. 1: Software Requirements Analysis & Definition
- Tutorial No. 6: Computer Networks & Distributed Processing
- Tutorial No. 11: Microprocessors

**TUESDAY, 12/11**

- Tutorial No. 2: Software Design Techniques
- Tutorial No. 7: Computer Communication Protocols
- Tutorial No. 12: Microcomputer Programming & Software Support

**WEDNESDAY, 12/12**

- Tutorial No. 3: Software Management
- Tutorial No. 8: Network Performance Analysis & Measurement
- Tutorial No. 13: LSI Testing

**THURSDAY, 12/13**

- Tutorial No. 4: Software Cost Estimating
- Tutorial No. 9: Computer Security & Integrity
- Tutorial No. 14: Image Processing

**FRIDAY, 12/14**

- Tutorial No. 5: Software Testing & Validation
- Tutorial No. 10: Concurrent Pascal
- Tutorial No. 15: Computer Graphics

I wish to register for the following tutorials (Please circle numbers):

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

□ Check enclosed □ Visa/BankAmericard □ Master Charge

Card #_________________________ Expiration Date_________________________ Signature_________________________

Name_________________________ IEEE or IEEE-CS Membership No._________________________

Street Address_________________________ Phone No._________________________

City/State/Zip_________________________

---

**ROOM RESERVATION REQUEST**

IEEE COMPUTER SOCIETY
December 9–14, 1979

Please check room preference

- □ SINGLE $39.00
- □ DOUBLE $49.00
- □ LANAI $75.00
- □ PARLOR SUITE $115.00

Plus prevailing city occupancy tax per room per day

Please include ONE night's advance room DEPOSIT to confirm the room reservation. No reservations by telephone, please.

Name_________________________

Address_________________________

City/State/Zip_________________________

I will arrive on (Date)______________Time______________

I will depart on (Date)______________

Reservations must be received by NOVEMBER 17, 1979, 21 days prior to conference date, in order to fulfill your room reservation request.

Please mail this form to: Reservations Manager
Hotel Del Coronado
1500 Orange Avenue
Coronado, CA 92118

MAIL TO:
Tutorial Week '79
P.O. Box 639
Silver Spring, MD 20901
Tutogria
Publication:
If you cannot attend Tutorial Week '79, use this order form to purchase any of the tutorial texts listed below. Mail the completed order form with your check or money order to (and payable to):

IEEE Computer Society
5855 Naples Plaza, Suite 301
Long Beach, CA 90803

<table>
<thead>
<tr>
<th>ORDER NO.</th>
<th>TUTORIAL TITLES</th>
<th>QTY.</th>
<th>MEMBER PRICE</th>
<th>NON-MEMBER PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>213</td>
<td>Microcomputer Programming and Software Support</td>
<td></td>
<td>$9.00</td>
<td>$12.00</td>
</tr>
<tr>
<td>207</td>
<td>Software Testing and Validation Techniques</td>
<td></td>
<td>$12.00</td>
<td>$16.00</td>
</tr>
<tr>
<td>215</td>
<td>LSI Testing (Second Edition)</td>
<td></td>
<td>$9.00</td>
<td>$12.00</td>
</tr>
<tr>
<td>075</td>
<td>Software Design Techniques (Second Edition)</td>
<td></td>
<td>$9.00</td>
<td>$12.00</td>
</tr>
<tr>
<td>290</td>
<td>Distributed Processing (Second Edition)</td>
<td></td>
<td>$12.00</td>
<td>$16.00</td>
</tr>
<tr>
<td>233</td>
<td>Computer Graphics</td>
<td></td>
<td>$18.75</td>
<td>$25.00</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Payment Options</th>
<th>charge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Enclosed</td>
<td></td>
</tr>
<tr>
<td>Bill Visa/BankAmericard</td>
<td></td>
</tr>
<tr>
<td>Bill Master Charge</td>
<td></td>
</tr>
</tbody>
</table>

California residents add 6½ sales tax.

Bill me & add $3.00 billing charge ($3.00)

TOTAL

Optional Shipping Charge
(4th class, no charge)

Total

Charge Card Number
Expiration Date
Signature

Name (please print)
Member No.

Address

City/State/Zip
Country

IEEE COMPUTER SOCIETY
IEEE Los Angeles Council
999 No. Sepulveda Blvd.
Suite 410
El Segundo, CA 90245