Core Curriculum
Education Program for Software Professionals

1. Modern Programming Methods

1.1 Objectives
To expose students to the basic properties of modern programming languages, particularly those of the C/C++/Java family.

1.2 Intended Audience
Programmers who need a refresher in basic issues of programming, who need to learn about different kinds of problem decomposition, and who need an introduction to C/C++/Java.

1.3 Software Used In Course
Watcom C/C++.

1.4 References
The course text is P.J. Plauger's "Programming on Purpose: Essays in Software Design". Reference texts are B.W. Kernighan and D.C. Ritchie's "The C Programming Language" and G. Satire and D. Brown's "C++ : The Core Language".

1.5 Course Outline-- 21 hours

1.5.1 Introduction (3 hours)
History of programming. Turing machines. The importance of decomposition. Types of decomposition. Introduction to C. Introduction to the Watcom environment.

1.5.2 Function modularity (3 hours)
Functions and arguments. Techniques for parameter passing. Recursive functions. Functions and macros. Functions in C++ and Java. Using function decomposition.

1.5.3 Data modularity (3 hours)
Data definition. Scope. Data typing. Abstract data types. Using data decomposition. Arrays and strings in C, C++, and Java.

1.5.4 Object modularity (3 hours)
Classes. Constructors, destructors, copy constructors. Inheritance. The rule of IS-A Containment. Overloading. Differences between C++ and Java.

1.5.5 Project modularity (3 hours)
Files, libraries, and other reusable components. Make and RCS. Integrated development environments. Designing for project modularity in C, C++, and Java.

1.5.6 Robust software (3 hours)
Principles of robust design. Language features for robust design. Exceptions. Garbage collection.

1.5.7 Measuring software (3 hours)
The purpose of measurement. Simple basics of order analysis. Factors affecting performance. Tools for measurement. Benchmarking.

2. Computer Science Structures

2.1 Objectives
This course is intended to introduce the student to some of the fundamental structures used in Computer Science. Upon successful completion of this course, the student will understand the fundamentals of

- SQL databases and sets;
- Compilers and finite state machines;
- Grammars for programming languages and documents;
- Program specifications and logic.

2.2 Intended Audience
This course is intended for all students who require some basic background in elementary logic, sets and their representations, syntax and semantics of languages, and graph theory.

2.3 References
The current course textbook is: Mathematical Structures for Computer Science, 3rd Edition, by Judith L. Gersting, Computer Science Press.

2.4 Outline -- 18 hours

2.4.1 Program Specifications and Logic (3 hours)
Basic logic and its use in describing the specifications of programs. The concept of program correctness and proofs of correctness. The notion of a specification language (such as Z or VDM).

2.4.2 Sets, their Representations and SQL Databases (3 hours)
An introduction to set theory and the foundations of SQL databases.

2.4.3 Efficiency of Computation (3 hours)
The notion of runtime as a function of input size, "Big-Oh" notation.
Data structures used in efficient representations of sets.

2.4.4 Compilers and Finite State Machines (3 hours)
Finite state machines and their relationship to translating programming languages. Introduction to LEX.

2.4.5 Grammars, Programming Languages and Documents (3 hours)

Context-free grammars and their relationship to programming languages and the structure of documents. Introduction to Backus Normal Form as a way of describing grammars. Application of grammars to programming languages, and documents, use of document grammars. Introduction to SGML and YACC.

2.4.6 Graph Theory and its Applications in Computing (3 hours)

A basic introduction to the notion of a graph and a broad selection of graph theoretic problems arising in practical computation.

3. Introduction to Database Management

3.1 Objectives

To study databases from three viewpoints, in particular, those of the database user, the database designer, and the database manager. To introduce students to the techniques that have been developed for processing very large collections of data.

3.2 Intended Audience

The Database Management course is intended for anyone who is involved with applications that use database services.

3.3 Software Used in Course

Sybase SQL Anywhere.

3.4 References

The course textbook is: Database Principles, Programming, Performance, by Patrick O'Neil, Morgan Kaufmann, 1995.

3.5 Course Outline -- 24 hours

3.5.1 Introduction (2 hours)

Overview of applications of database systems. Rationale for separating the data management function from applications. Architectural overview of database systems. Main functions of a database system.

3.5.2 Relational Databases (6 hours)

Basic concepts of the relational model. Relational algebra. SQL: data definition, queries, updates, views, access control. Embedded SQL and application programming. Hands-on experience with an existing relational system.

3.5.3 Database System Interfaces (1 hour)

Accessing and using database system interfaces from application programs.

3.5.4 Data Modeling (2 hours)
Basic concepts of the E-R model: entities, relationships, attributes, domains. Constraints: keys, cardinality constraints, weak entity sets. Extensions: aggregation, generalization and specialization.

3.5.5 Database Design (3 hours)
Logical database design: mapping entity-relationship designs to tables and constraints. The problem of good database design and anomalies. Redundancy and functional dependencies. Normal forms. Physical database design: partitioning, prejoining and indexing.

3.5.6 User views (2 hours)
Roles of views for user convenience, data independence, and security; processing of queries through views; problems of updates through views; advantages and disadvantages of materializing views.

3.5.7 Transaction Management (4 hours)
Database recovery: logging, checkpointing; transaction commit, rollback, undo and redo; recovery after a crash; two-phase commit. Concurrency control: serializibility; locking, deadlock handling; optimistic concurrency control; timestamping.

3.5.8 Distributed Databases (3 hours)
Transaction Processing Monitors (e.g. CICS, IMS/ITM), overview of client-server databases, (homogeneous) distributed databases, and multidatabase systems.

3.5.9 Advanced Data base Systems (1 hour)
Overview of object-oriented database systems, deductive databases, and active databases.

4. Operating Systems

4.1 Objectives
The course provides an introduction to the basic components of a modern operating system, including memory management, file systems, security, networks and distributed systems. Students will learn how to write complex programs that accomplish part of their operation through interaction with the operating system.

4.2 Intended Audience
The Operating Systems course is intended for anyone who is involved with applications that use operating systems services.

4.3 Software Used In Course:
C or C++ at the instructor's discretion. The Microsoft Windows NT operating system is used for practical illustration of operating system structures and services.

4.4 References

The current course textbook is: Operating Systems, by William Stallings, Prentice-Hall, 1995.

4.5 Course Outline -- 24 hours

4.5.1 Operating System Introduction (1 hour)

Block diagram of an Operating System. Peripherals: disks, tapes, terminals, local-area networks.

4.5.2 Processes (4 hours)

Concurrency among large programs: kernel, process control blocks, interface between application and kernel, heavy-weight context switch, scheduler.

4.5.3 Virtual Memory (5 hours)

An introduction to memory management: overlays, dynamic loading, segmentation, paging, demand segmentation, demand paging. Page replacement and working set.

4.5.4 Files and File Systems (3 hours)

Industry preferred standards of file naming and types. File operations. Layered file-system structure.

4.5.5 Interprocess Communication (5 hours)

Examining communication models, interfaces, synchronization, data exchange, and networks.

4.5.6 Operating System Interfaces (1 hour)

Accessing and using common operating system services from application programs.

4.5.7 Security and Protection (3 hours)

Examples of security flaws, basic design principles. External versus internal security. Access lists, capabilities, lock-and-key schemes. Dynamic protection structures. Formal models of protection.

4.5.8 Examples of Modern Operating Systems (2 hours)

Recent developments in commercial and research operating system design. Object-oriented operating systems. Distributed operating systems.

5. Distributed Systems

5.1 Objectives

This course provides an overview of computer networks and distributed systems from the perspective of the application developer.

5.2 Intended Audience
This course is intended for application developers who need to understand distributed systems and networks.

5.3 Prerequisites
The course on Operating Systems.

5.4 References
The current course textbooks are: Modern Operating Systems, by A.S. Tanenbaum, Prentice-Hall, 1992. Recommended: Computer Networks, 2nd ed., by A.S. Tanenbaum, Prentice-Hall, 1988. Course Notes.

5.5 Course Outline -- 24 hours

5.5.1 Network Architecture (4 hours)
Protocols, layered implementation, Network models, Open System Interconnection model (OSI), SNA, TCP/IP, DECNet, LANs, WANs.

5.5.2 Protocols (5 hours)
Physical layer. Medium-access sublayer, IEEE 802 protocols, CSMA/CD, token ring. Data-link layer, error detection and recovery, sliding-window protocols. Network layer, routing, congestion control, X.25 standard. Transport layer. Session layer. Presentation layer, encryption, ASN.1. Application layer, X.400, X.500.FTAM, other applications.

5.5.3 File and Directory Services (3 hours)
Naming issues, name servers, file servers.

5.5.4 Network Interfaces (2 hours)
Application program interfaces (APIs) to different network layers, accessing and using network services from application programs.

5.5.5 Network Management (2 hours)
Simple Network Management Protocol (SNMP-TCP/IP), Common Management Interface Protocol (CMIP-OSI).

5.5.6 New Network Technologies (2 hours)
ATM, FDDI, ISDN, OSF DCE.

5.5.7 Examples of Distributed Algorithms, Systems and Applications (6 hours)

6. Object-Oriented Design & Programming

6.1 Objectives
This course is intended to provide students with a strong background in object-oriented programming.

6.2 Intended Audience
This course is intended for students with strong programming skills.

6.3 Prerequisites
The course on Modern Programming Methods.

6.4 Software Used In Course:
Watcom C/C++.

6.5 References
The course textbook will be chosen from: A book on object-oriented programming with C++.

6.6 Course Outline -- 18 hours

6.6.1 Controlling access (6 hours)
Inheritance and Polymorphism. Base and derived classes. Overloaded functions. Virtual functions. Abstract base classes. Templates. Inheriting interface versus. inheriting implementation. Multiple inheritance.

6.6.2 Controlling redundancy (6 hours)
Classes and encapsulation. The notion of a class. Messages. Modifiers and readers. Inline functions. Constructors and destructors. Private, public, and protected access.

6.6.3 Using the concepts (6 hours)
Object-oriented methods. Choosing an object decomposition. Envelope and letter classes. Double dispatching. Mixins. Fat interfaces.

7. Course Resources and Materials

7.1 Software
All software is supplied by the University of Waterloo.

7.2 Textual materials
All textual materials, including textbooks and course notes, are supplied by the University of Waterloo.

7.3 Hardware
Access to personal computers will be arranged by the University of Waterloo.

7.4 Communication with Professors and Fellow students
Each student who has access to the Internet may communicate via electronic mail with both the instructor and the students enrolled in the course. This communication is for the purpose of asking questions or discussing specific course material among the participants. Alternate communications through FAX will be made available if necessary.