

CLASS 717, DATA PROCESSING: SOFTWARE DEVELOPMENT, INSTALLATION AND MANAGEMENT

SECTION I - CLASS DEFINITION

GENERAL STATEMENT OF THE CLASS DEFINITION

This class provides for software program development tool and techniques including processes and apparatus for controlling data processing operations pertaining to the development, maintenance, and installation of software programs. Such processes and apparatus include:

A. Processes and apparatus for program development functions such as specification, design, generation, and version management of source code programs.

B. Processes and apparatus for debugging of computer program including monitoring, simulation, emulation, and profiling of software programs.

C. Processes and apparatus for translating or compiling programs from a high-level representation to an intermediate code representation and finally into an object or machine code representation, including linking, and optimizing the program for subsequent execution.

D. Processes and apparatus for updating, installing, and version management of developed code.

SCOPE OF THE CLASS

- (1) Note. This class provides for operations performed during the generation of computer programs prior to actual deployment and/or execution on a target machine, except when the execution is performed for the purpose of debugging or optimizing the program, such as by simulation, emulation, or profiling.
- (2) Note. Classification herein requires more than nominal recitation of "programs", "programming", "debugging" or of development/programming environments and languages.
- (3) Note. Although this class includes operations including execution of programs for operations performed by execution of computer programs during optimizing or debugging programs, the operations per-

formed during actual intended use thereof are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

- (4) Note. Processes and apparatus for error detection, correction, and fault location and recovery are classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (5) Note. Processes and apparatus for generating and/or communicating through graphical user interfaces and/or application program interfaces of computer programs are classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (6) Note. This class provides for linking compiled load modules for the purpose of generating object or machine code representations of programs. Functions performed during dynamic linking of computer program modules performed at runtime are classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (7) Note. This class provides for debugging computer programs in order to ensure correct operation of software during its intended use. Debugging is performed at the time the program is under development. Operations for fault location and recovery during its actual use are classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (8) Note. This class provides for optimization of computer code by scheduling/reordering of instructions by a compiler during development. Scheduling and/or reordering instructions by hardware means such as superscalar processors are classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (9) Note. This class provides for processing of computer programming languages such as editing, parsing, semantic and syntactic analysis, translation, etc. Operations including translation or editing of natural language text and document are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

- (10) Note. This class provides for processes and apparatus for installation of software on computer systems, which may be performed in conjunction with downloading/transmitting the software and other installation-related information via communication networks and configuration of the computer system during installation of the software. Communication functions performed during downloading/transmitting of information and/or software and configuration of computer systems in a network are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SECTION II - REFERENCES TO OTHER CLASSES

SEE OR SEARCH CLASS:

- 340, Communications: Electrical, subclasses 1.1 through 16.1 for controlling one or more devices to obtain a plurality of results by transmission of a designated one of plural distinctive control signals over a smaller number of communication lines or channels, particularly subclasses 2.1-2.8 for channel selection, subclass 2.81 for tree or cascade selective communication, subclasses 3.1-3.9 for communication systems where status of a controlled device is communicated, subclass 3.51 for selective communication address polling control, subclasses 4.2 and 4.21 for synchronizing selective communication systems, subclasses 5.1-5.92 for security by intelligence comparison (e.g., authorization, etc.) in a selective communication system, subclasses 9.1-9.17 for addressing in selective system, and subclasses 12.1-12.55 for pulse responsive actuation in selective system.
- 345, Computer Graphics Processing and Selective Visual Display Systems, appropriate subclasses for visual display systems with selective electrical control including display memory organization and structure for storing image data and manipulating image data between a display memory and display peripheral, particularly subclasses 156 through 184 for display peripheral interface input device, subclasses 204-215 for display driving control circuitry, subclasses 418-475 for computer graphics processing, and subclasses 501-522 for computer graphic processing systems.
- 370, Multiplex Communications, appropriate subclasses for the simultaneous transmission of two or more signals over a common medium, particularly subclasses 254 through 258 for network configuration determination; subclasses 324, 350, and 503 through 520 for synchronization over free space or wire; subclasses 351 through 430 for path finding or routing including packet switching, circuit switching, and ATM switching; and subclasses 465 through 473 for adaptive communication protocol.
- 375, Pulse or Digital Communications, appropriate subclasses for generic pulse or digital communication systems and synchronization of clocking signals from input data, particularly subclasses 354 through 376 for synchronizing the operation of the receiving and transmitting mechanism including synchronization fault prevention and self synchronization.
- 379, Telephonic Communications, appropriate subclasses for two-way electrical communication of intelligible audio data of arbitrary content over a link including an electrical conductor, particularly subclass 145 for fraud or interference prevention, subclasses 188 through 200 for telephone call or terminal access alarm or control (e.g., access blocking equipment), subclasses 322 through 324 for a power supply in a centralized switching system, and subclass 413 for a power supply in a subscriber line or transmission line interface.
- 380, Cryptography, subclasses 255 through 276 for communication systems using cryptography.
- 455, Telecommunications, appropriate subclasses for modulated carrier wave communication, per se, particularly subclass 26.1 for subject matter which blocks access to a signal source or otherwise limits usage of modulated carrier equipment.
- 700, Data Processing: Generic Control Systems or Specific Applications, appropriate subclasses for data processing control systems and specific applications.
- 702, Data Processing: Measuring, Calibrating, or Testing, appropriate subclasses, particularly subclasses 60 through 63 for power parameter measuring system, subclass 125 for timing signal generation in a testing system, subclasses 176 through 178 for time duration or rate measuring system, and subclass 186 for computer and peripheral benchmarking.

- 704, Data Processing: Speech Signal Processing, Linguistics, Language Translation, and Audio Compression/Decompression, subclasses 1 through 10 for linguistics, subclasses 200 through 278 for speech processing, and subclasses 500 through 504 for audio signal time or bandwidth compression or expansion.
- 705, Data Processing: Financial, Business Practice, Management, or Cost/Price Determination, particularly subclass 18 for a point of sale terminal or an electronic cash register having security or user identification, subclass 44 for authentication or authorization in a credit or loan processing system, and subclasses 50 through 80 for business processing using cryptography.
- 706, Data Processing: Artificial Intelligence, subclasses 1 through 9 for fuzzy logic hardware, subclass 10 for plural processing intelligence systems, subclass 11 for artificial intelligence system having particular user interface, subclasses 12 and 13 for machine learning system, subclass 14 for adaptive system, subclasses 15 through 44 for neural network, and subclasses 45 through 62 for knowledge processing system.
- 707, Data Processing: Database, Data Mining, and File Management or Data Structures, particularly subclasses 609 through 686 for database maintenance including backup and synchronization, and subclasses 687 through 704 for data database integrity.
- 708, Electrical Computers: Arithmetic Processing and Calculating, appropriate subclasses for specialized function or calculation operation performed by an electrical analog computer, an electric hybrid computer, or an electric digital calculating computer.
- 709, Electrical Computers and Digital Processing Systems: Multicomputer Data Transferring or Plural Processor Synchronization, subclasses 200 through 253 for transferring data between a plurality of computers.
- 710, Electrical Computers and Digital Data Processing Systems: Input/Output, appropriate subclasses for interconnecting or transferring data among processors, memories, and peripherals of computers or digital data processing systems.
- 711, Electrical Computers and Digital Processing Systems: Memory, subclasses 1 through 6 for addressing combined with specific memory configurations (e.g., extended, expanded, dynamic, etc.) in a computer; subclasses 100 through 173 for accessing or controlling memories that are peripherals, for caching data, particularly subclass 164 for access limiting with password or key; and subclasses 200 through 221 for generalized address forming in a computer.
- 712, Electrical Computers and Digital Processing Systems: Processing Architectures and Instruction Processing (e.g., Processors), appropriate subclasses for processing architectures including virtual processors; multiple-instruction-multiple-data (MIMD); vector and array processors; single-chip microprocessors; and for fetching, buffering, decoding, or executing instruction data for operations other than I/O (e.g., logic functions).
- 713, Electrical Computers and Digital Processing Systems: Support, subclasses 1, 2, and 100 for digital data processing system initialization and configuration/ reconfiguration, per se, subclasses 150 through 181 for multiple-computer communication using cryptography, subclasses 182 through 186 for system access control based on user identification by cryptography, subclass 187 for computer program modification detection by cryptography, subclass 188 for computer virus detection by cryptography, and subclasses 189 through 194 for data processing protection using cryptography, including upgrade/install encryption.
- 714, Error Detection/Correction and Fault Detection/Recovery, particularly subclass 707 for synchronization control using an error rate, subclass 731 for a reference timing function or a clock pulse generator in a scan path testing system, subclass 744 for clock or synchronization in digital logic testing using a test pattern generator, and subclass 798 for error detection for synchronization control.
- 715, Data Processing: Presentation Processing of Document, Operator Interface Processing, and Screen Saver Display Processing, subclasses 700 through 866 for computer operator interface.
- 719, Electrical Computers and Digital Processing Systems: Interprogram Communication or Interprocess Communication (IPC), appropriate subclasses and particularly subclasses 331 through 332 for dynamic linking, late binding.
- 726, Information Security, subclasses 1 through 36 for information security in computers or digital processing system.

SUBCLASSES

100 SOFTWARE PROGRAM DEVELOPMENT TOOL (E.G., INTEGRATED CASE TOOL OR STAND-ALONE DEVELOPMENT TOOL):

This subclass is indented under the class definition. Subject matter comprising means or steps operating on a computer or digital data processing system which enables a user to create and manage computer program code.

- (1) Note. This subclass and those indented below accept computer programming tools, environments, and application programs which enable a programmer to create computer programs, for example, by use of a flowchart, pseudo code, a graphical description, program specifications, or by writing source code directly. However, particular software applications such as natural language translators, business tools, word processors, etc., are classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (2) Note. Integrated development environment (IDE) tools and Rapid Application Development (RAD) tools are classified herein.
- (3) Note. The tools classified here may translate input data into source code, translate source code into object code, link object code into executable code, or simply bring existing source code modules together to satisfy specified requirements. Code converters that change BCD to BINARY are classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (4) Note. The software development tools classifiable here may include the use of artificial intelligence in the construction of a program. However, artificial intelligence programming tools for developing expert systems are classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (5) Note. The general ability to link certain modules together to form a program is classifiable here. However, where the programming tool is directed to a particular application such as the control of a particular machine or device of another class, the proper classification is in said machine or device class. For example, programming for programmable positionable servo systems is classified with the positionable servo systems art. See the SEE OR SEARCH CLASS notes below. And, for example, programming for the machine control tools is classified with the machine control art. See the SEE OR SEARCH CLASS notes below.
- (6) Note. The software programming tools in this subclass often utilize graphical, symbolic, or iconic metaphors in the user interface. The combination of a programming tool with user interface metaphors is properly classified here. However, generalized operator interface inventions such as on-screen workspaces and features of GUI such as pop-up controls and menus are classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (7) Note. The software development tools here may include tools which perform the function of cross-compiling a program. However, a program which performs the cross-compiling on an application running on a platform is an emulator, per se, which is classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (8) Note. The software development tools here may include tools for developing object-oriented programs. However, operations performed during execution of object-oriented programs such as interprogram communication between objects are classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (9) Note. The general ability to link certain modules such as procedures or functions together to form a program is classifiable here. However, communication between procedures and functions during execution thereof such as function or proce-

dure calls is classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH CLASS:

- 318, Electricity: Motive Power Systems, subclasses 560 through 688 for programmable, positionable servo systems.
- 341, Coded Data Generation or Conversion, appropriate subclasses for digital code-to-digital code converters, A/D converters, and D/A converters.
- 399, Electrophotography, subclasses 76 through 78 for sequential and programmed control of photocopying machines.
- 434, Education and Demonstration, appropriate subclasses for means and method for teaching computer programming.
- 700, Data Processing: Generic Control Systems or Specific Applications, subclasses 17 and 18 for programming methods for general-purpose controllers and industrial controllers and subclasses 180 through 185 for machine tool operator interface.
- 703, Data Processing: Structural Design, Modeling, Simulation, and Emulation, subclass 22 for simulation of computer programs, subclasses 23 through 26 for emulator application programs, and subclasses 27 and 28 for in-system emulation of system components for compatibility.
- 705, Data Processing: Financial, Business Practice, Management, or Cost/Price Determination, appropriate subclasses for business data processing, per se.
- 706, Data Processing: Artificial Intelligence, appropriate subclasses for programming artificial intelligence tools and developing expert systems.
- 715, Data Processing: Presentation Processing of Document, Operator Interface Processing, and Screen Saver Display Processing, subclasses 200 through 277 for document and text processing; subclasses 700-866 for generalized operator interface; and cross-reference art collection 967 for visual or iconic programming used in process control and configuration.
- 719, Electrical Computers and Digital Processing Systems: Interprogram Communication or Interprocess Communication (IPC), appropriate subclasses for interprocess communication such as function calls and particularly subclass 315 and 316 for object-oriented messaging.

101 Software project management:

This subclass is indented under subclass 100. Subject matter comprising means or steps for planning, organizing, staffing, directing, and controlling a software development project.

- (1) Note. The subject matter herein may include frameworks, including object-oriented frameworks for developing software. Application frameworks for executing specific functions other than those associated with generation and management of software are classified in appropriate subclasses within the classes that provide for the specific application.
- (2) Note. Tools for measurement of software metrics related to a software development project, such as product metrics and process metrics, are classified here.
- (3) Note. For the purpose of this definition, software project management comprises management of all aspects related to a software development process. Business process management divorced from software development is classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH CLASS:

- 705, Data Processing: Financial, Business Practice, Management, or Cost/Price Determination, appropriate subclasses for business data processing, per se.

102 Enterprise based:

This subclass is indented under subclass 101. Subject matter wherein the software development project being managed is specific to a business enterprise or organization, including multiple personnel and software developers.

- (1) Note. Subject matter herein may include workflow management tools for monitoring the flow and progress of work between various groups of personnel participating in a software development project. Task management and control during the execution of a program is classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (2) Note. Subject matter herein includes collaborative program development environments wherein various groups of developers and other personnel work on parts of the software development project. Concurrently, established related or collaborative user interfaces including computer conferencing and computer supported cooperative work are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH CLASS:

- 709, Electrical Computers and Digital Processing Systems: Multicomputer Data Transferring or Plural Processor Synchronization, appropriate subclasses for details of data transfer among plurality of computers or digital data processing systems.
- 715, Data Processing: Presentation Processing of Document, Operator Interface Processing, and Screen Saver Display Processing, subclasses 733 through 759 for concurrently established related or collaborative user interfaces including computer conferencing and computer supported cooperative work.

103 Distributed:

This subclass is indented under subclass 102. Subject matter wherein the personnel and the developers are located in geographically dispersed environments.

- (1) Note. Included herein are distributed computer systems wherein plurality of developers and other personnel work on the project over a network such as the Internet or World Wide Web (WWW). Multicomputer data transfer over a network including the Internet and tools for

the preparation and viewing of documents over a network such as the Internet are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH CLASS:

- 709, Electrical Computers and Digital Processing Systems: Multicomputer Data Transferring or Plural Processor Synchronization, appropriate subclasses for details of data transfer among plurality of computers or digital data processing systems.
- 715, Data Processing: Presentation Processing of Document, Operator Interface Processing, and Screen Saver Display Processing, subclasses 200 through 277 for document and text processing, per se.

104 Modeling:

This subclass is indented under subclass 100. Subject matter comprising means or steps for designing and specifying a representation of the structure and desired behavior of a program to be developed.

- (1) Note. Specifications such as state diagrams and Top-Down model for representing a program are classified herein.
- (2) Note. Design templates that are customizable to generate a program are classified here.
- (3) Note. Subject matter under this subclass may include use of a modeling language such as Unified Modeling Language (UML).
- (4) Note. Subject matter herein may include object-oriented representation of a program to be created. Object-oriented data structures and databases, and communication among objects using messages are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH CLASS:

- 703, Data Processing: Structural Design, Modeling, Simulation, and Emulation, subclasses 1 through 22 for modeling and simulation of physical and electrical processes and objects.

- 707, Data Processing: Database, Data Mining, and File Management or Data Structures, subclasses 790 through 812 for database design including data structures and database structure management.
- 716, Computer-Aided Design and Analysis of Circuits and Semiconductor Masks, subclasses 50 through 56 for design and analysis of a semiconductor mask or reticle and subclasses 100 through 139 for the design and analysis of circuit systems and integrated circuit structure by data processing and computer programming techniques.
- 719, Electrical Computers and Digital Processing Systems: Interprogram Communication or Interprocess Communication (IPC), subclass 315 and 316 for interprogram communication using object-oriented message.
- 105 Visual:**
This subclass is indented under subclass 104. Subject matter comprising means or steps for generating a model or specification of a program using a graphically displayable representation.
- (1) Note. Subject matter under this definition includes visual or iconic representation of the program to be created.
- (2) Note. Visual display systems including operator interfaces are classified elsewhere. See the SEE OR SEARCH THIS CLASS, SUBCLASS notes below.
- SEE OR SEARCH THIS CLASS, SUBCLASS:
- 109, for code generating tools including visual generation of code.
- 113, for visual editing tools.
- 125, for visual debugging.
- SEE OR SEARCH CLASS:
- 715, Data Processing: Presentation Processing of Document, Operator Interface Processing, and Screen Saver Display Processing, subclasses 700 through 866 for visual display including operator interfaces and cross-reference art collection 967 for visual programming.
- 106 Code generation:**
This subclass is indented under subclass 100. Subject matter including means or steps for facilitating creation of a list of human-readable instruction data in a programming language.
- (1) Note. The code generating programs or tools classified herein may generate source code directly without using graphics, or object-oriented paradigm, for example, by use of a specification. However, program generation tools utilizing a specification directed to the physics of materials or interactions of constituent parts of a physical process, such as in the field of simulating chemical, electrical, or mechanical processes for the purpose of design and analysis, are classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- SEE OR SEARCH CLASS:
- 703, Data Processing: Structural Design, Modeling, Simulation, and Emulation, subclasses 1 through 22 for modeling and simulation of physical and electrical processes and objects.
- 707, Data Processing: Database, Data Mining, and File Management or Data Structures, subclasses 790 through 812 for database design including data structures and database structure management.
- 715, Data Processing: Presentation Processing of Document, Operator Interface Processing, and Screen Saver Display Processing, subclasses 700 through 866 for visual display including operator interfaces.
- 716, Computer-Aided Design and Analysis of Circuits and Semiconductor Masks, subclasses 50 through 56 for design and analysis of a semiconductor mask or reticle and subclasses 100 through 139 for the design and analysis of circuit systems and integrated circuit structure by data processing and computer programming techniques.
- 719, Electrical Computers and Digital Processing Systems: Interprogram Com-

munication or Interprocess Communication (IPC), subclass 315 and 316 for interprogram communication using object-oriented message.

107 Component based:

This subclass is indented under subclass 106. Subject matter including tools for generation of source code by combining modules such as procedures or functions together to form a program.

SEE OR SEARCH THIS CLASS, SUBCLASS:

162, for linking object code modules to generate executable programs including resolution of references.

SEE OR SEARCH CLASS:

719, Electrical Computers and Digital Processing Systems: Interprogram Communication or Interprocess Communication (IPC), appropriate subclasses for interprocess and interprogram communication, particularly subclass 331 and 332 for dynamic linking of objects to an executing program at run time.

108 Object oriented:

This subclass is indented under subclass 106. Subject matter including tools for developing source code using a programming paradigm in which program elements are conceptualized as objects that can pass messages to each other.

(1) Note. For this definition, the tools used to generate the source code utilizes object-oriented programming techniques. However, object-oriented database structures and inter-object communication performed during execution of object-oriented programs are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH THIS CLASS, SUBCLASS:

116, for software development tools for developing programs using object-oriented programming language constructs.

SEE OR SEARCH CLASS:

707, Data Processing: Database, Data Mining, and File Management or Data Structures, subclasses 790 through 812 for database design including data structures and database structure management.

719, Electrical Computers and Digital Processing Systems: Interprogram Communication or Interprocess Communication (IPC), appropriate subclasses for interprocess communication such as function calls, particularly subclass 315 and 316 for object-oriented messaging.

109 Visual:

This subclass is indented under subclass 106. Subject matter including means or steps for generating a displayable representation of the program code under development.

(1) Note. Subject matter herein includes tools which are used to facilitate software development and may include recitation and processes involving the use of graphical user interfaces, "visual" development environments, and flowcharts. However, general-purpose operator interfaces, graphical user interfaces (GUI), or other graphical displays are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH THIS CLASS, SUBCLASS:

105, for tools for modeling software including visual representation.

113, for visual editing tools.

125, for visual debugging.

SEE OR SEARCH CLASS:

715, Data Processing: Presentation Processing of Document, Operator Interface Processing, and Screen Saver Display Processing, subclasses 700 through 866 for operator interfaces, per se; and cross-reference art collection 967 for iconic programming.

110 Editing:

This subclass is indented under subclass 100. Subject matter comprising means or steps for

modifying code under development using a high-level representation of programs and instructions.

- (1) Note. Modifying and editing code for the purpose of debugging or optimizing an already generated program is classified here. Debugging and optimizing an already generated code per se, is classified elsewhere in this class. See the SEE OR SEARCH THIS CLASS, SUBCLASS notes below.
- (2) Note. Tools for modifying code for the purpose of generating an update for an older version of a program under development is classified here. Upgrading or updating a program after development is classified elsewhere within this class. See the SEE OR SEARCH THIS CLASS, SUBCLASS notes below.
- (3) Note. Subject matter under this subclass includes code modification in response to modifications inputted by a programmer or user and automatic editing or code modification by a computer.
- (4) Note. Modifying code simultaneously with program translation or execution is classified here. Dynamic compiling of code during execution and dynamic optimization of code during execution or compilation of code are classified elsewhere in this class. See the SEE OR SEARCH THIS CLASS, SUBCLASS notes below.

SEE OR SEARCH THIS CLASS, SUBCLASS:

- 124 through 135, for tools for debugging a program under development.
- 148, for dynamic compiling of program during execution.
- 151 through 161, for optimizing compilers, particularly subclass 153 for dynamic optimization.
- 168 through 173, for updating or upgrading an already developed program.

SEE OR SEARCH CLASS:

- 715, Data Processing: Presentation Processing of Document, Operator Inter-

face Processing, and Screen Saver Display Processing, subclasses 255 through 272 for editing of documents including editing text.

111 **Dynamic:**

This subclass is indented under subclass 110. Subject matter including means or steps for changing the code simultaneously with the input of the modifications desired by a user or a programmer.

SEE OR SEARCH THIS CLASS, SUBCLASS:

- 124 through 135, for tools for debugging a program under development.
- 141 through 144, for analysis of programs by a compiler.

SEE OR SEARCH CLASS:

- 714, Error Detection/Correction and Fault Detection/Recovery, subclasses 38.1 through 38.14 for analysis of software.

112 **Syntax based:**

This subclass is indented under subclass 111. Subject matter including means or steps for checking the grammar of a code modification for possible errors.

SEE OR SEARCH THIS CLASS, SUBCLASS:

- 124 through 135, for tools for debugging a program under development.
- 141 through 144, for analysis of programs by a compiler, including checking syntax in accordance with a grammar of a programming language.

SEE OR SEARCH CLASS:

- 704, Data Processing: Speech Signal Processing, Linguistics, Language Translation, and Audio Compression/Decompression, subclasses 1 through 10 for linguistics.
- 714, Error Detection/Correction and Fault Detection/Recovery, subclasses 38.1 through 38.14 for analysis of software.

113 **Visual:**

This subclass is indented under subclass 110. Subject matter comprising means or steps for

modifying a displayable representation of the code, in the form of text and/or graphics.

SEE OR SEARCH THIS CLASS, SUB-CLASS:

- 105, for tools for modeling software including visual representation.
- 109, for code generating tools including visual generation of code.
- 125, for visual debugging of programs.

SEE OR SEARCH CLASS:

- 715, Data Processing: Presentation Processing of Document, Operator Interface Processing, and Screen Saver Display Processing, subclasses 700 through 866 for operator interfaces, per se; and cross-reference art collection 967 for iconic programming.

114 **Programming language:**

This subclass is indented under subclass 100. Subject matter comprising means or steps for using programming language construct or other programming language specific attributes.

- (1) Note. Subject matter herein includes using extensions of existing programming language for the purpose of adding functionalities not already available in the existing programming language.
- (2) Note. Subject matter under this subclass includes specific types or classes of programming languages such as object-oriented, functional, rule-based, procedural and logic programming languages, as well as specific programming languages such as C, C++, LISP, Visual Basic, etc.
- (3) Note. Subject matter under this subclass may include markup languages such as HTML, XML, etc. Preparation and displaying/rendering of documents utilizing markup languages, including network-based document preparation on the Internet or the World Wide Web (WWW) are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH CLASS:

- 709, Electrical Computers and Digital Processing Systems: Multicomputer Data

Transferring or Plural Processor Synchronization, subclasses 200 through 253 for transferring data between a plurality of computers.

- 715, Data Processing: Presentation Processing of Document, Operator Interface Processing, and Screen Saver Display Processing, subclasses 700 through 866 for operator interfaces, per se; and cross-reference art collection 967 for iconic programming.
- 719, Electrical Computers and Digital Processing Systems: Interprogram Communication or Interprocess Communication (IPC), appropriate subclasses and particularly subclasses 331 through 332 for dynamic linking, late binding in interprogram communication.

115 **Script:**

This subclass is indented under subclass 114. Subject matter wherein the programming language comprises a set of instructions or commands along with simple control structures to be interpreted during execution of a developed program.

SEE OR SEARCH THIS CLASS, SUB-CLASS:

- 139, for interpreters for execution of a program in a script-based language by interpreting the program.

116 **Object oriented:**

This subclass is indented under subclass 114. Subject matter wherein the programming language used to develop the code is based on a programming paradigm in which a program is viewed as a collection of discrete objects that are self-contained collections of data structures and routines that interact with other objects.

- (1) Note. Subject matter may include use of object-oriented data structures such as objects and class hierarchies in the development of code. Object-oriented data structures and databases, per se, are classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (2) Note. Subject matter under this subclass includes development of programs including language or programming con-

structs for facilitating improved communication between objects or method calls. Object-oriented messaging, method calls, or method dispatching, per se, and communications between objects are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH THIS CLASS, SUBCLASS:

108, for software development tools for generating code using object-oriented environment.

SEE OR SEARCH CLASS:

707, Data Processing: Database, Data Mining, and File Management or Data Structures, subclasses 790 through 812 for database design including data structures and database structure management.

719, Electrical Computers and Digital Processing Systems: Interprogram Communication or Interprocess Communication (IPC), appropriate subclasses for interprocess communication such as function calls, particularly subclasses 315 through 316 for object-oriented messaging.

117 Declarative (e.g., rule based):

This subclass is indented under subclass 114. Subject matter wherein the programming language describes a set of facts and relationships and is queried for specific results.

(1) Note. A declarative programming language under this subclass is also known as nonprocedural language, since it does not follow the procedural paradigm of executing statements, subroutine calls, and control structures sequentially.

(2) Note. Subject matter includes development of programs including rule-based language or programming constructs for facilitating rule processing in a rule-based or declarative programming environment. Rule-based knowledge processing systems are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH CLASS:

706, Data Processing: Artificial Intelligence, subclasses 47 and 48 for knowledge processing systems including rule-based reasoning.

118 Bytecode (e.g., Java):

This subclass is indented under subclass 114. Subject matter wherein the programming language used to develop the code uses an intermediate and platform-independent form.

(1) Note. Examples of a bytecode form includes Java.

SEE OR SEARCH THIS CLASS, SUBCLASS:

139, for bytecode interpreter.

148, for dynamic compiling and executing bytecode or Java language program using a virtual machine.

119 Parallel:

This subclass is indented under subclass 114. Subject matter wherein the programming language used to develop the code includes constructs suitable for executing in parallel on a machine configured for parallel execution.

SEE OR SEARCH THIS CLASS, SUBCLASS:

149, for compiling code for parallel machines.

SEE OR SEARCH CLASS:

712, Electrical Computers and Digital Processing Systems: Processing Architectures and Instruction Processing (e.g., Processors), appropriate subclasses for processing architectures including multiple-instruction-multiple-data (MIMD), vector, and array processors.

120 Managing software components:

This subclass is indented under subclass 100. Subject matter including means or steps for maintaining software modules or files, preprocessing or integrating source code, generating design documentation from source code, or source code version management.

- (1) Note. The subject matter of this subclass is limited to managing components of software under development, including maintaining the source code modules. As evidenced by the many notes which follow, the generic concepts of version management, data consistency, and data coherency are distributed throughout other classes under various titles, such as document version management, file and database maintenance, consistency, etc. Therefore, a concept search for data management, per se, should include consideration of these related topics.
- (2) Note. For the purpose of this definition, preprocessing or integration of source code occurs when source code modules or files are combined or processed before compilation (e.g., using a preprocessor to process include files or MACROs or for inlining of procedures/functions, using a MAKEFILE to select and schedule source files for compilation).

SEE OR SEARCH THIS CLASS, SUBCLASS:

170, for updating and upgrading executable programs including plural version management.

SEE OR SEARCH CLASS:

707, Data Processing: Database, Data Mining, and File Management or Data Structures, particularly subclasses 609 through 686 for database maintenance including backup and synchronization, subclasses 687 through 704 for database integrity, and subclasses 790 through 812 for database design including data structures and database structure management.

715, Data Processing: Presentation Processing of Document, Operator Interface Processing, and Screen Saver Display Processing, subclass 229 for document version management under document processing.

121 **Software configuration:**

This subclass is indented under subclass 120. Subject matter including means or steps for

tracking and maintaining configuration of software components during the program development life cycle.

- (1) Note. For the purpose of this definition, changes to the components or configuration may be effected by many developers working on a software development project.

SEE OR SEARCH THIS CLASS, SUBCLASS:

101, for software project management tools.

122 **Source code version:**

This subclass is indented under subclass 120. Subject matter including means or steps for maintaining different versions of source code under development in a library to facilitate software development.

- (1) Note. For the purpose of this definition, the versions may have been created by a plurality of programmers and/or at different times.
- (2) Note. For the purpose of this definition, examples of source code version management include UNIX utilities Source Code Control System (SCCS) and Revision Control System (RCS).

SEE OR SEARCH THIS CLASS, SUBCLASS:

116, for updating and upgrading executable programs including plural version management.

SEE OR SEARCH CLASS:

707, Data Processing: Database, Data Mining, and File Management or Data Structures, particularly subclasses 609 through 686 for database maintenance including backup and synchronization, subclasses 687 through 704 for database integrity, and subclasses 790 through 812 for database design including data structures and database structure management.

715, Data Processing: Presentation Processing of Document, Operator Inter-

face Processing, and Screen Saver Display Processing, subclass 229 for document version management under document processing.

123 **Design documentation:**

This subclass is indented under subclass 120. Subject matter including means or steps for generating pseudo-code, structure charts, flowcharts, data flow diagrams, entity-relationship diagrams, class hierarchies, comment listings, or other graphs or programming aids from the source code.

SEE OR SEARCH THIS CLASS, SUBCLASS:

170, for updating and upgrading executable programs including plural version management.

SEE OR SEARCH CLASS:

705, Data Processing: Financial, Business Practice, Management, or Cost/Price Determination, subclasses 1.1 and 2 through 45 for automated financial or business practice or management systems

707, Data Processing: Database, Data Mining, and File Management or Data Structures, particularly subclasses 609 through 686 for database maintenance including backup and synchronization, subclasses 687 through 704 for database integrity, and subclasses 790 through 812 for database design including data structures and database structure management.

710, Electrical Computers and Digital Data Processing Systems: Input/Output, subclass 200 for access control, per se.

711, Electrical Computers and Digital Processing Systems: Memory, subclasses 141 through 146 for cache memory data coherency.

714, Error Detection/Correction and Fault Detection/Recovery, subclasses 15 through 22 for process and data state recovery, per se, in particular subclass 20 for use of plural data sets (i.e., logs) containing set interrelation data.

715, Data Processing: Presentation Processing of Document, Operator Inter-

face Processing, and Screen Saver Display Processing, subclass 229 for document version management under document processing.

718, Electrical Computers and Digital Processing Systems: Virtual Machine Task or Process Management or Task Management/Control, appropriate subclasses for identifying and dealing with run-time dependencies between executing programs, tasks, and processes (e.g., data dependencies, control flow dependencies, etc.), particularly subclass 106 for dependency based cooperative processing of multiple programs working together to accomplish a larger task.

726, Information Security, subclasses 1 through 36 for information security in computers or digital processing system.

124 **Testing or debugging:**

This subclass is indented under subclass 100. Subject matter including means or steps for testing program code for the purpose of determining correctness and performance of software or for locating and correcting errors under software development.

(1) Note. This subclass is for programming tools which help programmers debug executable code in the development phase of programming. Recovery from a run-time fault or exception in an already functioning and tested program is classified elsewhere. See the SEE OR SEARCH CLASS notes below.

(2) Note. Subject matter under this subclass includes symbolic debugging and debugging of object-oriented programs. Object-oriented data structures and databases, and communication between objects using object-oriented messaging during execution of object-oriented programs are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

(3) Note. Subject matter under this subclass may include reverse analysis, de-assembly, or decompilation of machine language program for aiding a programmer in debugging the program. Reverse com-

pilation/reverse assembly, per se, is classified elsewhere within this class. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH CLASS:

- 380, Cryptography, subclass 2 for equipment test or malfunction indication, subclasses 200 through 242 for video cryptography, and subclasses 255 through 276 for communication systems using cryptography.
- 702, Data Processing: Measuring, Calibrating, or Testing, appropriate subclasses for the application of computer data processing in measuring, calibrating, or testing, particularly subclass 119 for program loading or creating in a testing system, subclass 123 for program setup in a testing system, and subclasses 176 through 178 for a time duration or rate measurement system.
- 703, Data Processing: Structural Design, Modeling, Simulation, and Emulation, subclasses 23 through 28 for emulation, per se.
- 712, Electrical Computers and Digital Processing Systems: Processing Architectures and Instruction Processing (e.g., Processors), subclass 227 for specialized instruction processing in support of testing, debugging, and emulation.
- 713, Electrical Computers and Digital Processing Systems: Support, subclasses 150 through 181 for multiple-computer communication using cryptography, subclasses 182 through 186 for system access control based on user identification by cryptography, particularly subclass 187 for computer program modification detection by cryptography, subclass 188 for computer virus detection by cryptography, and subclasses 189 through 194 for data processing protection using cryptography.
- 714, Error Detection/Correction and Fault Detection/Recovery, for error detection/correction not related to software program development, particularly subclasses 799 through 824 directed to fault locating; subclasses 15

through 24 for state recovery such as re-executing instructions in a computer process; subclass 34 for generating a halt, clock, or interrupt signal in the detection of a fault; subclass 35 for substituting or adding an instruction in the detection of a fault; and subclasses 37 through 39 for analyzing outputs, state, or design in the detection of a fault including computer software faults.

- 726, Information Security, subclasses 1 through 36 for information security in computers or digital processing system.

125 Having interactive or visual:

This subclass is indented under subclass 124. Subject matter including means or steps for displaying a representation of the program being debugged to a programmer or means or steps for communicating between a programmer and the computer system on which the program is being debugged, so as to enable the programmer to identify and apply the necessary corrections to the program.

SEE OR SEARCH THIS CLASS, SUBCLASS:

- 105, for tools for modeling software including visual representation.
- 109, for code generating tools including visual generation of code.
- 113, for visual editing tools.

SEE OR SEARCH CLASS:

- 715, Data Processing: Presentation Processing of Document, Operator Interface Processing, and Screen Saver Display Processing, subclasses 733 through 759 for operator interface aspects of work groups for plural users or sites and cross-reference art collection 971 for interface aspects of cooperative decision support systems for groups of users.

126 Program verification:

This subclass is indented under subclass 124. Subject matter including means or steps for proving that the program code demonstrates certain desirable properties.

- (1) Note. Subject matter under this subclass includes use of formal methods and models.

SEE OR SEARCH CLASS:

- 380, Cryptography, subclass 2 for equipment test or malfunction indication, subclasses 200 through 242 for video cryptography, and subclasses 255 through 277 for communication systems using cryptography.
- 714, Error Detection/Correction and Fault Detection/Recovery, subclasses 15 through 24 for state recovery such as re-executing instructions in a computer process; subclass 34 for generating a halt, clock, or interrupt signal in the detection of a fault; subclass 35 for substituting or adding an instruction in the detection of a fault; and subclasses 37 through 39 for analyzing outputs, state, or design in the detection of a fault including computer software faults.

127 Monitoring program execution:

This subclass is indented under subclass 124. Subject matter comprising means or steps for examining effects of execution of program code.

- (1) Note. Subject matter under this subclass may include use of breakpoints, hooks, tracing, or instruction substitution. Implementation of breakpoints, hooks, tracing, etc. for locating errors in an already developed and executing program is classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (2) Note. Subject matter under this subclass may include program status monitors for monitoring the status of a program during execution. Performance analyzers for analyzing the performance of program execution for debugging or optimizing purposes are classified elsewhere within this subclass. See the SEE OR SEARCH THIS CLASS, SUBCLASS notes below.

SEE OR SEARCH THIS CLASS, SUBCLASS:

- 131, for program execution performance analysis for debugging a program.
- 154, for program execution performance analysis for optimization of code.

SEE OR SEARCH CLASS:

- 702, Data Processing: Measuring, Calibrating, or Testing, appropriate subclasses for the application of computer data processing in measuring, calibrating, or testing, particularly subclass 119 for program loading or creating in a testing system, subclass 123 for program setup in a testing system, and subclasses 176 through 178 for a time duration or rate measurement system.
- 709, Electrical Computers and Digital Processing Systems: Multicomputer Data Transferring or Plural Processor Synchronization, subclasses 223 through 226 for computer network managing including network monitoring.
- 714, Error Detection/Correction and Fault Detection/Recovery, subclass 39 for logic state analyzer analyzing outputs, state, or design in the detection of a fault including computer software faults.

128 Tracing:

This subclass is indented under subclass 127. Subject matter including means or steps for recording outputs and memory contents during execution of a program code.

SEE OR SEARCH CLASS:

- 714, Error Detection/Correction and Fault Detection/Recovery, for error detection/correction, particularly subclass 45 for fault locating by output recording (i.e., tracing).

129 Using breakpoint:

This subclass is indented under subclass 127. Subject matter including means or steps for halting, stopping, or interrupting execution of a program code such that the effects of execution thereof may be recorded or observed.

SEE OR SEARCH CLASS:

714, Error Detection/Correction and Fault Detection/Recovery, subclass 34 for generating a halt, clock, or interrupt signal in the detection of a fault; and subclass 35 for substituting or adding an instruction in the detection of a fault.

130 Including instrumentation and profiling:

This subclass is indented under subclass 124. Subject matter including means or steps for inserting monitoring instructions at selected locations in the program code and executing the modified program code along with the inserted instructions, for the purpose of identifying portions of the code that need to be corrected.

- (1) Note. Subject matter under this subclass includes execution time profilers and program resource utilization monitors (e.g., memory use profilers or pointer utilization profilers).
- (2) Note. Instrumentation herein refers to adding software code to a selected portion of an existing program code for the purpose of monitoring run-time behavior of the program at the selected portion.
- (3) Note. Optimization of program code including profiling or instrumenting a program is classified elsewhere within this class. See the SEE OR SEARCH THIS CLASS, SUBCLASS notes below.

SEE OR SEARCH THIS CLASS, SUBCLASS:

154 through 158, for optimization of program including instrumentation and profiling of the program by a compiler.

131 Including analysis of program execution:

This subclass is indented under subclass 124. Subject matter including means or steps for determining efficiency of a program during execution, so as to utilize the determination in debugging of the software during the development process.

- (1) Note. Analysis or monitoring of program code execution for the purpose of optimizing program efficiency by an optimizing compiler is classified elsewhere in this class. See the SEE OR SEARCH THIS CLASS, SUBCLASS notes below.
- (2) Note. Analysis or monitoring of program code execution for the purpose of fault location and recovery during actual use of computer software subsequent to its development are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH THIS CLASS, SUBCLASS:

141 through 144, for analyzing syntax, data, or an expression of computer code for compilation.
 146, for translation of computer code using an intermediate form or language.
 154 through 158, for optimization of computer programs including analysis of program.

SEE OR SEARCH CLASS:

714, Error Detection/Correction and Fault Detection/Recovery, subclasses 37 through 39 for analyzing outputs, state, or design in the detection of a fault including computer software faults.

132 Using program flow graph:

This subclass is indented under subclass 131. Subject matter including means or steps for generating a graph representing the data and control flow of the program being analyzed.

- (1) Note. For the purpose of this definition, examples of flow graphs are program flow graph (PFG) and control flow graph (CFG).
- (2) Note. Using a flow graph for analysis of code for locating bugs in a program is classified herein. Use of flow graphs for analysis of code by an optimizing compiler is classified elsewhere within this class. See the SEE OR SEARCH THIS CLASS, SUBCLASS notes below.

SEE OR SEARCH THIS CLASS, SUB-CLASS:

144, for analysis of code by a compiler using graph representation.

156, for optimization of program including program analysis using program flow graph by an optimizing compiler.

133 Using procedure or function call graph:

This subclass is indented under subclass 132. Subject matter wherein the program flow graph used in the analysis is a function or procedure call graph depicting the calling sequence of functions or procedures during program execution.

SEE OR SEARCH THIS CLASS, SUB-CLASS:

142, for analyzing syntax, data, or an expression of computer code for compilation.

156, and 157, for optimizing compiling including analysis of code using program flow graph.

SEE OR SEARCH CLASS:

709, Electrical Computers and Digital Processing Systems: Multicomputer Data Transferring or Plural Processor Synchronization, subclasses 310 through 332 for interprogram and interprocess communication, including remote procedure call.

134 Including emulation:

This subclass is indented under subclass 124. Subject matter including means or steps for translating a program under development for execution of the program in a target environment on a host platform during testing or debugging of a program.

SEE OR SEARCH THIS CLASS, SUB-CLASS:

138, for translation of code by emulation, per se.

SEE OR SEARCH CLASS:

703, Data Processing: Structural Design, Modeling, Simulation, and Emulation, subclasses 26 through 28 for emulation of instruction and compatibility emulation.

712, Electrical Computers and Digital Processing Systems: Processing Architectures and Instruction Processing (e.g., Processors), subclass 209 for decoding instruction to accommodate plural instruction interpretation including different dialects, languages, and emulation; and subclass 227 for specialized instruction processing in support of testing, debugging, and emulation.

135 Including simulation:

This subclass is indented under subclass 124. Subject matter wherein testing and debugging of a program is performed by modeling (i.e., artificially mimicking) the execution of the program in a target environment.

SEE OR SEARCH THIS CLASS, SUB-CLASS:

151, for optimizing compiling including profiling and code instrumenting.

SEE OR SEARCH CLASS:

702, Data Processing: Measuring, Calibrating, or Testing, appropriate subclasses for the application of computer data processing in measuring, calibrating, or testing, particularly subclass 119 for program loading or creating in a testing system, subclass 123 for program setup in a testing system, and subclasses 176 through 178 for a time duration or rate measurement system.

703, Data Processing: Structural Design, Modeling, Simulation, and Emulation, subclass 22 for simulation of software program, per se.

714, Error Detection/Correction and Fault Detection/Recovery, particularly subclasses 799 through 824 directed to fault locating, subclasses 15 through 24 for state recovery such as re-executing instructions in a computer process, subclass 34 for generating a halt, clock, or interrupt signal in the detection of a fault, subclass 35 for substituting or adding an instruction in the detection of a fault, and subclasses 37 through 39 for analyzing outputs, state, or design in the detection of a

fault including computer software faults.

136 Translation of code:

This subclass is indented under subclass 100. Subject matter comprising means or steps for converting program code from one form to another.

- (1) Note. The translation can vary in degree of complexity from a simple reordering of program code blocks or statements to reverse compiling from executable code to source code. The following are examples of program code translators:

(a)Interpreters -:[source code] -> the instructions are interpreted and executed one instruction at a time;

(b)Compiler -:[source code] -> [object code];

(c)Reverse compilers -:[executable code]-> [source code];

(d)Cross-compilers -:[code on a first system]->[execution on a second system];

(e)Cross platform -:[code for HOST1 (e.g., IBM)] -> [Code for HOST2 (e.g., VAX)];

(f)Cross language -:[source code in a first language; e.g., Pascal] -> [source code in a second language; e.g., "C"].

SEE OR SEARCH THIS CLASS, SUBCLASS:

- 141 through 144, for compiling code including analysis of code.
 145, for compiling code including recompilation.
 146 through 148, for compiling including intermediate form or language.
 149, for compiling for a parallel or multi-processor system.
 151 through 161, for optimization of computer programs.

SEE OR SEARCH CLASS:

- 341, Coded Data Generation or Conversion, appropriate subclasses for A-to-D/D-to-A converters or D-to-D converters (e.g., BCD-to-decimal).
 703, Data Processing: Structural Design, Modeling, Simulation, and Emulation, subclasses 23 through 28 for emulators and emulation of a program or device.
 704, Data Processing: Speech Signal Processing, Linguistics, Language Translation, and Audio Compression/Decompression, subclasses 1 through 10 for parsers and syntax checkers for linguistic and natural language applications (e.g., parsing, grammatical rules, etc.).
 707, Data Processing: Database, Data Mining, and File Management or Data Structures, particularly subclasses 713 through 721 for query optimization, subclasses 759 through 768 for query statement modification, and subclasses 802 and 809 through 811 for database and data structure management involving transformations.
 712, Electrical Computers and Digital Processing Systems: Processing Architectures and Instruction Processing (e.g., Processors), appropriate subclasses for processing architectures including virtual processors; multiple-instruction-multiple-data (MIMD); vector and array processors; single-chip microprocessors; and for fetching, buffering, decoding, or executing instruction data for operations other than I/O (e.g., logic functions).
 714, Error Detection/Correction and Fault Detection/Recovery, for error detection/correction not related to software program development, particularly subclasses 799 through 824 directed to fault locating; subclasses 15 through 24 for state recovery such as re-executing instructions in a computer process; subclass 34 for generating a halt, clock, or interrupt signal in the detection of a fault; subclass 35 for substituting or adding an instruction in the detection of a fault; and subclasses 37 through 39 for analyz-

- ing outputs, state, or design in the detection of a fault including computer software faults.
- 716, Data Processing: Design and Analysis of Circuit or Semiconductor Mask, subclasses 1 through 18 for “silicon compilers” (i.e., tools used in the design and analysis of integrated circuits).
- 137 Source-to-source programming language translation:**
This subclass is indented under subclass 136. Subject matter wherein the code is translated from one source programming language to another source programming language.
- SEE OR SEARCH THIS CLASS, SUBCLASS:
114 through 119, for program development using programming language constructs.
- SEE OR SEARCH CLASS:
704, Data Processing: Speech Signal Processing, Linguistics, Language Translation, and Audio Compression/Decompression, subclasses 1 through 10 for parsers and syntax checkers for linguistic and natural language applications (e.g., parsing, grammatical rules, etc.).
- 138 Emulation:**
This subclass is indented under subclass 136. Subject matter wherein the code translation for a target environment is performed by emulation of software program (i.e., interpretation and execution) on a host environment.
- SEE OR SEARCH THIS CLASS, SUBCLASS:
134, for debugging a program during development including emulation on a host machine.
- SEE OR SEARCH CLASS:
703, Data Processing: Structural Design, Modeling, Simulation, and Emulation, subclasses 26 through 28 for emulation of instruction and compatibility emulation.
712, Electrical Computers and Digital Processing Systems: Processing Archi-
- tectures and Instruction Processing (e.g., Processors), subclass 209 for decoding instruction to accommodate plural instruction interpretation including different dialects, languages, and emulation; and subclass 227 for specialized instruction processing in support of testing, debugging, and emulation.
- 139 Interpreter:**
This subclass is indented under subclass 136. Subject matter including means or steps for translating each instruction of a program into machine executable form and executing the instruction, one instruction at a time.
- SEE OR SEARCH THIS CLASS, SUBCLASS:
114 through 119, for software development tools utilizing specific programming language constructs, including interpretable languages such as command language, scripting languages, and bytecode.
- SEE OR SEARCH CLASS:
712, Electrical Computers and Digital Processing Systems: Processing Architectures and Instruction Processing (e.g., Processors), subclasses 208 through 213 for decoding instruction during instruction processing of an already developed and executing program.
- 140 Compiling code:**
This subclass is indented under subclass 136. Subject matter including means or steps for converting a high-level source code into lower-level machine or object code.
- (1) Note. Compilers convert whole or portions of a program into a machine-level form, thus generating a machine-level program which is executable, whereas interpreters convert and execute instructions one at a time, at the time of execution of a program.
- (2) Note. Database query language translators (e.g., SQL to QBE) and compilers of embedded SQL database query statements (nonprocedural language) are

classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH THIS CLASS, SUB-CLASS:

139, for interpreters.

SEE OR SEARCH CLASS:

704, Data Processing: Speech Signal Processing, Linguistics, Language Translation, and Audio Compression/Decompression, subclasses 2 through 9 for translation of natural languages.
707, Data Processing: Database, Data Mining, and File Management or Data Structures, particularly subclasses 713 through 721 for query optimization, subclasses 759 through 768 for query statement modification, and subclasses 802 and 809 through 811 for database and data structure management involving transformations.

141 **Analysis of code form:**

This subclass is indented under subclass 140. Subject matter including means or steps for analyzing the grammar of a code form for compliance with established rules of the code form in preparation for the conversion of the code form.

- (1) Note. Products of preconversion analysis functions are manipulated and refined code forms. Therefore, a concept search for code refinement techniques, per se, may include a search in the preceding related translation subclasses, per se.
- (2) Note. This subclass is for subject matter directed to grammar analysis in the environment of software programming and development tools. Parsers and syntax checkers for linguistic and natural language applications (e.g., parsing, grammatical rules, etc.) are classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (3) Note. This subclass is for subject matter directed to grammar analysis of procedural languages in the environment of software programming and development tools. Database query language translators (e.g., SQL to QBE) and com-

pilars of embedded SQL database query statements (nonprocedural language) are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH THIS CLASS, SUB-CLASS:

132, for debugging including analysis of program using program flow graph.
133, for debugging including analysis of program using procedure or function call graph.
146, for compiling using an intermediate form.
156, and 157, for optimizing a program by a compiler including data and control flow analysis of program.

SEE OR SEARCH CLASS:

704, Data Processing: Speech Signal Processing, Linguistics, Language Translation, and Audio Compression/Decompression, subclasses 1 through 10 for parsers and syntax checkers for linguistic and natural language applications (e.g., parsing, grammatical rules, etc.).
707, Data Processing: Database, Data Mining, and File Management or Data Structures, particularly subclasses 713 through 721 for query optimization, subclasses 759 through 768 for query statement modification, and subclasses 802 and 809 through 811 for database and data structure management involving transformations.

142 **Scanning and lexical analysis:**

This subclass is indented under subclass 141. Subject matter including means or steps for reading a program code text and generating an output sequence of tokens based on a lexical analysis of the text.

- (1) Note. Subject matter under this subclass may include use of a symbol table for storing identifiers for character strings analyzed.

SEE OR SEARCH CLASS:

704, Data Processing: Speech Signal Processing, Linguistics, Language Trans-

lation, and Audio Compression/Decompression, subclasses 1 through 10 for parsers and syntax checkers for linguistic and natural language applications (e.g., parsing, grammatical rules, etc.).

143 Parsing, syntax analysis, and semantic analysis:

This subclass is indented under subclass 141. Subject matter including means or steps for analyzing program code text to determine whether the program code conforms to grammatical rules of the programming language.

SEE OR SEARCH CLASS:

704, Data Processing: Speech Signal Processing, Linguistics, Language Translation, and Audio Compression/Decompression, subclasses 1 through 10 for parsers and syntax checkers for linguistic and natural language applications (e.g., parsing, grammatical rules, etc.).

144 Including graph or tree representation (e.g., abstract syntax tree or AST):

This subclass is indented under subclass 141. Subject matter including means or steps for generating a tree or graph representing the structure of a program by organizing code elements in the source code text of a program.

SEE OR SEARCH THIS CLASS, SUBCLASS:

132, for debugging including analysis of code using program flow graph.
133, for debugging including analysis of code using procedure or function call graph.
156, and 157, for optimization of program including data flow and control flow analysis of a program.

SEE OR SEARCH CLASS:

707, Data Processing: Database, Data Mining, and File Management or Data Structures, subclasses 790 through 812 for database design including data structures and database structure management.

145 Including recompilation:

This subclass is indented under subclass 140. Subject matter comprising means or steps for repeatedly compiling a number of times after any modifications are made to the program during development.

- (1) Note. The modifications leading to the recompilations may include editing source code for the purpose of debugging or optimizing the program. Tools for editing source code are classified elsewhere in this class. See the SEE OR SEARCH THIS CLASS, SUBCLASS notes below.
- (2) Note. Subject matter under this subclass may include incremental compilation wherein only the modified modules or sections of code are recompiled upon modification thereof.

SEE OR SEARCH THIS CLASS, SUBCLASS:

110 through 113, for utilizing editing tools for editing source code.
122, for source code version management.
151 through 161, for optimizing compilers.
168 through 173, for updating and upgrading of software including version management.

146 Including intermediate code:

This subclass is indented under subclass 140. Subject matter comprising means or steps for performing at least two levels of code conversion during the conversion of program code from an initial form to a target form.

- (1) Note. For the purpose of this definition, it is not required that the program code in the target form be executable code.
- (2) Note. Classification here is proper even if an intermediate form of the code is produced during a transitional process of code translation from the initial form to the target form. However, for the purpose of this definition, a symbol table generated by a multipass compiler is not considered to be an intermediate code form and is classified with compilers,

per se. For such excluded subject matter, see the SEE OR SEARCH THIS CLASS, SUBCLASS notes below.

- (3) Note. Linkers are not found in this subclass. For such excluded subject matter, see the SEE OR SEARCH THIS CLASS, SUBCLASS notes below.
- (4) Note. Subject matter under this subclass includes Static Single Assignment form or SSA form of program code being compiled.

SEE OR SEARCH THIS CLASS, SUBCLASS:

- 140, for generating symbol tables by a compiler.
- 162 through 167, for linking object code.

SEE OR SEARCH CLASS:

- 715, Data Processing: Presentation Processing of Document, Operator Interface Processing, and Screen Saver Display Processing, subclass 249 for format transformation of document types.
- 719, Electrical Computers and Digital Processing Systems: Interprogram Communication or Interprocess Communication (IPC), particularly subclass 331 and 332 for dynamic linking, late binding.

147 Platform-independent form (e.g., abstract code):

This subclass is indented under subclass 146. Subject matter wherein the intermediate form is independent of a target machine on which the compiled program is to be executed.

SEE OR SEARCH THIS CLASS, SUBCLASS:

- 134, for debugging a program including emulation.
- 138, for translation of code by emulation, per se.

SEE OR SEARCH CLASS:

- 703, Data Processing: Structural Design, Modeling, Simulation, and Emulation, subclasses 26 through 28 for emulation of instruction and compatibility emulation.

- 712, Electrical Computers and Digital Processing Systems: Processing Architectures and Instruction Processing (e.g., Processors), subclass 209 for decoding instruction to accommodate plural instruction interpretation including different dialects, languages, and emulation; and subclass 227 for specialized instruction processing in support of testing, debugging, and emulation.

148 Just-in-time compiling or dynamic compiling (e.g., compiling Java bytecode on a virtual machine):

This subclass is indented under subclass 146. Subject matter wherein the compilation is performed on a target machine at the time of execution of the program.

- (1) Note. Subject matter may include compiling in combination with emulation of program on a target architecture.

SEE OR SEARCH THIS CLASS, SUBCLASS:

- 118, for software development tools utilizing programming language constructs including Java or bytecode.
- 134, for debugging a program including emulation.
- 138, for translation of code by emulation, per se.

SEE OR SEARCH CLASS:

- 703, Data Processing: Structural Design, Modeling, Simulation, and Emulation, subclasses 26 through 28 for emulation of instruction and compatibility emulation.
- 712, Electrical Computers and Digital Processing Systems: Processing Architectures and Instruction Processing (e.g., Processors), subclass 209 for decoding instruction to accommodate plural instruction interpretation including different dialects, languages, and emulation; and subclass 227 for specialized instruction processing in support of testing, debugging, and emulation.

149 For a parallel or multiprocessor system:

This subclass is indented under subclass 140. Subject matter including means or steps for analyzing and compiling code which is intended for execution on a parallel or multiprocessor system.

- (1) Note. For the purpose of this definition, the code is generally analyzed for parallel sections, multiprocessor execution relationships, and parallel or cooperative data dependencies. Code is then compiled to create one or more programs for a parallel, cooperative, or distributed data processing system.

SEE OR SEARCH THIS CLASS, SUBCLASS:

- 151 through 161, for optimizing compilers, specifically subclasses 160 and 161 for optimizing code including loops.

SEE OR SEARCH CLASS:

- 708, Electrical Computers: Arithmetic Processing and Calculating, subclasses 200 through 714 for calculating and computing systems for performing arithmetic operations.
- 709, Electrical Computers and Digital Processing Systems: Multicomputer Data Transferring or Plural Processor Synchronization, appropriate subclasses for a distributed data processing system having multicomputer data transfer.
- 718, Electrical Computers and Digital Processing Systems: Virtual Machine Task or Process Management or Task Management/Control, appropriate subclasses for identifying and dealing with run-time dependencies between executing programs, tasks, and processes (e.g., data dependencies, control flow dependencies, etc.), particularly subclass 106 for dependency based cooperative processing of multiple programs working together to accomplish a larger task.

150 Loop compiling:

This subclass is indented under subclass 149. Subject matter wherein the program includes loops to be executed on a parallel machine.

SEE OR SEARCH THIS CLASS, SUBCLASS:

- 160, and 161, for optimizing and scheduling loops by a compiler for execution on a target machine.

SEE OR SEARCH CLASS:

- 712, Electrical Computers and Digital Processing Systems: Processing Architectures and Instruction Processing (e.g., Processors), subclass 241 for instruction processing during execution of programs including loop processing.

151 Optimization:

This subclass is indented under subclass 140. Subject matter comprising means or steps for making program code compilation or execution more efficient with respect to a performance parameter such as speed, memory usage, or other resource usage, through the selection and design of data structures and algorithms.

- (1) Note. This subclass is for optimizing compilers. For the purpose of this definition, it should be understood that optimization steps may occur during any phase of the compilation. And, if a compiler does not perform an optimization, its code would still run, however, not as efficiently as code that has been optimized with respect to a parameter or set of parameters. Further, for the purpose of this definition, a "fast" compiler is considered to be an optimizing compiler in this subclass.
- (2) Note. This subclass is for optimizing compilers. Classification is proper here if the compiler addresses a performance cost function with respect to the compilation process or the run-time parameters of an executable program code that is generated by the compiler. For example, optimizing should be to reduce or eliminate redundant code, redundant arithmetic operations, or unnecessary memory

accesses. These types of optimizations result in a smaller executable target program. In contrast, optimizing to decrease target program execution time by using inline code to eliminate the overhead of function calls will satisfy speed requirements but result in a much larger executable program. Generic systems dealing with cost functions and optimization of processes implemented on or outside a computer, per se, are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

- (3) Note. For the purpose of this definition, optimizing branch execution and register allocation are performed by an optimizing compiler so as to improve the efficiency of execution of the resulting executable code on a target machine and are classified herein. Processing of branch and conditional instructions, per se, and context switching and preserving using registers during execution of a program (which may or may not have been optimized by a compiler) are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH THIS CLASS, SUBCLASS:

- 159, for optimization including restructuring a program code.

SEE OR SEARCH CLASS:

- 707, Data Processing: Database, Data Mining, and File Management or Data Structures, particularly subclasses 713 through 721 for query optimization, subclasses 759 through 768 for query statement modification, and subclasses 802 and 809 through 811 for database and data structure management involving transformations.

- 712, Electrical Computers and Digital Processing Systems: Processing Architectures and Instruction Processing (e.g., Processors), subclasses 233 through 240 for branch processing, subclass 228 for context preserving including register windowing, and subclass 300 for byte-word rearranging.

- 718, Electrical Computers and Digital Processing Systems: Virtual Machine Task or Process Management or Task Management/Control, appropriate subclasses for identifying and dealing with run-time dependencies between executing programs, tasks, and processes (e.g., data dependencies, control flow dependencies, etc.), particularly subclass 104 for resource allocation, subclass 106 for dependency based cooperative processing of multiple programs working together to accomplish a larger task, and subclass 108 for context switching of processes or tasks during program execution.

152 Static (source or intermediate level):

This subclass is indented under subclass 151. Subject matter wherein the compiler performs optimization at the source level or intermediate level, without including runtime information and without specifics of the target machine.

153 Dynamic (i.e., machine or object level):

This subclass is indented under subclass 151. Subject matter wherein the compiler performs optimization at the executable or object code level.

- (1) Note. For the purpose of this definition, dynamic optimization typically includes runtime information specific to the target machine.

SEE OR SEARCH THIS CLASS, SUBCLASS:

- 148, for dynamic compiling at the time of execution.

154 Including analysis of program:

This subclass is indented under subclass 151. Subject matter including means or steps for determining the interdependence of all elements of a computer program prior to optimization, so as to identify program portions to be optimized and to identify the necessary optimization.

- (1) Note. Subject matter under this definition may include branch analysis and function or procedure call analysis.

SEE OR SEARCH THIS CLASS, SUB-CLASS:

- 131, for debugging of program including analysis of program performance.
- 141, for analysis of code including analysis of the grammar of a code form before compilation for compliance with established rules of a programming language.
- 159, for code restructuring techniques for optimizing a program including branch optimization.

SEE OR SEARCH CLASS:

- 712, Electrical Computers and Digital Processing Systems: Processing Architectures and Instruction Processing (e.g., Processors), subclasses 233 through 240 for branch processing.
- 714, Error Detection/Correction and Fault Detection/Recovery, subclasses 38.1 through 38.14 for analysis of software for the purpose of locating fault/error during execution of a developed software.

155 **Data flow analysis:**

This subclass is indented under subclass 154. Subject matter including means or steps for analyzing data flow information performed in order to identify program portions to be optimized.

- (1) Note. Examples of data flow analysis include data dependency analysis, live variable analysis, use-definition analysis, etc.

SEE OR SEARCH THIS CLASS, SUB-CLASS:

- 159, for code restructuring techniques for optimizing a program code based on data flow analysis.

156 **Using flow graph:**

This subclass is indented under subclass 154. Subject matter wherein the analysis is performed using a flow graph representing the data and control flow of the program being analyzed.

- (1) Note. For the purpose of this definition, examples of flow graphs are program

flow graph (PFG), interference graph, and control flow graph (CFG).

- (2) Note. Using a flow graph for analysis of code for optimizing a program is classified herein. Use of flow graphs for analysis of code by a debugger is classified elsewhere within this class. See the SEE OR SEARCH THIS CLASS, SUB-CLASS notes below.

SEE OR SEARCH THIS CLASS, SUB-CLASS:

- 132, for debugging including program analysis using program flow graph by a debugger.
- 144, for syntax analysis using graph representation.

157 **Using procedure or function call graph:**

This subclass is indented under subclass 156. Subject matter wherein the program flow graph used in the analysis is a function or procedure call graph depicting the calling sequence of functions or procedures during program execution.

SEE OR SEARCH THIS CLASS, SUB-CLASS:

- 132, for debugging including analysis of code using program flow graph.
- 133, for debugging including analysis of code using procedure or function call graph.

SEE OR SEARCH CLASS:

- 714, Error Detection/Correction and Fault Detection/Recovery, subclasses 38.1 through 38.14 for analysis of software for the purpose of locating fault/error during execution of a developed software.

158 **Including instrumentation and profiling:**

This subclass is indented under subclass 154. Subject matter wherein the analysis is performed by inserting monitoring code in the program and gathering runtime information by executing the program along with the monitoring code.

SEE OR SEARCH THIS CLASS, SUB-CLASS:

130, for debugging a program during development by instrumentation and profiling by a debugger.

SEE OR SEARCH CLASS:

714, Error Detection/Correction and Fault Detection/Recovery, subclass 35 for instrumentation of code for locating fault/error during execution of program during actual use of developed program.

159 Code restructuring:

This subclass is indented under subclass 151. Subject matter wherein all or portions of a program code is reworked in order to improve the efficiency of storage and execution of the executable code generated by the compiler.

(1) Note. Examples of code restructuring techniques include procedure reordering, dead code removal, and common subexpression elimination.

160 Including loop:

This subclass is indented under subclass 159. Subject matter wherein a code portion to be restructured comprises at least one construct representing a set of repeatedly executing instructions.

(1) Note. Examples of loop optimization under this definition include code motion, strength reduction, loop fusion, etc.

SEE OR SEARCH THIS CLASS, SUB-CLASS:

150, for compiling loops in a program for execution on a parallel machine or multiprocessor.

SEE OR SEARCH CLASS:

712, Electrical Computers and Digital Processing Systems: Processing Architectures and Instruction Processing (e.g., Processors), subclass 241 for processing of instructions for loop execution in a program being executed after compiling.

161 Including scheduling instructions:

This subclass is indented under subclass 160.

Subject matter comprising means or steps for scheduling iterations of a loop for optimized execution on a target machine.

(1) Note. Scheduling iterations of loops, including nested loops, when performed by a compiler prior to execution of the program including loops, is classified herein. Instruction processing for loops in a program being executed, which may include scheduling instructions corresponding to iterations of loops on a particular processing architecture such as a superscalar processor is classified elsewhere.

SEE OR SEARCH THIS CLASS, SUB-CLASS:

150, for compiling loops in a program for execution on a parallel machine or multiprocessor.

SEE OR SEARCH CLASS:

709, Electrical Computers and Digital Processing Systems: Multicomputer Data Transferring or Plural Processor Synchronization, subclasses 102 through 108 for scheduling of processes on one or more processors during execution of a program.

712, Electrical Computers and Digital Processing Systems: Processing Architectures and Instruction Processing (e.g., Processors), subclass 241 for processing of instructions for loop execution in a program being executed after compiling.

162 Linking:

This subclass is indented under subclass 100.

Subject matter comprising means or steps for joining files or modules of object code to create executable program code.

(1) Note. For the purpose of this definition, processes for linking include resolving address data references and external references between separate files or modules of object code to create an executable program.

- (2) Note. Static linking is classified in this subclass, since images of all linked object files and object modules become an integral part of the executable program. Dynamic linking of objects during program execution is classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (3) Note. The subject matter of this subclass is directed to linking object code to create an executable program. The linking of related data, objects, text data (e.g., hypertext), and other information which is intended for graphical display is classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (4) Note. Subject matter under this definition may include forming a load module which may be subsequently executed, by a linkage editor or loader. Loading initialization programs during booting/initialization of a computer is classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH THIS CLASS, SUBCLASS:

- 107, for component-based code generation tools for generating programs by combining program components or modules.
- 168 through 173, for software updating including updating and version management of files and libraries.

SEE OR SEARCH CLASS:

- 709, Electrical Computers and Digital Processing Systems: Multicomputer Data Transferring or Plural Processor Synchronization, appropriate subclasses for multicomputer data transferring
- 713, Electrical Computers and Digital Processing Systems: Support, subclass 2 for loading of initialization programs (e.g., booting, rebooting, initial program load).
- 715, Data Processing: Presentation Processing of Document, Operator Inter-

- face Processing, and Screen Saver Display Processing, subclasses 804 through 805 for linking of related data, objects, text data (e.g., hypertext), and other information which is intended for graphical display.
- 719, Electrical Computers and Digital Processing Systems: Interprogram Communication or Interprocess Communication (IPC), subclass 330 for a remote procedure call, and subclasses 331 and 332 for dynamic linking, late binding.

163 Including library:

This subclass is indented under subclass 162. Subject matter wherein the modules or files to be linked reside in a library.

SEE OR SEARCH THIS CLASS, SUBCLASS:

- 107, for component-based code generation tools for generating programs by combining components or modules stored in a library.
- 168 through 173, for software updating including updating and version management of files and libraries.

164 Shared:

This subclass is indented under subclass 163. Subject matter wherein the library is shared among multiple programs.

- (1) Note. Subject matter under this definition includes libraries containing modules to be linked by a linker for development of executable code. Dynamic linking using dynamic link libraries or DLLs containing objects to be linked to a program during execution thereof are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH CLASS:

- 719, Electrical Computers and Digital Processing Systems: Interprogram Communication or Interprocess Communication (IPC), subclass 331 and 332 for dynamic linking, late binding.

165 Object oriented:

This subclass is indented under subclass 162. Subject matter comprising means or steps for linking object-oriented components.

- (1) Note. Subject matter under this definition may include resolving references to object-oriented classes or methods at the time of linking the classes and methods to a program prior to execution thereof. Generating object-oriented programs and object-oriented programming language tools are classified elsewhere in this class. See the SEE OR SEARCH THIS CLASS, SUBCLASS notes below. Object-oriented messaging and object-oriented databases and data structures are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH THIS CLASS, SUBCLASS:

- 108, for tools for generating code using an object-oriented environment.
116, for software development tools using an object-oriented programming language or constructs.

SEE OR SEARCH CLASS:

- 707, Data Processing: Database, Data Mining, and File Management or Data Structures, subclasses 790 through 812 for database design including data structures and database structure management.
719, Electrical Computers and Digital Processing Systems: Interprogram Communication or Interprocess Communication (IPC), subclass 331 and 332 for dynamic linking during execution of an already developed program.

166 Using class loader:

This subclass is indented under subclass 165. Subject matter including the use of a separate program for linking and loading classes referenced and utilized by an object-oriented program before runtime.

- (1) Note. Class loaders for facilitating linking and loading of object-oriented classes referenced by a program to form

a load module which may be subsequently executed are classified herein. Object-oriented databases and data structures and communication between objects using messages during program execution are classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH THIS CLASS, SUBCLASS:

- 168 through 173, for software updating including updating and version management of files and libraries.

SEE OR SEARCH CLASS:

- 713, Electrical Computers and Digital Processing Systems: Support, subclass 2 for loading of initialization programs (e.g., booting, rebooting, initial program load).
719, Electrical Computers and Digital Processing Systems: Interprogram Communication or Interprocess Communication (IPC), subclasses 315 through 316 for interprogram and interprocess communication including object-oriented message and subclass 332 for object-oriented dynamic linking.

167 Remote:

This subclass is indented under subclass 162. Subject matter wherein the libraries and objects to be linked are located at a site different from the system where the program resides.

SEE OR SEARCH CLASS:

- 707, Data Processing: Database, Data Mining, and File Management or Data Structures, subclass 770 for distributed or remote database or file access.
709, Electrical Computers and Digital Processing Systems: Multicomputer Data Transferring or Plural Processor Synchronization, subclasses 200 through 253 for multicomputer data transferring.

168 SOFTWARE UPGRADING OR UPDATING:

This subclass is indented under the class definition. Subject matter comprising means or steps for modifying an existing operating system,

application program, or other executable program, in order to produce an upgraded or updated program.

- (1) Note. The subject matter for this subclass is limited to updating and upgrading of existing executable programs and operating systems (e.g., to a newer version of the program). Tools for editing program code under development such as source code are classified elsewhere in this class. See the SEE OR SEARCH THIS CLASS, SUBCLASS notes below.
- (2) Note. For the purpose of this definition, the updating or upgrading of an existing program may be performed by applying a patch to the original executable program, wherein the patch corresponds to a new or modified functionality to be affected by the update/upgrade. Error correction of a memory or program during execution by using patches is classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (3) Note. This subclass is directed to the processes for updating and upgrading of executable software. Medium for delivering software, per se, such as punched cards, paper or magnetic tapes, magnetic and optical disks, pluggable modules, cartridges and cards, networks, and memories are classified in the appropriate device class.
- (4) Note. This subclass accepts generic updating and upgrading of software locally or over a network. However, purchasing software over a network divorced from upgrading is classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH THIS CLASS, SUBCLASS:

110 through 113, for editing tools for editing a program under development.

SEE OR SEARCH CLASS:

340, Communications: Electrical, subclasses 1.1 through 16.1 for controlling one or more devices to obtain a plurality of results by transmission of

a designated one of plural distinctive control signals over a smaller number of communication lines or channels, particularly subclasses 2.1-2.8 for channel selection, subclass 2.81 for tree or cascade selective communication, subclasses 3.1-3.9 for communication systems where status of a controlled device is communicated, subclass 3.51 for selective communication address polling control, subclasses 4.2 and 4.21 for synchronizing selective communication systems, subclasses 5.1-5.92 for security by intelligence comparison (e.g., authorization, etc.) in a selective communication system, subclasses 9.1-9.17 for addressing in selective system, and subclasses 12.1-12.55 for pulse responsive actuation in selective system.

370, Multiplex Communications, appropriate subclasses for the simultaneous transmission of two or more signals over a common medium, particularly subclasses 254 through 258 for network configuration determination; subclasses 324, 350, and 503 through 520 for synchronization over free space or wire; subclasses 351 through 430 for path finding or routing including packet switching, circuit switching, and ATM switching; and subclasses 465 through 473 for adaptive communication protocol.

375, Pulse or Digital Communications, appropriate subclasses for generic pulse or digital communication systems and synchronization of clocking signals from input data, particularly subclasses 354 through 376 for synchronizing the operation of the receiving and transmitting mechanism including synchronization fault prevention and self-synchronization.

379, Telephonic Communications, appropriate subclasses for two-way electrical communication of intelligible audio data of arbitrary content over a link including an electrical conductor, particularly subclass 145 for fraud or interference prevention, subclasses 188 through 200 for telephone call or terminal access alarm or control (e.g.,

- access blocking equipment), subclasses 322 through 324 for a power supply in a centralized switching system, and subclass 413 for a power supply in a subscriber line or transmission line interface.
- 380, Cryptography, subclasses 255 through 276 for communication systems using cryptography.
- 455, Telecommunications, appropriate subclasses for modulated carrier wave communication, per se, particularly subclass 26.1 for subject matter which blocks access to a signal source or otherwise limits usage of modulated carrier equipment.
- 705, Data Processing: Financial, Business Practice, Management, or Cost/Price Determination, appropriate subclasses for automated financial or business practice or management systems, including purchasing software over a network such as the Internet, and business processing using cryptography.
- 707, Data Processing: Database, Data Mining, and File Management or Data Structures, particularly subclasses 609 through 686 for database maintenance including backup and synchronization, subclasses 687 through 704 for database integrity, and subclasses 790 through 812 for database design including data structures and database structure management for version management of nonexecutable files and databases.
- 709, Electrical Computers and Digital Processing Systems: Multicomputer Data Transferring or Plural Processor Synchronization, subclasses 220 through 222 for network computer configuring.
- 710, Electrical Computers and Digital Data Processing Systems: Input/Output, subclasses 8 through 14 for peripheral configuring and subclasses 62 through 74 for peripheral adapting.
- 713, Electrical Computers and Digital Processing Systems: Support, subclasses 1, 2, and 100 for digital data processing system initialization and configuration/reconfiguration, per se; subclasses 150 through 181 for multi-
 ple-computer communication using cryptography; subclasses 182 through 186 for system access control based on user identification by cryptography; subclass 187 for computer program modification detection by cryptography; subclass 188 for computer virus detection by cryptography; and subclasses 189 through 194 for data processing protection using cryptography, including upgrade/install encryption.
- 714, Error Detection/Correction and Fault Detection/Recovery, particularly subclasses 2 through 24 for fault recovery, per se.
- 726, Information Security, subclasses 1 through 36 for information security in computers or digital processing system.
- 169 Including multiple files:**
 This subclass is indented under subclass 168. Subject matter wherein the updating or upgrading is performed on software including multiple files.
- (1) Note. For purposes of this definition, the updating or upgrading may involve multiple files where the files are associated with different versions of the program. The generic concept of version management is classified in different classes under various titles, such as document version management, file and database maintenance, and managing source code. Therefore, a concept search for version management, per se, should include consideration of these related topics in the other classes. See the SEE OR SEARCH CLASS notes below.
- SEE OR SEARCH THIS CLASS, SUBCLASS:
 122, for source code version management under software development which may include UNIX utilities such as Source Code Control System (SCCS) and Revision Control System (RCS).
 170, for plural version management of program.
 175, for installation of software on a computer system including multiple files.

SEE OR SEARCH CLASS:

707, Data Processing: Database, Data Mining, and File Management or Data Structures, particularly subclasses 609 through 686 for database maintenance including backup and synchronization, subclasses 687 through 704 for database integrity, and subclasses 790 through 812 for database design including data structures and database structure management for version management of nonexecutable files and databases.

170 Plural version management:

This subclass is indented under subclass 168. Subject matter comprising means or steps for simultaneously maintaining a plurality of versions of a program in executable form.

- (1) Note. Version management includes checking for compatibility between specific versions of application programs and operating systems. Means and steps for making systems compatible with one another, per se, are classified in different classes under topics such as emulation, peripheral configuring, peripheral adapting, network computer configuring, and digital data processing system initialization and configuration, per se (e.g., initializing, setup, configuration, and resetting). A concept search on the topic of compatibility should consider these related areas in the other classes. See the SEE OR SEARCH CLASS notes below.
- (2) Note. For the purpose of this definition, version management includes keeping track of already installed versions of an operating system, application program, or other executable program on a computer system. Operations including installing one or more version of an operating system, application program, or other executable programs are classified elsewhere in this class. See the SEE OR SEARCH THIS CLASS, SUBCLASS notes below.
- (3) Note. The identification of the specific hardware and/or software resources required by a particular version of oper-

ating system or application program is also properly classified here. However, recovering from faults in digital data processing systems, locating faults, or detecting faults in digital data processing systems to ensure reliability and availability is classified elsewhere. See the SEE OR SEARCH CLASS notes below.

- (4) Note. This subclass is directed to the processes for managing versions of executable software. Medium for delivering software, per se, such as punched cards, paper or magnetic tapes, magnetic and optical disks, pluggable modules, cartridges and cards, networks, and memories are classified in the appropriate device class.

SEE OR SEARCH THIS CLASS, SUBCLASS:

- 122, for source code version management under software development which may include UNIX utilities such as Source Code Control System (SCCS) and Revision Control System (RCS).
- 174, for installation of software.

SEE OR SEARCH CLASS:

- 707, Data Processing: Database, Data Mining, and File Management or Data Structures, particularly subclasses 609 through 686 for database maintenance including backup and synchronization, subclasses 687 through 704 for database integrity, and subclasses 790 through 812 for database design including data structures and database structure management for version management of nonexecutable files and databases.
- 713, Electrical Computers and Digital Processing Systems: Support, subclasses 1, 2, and 100 for digital data processing system initialization and configuration/reconfiguration, per se; subclasses 150 through 181 for multiple-computer communication using cryptography; subclasses 182 through 186 for system access control based on user identification by cryptography; subclass 187 for computer program modification detection by cryptography; subclass 188 for com-

puter virus detection by cryptography; and subclasses 189 through 194 for data processing protection using cryptography, including upgrade/install encryption.

726, Information Security, subclasses 1 through 36 for information security in computers or digital processing system.

171 Network:

This subclass is indented under subclass 168. Subject matter wherein the upgrading or updating is performed in one or more computer system located within a network of multiple computers.

SEE OR SEARCH THIS CLASS, SUBCLASS:

176, for installation of programs in a network environment.

172 Including distribution of software (e.g., push-down, pull-down):

This subclass is indented under subclass 171. Subject matter comprising means or steps for distributing the upgrades or updates to a computer on which the software is to be updated.

(1) Note. The distribution under this subclass includes distribution of the updates or upgrades over a network. However, multicomputer data transferring in networks is classified elsewhere. See the SEE OR SEARCH THIS CLASS, SUBCLASS notes below.

(2) Note. This subclass includes distributing upgrades and updates to be applied to an existing software, whereupon the software is updated by applying the distributed updates. Installation of software by distributing software components such as files to be installed on a target computer is classified elsewhere within this class. See the SEE OR SEARCH THIS CLASS, SUBCLASS notes below.

SEE OR SEARCH THIS CLASS, SUBCLASS:

177, for installation of programs in a network environment.

SEE OR SEARCH CLASS:

709, Electrical Computers and Digital Processing Systems: Multicomputer Data Transferring or Plural Processor Synchronization, subclasses 220 through 222 for multicomputer data transferring including network computer configuring.

173 Including downloading:

This subclass is indented under subclass 172. Subject matter wherein the updates and upgrades are distributed by downloading them to a computer system via network.

(1) Note. This subclass includes downloading upgrades and updates to be applied to an existing software, whereupon the software is updated by applying the downloaded updates. Installation of software by downloading software components such as files to be installed on a target computer is classified elsewhere within this class. See the SEE OR SEARCH THIS CLASS, SUBCLASS notes below.

(2) Note. The downloading of updates or upgrades under this definition may include distribution of the updates or upgrades over a network. However, multicomputer data transferring in networks is classified elsewhere. See the SEE OR SEARCH THIS CLASS, SUBCLASS notes below.

SEE OR SEARCH THIS CLASS, SUBCLASS:

178, for installation of programs including downloading of software components.

SEE OR SEARCH CLASS:

340, Communications: Electrical, subclasses 1.1 through 16.1 for controlling one or more devices to obtain a plurality of results by transmission of a designated one of plural distinctive control signals over a smaller number of communication lines or channels, particularly subclass 2.81 for tree or cascade selective communication, subclasses 2.1-2.8 for channel selection, subclasses 3.1-3.9 for communi-

- cation systems where status of a controlled device is communicated, subclass 3.51 for selective communication address polling control, subclasses 4.2 and 4.21 for synchronizing selective communication systems, subclasses 5.1-5.92 for security by intelligence comparison (e.g., authorization, etc.) in a selective communication system, subclasses 9.1-9.17 for addressing in selective system, and subclasses 12.1-12.55 for pulse responsive actuation in selective system.
- 370, Multiplex Communications, appropriate subclasses for the simultaneous transmission of two or more signals over a common medium, particularly subclasses 254 through 258 for network configuration determination; subclasses 324, 350, and 503 through 520 for synchronization over free space or wire; subclasses 351 through 430 for path finding or routing including packet switching, circuit switching, and ATM switching; and subclasses 465 through 473 for adaptive communication protocol.
- 375, Pulse or Digital Communications, appropriate subclasses for generic pulse or digital communication systems and synchronization of clocking signals from input data, particularly subclasses 354 through 376 for synchronizing the operation of the receiving and transmitting mechanism including synchronization fault prevention and self-synchronization.
- 379, Telephonic Communications, appropriate subclasses for two-way electrical communication of intelligible audio data of arbitrary content over a link including an electrical conductor, particularly subclass 145 for fraud or interference prevention, subclasses 188 through 200 for telephone call or terminal access alarm or control (e.g., access blocking equipment), subclasses 322 through 324 for a power supply in a centralized switching system, and subclass 413 for a power supply in a subscriber line or transmission line interface.
- 380, Cryptography, subclasses 255 through 276 for communication systems using cryptography.
- 455, Telecommunications, appropriate subclasses for modulated carrier wave communication, per se, particularly subclass 26.1 for subject matter which blocks access to a signal source or otherwise limits usage of modulated carrier equipment.
- 705, Data Processing: Financial, Business Practice, Management, or Cost/Price Determination, appropriate subclasses for automated financial or business practice or management systems including purchasing software over a network, such as the Internet, and business processing using cryptography.
- 707, Data Processing: Database, Data Mining, and File Management or Data Structures, particularly subclasses 609 through 686 for database maintenance including backup and synchronization, subclasses 687 through 704 for database integrity, and subclasses 790 through 812 for database design including data structures and database structure management for version management of nonexecutable files and databases.
- 709, Electrical Computers and Digital Processing Systems: Multicomputer Data Transferring or Plural Processor Synchronization, subclasses 220 through 222 for multicomputer data transferring including network computer configuring.
- 713, Electrical Computers and Digital Processing Systems: Support, subclasses 1, 2, and 100 for digital data processing system initialization and configuration/reconfiguration, per se; subclasses 150 through 181 for multiple-computer communication using cryptography; subclasses 182 through 186 for system access control based on user identification by cryptography; subclass 187 for computer program modification detection by cryptography; subclass 188 for computer virus detection by cryptography; and subclasses 189 through 194 for data processing protection using cryp-

tography, including upgrade/install encryption.

- 726, Information Security, subclasses 1 through 36 for information security in computers or digital processing system.

174 SOFTWARE INSTALLATION:

This subclass is indented under the class definition. Subject matter including means or steps for setting up and preparing an operating system, application program, or other executable program for operation on a computer program.

- (1) Note. Subject matter under this definition may include preparation of installation packages including program files to be installed and other files such as batch files or scripts necessary to implement the installation of the program files on a target computer. Code generation and development of programs using software development tools are classified elsewhere in this class. See the SEE OR SEARCH THIS CLASS, SUBCLASS notes below.
- (2) Note. The subject matter for this subclass is limited to installation of different versions of executable programs and operating systems. As evidenced by the many notes which follow, the generic concept version management is classified in different classes under various titles, such as document version management, file and database maintenance, and managing source code. Therefore, a concept search for version management, per se, should include consideration of these related topics in the other classes. See the SEE OR SEARCH CLASS notes below.
- (3) Note. Installation under this definition may include checking for compatibility between specific versions of application programs and operating systems already installed or to be installed on a target system. Means and steps for making systems compatible with one another, per se, are classified in different classes under topics such as emulation, peripheral configuring, peripheral adapting, network computer configuring, and digi-

tal data processing system initialization and configuration, per se (e.g., initializing, setup, configuration, and resetting). A concept search on the topic of compatibility should consider these related areas in the other classes. See the SEE OR SEARCH CLASS notes below.

- (4) Note. For the purpose of this definition, installation is defined as a one-time loading event which occurs only when a new version of an operating system or application program is updated and released.
- (5) Note. For the purpose of this definition, the installation of software includes the local and remote loading or copying of an operating system, application program, or other executable program onto a target storage medium such as a hard disk, tape drive, or memory device; and version management includes keeping track of the installed version of an operating system, application program, or other executable program.
- (6) Note. This subclass is directed to the processes for installing software. Medium for delivering software, per se, such as punched cards, paper or magnetic tapes, magnetic and optical disks, pluggable modules, cartridges and cards, networks, and memories are classified in the appropriate device class. When claimed in combination with installation they are classified here.
- (7) Note. This subclass accepts generic installation of software locally or over a network. However, purchasing software over a network divorced from installation is classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (8) Note. The identification of the specific hardware and/or software resources required by a particular version of operating system or application program being installed is also properly classified here. However, recovering from faults in digital data processing systems, locating faults, or detecting faults in digital data processing systems to ensure reliability and availability is classified elsewhere.

See the SEE OR SEARCH CLASS notes below.

- (9) Note. Subject matter under this definition may include compression and decompression of files to be installed on a target system. Compression and decompression of data, per se, is classified elsewhere.

SEE OR SEARCH THIS CLASS, SUBCLASS:

- 122, for source code version management under software development which may include UNIX utilities such as Source Code Control System (SCCS) and Revision Control System (RCS).

SEE OR SEARCH CLASS:

- 340, Communications: Electrical, subclasses 1.1 through 16.1 for controlling one or more devices to obtain a plurality of results by transmission of a designated one of plural distinctive control signals over a smaller number of communication lines or channels, particularly subclasses 2.1-2.8 for channel selection, subclass 2.81 for tree or cascade selective communication; subclasses 3.1-3.9 for communication systems where status of a controlled device is communicated, subclass 3.51 for selective communication address polling control; subclasses 4.2 and 4.21 for synchronizing selective communication systems; subclasses 5.1-5.92 for security by intelligence comparison (e.g., authorization, etc.) in a selective communication system; subclasses 9.1-9.17 for addressing in selective system; and subclasses 12.1-12.55 for pulse responsive actuation in selective system.
- 370, Multiplex Communications, appropriate subclasses for the simultaneous transmission of two or more signals over a common medium, particularly subclasses 254 through 258 for network configuration determination; subclasses 324, 350, and 503 through 520 for synchronization over free space or wire; subclasses 351 through 430 for path finding or routing includ-

ing packet switching, circuit switching, and ATM switching; and subclasses 465 through 473 for adaptive communication protocol.

- 375, Pulse or Digital Communications, appropriate subclasses for generic pulse or digital communication systems and synchronization of clocking signals from input data, particularly subclasses 354 through 376 for synchronizing the operation of the receiving and transmitting mechanism including synchronization fault prevention and self-synchronization.
- 379, Telephonic Communications, appropriate subclasses for two-way electrical communication of intelligible audio data of arbitrary content over a link including an electrical conductor, particularly subclass 145 for fraud or interference prevention, subclasses 188 through 200 for telephone call or terminal access alarm or control (e.g., access blocking equipment), subclasses 322 through 324 for a power supply in a centralized switching system, and subclass 413 for a power supply in a subscriber line or transmission line interface.
- 380, Cryptography, subclasses 255 through 276 for communication systems using cryptography.
- 455, Telecommunications, appropriate subclasses for modulated carrier wave communication, per se, particularly subclass 26.1 for subject matter which blocks access to a signal source or otherwise limits usage of modulated carrier equipment.
- 703, Data Processing: Structural Design, Modeling, Simulation, and Emulation, subclasses 23 through 28 for emulation of system components.
- 705, Data Processing: Financial, Business Practice, Management, or Cost/Price Determination, appropriate subclasses for automated financial or business practice or management systems including purchasing software over a network, such as the Internet, and business processing using cryptography.

- 707, Data Processing: Database, Data Mining, and File Management or Data Structures, particularly subclasses 609 through 686 for database maintenance including backup and synchronization, subclasses 687 through 704 for database integrity, and subclasses 790 through 812 for database design including data structures and database structure management for version management of nonexecutable files and databases.
- 709, Electrical Computers and Digital Processing Systems: Multicomputer Data Transferring or Plural Processor Synchronization, subclasses 220 through 222 for network computer configuring.
- 710, Electrical Computers and Digital Data Processing Systems: Input/Output, subclasses 8 through 14 for peripheral configuring and subclasses 62 through 74 for peripheral adapting.
- 713, Electrical Computers and Digital Processing Systems: Support, subclasses 1, 2, and 100 for digital data processing system initialization and configuration/reconfiguration, per se; subclasses 150 through 181 for multiple-computer communication using cryptography; subclasses 182 through 186 for system access control based on user identification by cryptography; subclass 187 for computer program modification detection by cryptography; subclass 188 for computer virus detection by cryptography; and subclasses 189 through 194 for data processing protection using cryptography, including upgrade/install encryption.
- 714, Error Detection/Correction and Fault Detection/Recovery, particularly subclasses 2 through 24 for fault recovery, per se.
- 726, Information Security, subclasses 1 through 36 for information security in computers or digital processing system.
- 175 Including multiple files:**
This subclass is indented under subclass 174. Subject matter wherein the installation of software is performed by using software components including multiple files.
- (1) Note. For purposes of this definition, the installation may involve multiple files where the files are associated with different versions of the program. The generic concept of version management is classified in different classes under various titles, such as document version management, file and database maintenance, and managing source code. Therefore, a concept search for version management, per se, should include consideration of these related topics in the other classes. See the SEE OR SEARCH CLASS notes below.
- SEE OR SEARCH THIS CLASS, SUBCLASS:**
- 122, for source code version management under software development which may include UNIX utilities such as Source Code Control System (SCCS) and Revision Control System (RCS).
- 169, for updating of software on a computer system including multiple files.
- 170, for plural version management of executable program.
- SEE OR SEARCH CLASS:**
- 707, Data Processing: Database, Data Mining, and File Management or Data Structures, particularly subclasses 609 through 686 for database maintenance including backup and synchronization, subclasses 687 through 704 for database integrity, and subclasses 790 through 812 for database design including data structures and database structure management for version management of nonexecutable files and databases.
- 176 Network:**
This subclass is indented under subclass 174. Subject matter wherein the installation of software is performed in one or more computers within a network of multiple computers.
- SEE OR SEARCH THIS CLASS, SUBCLASS:**
- 171, for updating of programs in a network environment.

177 Including distribution of software:

This subclass is indented under subclass 176. Subject matter wherein the installation of software includes distribution of software components to the computers on which the software is to be installed.

- (1) Note. The distribution under this definition may include distribution of the updates or upgrades over a network. However, multicomputer data transferring in networks is classified elsewhere. See the SEE OR SEARCH CLASS notes below.
- (2) Note. This subclass includes distributing software components and other programs necessary to install the distributed software on a target machine. Updating of software by distributing software updates and upgrades is classified elsewhere within this class. See the SEE OR SEARCH THIS CLASS, SUBCLASS notes below.

SEE OR SEARCH THIS CLASS, SUBCLASS: END

172, for updating and upgrading of programs including distribution of software.

SEE OR SEARCH CLASS:

709, Electrical Computers and Digital Processing Systems: Multicomputer Data Transferring or Plural Processor Synchronization, subclasses 220 through 222 for multicomputer data transferring including network computer configuring.

178 Including downloading:

This subclass is indented under subclass 177. Subject matter wherein the software components are distributed by downloading them to a computer system via network.

- (1) Note. This subclass includes downloading software components such as files to be installed on a target computer. Updating and upgrading of software by downloading upgrades and updates to be applied to an existing software is classified elsewhere within this class. See the

SEE OR SEARCH THIS CLASS, SUBCLASS notes below.

- (2) Note. The downloading of software components under this definition may include distribution of the software components over a network. However, multicomputer data transferring in networks is classified elsewhere. See the SEE OR SEARCH CLASS notes below.

SEE OR SEARCH THIS CLASS, SUBCLASS:

173, for upgrading and updating of programs including downloading of software components.

SEE OR SEARCH CLASS:

709, Electrical Computers and Digital Processing Systems: Multicomputer Data Transferring or Plural Processor Synchronization, subclasses 220 through 222 for multicomputer data transferring including network computer configuring.