Macintosh Programmer's Workshop C++

```
  🍎  File  Edit  Find  Mark  Window  Project  Directory  Build

                    Shape Land            North40:MPW:Examples:CPlusExamples:Shap

                                          const short width = 25;
         ●    ●         ●          ●      const short height = 30;

              ▢                    ●      TShape::TShape(Rect *r)
                                          {
      ●            ●                            RandomRect(r);
                                          }

                                          // Assign a random fBoundRect for the shape.

                          ●               void TShape::RandomRect(Rect *drawRect)
                                          {
             ▢                                 short rand1, rand2;

                                               rand1 = abs(Random()) % (drawRect->right - width);
                                               fBoundRect.left = rand1;
                    North40:MPW:Works          rand2 = abs(Random()) % (drawRect->bottom - (height
                                               fBoundRect.top = rand2 + drawRect->top;
  # 1:23:41 AM ----- Build of CPlusShapesApp.   fBoundRect.right = fBoundRect.left + width;
  # 1:23:42 AM ----- Analyzing dependencies.    fBoundRect.bottom = fBoundRect.top + height;
  # 1:23:44 AM ----- Executing build commands }
    CPlus Shapes.cp -o Shapes.cp.o -sym on
    Set Echo 0                            void TShape::Move(Rect *r)
    Link -d -o CPlusShapesApp -sym on Shape {
    SetFile CPlusShapesApp -t APPL -c 'MOOT       RandomRect(r);
    Rez -append -o CPlusShapesApp ShapesApp }
    Rez -append -o CPlusShapesApp TApplicat
  # 1:24:38 AM ----- Done.                 TArc::TArc(Rect *r) : (r)        // Calls base class cons
    CPlusShapesApp                         {
                                               short rand1, rand2;

                                               rand1 = abs(Random()) % 270;
                                               fStartAngle = rand1;
                                               rand2 = abs(Random()) % 270;
                                               fArcAngle = rand2;
                                           }
                                           MPW Shell
```
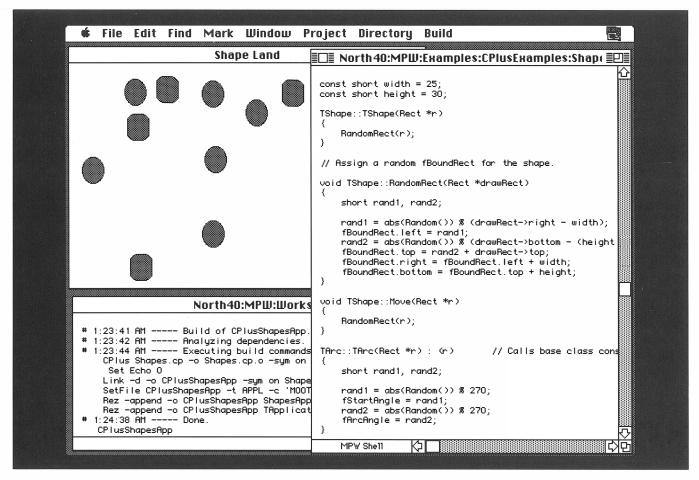
Overview

C++ is the latest object-oriented programming language available for use with the Macintosh® Programmer's Workshop (MPW™) development environment.

Apple's implementation fully supports the industry standard for object-oriented C as defined by AT&T's C++ Release 2.0. Apple has extended the language to support the Macintosh Toolbox and operating system, Object Pascal-based functions and procedures (such as those found in MacApp®), and the Standard Apple Numerics Environment (SANE®). MPW C++ can be debugged at the C++ source level using Apple's Symbolic Application Debugging Environment (SADE™). Applications built using MPW C++ can be compiled to run on the complete line of Apple® Macintosh personal computers, or to take advantage of the powerful hardware found in the high-end Macintosh models.

MPW C++ provides full support for object-oriented programming for C-based applications. The use of object-oriented programming techniques helps to reduce development time while increasing the reliability of the resulting applications.

Features	Benefits
▷ Support for object-oriented programming	▷ Reduces development time. ▷ Makes it easier to maintain applications. ▷ Increases the reliability of applications. ▷ Facilitates the creation of reusable code. ▷ Offers a better model for building applications than procedural programming can provide.
▷ Based on AT&T Release 2.0 C++	▷ Provides data abstraction, multiple inheritance, and message-passing capabilities. ▷ Offers operator overloading and protected variables within classes. ▷ Provides strong type-checking for C-based applications.
▷ Extensions for the Macintosh environment	▷ Supplies full access to the Macintosh Toolbox and operating system. ▷ Supports Object Pascal functions and procedures, for compatibility with MacApp. ▷ Provides access to SANE for numerical accuracy. ▷ Supports SADE for source-level debugging. ▷ Includes Apple's Commando interface for ease of use.
▷ CFront tool is integrated with MPW C	▷ Includes the MPW C scanner and preprocessor. ▷ Allows MPW C++ to produce tokenized C resulting in reduced build times.
▷ Support for multilingual Applications	▷ Lets you call Object Pascal functions and procedures from MPW C++. ▷ Allows C++ to be used with MacApp, further enhancing the programmer's productivity.
▷ Sample programs	▷ Provides examples of two stand-alone, MultiFinder-compatible applications. ▷ Provides an example of an MPW tool that is written in C++. ▷ Can be used as a learning aid or as the foundation for actual applications and tools.

Product Details

Object-Oriented Language Extensions

The MPW C++ system offers object-oriented programming to programmers using C. Multiple inheritance, operator overloading, and protected variables and members within classes are but a few of the object-oriented facilities of MPW C++.

C++ Translator

C++ source code is translated to C source code by the CFront tool. The resulting C source code is then compiled by MPW C. All of this is "automated" by CPlus, an MPW script provided with MPW C++. CPlus calls both CFront and MPW C, passing appropriate parameters. This results in a complete compilation of C++ source code.

MPW C++ uses the same preprocessor and scanner as MPW C. This allows MPW C++ to output tokenized C source code (as well as "standard" C source code), that reduces the build times typically associated with C++.

MPW C is available from the Apple Programmers and Developers Association (APDA™).

Source Level Debugging

MPW C++ works with Apple's Symbolic Application Debugging Environment (SADE). SADE can be used at either the source or the assembly level to debug applications and MPW tools. During compilation, MPW C++ can create the symbol files that are needed by SADE to debug C++ applications at the C++ source code level. This allows the powerful scripting language of SADE to be harnessed by C++ programmers during the development cycle, to further increase application reliability and decrease development time.

SADE is available from APDA.

Libraries

MPW C++ includes libraries for complex math and I/O stream processing. Apple has completely redone the Complex library. It retains the functionality of AT&T's Complex library and expands on it, using SANE as the basis for superior numerical accuracy.

Unmangler

Error messages produced while linking C++–based files can be very cryptic. MPW C++ comes with a tool for converting these "mangled" error messages into messages that are much easier to read. Also included is a resource for use with the MacsBug that allows MacsBug debugger to unmangle C++ function names.

Sample Programs

Three sample programs are included with MPW C++. Two of them are complete Macintosh applications and the third is a counting tool for MPW. These samples make excellent starting points for the development of other applications and tools.

C++ and MacApp

MacApp provides an object-oriented framework that implements the standard Macintosh user interface, including scrollable, resizable windows and multipage printing. MacApp fosters development of robust, professional-quality applications by providing you with extensive memory management support, exception-handling mechanisms, support for "undo" commands, and a large body of ready-to-use, high-quality code that can be inherited by your application.

A future release of MacApp will allow programmers to use C++ in place of Object Pascal. This will be accomplished through the use of special C++ interface files, since MPW C++ can call Object Pascal–based procedures and functions. These special interface files will be offered separately.

For more information on MacApp, refer to the MacApp data sheet (order number M0243LL/A) or contact APDA.

Training and Support

Apple offers courses in C++ programming. For details, please contact:

Registrar
Apple Developer University
20525 Mariani Avenue, M/S 75-2B
Cupertino, CA 95014
(408) 974-6215
AppleLink®: DEVUNIV

System Requirements	To use MPW C++ you will need the following:	▷ An Apple Macintosh Plus, Macintosh SE, or Macintosh II personal computer with at least 2 MB RAM (4 MB or more highly recommended). A 68020 or 68030 microprocessor is recommended.	▷ A hard disk ▷ Macintosh system software 6.0.2 or later ▷ MPW v.3.0 or later ▷ MPW C v.3.0 or later
Ordering Information	**MPW C++ v.3.1B1** APDA Order No. M0346LL/A	With your order, you'll receive: ▷ Two disks containing the Macintosh Programmer's Workshop C++ translator, C++ interfaces and libraries, an unmangler for CFront error messages, and sample programs.	▷ *Macintosh Programmer's Workshop C++ Reference* ▷ *AT&T C++ Release 2.0 Product Reference* ▷ *AT&T C++ Release 2.0 Library Manual* ▷ *AT&T C++ Release 2.0 Selected Readings*
Apple Programmers and Developers Association	Apple Computer, Inc. 20525 Mariani Avenue, M/S 33G Cupertino, CA 95014 TLX: 171-576 800-282-APDA (800-282-2732)	AppleLink®: APDA CompuServe: 766,2045 MCI: Postrom Fax: (408) 562-3971 GEnie: A.DEVELOPER3	