



hp services

education

course description

TAL/pTAL programming U4198S

course overview

This 4-day course teaches a basic understanding of the Transaction Application Language (TAL) and its usage. Through a series of exercises and labs, students will gain sufficient understanding of syntax and operations to develop and maintain TAL and pTAL programs.

audience

Systems programmers or maintainers who want to become proficient TAL or pTAL programmers.

benefits to you

- Introduction to TAL/pTAL
- Program organization and general syntax
- Simple data types and arrays
- Data transfer, program control, and data scan statements
- Operators and expressions
- Pointers and addressing
- Procedures and subprocedures
- The Common Run-Time Environment; Interfacing C and TAL
- Building native mode programs
- Debugging with the Inspect tool

pre-requisites

- Concepts and Facilities course

Recommended: At least six months of programming experience, preferably with C, PASCAL, or other block-structured procedure-based languages. Alternatively, one year of COBOL 85 programming experience

next steps

Guardian API Programming

to order

You can order this course online at <http://education.hp.com>. At the site, select a country, then choose "registration" or "Book a course" and fill out the online registration form.

why hp education?

- Experienced and best-in-the-field HP instructors
- Comprehensive student materials
- State-of-the-art classroom facilities
- Hands-on practice
- Focus on job-specific skills
- More than 120 locations worldwide
- Customized on-site delivery
- Online instructor-led and self-paced training at <http://itresourcecenter.hp.com>

detailed course outline: TAL/pTAL programming (U4198S)

module	key topics
introduction to TAL/pTAL	<ul style="list-style-type: none">• Introduce the Tandem® Transaction Application Language (TAL and pTAL)• Describe TAL/pTAL programming in the Guardian environment• Describe tools to assist the developer in writing and debugging TAL and pTAL programs
program organization and general syntax	<ul style="list-style-type: none">• Discuss basic syntax and organization of TAL programs• Discuss identifiers, begin-end blocks, constants, operators, variable declarations, and procedure declarations.• Describe compiler (both TAL and pTAL) directives that control listings• Describe sourcing from another file• Explain the environment set up for the Inspect product and compile for syntax only• Demonstrate basic Inspect software commands
simple data types and arrays	<ul style="list-style-type: none">• Describe how data is stored and what facilities TAL provides to access data• Discuss correct data types for various purposes, definitions, literals, labels, and data equivalencing
program flow statements	<ul style="list-style-type: none">• Describe types of program flow statements and their usage• Describe statements for CASE, IF, program control, bit extraction, and manipulation• Lab Exercise• Use constructs such as the CASE statement and WHILE loop
terminal I/O	<ul style="list-style-type: none">• Discuss how to perform terminal I/O in programs• Discuss error handling and data conversion• Lab Exercise• Perform terminal I/O to prompt for data and implement data conversion
operators and expressions	<ul style="list-style-type: none">• Describe types of operators and expressions• Discuss arithmetic expressions and conditional expressions• Discuss special expressions such as: assignment, CASE, IF, and group comparison expressions
pointers and addressing	<ul style="list-style-type: none">• Describe direct and indirect data access, pointers, and structures.• Discuss additional pTAL pointer data types• Discuss data transfer and scan statements to illustrate use of pointers and structures• Describe pTAL caveats• Lab Exercise• Use pointers, addressing, and data movement
procedures and subprocedures	<ul style="list-style-type: none">• Describe procedures and subprocedures, with or without parameter passing• Discuss procedure declaration, subprocedure declaration, and procedure calls and returns• Explain the use of formal and actual parameters in procedure calls and returns• Discuss TAL and pTAL considerations for Guardian procedures that are obsolete for pTAL procedures• Lab Exercise• Code procedures, with and without parameters• Invoke procedures, with and without parameters
interfacing C and TAL	<ul style="list-style-type: none">• Describe the Common Run-Time Environment (CRE)• Demonstrate how to interface C and TAL and memory model considerations• Discuss the use of development tools such as Bind, nld, and noft• Lab Exercise• Build runnable units from separately compiled C and/or TAL objects

for more information

For more information on HP Education Services, contact any of our worldwide offices or visit our worldwide web site on the internet at <http://education.hp.com>

Technical information in this document is subject to change without notice.

Microsoft®, Windows®, MS Windows®, and Windows NT® are U.S. registered trademarks of Microsoft Corporation. UNIX® is a registered trademark of the Open Group.

©Copyright Hewlett-Packard Company 2000. All Rights Reserved. Reproduction, adaptation, or translation without prior written permission is prohibited except as allowed under the copyright laws.

9/02 U4198S

