



TAL/pTAL programming U4198S

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.

TAL/pTAL programming

Price USD \$3,200

**Links to local
schedules,
pricing and** [US/Canada](#)
[Mexico/Latin America](#)
[Brazil](#)

HP course # U4198S

Category NonStop

Duration 4 days

Audience

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

Prerequisites

- 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

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

Next steps

- Guardian API programming

Course outline

Introduction to TAL/pTAL

- Transaction Application Language (TAL and pTAL)
- TAL/pTAL programming in the Guardian environment of the HP NonStop Kernel operating system
- Tools to assist the developer in writing and debugging TAL and pTAL programs

Program organization and general syntax

- Basic syntax and organization of TAL programs
- Identifiers, begin-end blocks, constants, operators, variable declarations, and procedure declarations
- Compiler (both TAL and pTAL) directives that control listings
- Sourcing from another file
- The environment set up for the Inspect product and compile for syntax only
- Basic Inspect software commands

Simple data types and arrays

- How data is stored and what facilities TAL provides to access data
- Correct data types for various purposes, definitions, literals, labels, and data equivalencing
- Program flow statements Types of program flow statements and their usage
- 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

- How to perform terminal I/O in programs
- Error handling and data conversion
- Lab exercise:
 - Perform terminal I/O to prompt for data and implement data conversion

Operators and expressions

- Types of operators and expressions
- Arithmetic expressions and conditional expressions
- Special expressions such as: assignment, CASE, IF, and group comparison expressions

Pointers and addressing

- Direct and indirect data access, pointers, and structures
- Additional pTAL pointer data types
- Data transfer and scan statements to illustrate use of pointers and structures
- PTAL caveats
- Lab exercise:
 - Use pointers, addressing, and data movement

Procedures and subprocedures

- Procedures and subprocedures, with or without parameter passing
- Procedure declaration, subprocedure declaration, and procedure calls and returns
- The use of formal and actual parameters in procedure calls and returns
- 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

- Common Run-Time Environment (CRE)
- How to interface C and TAL and memory model considerations
- Use of development tools such as Bind, nld, and noft
- Lab exercise:
 - Build runnable units from separately compiled C and/or TAL objects

Learn more at

hpe.com/us/training/nonstop