



# TAL/pTAL Programming U4198S

This 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.

HPE course number	U4198S
Course length	4 days
Delivery mode	ILT/VILT
View schedule, local pricing, and register	<a href="#">View now</a>
View related courses	<a href="#">View now</a>

## Why HPE Education Services?

- IDC MarketScape leader 5 years running for IT education and training\*
- Recognized by IDC for leading with global coverage, unmatched technical expertise, and targeted education consulting services\*
- Key partnerships with industry leaders OpenStack®, VMware®, Linux®, Microsoft®, ITIL, PMI, CSA, and SUSE
- Complete continuum of training delivery options—self-paced eLearning, custom education consulting, traditional classroom, video on-demand instruction, live virtual instructor-led with hands-on lab, dedicated onsite training
- Simplified purchase option with HPE Training Credits

## 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

## 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

\*Realize Technology Value with Training, IDC Infographic 2037, Sponsored by HPE, October 2017



## Detailed course outline

<b>Module 1: Introduction to TAL/pTAL</b>	<ul style="list-style-type: none"> <li>• Introduce the Tandem Transaction Application Language (TAL and pTAL)</li> <li>• Describe TAL/pTAL programming in the Guardian environment</li> <li>• Describe tools to assist the developer in writing and debugging TAL and pTAL programs</li> </ul>
<b>Module 2: Program organization and general syntax</b>	<ul style="list-style-type: none"> <li>• Discuss basic syntax and organization of TAL programs</li> <li>• Discuss identifiers, begin-end blocks, constants, operators, variable declarations, and procedure declarations</li> <li>• Describe compiler (both TAL and pTAL) directives that control listings</li> <li>• Describe sourcing from another file</li> <li>• Explain the environment set up for the Inspect product and compile for syntax only</li> <li>• Demonstrate basic Inspect software commands</li> </ul>
<b>Module 3: Simple data types and arrays program flow statements</b>	<ul style="list-style-type: none"> <li>• Describe how data is stored and what facilities TAL provides to access data</li> <li>• Discuss correct data types for various purposes, definitions, literals, labels, and data equivalencing</li> <li>• Describe types of program flow statements and their usage</li> <li>• Describe statements for CASE, IF, program control, bit extraction, and manipulation</li> <li>• Lab Exercise</li> <li>• Use constructs such as the CASE statement and WHILE loop</li> </ul>
<b>Module 4: Terminal I/O</b>	<ul style="list-style-type: none"> <li>• Discuss how to perform terminal I/O in programs</li> <li>• Discuss error handling and data conversion</li> <li>• Lab Exercise</li> <li>• Perform terminal I/O to prompt for data and implement data conversion</li> </ul>
<b>Module 5: Operators and expressions</b>	<ul style="list-style-type: none"> <li>• Describe types of operators and expressions</li> <li>• Discuss arithmetic expressions and conditional expressions</li> <li>• Discuss special expressions such as: assignment, CASE, IF, and group comparison expressions</li> </ul>
<b>Module 6: Pointers and addressing</b>	<ul style="list-style-type: none"> <li>• Describe direct and indirect data access, pointers, and structures</li> <li>• Discuss additional pTAL pointer data types</li> <li>• Discuss data transfer and scan statements to illustrate use of pointers and structures</li> <li>• Describe pTAL caveats</li> <li>• Lab Exercise</li> <li>• Use pointers, addressing, and data movement</li> </ul>
<b>Module 7: Procedures and subprocedures</b>	<ul style="list-style-type: none"> <li>• Describe procedures and subprocedures, with or without parameter passing</li> <li>• Discuss procedure declaration, subprocedure declaration, and procedure calls and returns</li> <li>• Explain the use of formal and actual parameters in procedure calls and returns</li> <li>• Discuss TAL and pTAL considerations for Guardian procedures that are obsolete for pTAL procedures</li> <li>• Lab Exercise</li> <li>• Code procedures, with and without parameters</li> <li>• Invoke procedures, with and without parameters</li> </ul>
<b>Module 8: Interfacing C and TAL</b>	<ul style="list-style-type: none"> <li>• Describe the Common Run-Time Environment (CRE)</li> <li>• Demonstrate how to interface C and TAL and memory model considerations</li> <li>• Discuss the use of development tools such as Bind, nld, and noft</li> <li>• Lab Exercise</li> <li>• Build runnable units from separately compiled C and/or TAL objects</li> </ul>

## Next steps

Guardian API Programming.



Learn more at  
[hpe.com/ww/learnnonstop](https://hpe.com/ww/learnnonstop)

Follow us:



---

© Copyright 2020 Hewlett Packard Enterprise Development LP. The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Microsoft is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries. The OpenStack Word Mark is either a registered trademark/service mark or trademark/service mark of the OpenStack Foundation, in the United States and other countries and is used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation or the OpenStack community. Pivotal and Cloud Foundry are trademarks and/or registered trademarks of Pivotal Software, Inc. in the United States and/or other countries. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other third-party trademark(s) is/are property of their respective owner(s).

U4198S D.01, January 2020