PD ISO/IEC TS 19217:2015



BSI Standards Publication

Information technology — Programming languages — C++ Extensions for concepts

bsi.

...making excellence a habit.™

National foreword

This Published Document is the UK implementation of ISO/IEC TS 19217:2015.

The UK participation in its preparation was entrusted to Technical Committee IST/5, Programming languages, their environments and system software interfaces.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2015.
Published by BSI Standards Limited 2015

ISBN 978 0 580 92559 7
ICS 35.060

Compliance with a British Standard cannot confer immunity from legal obligations.

This Published Document was published under the authority of the Standards Policy and Strategy Committee on 29 February 2016.

Amendments/corrigenda issued since publication

Date	Text affected

PD ISO/IEC TS 19217:2015

TECHNICAL SPECIFICATION

ISO/IEC TS 19217

First edition
2015-11-15

Information technology — Programming languages — C++ Extensions for concepts

Technologie de l'information — Langages de programmation — Extensions C++ pour les concepts warning

Reference number
ISO/IEC TS 19217:2015(E)

© ISO/IEC 2015



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2015, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

N4549

Contents

Contents iii

List of Tables iv

Foreword v

1 General 1
- 1.1 Scope . . . 1
- 1.2 Normative references . . . 1
- 1.3 Terms and definitions . . . 1
- 1.4 Implementation compliance . . . 2
- 1.5 Feature-testing recommendations . . . 2
- 1.6 Acknowledgments . . . 2

2 Lexical conventions 3
- 2.1 Keywords . . . 3

5 Expressions 4
- 5.1 Primary expressions . . . 4

7 Declarations 11
- 7.1 Specifiers . . . 11

8 Declarators 21
- 8.3 Meaning of declarators . . . 21

10 Derived classes 25
- 10.3 Virtual functions . . . 25

13 Overloading 26
- 13.1 Overloadable declarations . . . 26
- 13.3 Overload resolution . . . 26
- 13.4 Address of overloaded function . . . 27

14 Templates 28
- 14.1 Template parameters . . . 28
- 14.2 Introduction of template parameters . . . 30
- 14.3 Names of template specializations . . . 32
- 14.4 Template arguments . . . 33
- 14.6 Template declarations . . . 33
- 14.7 Name resolution . . . 39
- 14.8 Template instantiation and specialization . . . 40
- 14.9 Function template specializations . . . 41
- 14.10 Template constraints . . . 43

A Compatibility 53
- A.1 C++ extensions for Concepts and ISO C++ 2014 . . . 53

N4549

List of Tables

A Feature-test macro(s) 2

10 *simple-type-specifier*s and the types they specify 12

B Value of folding empty sequences 36

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, SC 22, *Programming languages, their environments and system software interfaces.*

1 General [intro]

1.1 Scope [intro.scope]

1 This Technical Specification describes extensions to the C++ Programming Language (1.2) that enable the specification and checking of constraints on template arguments, and the ability to overload functions and specialize class templates based on those constraints. These extensions include new syntactic forms and modifications to existing language semantics.

2 The International Standard, ISO/IEC 14882, provides important context and specification for this Technical Specification. This document is written as a set of changes against that specification. Instructions to modify or add paragraphs are written as explicit instructions. Modifications made directly to existing text from the International Standard use underlining to represent added text and ~~strikethrough~~ to represent deleted text.

3 WG21 paper N4191 defines "fold expressions", which are used to define constraint expressions resulting from the use of *constrained-parameter*s that declare template parameter packs. This feature is not present in ISO/IEC 14882:2014, but it is planned to be included in the next revision of that International Standard. The specification of that feature is included in this document.

1.2 Normative references [intro.refs]

1 The following referenced document is indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

(1.1) — ISO/IEC 14882:2014, *Programming Languages – C++*

ISO/IEC 14882:2014 is hereafter called the *C++ Standard*. The numbering of Clauses, sections, and paragraphs in this document reflects the numbering in the C++ Standard. References to Clauses and sections not appearing in this Technical Specification refer to the original, unmodified text in the C++ Standard.

1.3 Terms and definitions [intro.defs]

Modify the definitions of "signature" to include associated constraints (14.10.2). This allows different translation units to contain definitions of functions with the same signature, excluding associated constraints, without violating the one definition rule (3.2). That is, without incorporating the constraints in the signature, such functions would have the same mangled name, thus appearing as multiple definitions of the same function.

1.3.1 [defns.signature]

signature
<function> name, parameter type list (8.3.5), ~~and~~ enclosing namespace (if any), and any associated constraints (14.10.2)
[*Note:* Signatures are used as a basis for name mangling and linking. — *end note*]

1.3.2 [defns.signature.templ]

signature
<function template> name, parameter type list (8.3.5), enclosing namespace (if any), return type, ~~and~~ template parameter list, and any associated constraints (14.10.2)

1.3.3 [defns.signature.member]