

ISO/IEC TR 24772-2:2020-04 (E)

Programming languages - Guidance to avoiding vulnerabilities in programming languages - Part 2: Ada

Contents		Page
Foreword		vii
Introduction		viii
1 Scope		1
2 Normative references		1
3 Terms and definitions		1
4 Language concepts		6
4.1 Enumeration type.....		6
4.2 Exception.....		6
4.3 Hiding.....		6
4.4 Implementation defined.....		6
4.5 Type conversions.....		6
4.6 Operational and Representation Attributes.....		7
4.7 User defined types.....		7
4.8 Pragma compiler directives.....		7
4.8.1 Pragma Atomic		7
4.8.2 Pragma Atomic_Components		7
4.8.3 Pragma Convention		7
4.8.4 Pragma Detect_Blocking		7
4.8.5 Pragma Discard_Names		7
4.8.6 Pragma Export		8
4.8.7 Pragma Import		8
4.8.8 Pragma Normalize_Scalars		8
4.8.9 Pragma Pack		8
4.8.10 Pragma Restrictions		8
4.8.11 Pragma Suppress		8
4.8.12 Pragma Unchecked_Union		8
4.8.13 Pragma Volatile		8
4.8.14 Pragma Volatile_Components		8
4.9 Separate compilation.....		8
4.10 Storage pool.....		8
4.11 Unsafe programming.....		9
5 General guidance for Ada		9
5.1 Ada language design.....		9
5.2 Top avoidance mechanisms.....		10
6 Specific guidance for Ada		11
6.1 General.....		11
6.2 Type system [IHN].....		11
6.2.1 Applicability to language.....		11
6.2.2 Guidance to language users.....		11
6.3 Bit representation [STR].....		11
6.3.1 Applicability to language.....		11
6.3.2 Guidance to language users.....		11
6.4 Floating-point arithmetic [PLF].....		12
6.4.1 Applicability to language.....		12
6.4.2 Guidance to language users.....		12
6.5 Enumerator issues [CCB].....		12
6.5.1 Applicability to language.....		12
6.5.2 Guidance to language users.....		13

6.6	Conversion errors [FLC]	13
6.6.1	Applicability to language	13
6.6.2	Guidance to language users	13
6.7	String termination [CJM]	14
6.8	Buffer boundary violation (buffer overflow) [HCB]	14
6.9	Unchecked array indexing [XYZ]	14
6.9.1	Applicability to language	14
6.9.2	Guidance to language users	14
6.10	Unchecked array copying [XYW]	14
6.11	Pointer type conversions [HFC]	14
6.11.1	Applicability to language	14
6.11.2	Guidance to language users	15
6.12	Pointer arithmetic [RVG]	15
6.13	Null pointer dereference [XYH]	15
6.13.1	Applicability to the language	15
6.13.2	Guidance to language users	15
6.14	Dangling reference to heap [XYK]	15
6.14.1	Applicability to language	15
6.14.2	Guidance to language users	16
6.15	Arithmetic wrap-around error [FIF]	16
6.16	Using shift operations for multiplication and division [PIK]	16
6.17	Choice of clear names [NAI]	16
6.17.1	Applicability to language	16
6.17.2	Guidance to language users	17
6.18	Dead store [WXQ]	17
6.18.1	Applicability to language	17
6.18.2	Guidance to language users	17
6.19	Unused variable [YZS]	17
6.19.1	Applicability to language	17
6.19.2	Guidance to language users	17
6.20	Identifier name reuse [YOW]	18
6.20.1	Applicability to language	18
6.20.2	Guidance to language users	18
6.21	Namespace issues [BJL]	18
6.22	Initialization of variables [LAV]	18
6.22.1	Applicability to language	18
6.22.2	Guidance to language users	19
6.23	Operator precedence/order of evaluation [JCW]	19
6.23.1	Applicability to language	19
6.23.2	Guidance to language users	19
6.24	Side-effects and order of evaluation [SAM]	20
6.24.1	Applicability to language	20
6.24.2	Guidance to language users	20
6.25	Likely incorrect expression [KOA]	20
6.25.1	Applicability to language	20
6.25.2	Guidance to language users	21
6.26	Dead and deactivated code [XYQ]	21
6.26.1	Applicability to language	21
6.26.2	Guidance to language users	21
6.27	Switch statements and static analysis [CLL]	21
6.27.1	Applicability to language	21
6.27.2	Guidance to language users	22
6.28	Demarcation of control flow [EOJ]	22
6.29	Loop control variables [TEX]	22
6.30	Off-by-one error [XZH]	22
6.30.1	Applicability to language	22
6.30.2	Guidance to language users	23
6.31	Unstructured programming [EWD]	23
6.31.1	Applicability to language	23
6.31.2	Guidance to language users	23
6.32	Passing parameters and return values [CSJ]	23
6.32.1	Applicability to language	23

6.32.2	Guidance to language users.....	23
6.33	Dangling references to stack frames [DCM].....	23
6.33.1	Applicability to language.....	23
6.33.2	Guidance to language users.....	24
6.34	Subprogram signature mismatch [OTR].....	24
6.34.1	Applicability to language.....	24
6.34.2	Guidance to language users.....	24
6.35	Recursion [GDL].....	25
6.35.1	Applicability to language.....	25
6.35.2	Guidance to language users.....	25
6.36	Ignored error status and unhandled exceptions [OYB].....	25
6.36.1	Applicability to language.....	25
6.36.2	Guidance to language users.....	25
6.37	Type-breaking reinterpretation of data [AMV].....	25
6.37.1	Applicability to language.....	25
6.37.2	Guidance to language users.....	26
6.38	Deep vs. shallow copying [YAN].....	26
6.38.1	Applicability to language.....	26
6.38.2	Guidance to language users.....	26
6.39	Memory leak and heap fragmentation [XYL].....	26
6.39.1	Applicability to language.....	26
6.39.2	Guidance to language users.....	27
6.40	Templates and generics [SYM].....	27
6.41	Inheritance [RIP].....	27
6.41.1	Applicability to language.....	27
6.41.2	Guidance to language users.....	27
6.42	Violations of the Liskov substitution principle or the contract model [BLP].....	28
6.42.1	Applicability to language.....	28
6.42.2	Guidance to language users.....	28
6.43	Redispatching [PPH].....	28
6.43.1	Applicability to language.....	28
6.43.2	Guidance to language users.....	28
6.44	Polymorphic variables [BKK].....	29
6.44.1	Applicability to language.....	29
6.44.2	Guidance to language users.....	29
6.45	Extra intrinsics [LRM].....	29
6.46	Argument passing to library functions [TR].....	29
6.46.1	Applicability to language.....	29
6.46.2	Guidance to language users.....	30
6.47	Inter-language calling [DJS].....	30
6.47.1	Applicability to language.....	30
6.47.2	Guidance to language users.....	30
6.48	Dynamically-linked code and self-modifying code [NYY].....	30
6.49	Library signature [NSQ].....	30
6.49.1	Applicability to language.....	30
6.49.2	Guidance to language users.....	31
6.50	Unanticipated exceptions from library routines [HJW].....	31
6.50.1	Applicability to language.....	31
6.50.2	Guidance to language users.....	31
6.51	Pre-processor directives [NMP].....	31
6.52	Suppression of language-defined run-time checking [MXB].....	31
6.52.1	Applicability to Language.....	31
6.52.2	Guidance to language users.....	32
6.53	Provision of inherently unsafe operations [SKL].....	32
6.53.1	Applicability to Language.....	32
6.53.2	Guidance to language users.....	32
6.54	Obscure language features [BRS].....	32
6.54.1	Applicability to language.....	32

6.54.2	Guidance to language users	32
6.55	Unspecified behaviour [BQF]	32
6.55.1	Applicability to language	32
6.55.2	Guidance to language users	33
6.56	Undefined behaviour [EWF]	33
6.56.1	Applicability to language	33
6.56.2	Guidance to language users	34
6.57	Implementation-defined behaviour [FAB]	34
6.57.1	Applicability to language	34
6.57.2	Guidance to language users	35
6.58	Deprecated language features [MEM]	35
6.58.1	Applicability to language	35
6.58.2	Guidance to language users	35
6.59	Concurrency — Activation [CGA]	35
6.59.1	Applicability to language	35
6.59.2	Guidance to language users	35
6.60	Concurrency — Directed termination [CGT]	36
6.60.1	Applicability to language	36
6.60.2	Guidance to language users	36
6.61	Concurrent data access [CGX]	36
6.61.1	Applicability to language	36
6.61.2	Guidance to language users	36
6.62	Concurrency — Premature termination [CGS]	36
6.62.1	Applicability to language	36
6.62.2	Guidance to language users	36
6.63	Protocol lock errors [CGM]	37
6.63.1	Applicability to language	37
6.63.2	Guidance to language users	37
6.64	Reliance on external format strings [SHL]	37
7	Language-specific vulnerabilities for Ada	37
8	Implications for standardization	37
	Bibliography	39
	Index	41