4.8 — Floating point numbers

ALEX AUGUST 11, 2021

H rcecpqe pce pc_ rehnpeansl rg eeu f mjce sk `cpqê s rqmk crgk cqu ce ccberqrpc *tcpvi*_pece sk `cpqmpe sk `cpqu grf e eb_argml _jemk nml cl r @ floating point e wnc t_pg `jce e _pg `jcef _re_l e mjbe pc_je sk `cpq s af e qe Ï Ï Ï ñ ñ ñ Ï ñ ñ Ï Ï Ï Ï Sf ce *djm_rgl ee*_pre f ce _k ce *djm_rgl eemnmgl r* pccdcpqe f ce _arf _re f cebcagk _jemngl r a_l e djm_r Ñ f _reqgrea_l eqs nnmpre e _pg `jce sk `cpemdcqgrqe cdmpce l be drcpef c bcagk _jemngl r

Sf cpce pce f pcce bgdtpcl re djm_rgl eemngl rebe_r_e wncq float double e l be long double @qu grf e gl rcecpq Bgggmcqe mebcdj ce f ce ars _jeg ce f ce cq e wncq s rqreq es _p_l rcek g gk sk eqkcq Nl ek mbcpl e paf gcars pcq djm_rgl eemngl rencnpcqcl r_rgml _jk mre ju _wqemjju q DDD e Ï Ïe g _pwmk _ r H ef g mk _r e djm_r e Ïe wrcq bms `jce Ïe l be jml e bms `jce_l e ce cosgt_jcl reme bms `jce e Ïe wrcqg Ï Ïñ gqe rcl n_bbcberq Ïe wrcq Ï Ïe wrcq

Ejm_rgl eemngl reb_r_e wncq e pce ju _wqeel cbe _l ef mjbemggqce l be ce_rgce _js cq

Category	Type	Minimum Size	Typical Size
djm_rgl eemngl r	djm_r	Ïe wrcq	Ïe wrcq
	bms `jc	Ïe wrcq	Ïe wrcq
	jml e bms `jc	Ïe wrcq	Ï Ï Ïe wrcq

Gcpce pcemk cebcdgl gml qemdjm_rgl eemngl re sk `cpq

```
float fValue;
double dValue;
long double ldValue;
```

V f cl e sqgl e djm_rgl eemngl re gcp_jqe ju_wqee ajsbce re c_qreml cebcagk _jej_ace tcl e f cebcagk _je Ï Sf gqf cjnqef cemk ngcp sl bcpqr_l bef _ref ce sk `cpe e djm_rgl eemngl re sk `cpe l be mre l e rcecp

```
int x{5}; // 5 means integer
double y{5.0}; // 5.0 is a floating point literal (no suffix means double type by
default)
float z{5.0f}; // 5.0 is a floating point literal, f suffix means float type
```

Mmrcef _ rę`wþcd_sjrłęjm_rg eęnmg ręjgcp_jqęþcd_sjręmęwncęþms `jcł@l ęęps ddyęgęs qcbęmęþcl mrcę ęjgcp_jęmdęwncęjm_rł

Best practice

@ju _wqęk _i cęąs pcęf cęwncęndęwns pęjgcp_jqęk _raf ęf cęwncęndęf cęt_pg `jcqęf cwŊpcę`cgl eeę qqgel cbęmęmęps qcbęmęgl gg jgxcł Nrf cpu gqcę_l ęs l l cacqq_pwęaml tcpqgml ęu gjjęcqs jrłęnmqqg`jwęu grf ę ęjmqqęndęnpcaggml ł

Warning

L _i cęąs pcęwns ęþml Ŋęs qcęg rcecpęjgcp_jqęu f cpcęjm_rg eęnmg ręjgcp_jqęf ms jbę`cęs qcbłSf gqęg ajs bcqęu f cl ęg gg jgxg eęmpęqqgel g e t_js cqęmęjm_rg eęnmg ręm`hcarqłbmg eęjm_rg eęnmg ręprf k crgaęl bę_jjg eęs l argml qęf _ręcvncaręjm_rg eęnmg ręt_js cqł

Printing floating point numbers

Mmu ęaml qgbcpęf gąęjgk njcęnpmep_k IJ

```
#include <iostream>

int main()
{
 std::cout << 5.0 << '\n';
 std::cout << 6.7f << '\n';
 std::cout << 9876543.21 <<
'\n';

 return 0;
}
```

Sf cęęcqs jrqęndęf gqęcck g ejwęqgk njcęnpmep_k ęk _węqs pnpgqcęwns IJ

```
5
6.7
9.87654e+06
```

H ęrf cęlgoq rę_ qckęrf cęrbIJams ręnpg rcbę kęctcl ęrf ms ef ęu cęwncbeg ęĭĦĦęAwbcd_sjrkęrbIJams ręu gjję mręnpg ręrf cęb_argml _jęn_ prem ę l sk `cpegęrf cęb_argml _jęn_ premęĩĦ

H ęrf cęcaml bę_ qckęrf cę l sk `cpęnpg rqę qęu cęvncarĦ

H ęrf cęrf gobę_ qckęnpg rcbęrf cę l sk `cpęl ęqagcl rgdgaęl mr_ rgml ęgdęwms ęccbę ęcdęcqęf cpęl ęqagcl rgdgaęl mr_ rgml kęccęjcqqml ĭĦęĥ H rpmbs argml ęmęqagcl rgdgaęl mr_ rgml ęĦ

Floating point range

@qqsk g eęHDDDęĭĬęcnpcqcl r_ rgml IJ

Size	Range	Precision
Ĭę`wrcq	®ĩĦİęvęĩĥİęmę®ĩĦĭęvęĩīİ	Įĥ ęqgel gdga_l rębgegrqęrwnga_jjwę
İę`wrcq	®ĪĦīęvęĩĥĬİęmę®ĩĦİĭęvęĩīĬİ	īīĥİęqgel gdga_l rębgegrqęrwnga_jjwęĮ
İĬĥ`grqęrwnga_jjwęqcqęĪęmpęĮę`wrcqę	®ĩĦĮęvęĩĥıīĪęmę®ĩĦİęvęĩĬıīĪ	īİĥĩīęqgel gdga_l rębgegrq
īĮę`wrcq	®ĩĦĮęvęĩĥıīĪęmę®ĩĦİęvęĩĬıīĪ	īīĥĮęqgel gdga_l rębgegrq

[illegible]

[illegible]

Floating point precision

[illegible]

[illegible]

[illegible] precision [illegible]

[illegible]

[illegible]

```cpp
#include <iostream>

int main()
{
    std::cout << 9.87654321f << '\n';
    std::cout << 987.654321f << '\n';
    std::cout << 987654.321f << '\n';
    std::cout << 9876543.21f << '\n';
    std::cout << 0.0000987654321f << '\n';

    return 0;
}
```

[illegible]

```
9.87654
987.654
987654
9.87654e+006
9.87654e-005
```

[illegible]

[illegible]

[illegible]

[illegible] `output manipulator` [illegible] `std::setprecision()` [illegible] Output manipulators [illegible]

```cpp
#include <iostream>
#include <iomanip> // for output manipulator std::setprecision()

int main()
{
    std::cout << std::setprecision(16); // show 16 digits of precision
    std::cout << 3.33333333333333333333333333333333333333f <<'\n'; // f suffix means float
    std::cout << 3.33333333333333333333333333333333333333 << '\n'; // no suffix means double

    return 0;
}
```

[illegible]

```
3.333333253860474
3.333333333333334
```

[illegible] `std::setprecision()` [illegible]

[illegible]

```cpp
#include <iomanip> // for std::setprecision()
#include <iostream>

int main()
{
    float f { 123456789.0f }; // f has 10 significant digits
    std::cout << std::setprecision(9); // to show 9 digits in f
    std::cout << f << '\n';

    return 0;
}
```

[illegible]

```
123456792
```

[illegible] rounding error.

[illegible]

Best practice

[illegible]

Rounding errors make floating point comparisons tricky

[illegible]

[illegible]

```cpp
#include <iomanip> // for std::setprecision()
#include <iostream>

int main()
{
    double d{0.1};
    std::cout << d << '\n'; // use default cout precision of 6
    std::cout << std::setprecision(17);
    std::cout << d << '\n';

    return 0;
}
```

[illegible]

```
0.1
0.10000000000000001
```

[illegible]

[illegible]

[illegible]

```
#include <iomanip> // for std::setprecision()
#include <iostream>

int main()
{
    std::cout << std::setprecision(17);

    double d1{ 1.0 };
    std::cout << d1 << '\n';

    double d2{ 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 }; // should equal
1.0
    std::cout << d2 << '\n';

    return 0;
}
```

```
1
0.99999999999999989
```

[illegible]

[illegible]

Key insight

[illegible]

[illegible]

NaN and Inf

[illegible]

GcpcNqe enpmep_k eqf mu g eejjeqf pccIJ

```cpp
#include <iostream>

int main()
{
    double zero {0.0};
    double posinf { 5.0 / zero }; // positive infinity
    std::cout << posinf << '\n';

    double neginf { -5.0 / zero }; // negative infinity
    std::cout << neginf << '\n';

    double nan { zero / zero }; // not a number (mathematically invalid)
    std::cout << nan << '\n';

    return 0;
}
```

@l beqf cepcqsjrqesqg eeUgps _jeRrs bgmeĪĨĨI eml eV g bmu qIJ

```
1.#INF
-1.#INF
1.#IND
```

*HMEeq*r_l bqedhpeg dg grwÎe l be*HMCeq*r_l bqedhpeg bcrcpk g _rcH Mmrcef _ref cepcqs jrqendpg rg eeH de l beM_ Me pcenj_rdmpk encagga qewns p pcqs jrqek _we _pwH

Best practice

@tmgbebg ggml e wÃe jrmecrf cpÎtcl egdwns peamk ngjcpes nnmprqegH

Conclusion

Smes k k _pgxcÎef ceu mef g eqewns ef ms jbepck ck `cpe `ms redjm_rg eenmg re sk `cpqIJ

ĩ HEjm_rg eenmg re sk `cpqe pcesqcds jedmpermpg eetcpwj_pecempetcpwek _jje sk `cpqÎg ajsbg eef mqceu gf eb_argml _jeamk nml cl rqH

Ĩ HEjm_rg eenmg re sk `cpqendrcl ef _tcek _jjepns l bg eeppmpqÎtcl eu f cl ef ce sk `cpf _qetu cpegel gga_l rebgegrqef _l ef cenpcaggml H
L _l wegk cqef cqcemesl l mrgacbe`ca_sqcef cwe pcenmek _jjÎe l be`ca_sqcef ce sk `cpqe pcepsl a_rcbedmpens rns rHGmu ctcpÎ
amk n_pgml qendejm_rg eenmg re sk `cpqek _we mregtcef cecvncarcbepcqs jrqH Ocpdmpk g eek _rf ck _rga_jemcp_rgml qeml ef cqcet_jscq u gjea_sqcef cepns l bg eeppmpqermepmu e_pecpH

Mcvregjcqqml

ĬH Amnjc_l et_jscq

A_ai ermer_ `jcemdeaml rcl rq

Opct gms qejcqqml

ĬH H rpmbs argml ermepagcl rgdbae mr_rgml

Leave a comment... Put C++ code between triple-backticks (markdown style):```Your C++ cod

Name*

Email*

@t_r_pqęþmk ę ƒ rrnqĮĵħep_t_r_pĦmk ħę pcęml l carcbęrnęwns pęnpmtgbcbęck _gię bbpcqqĦ

Mmrgdwęk cę `ms rępcnjgcqĮęę ę **POST COMMENT**

444 COMMENTS

Newest

©2021 Learn C++

X

X