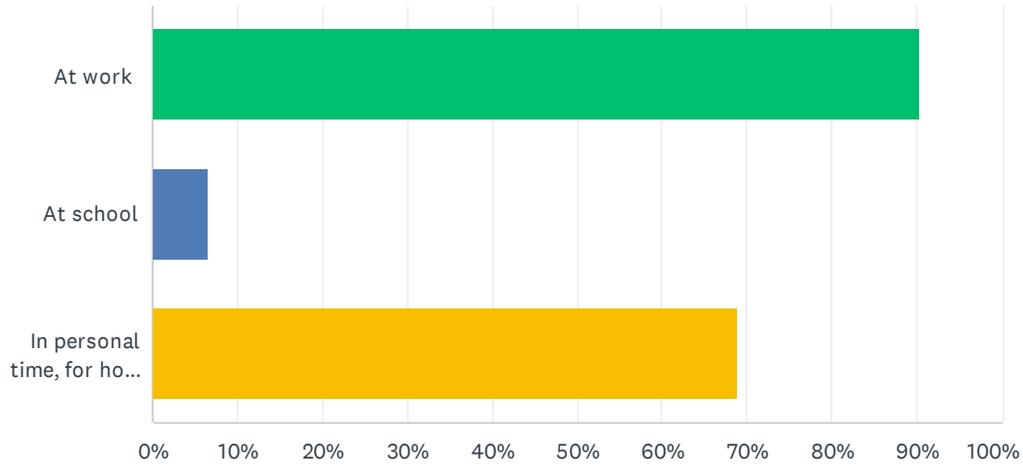


## Q1 Where do you use C++? (select all that apply)

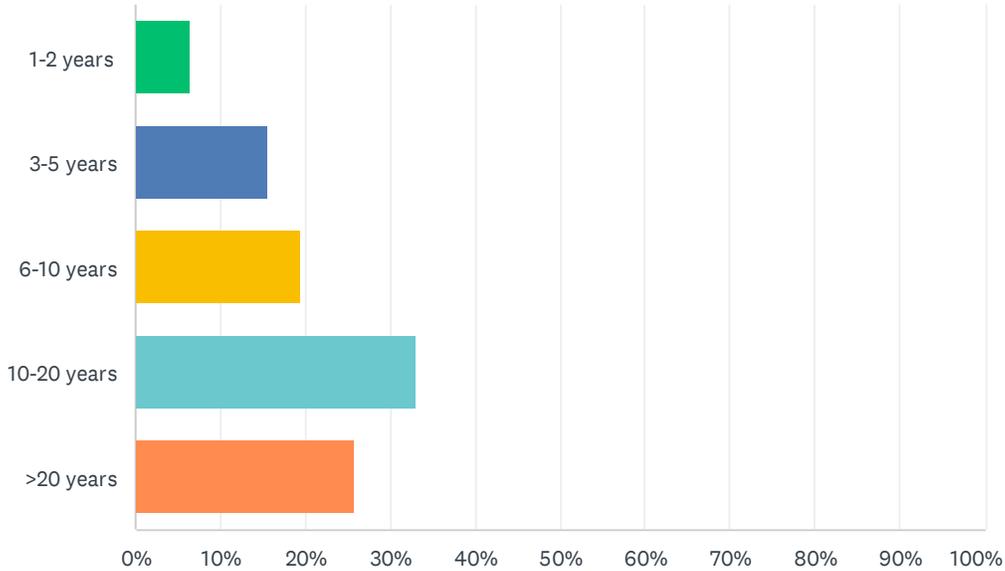
Answered: 1,188 Skipped: 1



ANSWER CHOICES	RESPONSES	
At work	90.40%	1,074
At school	6.57%	78
In personal time, for hobby projects or to try new things	68.94%	819
Total Respondents: 1,188		

## Q2 How many years of programming experience do you have in C++ specifically?

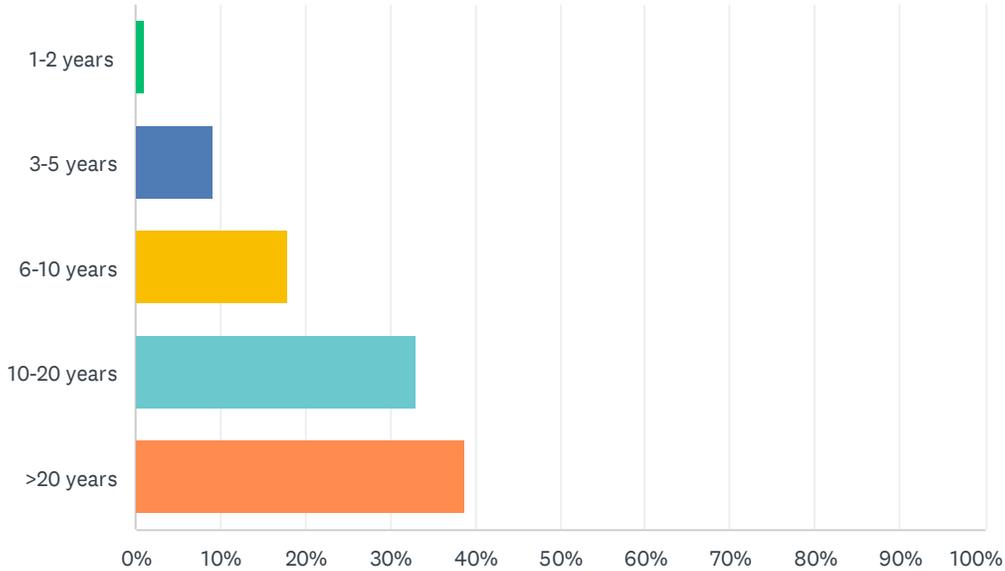
Answered: 1,184 Skipped: 5



ANSWER CHOICES	RESPONSES
1-2 years	6.33% 75
3-5 years	15.54% 184
6-10 years	19.34% 229
10-20 years	33.02% 391
>20 years	25.76% 305
<b>TOTAL</b>	<b>1,184</b>

### Q3 How many years of programming experience do you have overall (all languages)?

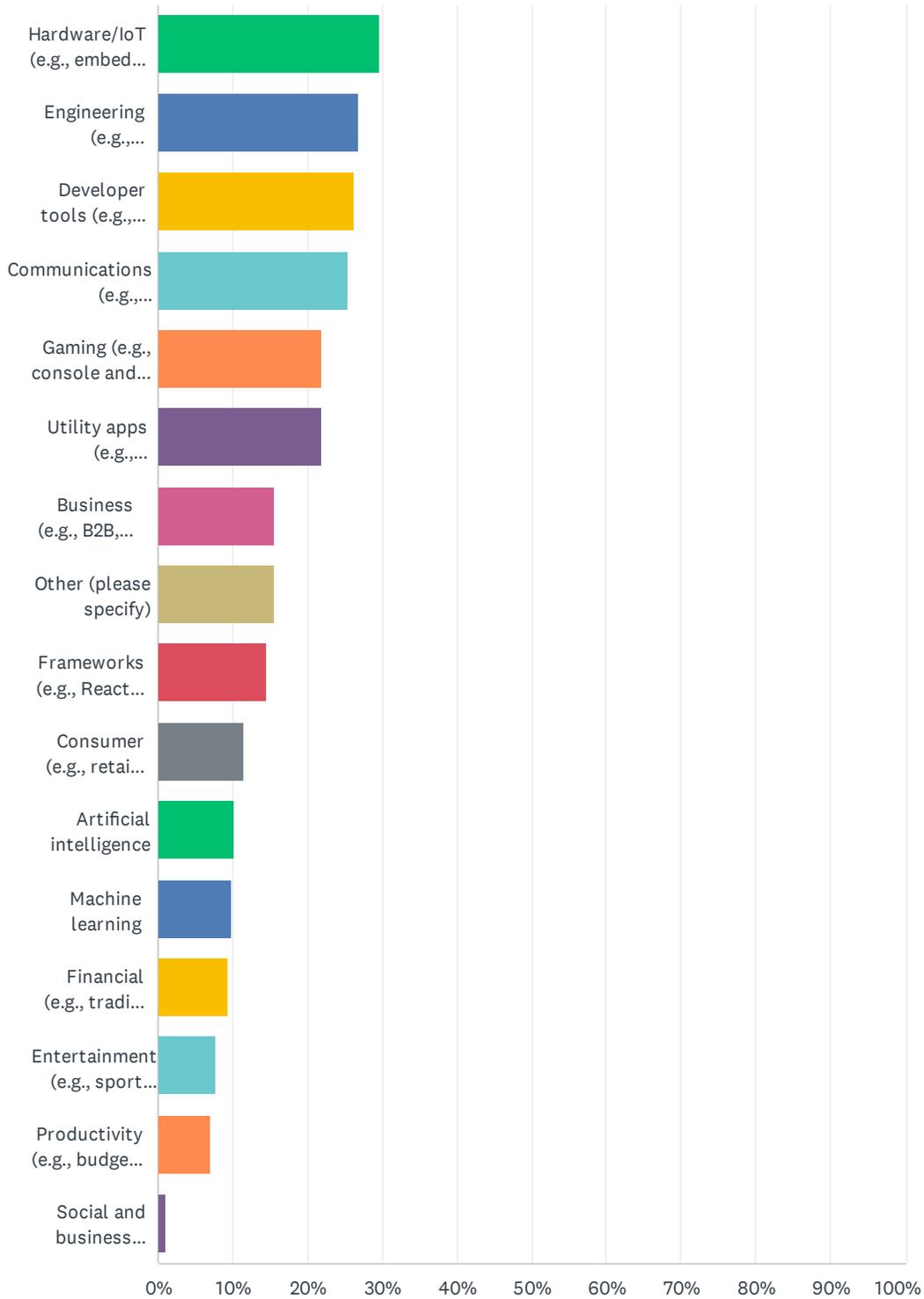
Answered: 1,183 Skipped: 6



ANSWER CHOICES	RESPONSES	
1-2 years	1.01%	12
3-5 years	9.13%	108
6-10 years	17.92%	212
10-20 years	33.05%	391
>20 years	38.88%	460
<b>TOTAL</b>		<b>1,183</b>

### Q4 What types of projects do you work on? (select all that apply)

Answered: 1,186 Skipped: 3

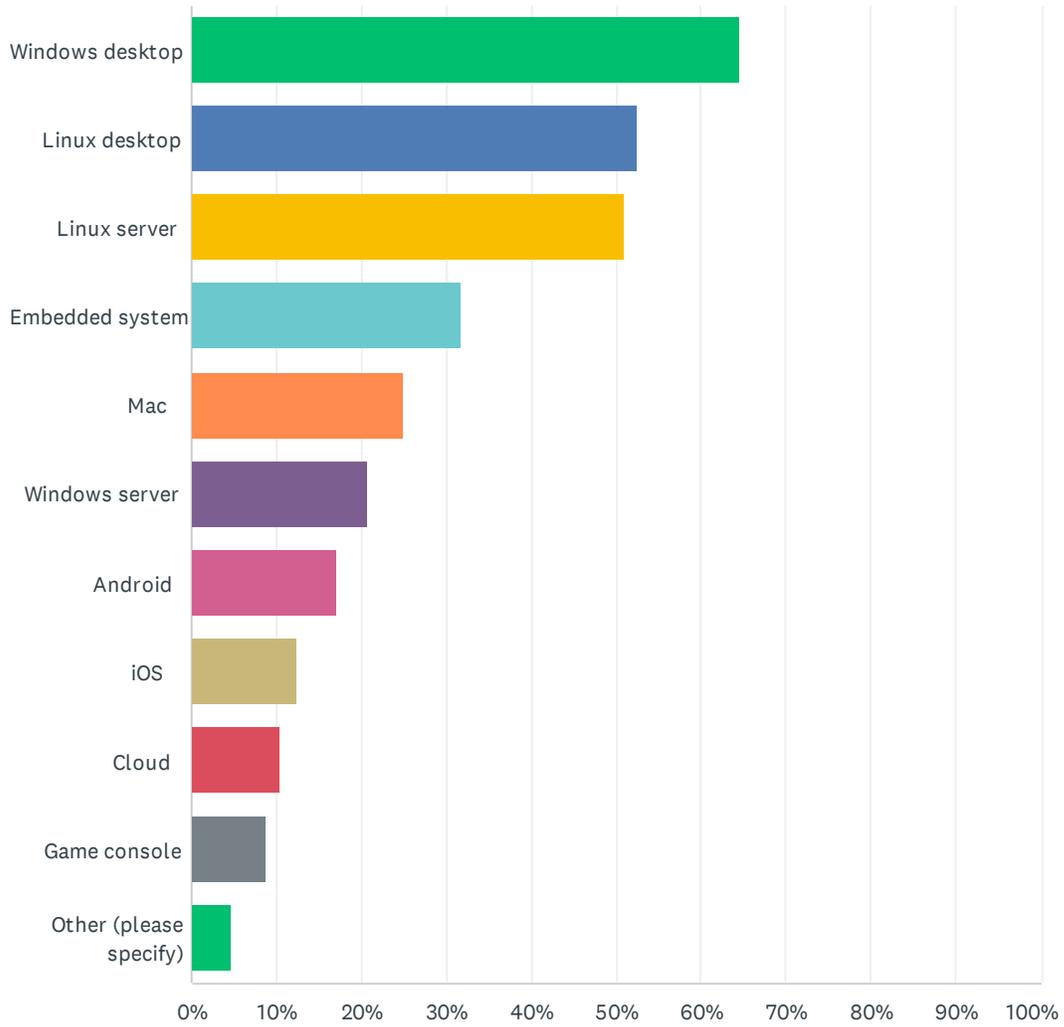


## 2022 Annual C++ Developer Survey "Lite"

ANSWER CHOICES	RESPONSES	
Hardware/IoT (e.g., embedded systems, home automation)	29.68%	352
Engineering (e.g., avionics, power management)	26.90%	319
Developer tools (e.g., compilers, code editors)	26.14%	310
Communications (e.g., networking, email)	25.38%	301
Gaming (e.g., console and mobile games)	21.92%	260
Utility apps (e.g., calculators, simple image editors)	21.92%	260
Business (e.g., B2B, B2E)	15.60%	185
Other (please specify)	15.60%	185
Frameworks (e.g., React, Unity)	14.42%	171
Consumer (e.g., retail websites, mobile apps)	11.47%	136
Artificial intelligence	10.20%	121
Machine learning	9.78%	116
Financial (e.g., trading, mortgage, asset management)	9.36%	111
Entertainment (e.g., sports apps, video streaming)	7.59%	90
Productivity (e.g., budget tracking, note taking)	7.00%	83
Social and business networking (e.g., Facebook, Twitter)	1.01%	12
<b>Total Respondents: 1,186</b>		

### Q5 What platforms do you develop for? (select all that apply)

Answered: 1,183 Skipped: 6

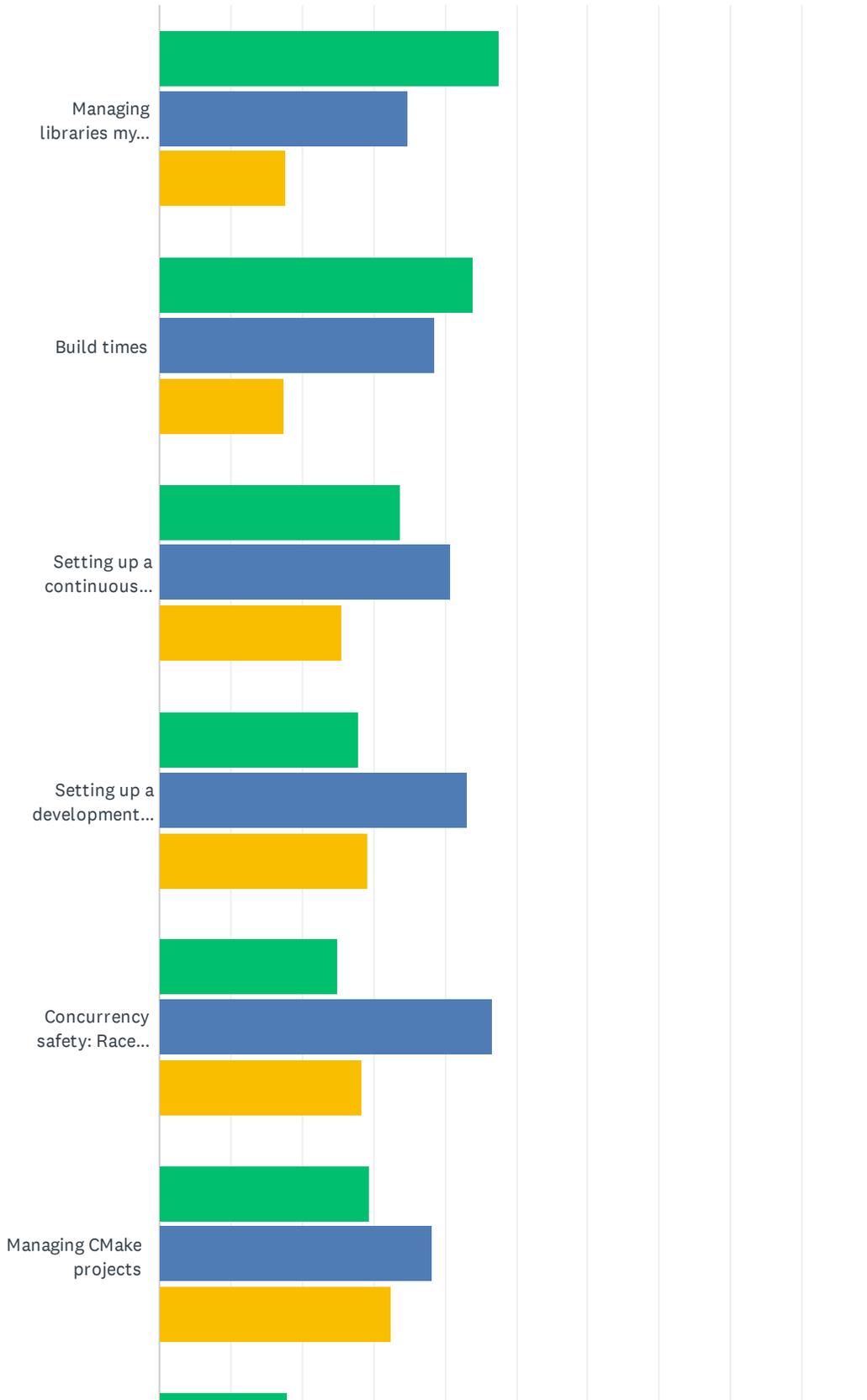


## 2022 Annual C++ Developer Survey "Lite"

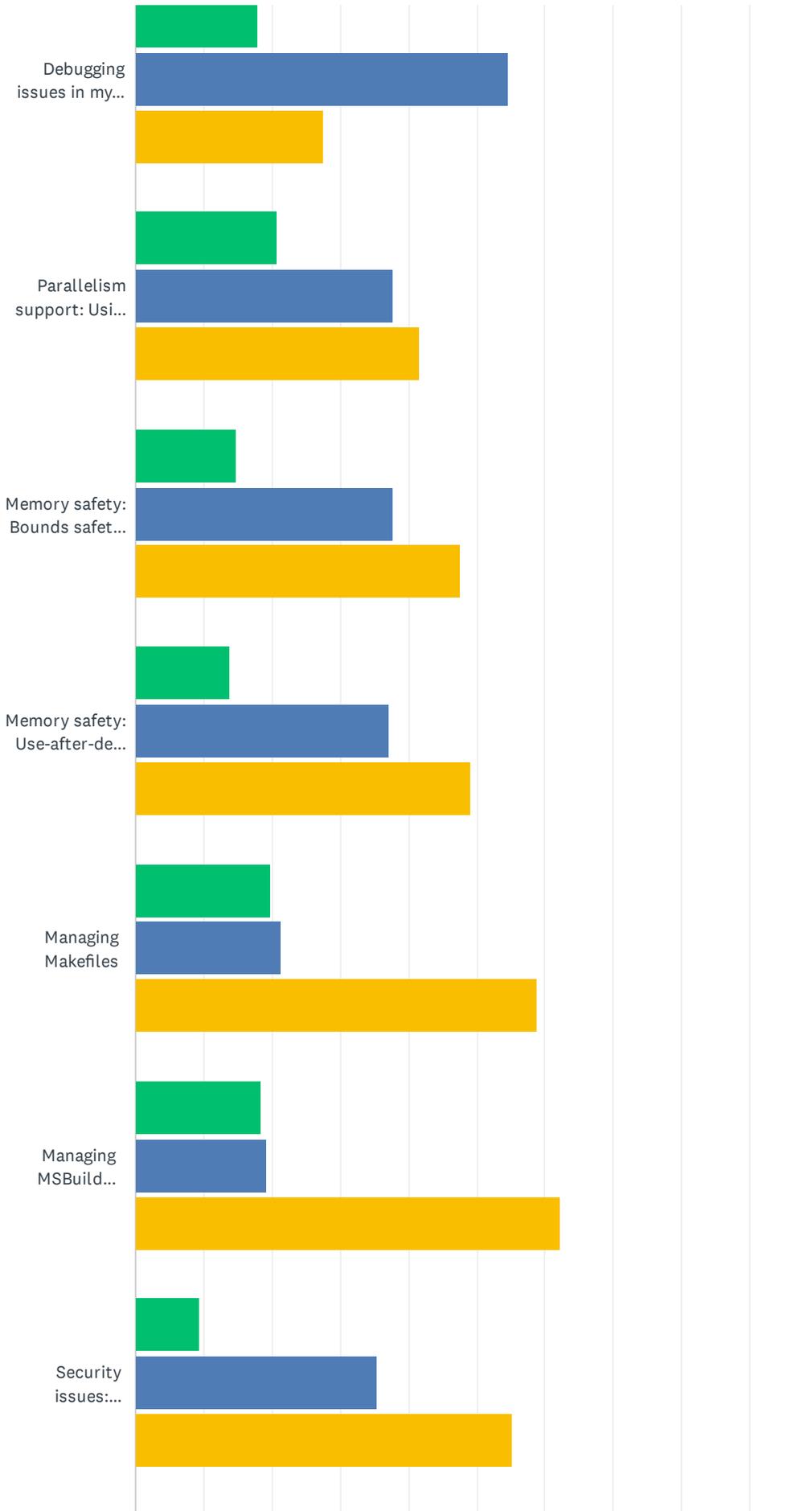
ANSWER CHOICES	RESPONSES	
Windows desktop	64.67%	765
Linux desktop	52.41%	620
Linux server	51.06%	604
Embedded system	31.87%	377
Mac	24.85%	294
Windows server	20.71%	245
Android	16.99%	201
iOS	12.34%	146
Cloud	10.40%	123
Game console	8.79%	104
Other (please specify)	4.73%	56
Total Respondents: 1,183		

# Q6 Which of these do you find frustrating about C++ development?

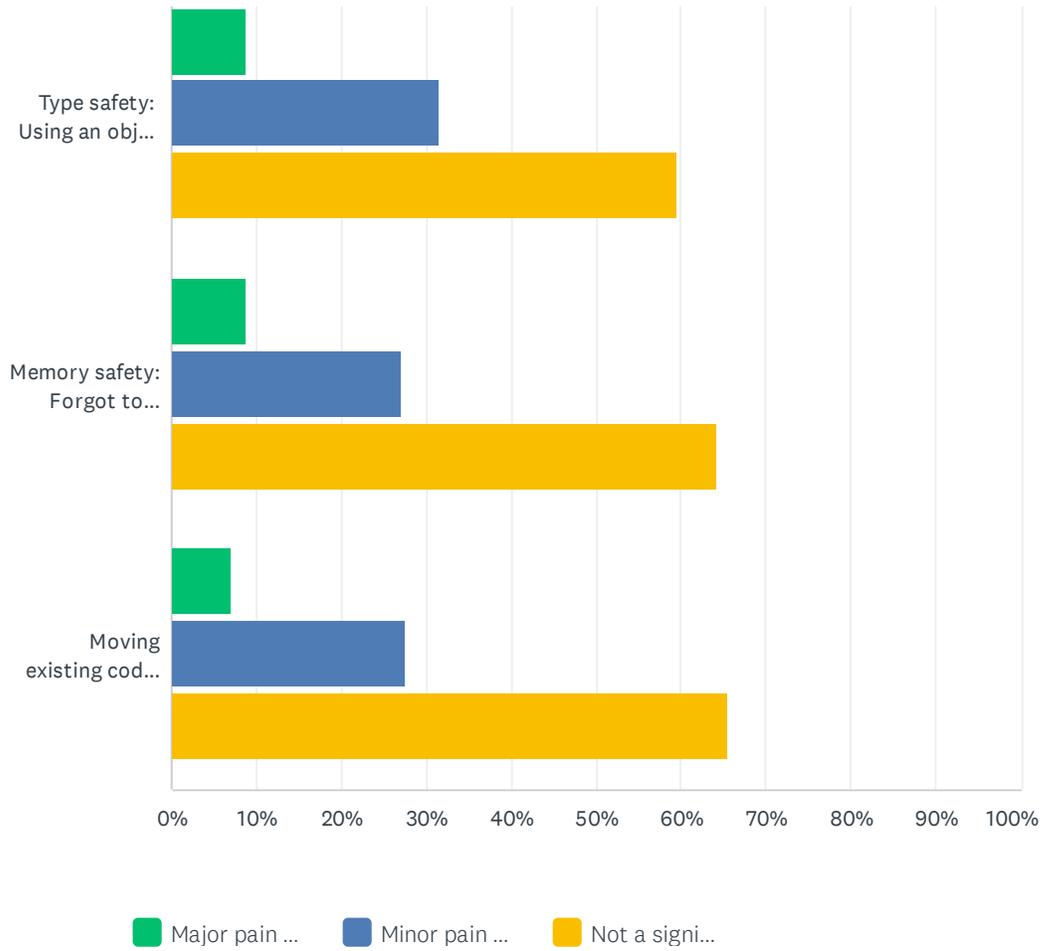
Answered: 1,185 Skipped: 4



# 2022 Annual C++ Developer Survey "Lite"



# 2022 Annual C++ Developer Survey "Lite"

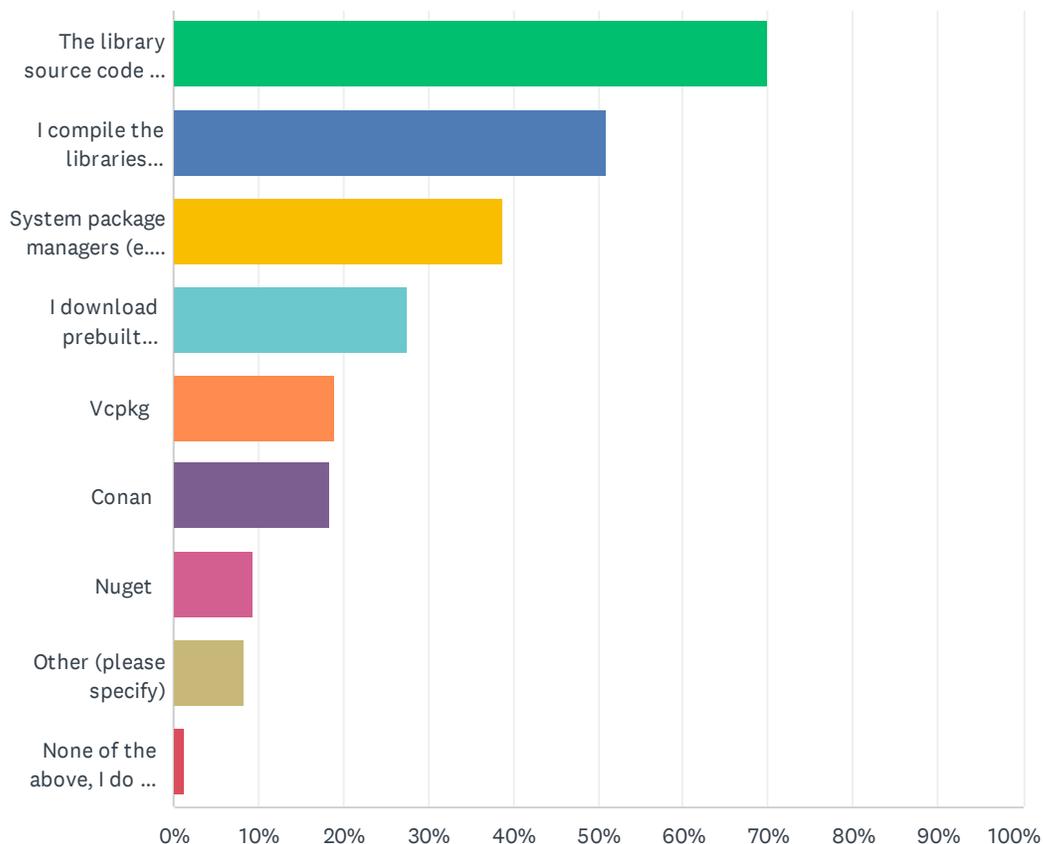


## 2022 Annual C++ Developer Survey "Lite"

	MAJOR PAIN POINT	MINOR PAIN POINT	NOT A SIGNIFICANT ISSUE FOR ME	TOTAL	WEIGHTED AVERAGE
Managing libraries my application depends on	47.63% 563	34.77% 411	17.60% 208	1,182	2.30
Build times	43.94% 515	38.65% 453	17.41% 204	1,172	2.27
Setting up a continuous integration pipeline from scratch (automated builds, tests, ...)	33.73% 394	40.75% 476	25.51% 298	1,168	2.08
Setting up a development environment from scratch (compiler, build system, IDE, ...)	27.83% 329	42.98% 508	29.19% 345	1,182	1.99
Concurrency safety: Races, deadlocks, performance bottlenecks	25.04% 293	46.67% 546	28.29% 331	1,170	1.97
Managing CMake projects	29.34% 343	38.15% 446	32.51% 380	1,169	1.97
Debugging issues in my code	17.85% 209	54.57% 639	27.58% 323	1,171	1.90
Parallelism support: Using more CPU/GPU/other cores to compute an answer faster	20.74% 242	37.79% 441	41.47% 484	1,167	1.79
Memory safety: Bounds safety issues (read/write beyond the bounds of an object or array)	14.81% 174	37.70% 443	47.49% 558	1,175	1.67
Memory safety: Use-after-delete/free (dangling pointers, iterators, spans, ...)	13.93% 163	37.01% 433	49.06% 574	1,170	1.65
Managing Makefiles	19.88% 226	21.37% 243	58.75% 668	1,137	1.61
Managing MSBuild projects	18.41% 209	19.30% 219	62.29% 707	1,135	1.56
Security issues: Overlaps with "safety" but includes other issues (secret disclosure, vulnerabilities, exploits, ...)	9.36% 109	35.40% 412	55.24% 643	1,164	1.54
Type safety: Using an object as the wrong type (unsafe downcasts, unsafe unions, ...)	8.77% 103	31.66% 372	59.57% 700	1,175	1.49
Memory safety: Forgot to delete/free (memory leaks)	8.68% 102	27.06% 318	64.26% 755	1,175	1.44
Moving existing code to the latest language standard	7.03% 83	27.54% 325	65.42% 772	1,180	1.42

## Q7 How do you manage your C++ 1st and 3rd party libraries? (Check all that apply)

Answered: 1,183 Skipped: 6

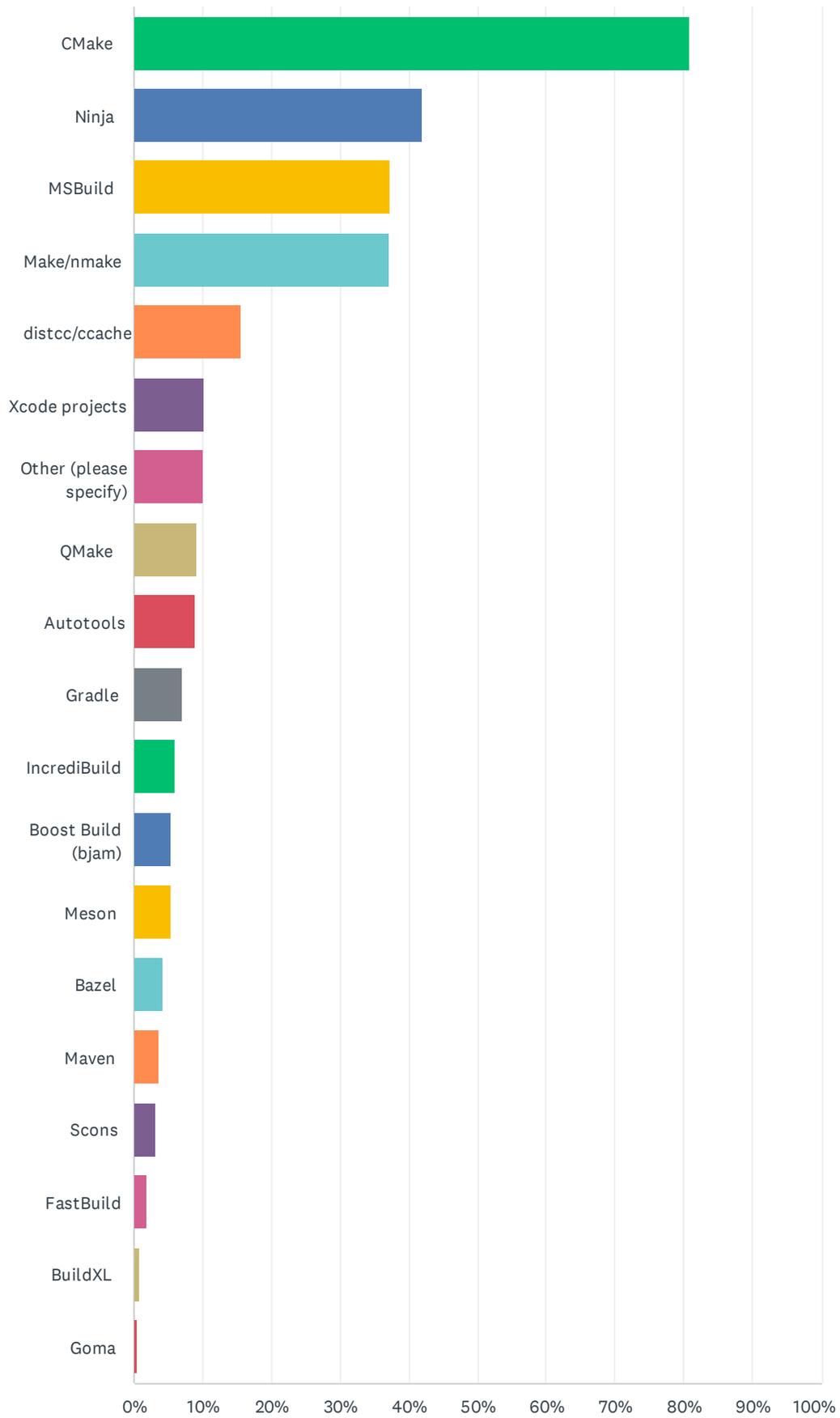


ANSWER CHOICES	RESPONSES	
The library source code is part of my build	69.91%	827
I compile the libraries separately using their instructions	50.89%	602
System package managers (e.g., apt, brew, ...)	38.80%	459
I download prebuilt libraries from the Internet	27.56%	326
Vcpkg	18.93%	224
Conan	18.34%	217
Nuget	9.30%	110
Other (please specify)	8.37%	99
None of the above, I do not have any dependencies	1.35%	16
Total Respondents: 1,183		

## Q8 What build tools do you use? (Check all that apply)

Answered: 1,176 Skipped: 13

# 2022 Annual C++ Developer Survey "Lite"

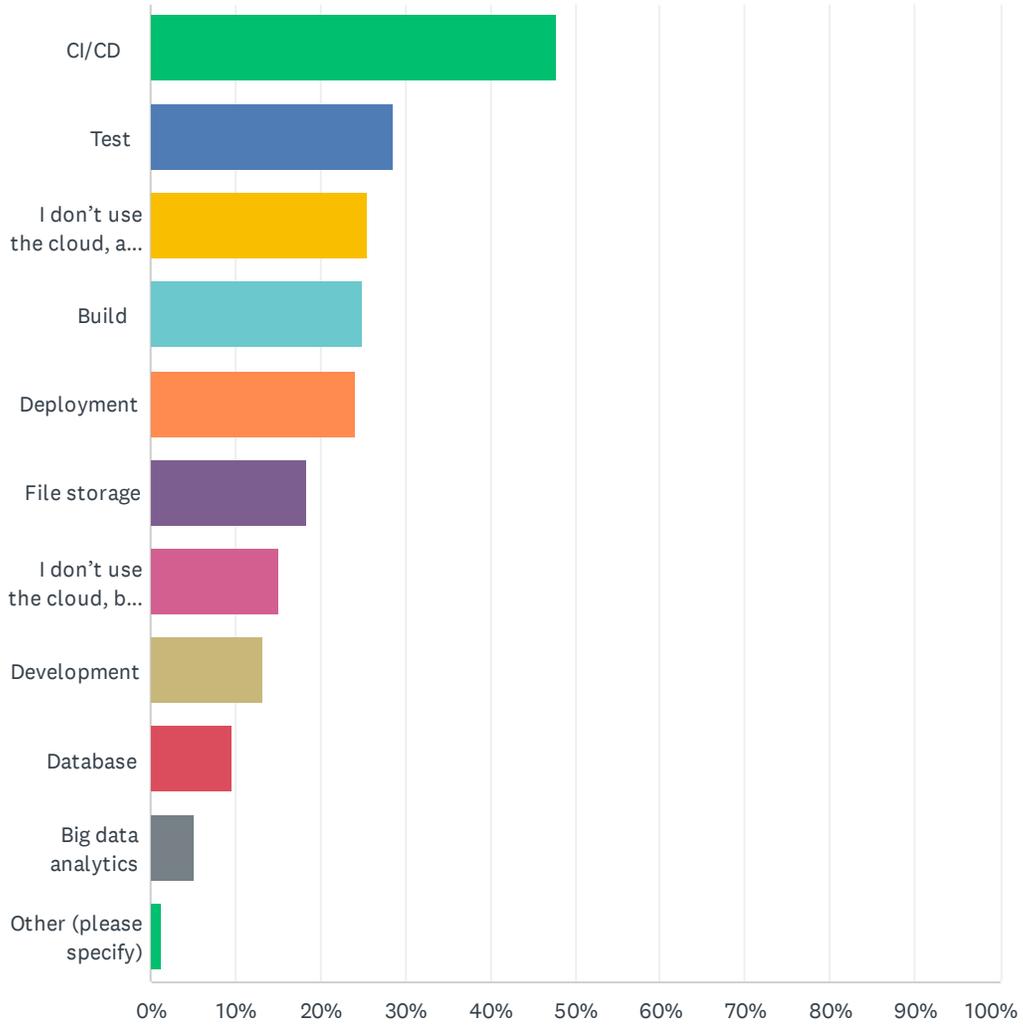


## 2022 Annual C++ Developer Survey "Lite"

ANSWER CHOICES	RESPONSES	
CMake	80.87%	951
Ninja	41.92%	493
MSBuild	37.41%	440
Make/nmake	37.07%	436
distcc/ccache	15.65%	184
Xcode projects	10.20%	120
Other (please specify)	10.03%	118
QMake	9.27%	109
Autotools	8.93%	105
Gradle	6.97%	82
IncrediBuild	6.04%	71
Boost Build (bjam)	5.27%	62
Meson	5.27%	62
Bazel	4.17%	49
Maven	3.57%	42
Scons	3.23%	38
FastBuild	1.87%	22
BuildXL	0.77%	9
Goma	0.51%	6
Total Respondents: 1,176		

# Q9 What parts of your development lifecycle use the cloud? (Check all that apply)

Answered: 1,167 Skipped: 22

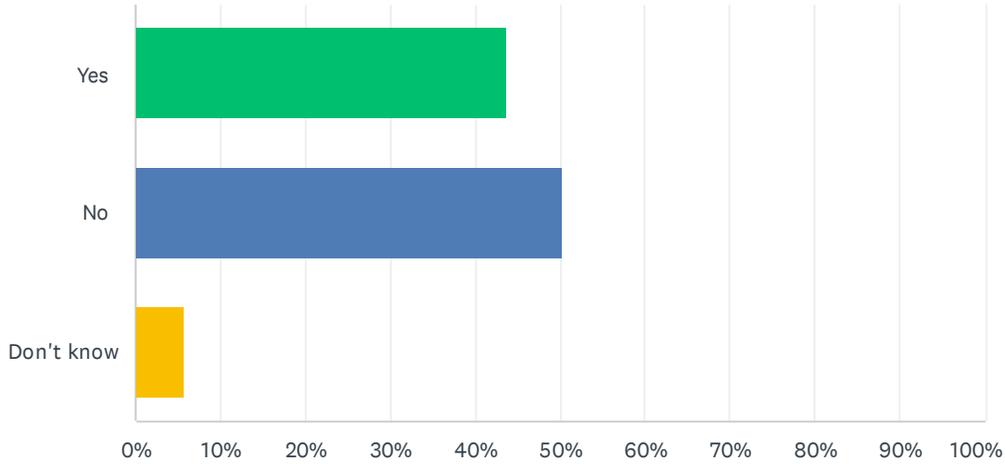


## 2022 Annual C++ Developer Survey "Lite"

ANSWER CHOICES	RESPONSES	
CI/CD	47.73%	557
Test	28.62%	334
I don't use the cloud, and I am not interested	25.54%	298
Build	24.94%	291
Deployment	24.08%	281
File storage	18.42%	215
I don't use the cloud, but I am interested	15.17%	177
Development	13.28%	155
Database	9.60%	112
Big data analytics	5.06%	59
Other (please specify)	1.37%	16
Total Respondents: 1,167		

## Q10 Does your current project use sanitizers and/or fuzzing as part of its normal development and release cycle?

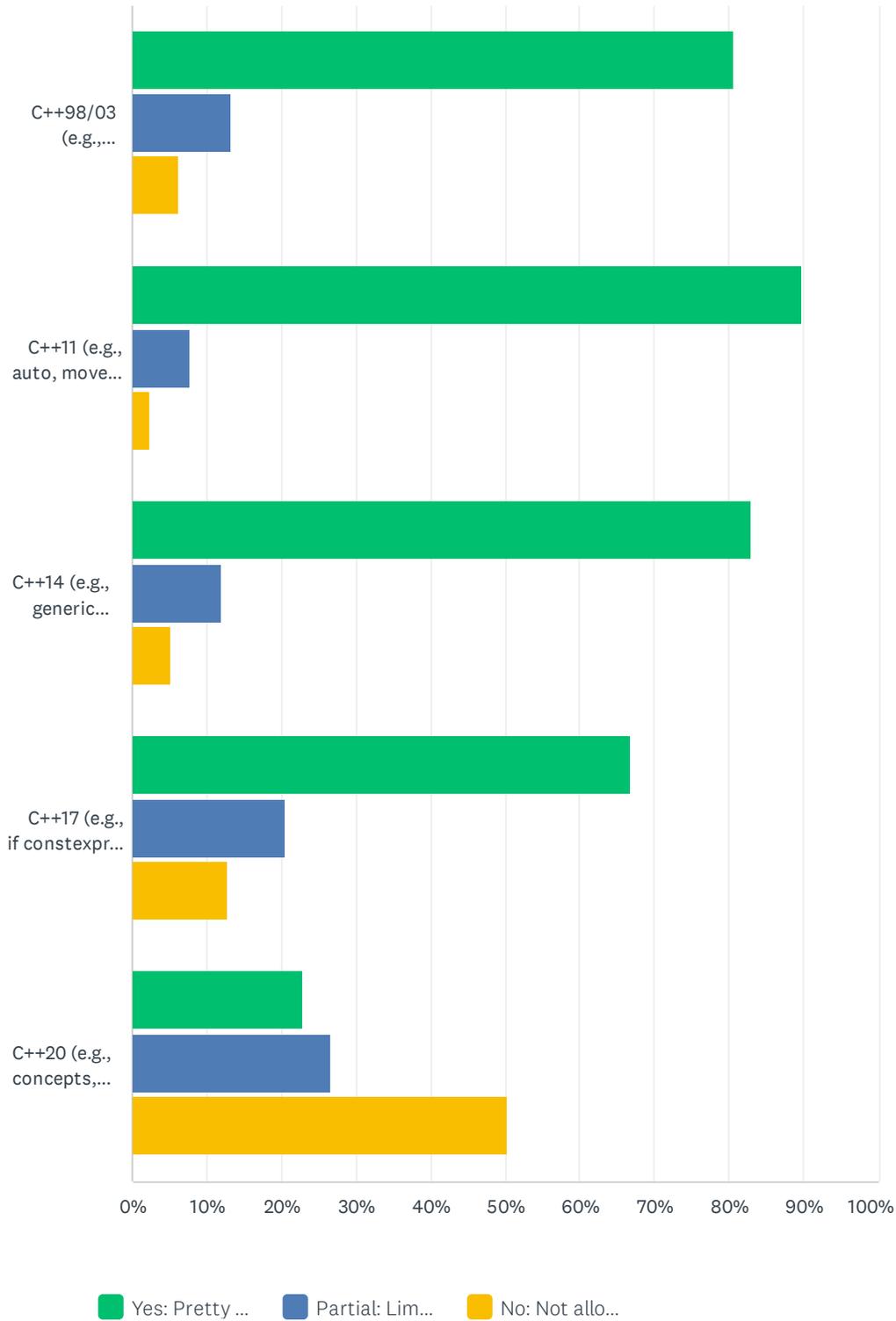
Answered: 1,176 Skipped: 13



ANSWER CHOICES	RESPONSES	
Yes	43.79%	515
No	50.43%	593
Don't know	5.78%	68
<b>TOTAL</b>		<b>1,176</b>

# Q11 What version(s) of C++ are you allowed to use on your current project (work or school)?

Answered: 1,178 Skipped: 11

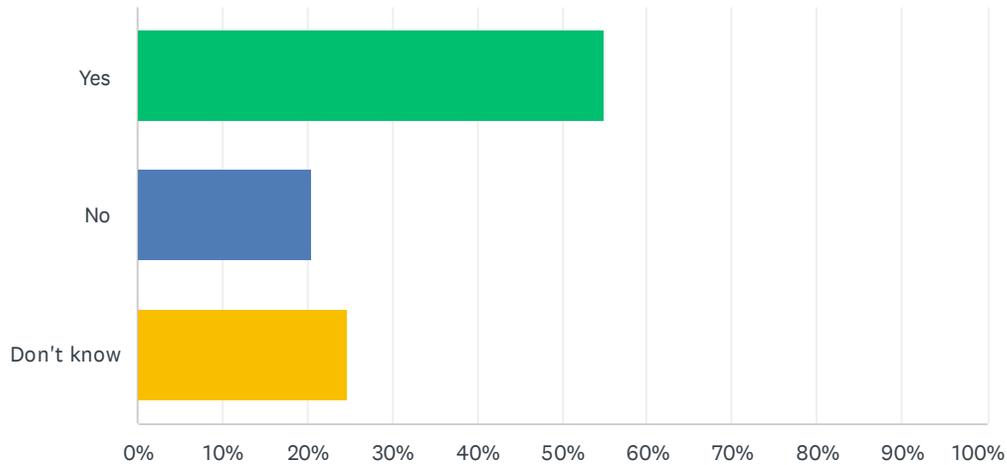


2022 Annual C++ Developer Survey "Lite"

	YES: PRETTY MUCH ALL	PARTIAL: LIMITED FEATURES/USAGE	NO: NOT ALLOWED	TOTAL	WEIGHTED AVERAGE
C++98/03 (e.g., exceptions, templates, RTTI)	80.62% 890	13.13% 145	6.25% 69	1,104	2.74
C++11 (e.g., auto, move semantics, =delete/=default, shared_ptr, lambdas)	89.86% 1,010	7.74% 87	2.40% 27	1,124	2.87
C++14 (e.g., generic lambdas, auto return types, general constexpr functions)	82.89% 940	11.99% 136	5.11% 58	1,134	2.78
C++17 (e.g., if constexpr, if/switch scoped variables, structured bindings, string_view, optional/any/variant, Parallel STL)	66.81% 775	20.43% 237	12.76% 148	1,160	2.54
C++20 (e.g., concepts, coroutines, modules)	22.85% 258	26.75% 302	50.40% 569	1,129	1.72

### Q12 In the next 12 months, does your current project plan to start allowing additional use of newer C++ standard features (i.e., more than in the previous answer)?

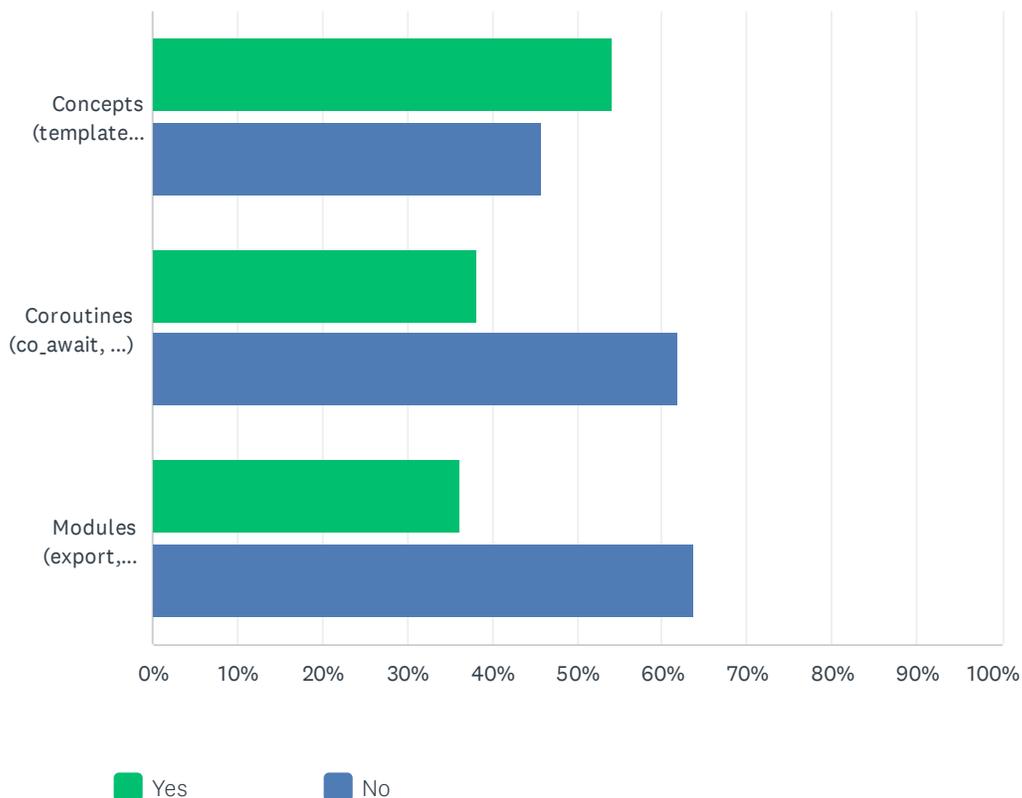
Answered: 1,176 Skipped: 13



ANSWER CHOICES	RESPONSES	
Yes	54.93%	646
No	20.41%	240
Don't know	24.66%	290
<b>TOTAL</b>		<b>1,176</b>

## Q13 Specifically for upcoming major C++20 features: In the next 12 months, does your current project plan to allow use of these C++20 features?

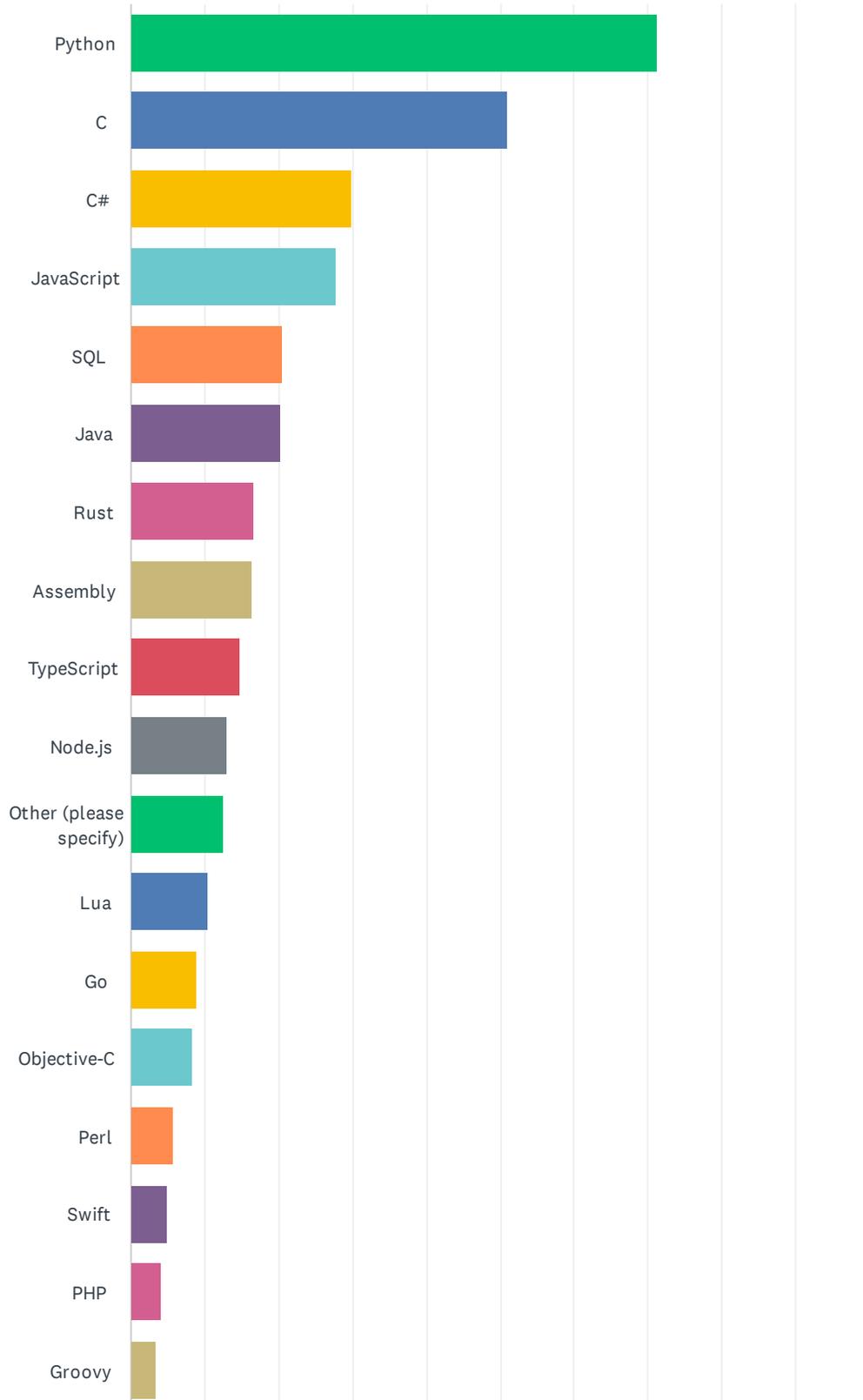
Answered: 1,141 Skipped: 48



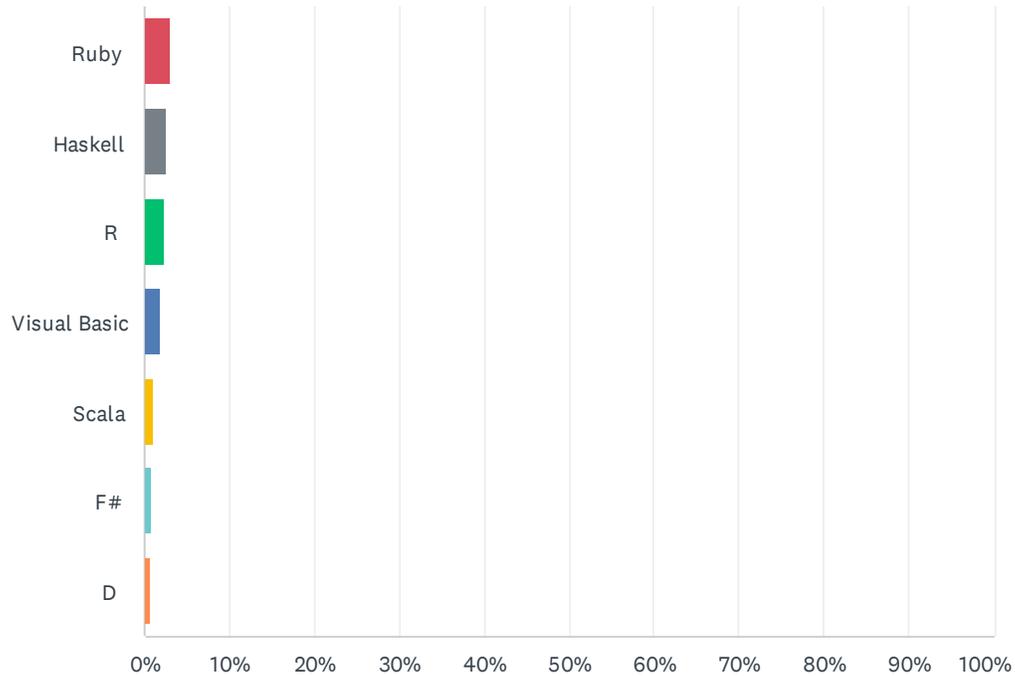
	YES	NO	TOTAL	WEIGHTED AVERAGE
Concepts (template constraints, requires, ...)	54.26% 612	45.74% 516	1,128	1.91
Coroutines (co_await, ...)	38.24% 429	61.76% 693	1,122	2.24
Modules (export, import, ...)	36.24% 411	63.76% 723	1,134	2.28

# Q14 Besides C++, what programming languages/environments do you use in your current and recent projects? (select all that apply)

Answered: 1,167 Skipped: 22



# 2022 Annual C++ Developer Survey "Lite"

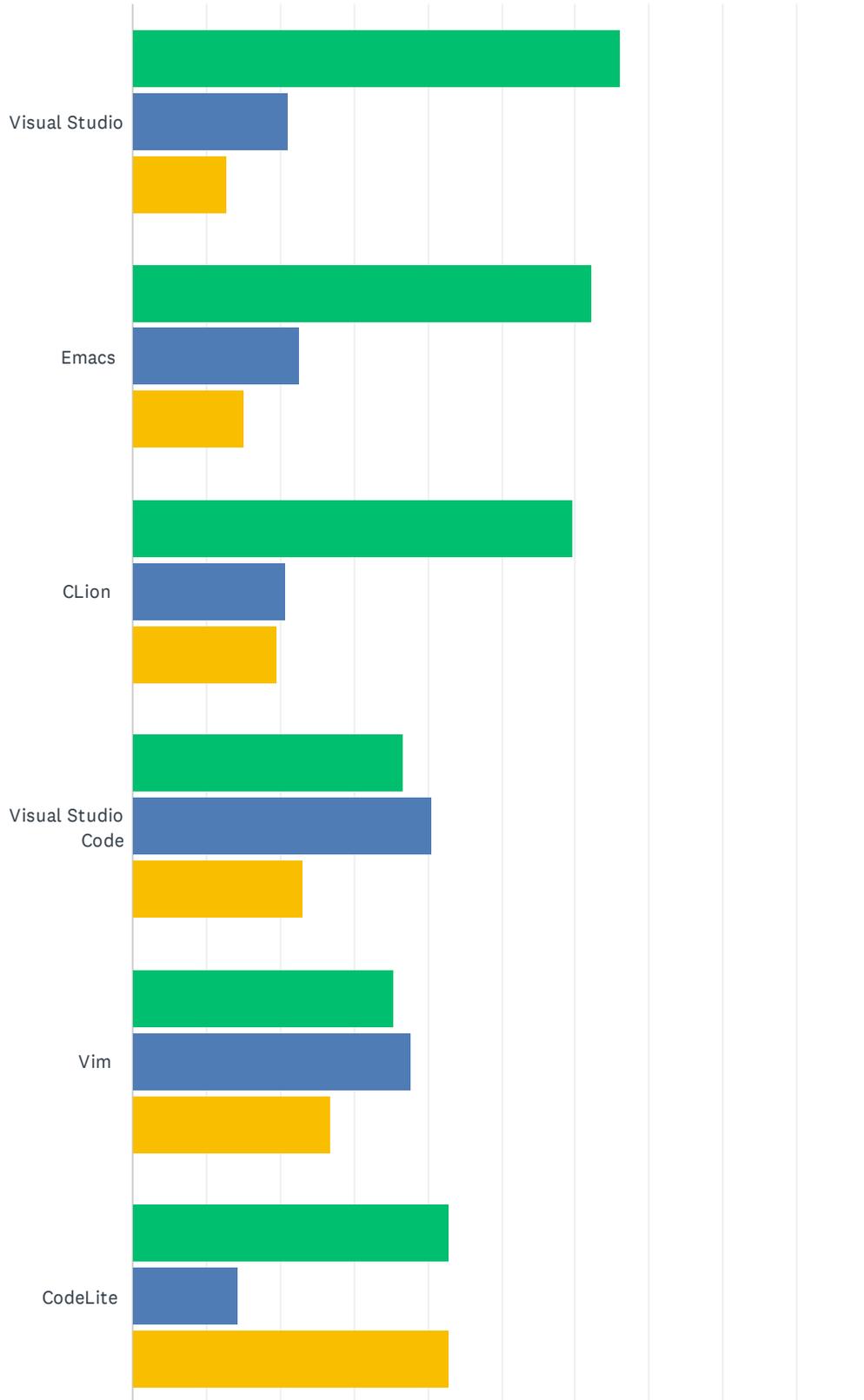


## 2022 Annual C++ Developer Survey "Lite"

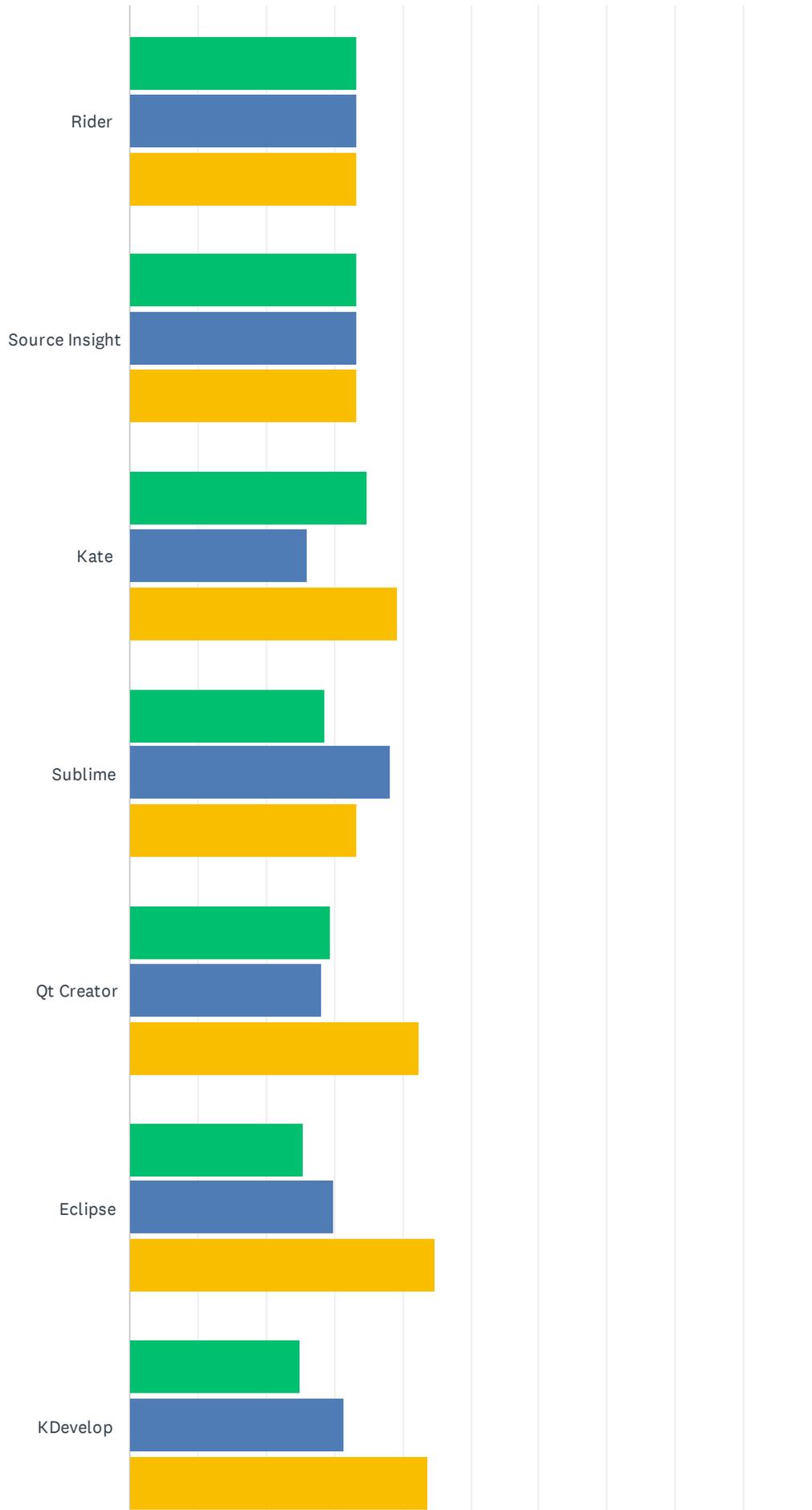
ANSWER CHOICES	RESPONSES	
Python	71.12%	830
C	50.90%	594
C#	29.82%	348
JavaScript	27.68%	323
SQL	20.39%	238
Java	20.22%	236
Rust	16.71%	195
Assembly	16.45%	192
TypeScript	14.74%	172
Node.js	12.94%	151
Other (please specify)	12.68%	148
Lua	10.54%	123
Go	8.91%	104
Objective-C	8.40%	98
Perl	5.74%	67
Swift	4.97%	58
PHP	4.11%	48
Groovy	3.34%	39
Ruby	3.00%	35
Haskell	2.49%	29
R	2.40%	28
Visual Basic	1.97%	23
Scala	1.11%	13
F#	0.94%	11
D	0.69%	8
Total Respondents: 1,167		

# Q15 Which development environments (IDEs) or editors do you use for C++ development?

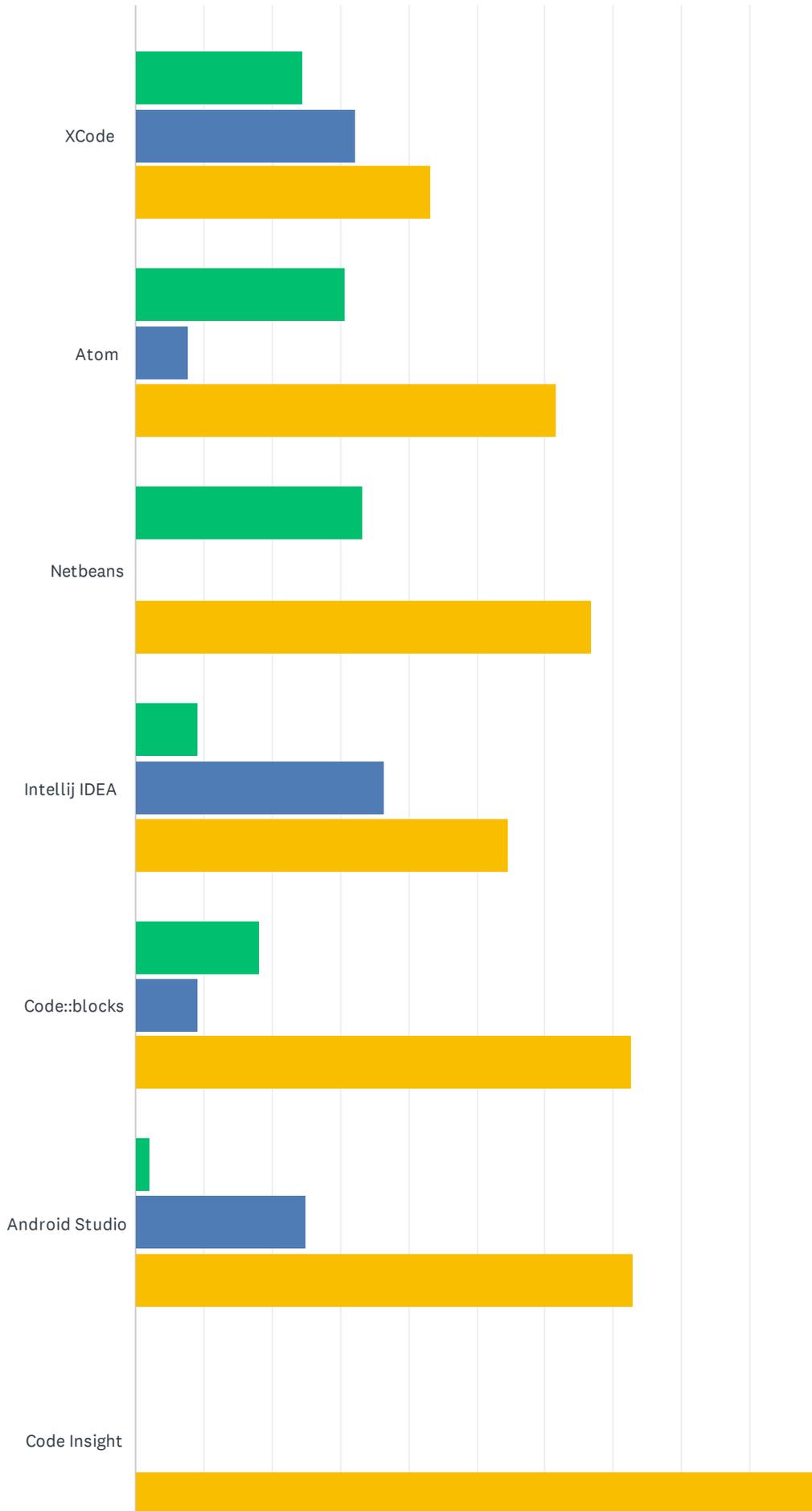
Answered: 1,177 Skipped: 12



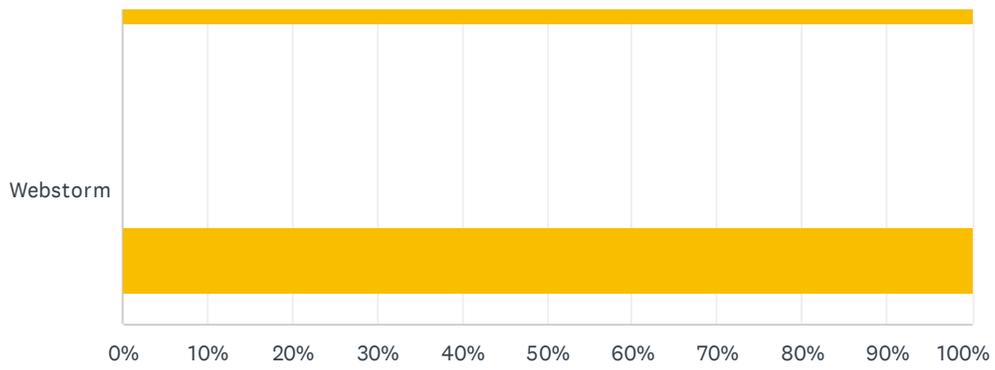
# 2022 Annual C++ Developer Survey "Lite"



# 2022 Annual C++ Developer Survey "Lite"



# 2022 Annual C++ Developer Survey "Lite"



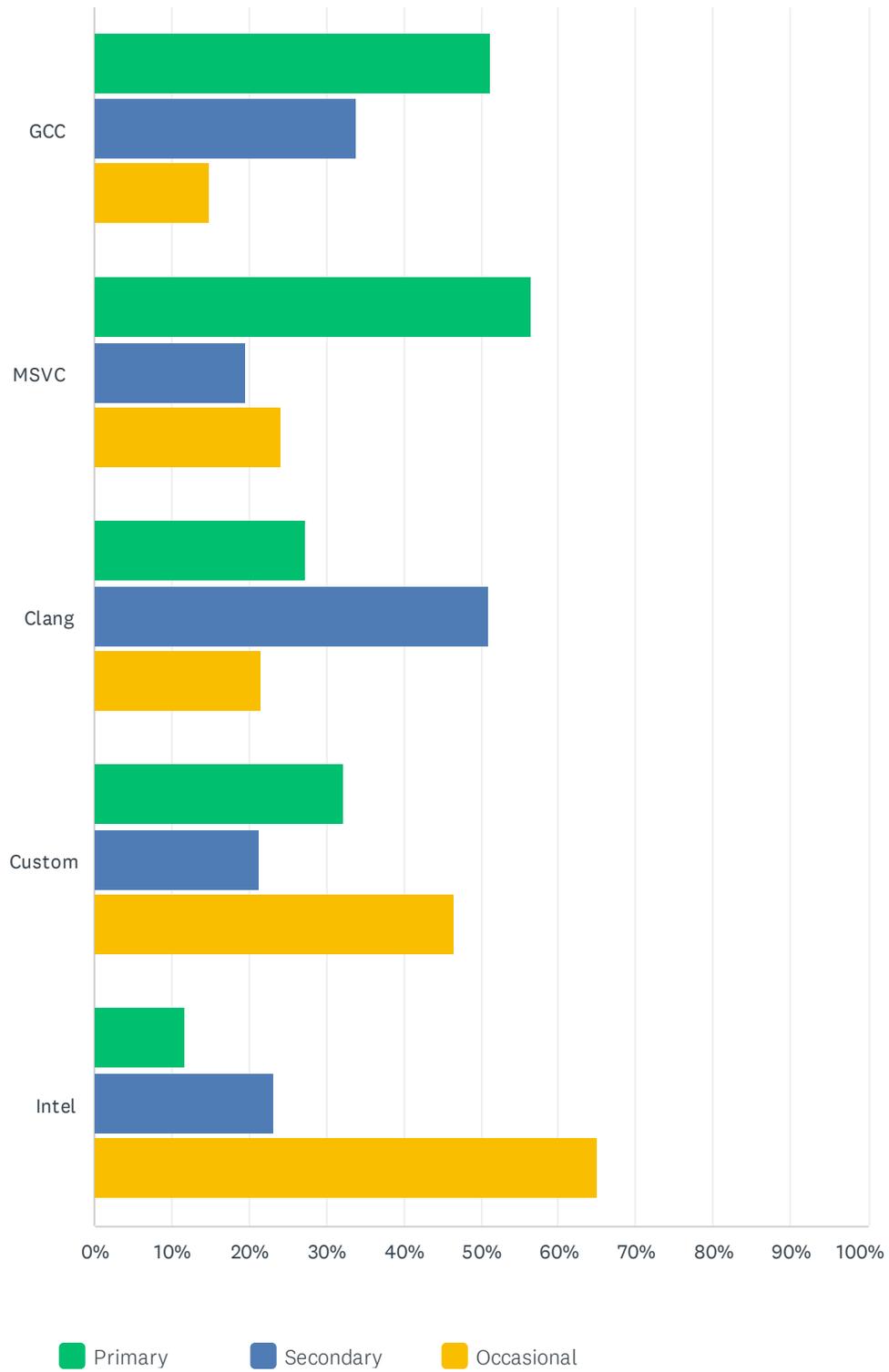
■ Primary    ■ Secondary    ■ Occasional

2022 Annual C++ Developer Survey "Lite"

	PRIMARY	SECONDARY	OCCASIONAL	TOTAL	WEIGHTED AVERAGE
Visual Studio	66.02% 410	21.10% 131	12.88% 80	621	2.53
Emacs	62.26% 66	22.64% 24	15.09% 16	106	2.47
CLion	59.62% 124	20.67% 43	19.71% 41	208	2.40
Visual Studio Code	36.59% 266	40.44% 294	22.97% 167	727	2.14
Vim	35.37% 139	37.66% 148	26.97% 106	393	2.08
CodeLite	42.86% 3	14.29% 1	42.86% 3	7	2.00
Rider	33.33% 4	33.33% 4	33.33% 4	12	2.00
Source Insight	33.33% 1	33.33% 1	33.33% 1	3	2.00
Kate	34.78% 8	26.09% 6	39.13% 9	23	1.96
Sublime	28.57% 18	38.10% 24	33.33% 21	63	1.95
Qt Creator	29.52% 62	28.10% 59	42.38% 89	210	1.87
Eclipse	25.37% 17	29.85% 20	44.78% 30	67	1.81
KDevelop	25.00% 4	31.25% 5	43.75% 7	16	1.81
XCode	24.58% 29	32.20% 38	43.22% 51	118	1.81
Atom	30.77% 4	7.69% 1	61.54% 8	13	1.69
Netbeans	33.33% 1	0.00% 0	66.67% 2	3	1.67
Intellij IDEA	9.09% 2	36.36% 8	54.55% 12	22	1.55
Code::blocks	18.18% 2	9.09% 1	72.73% 8	11	1.45
Android Studio	2.08% 1	25.00% 12	72.92% 35	48	1.29
Code Insight	0.00% 0	0.00% 0	100.00% 1	1	1.00
Webstorm	0.00% 0	0.00% 0	100.00% 3	3	1.00

## Q16 Which compilers do you use for C++ development?

Answered: 1,177 Skipped: 12



## 2022 Annual C++ Developer Survey "Lite"

	PRIMARY	SECONDARY	OCCASIONAL	TOTAL	WEIGHTED AVERAGE
GCC	51.17% 481	33.94% 319	14.89% 140	940	2.36
MSVC	56.45% 433	19.56% 150	23.99% 184	767	2.32
Clang	27.35% 244	51.01% 455	21.64% 193	892	2.06
Custom	32.14% 9	21.43% 6	46.43% 13	28	1.86
Intel	11.67% 7	23.33% 14	65.00% 39	60	1.47

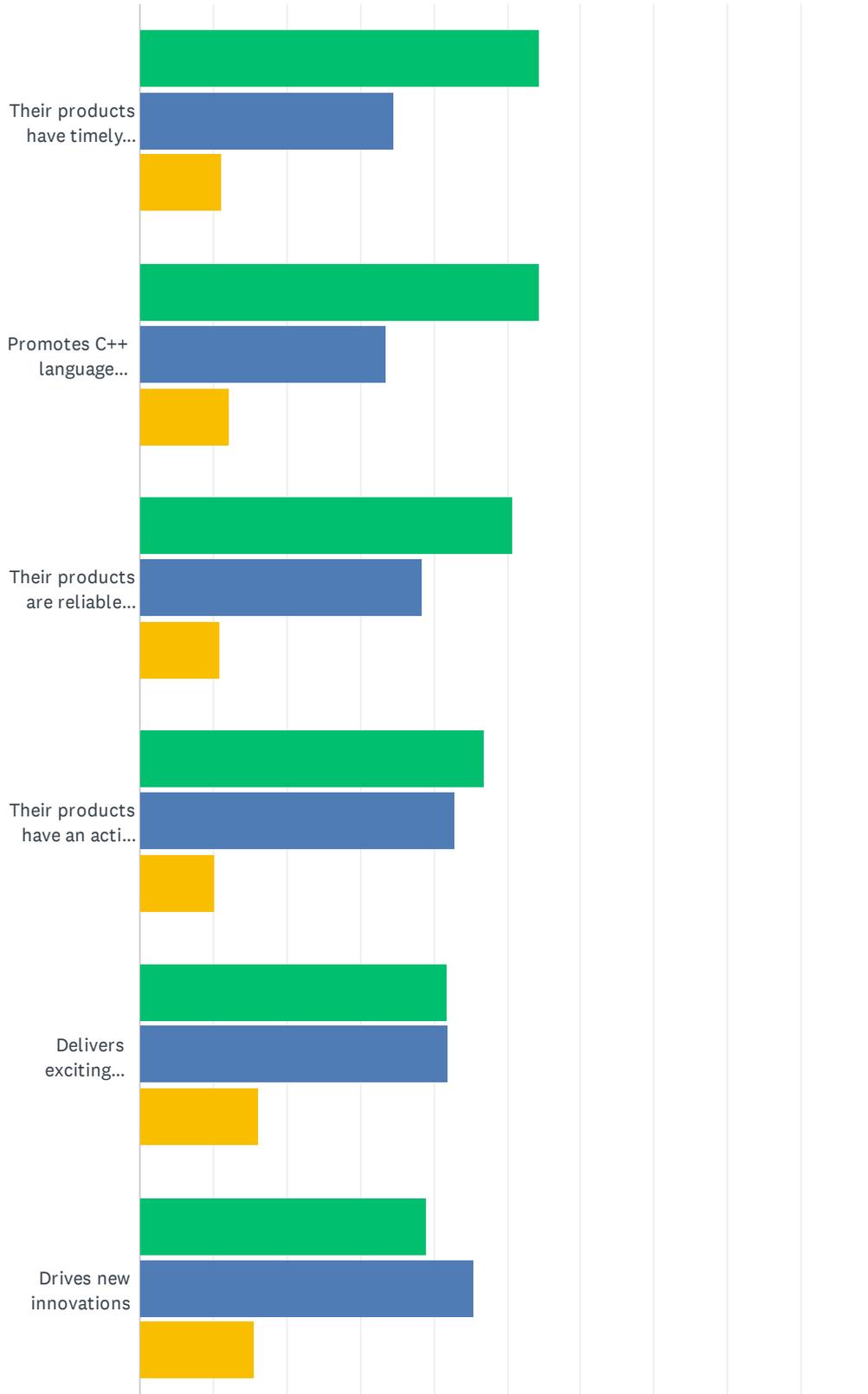
# Q17 What organizations come to your mind the most when you think about C++? Why?

Answered: 639 Skipped: 550

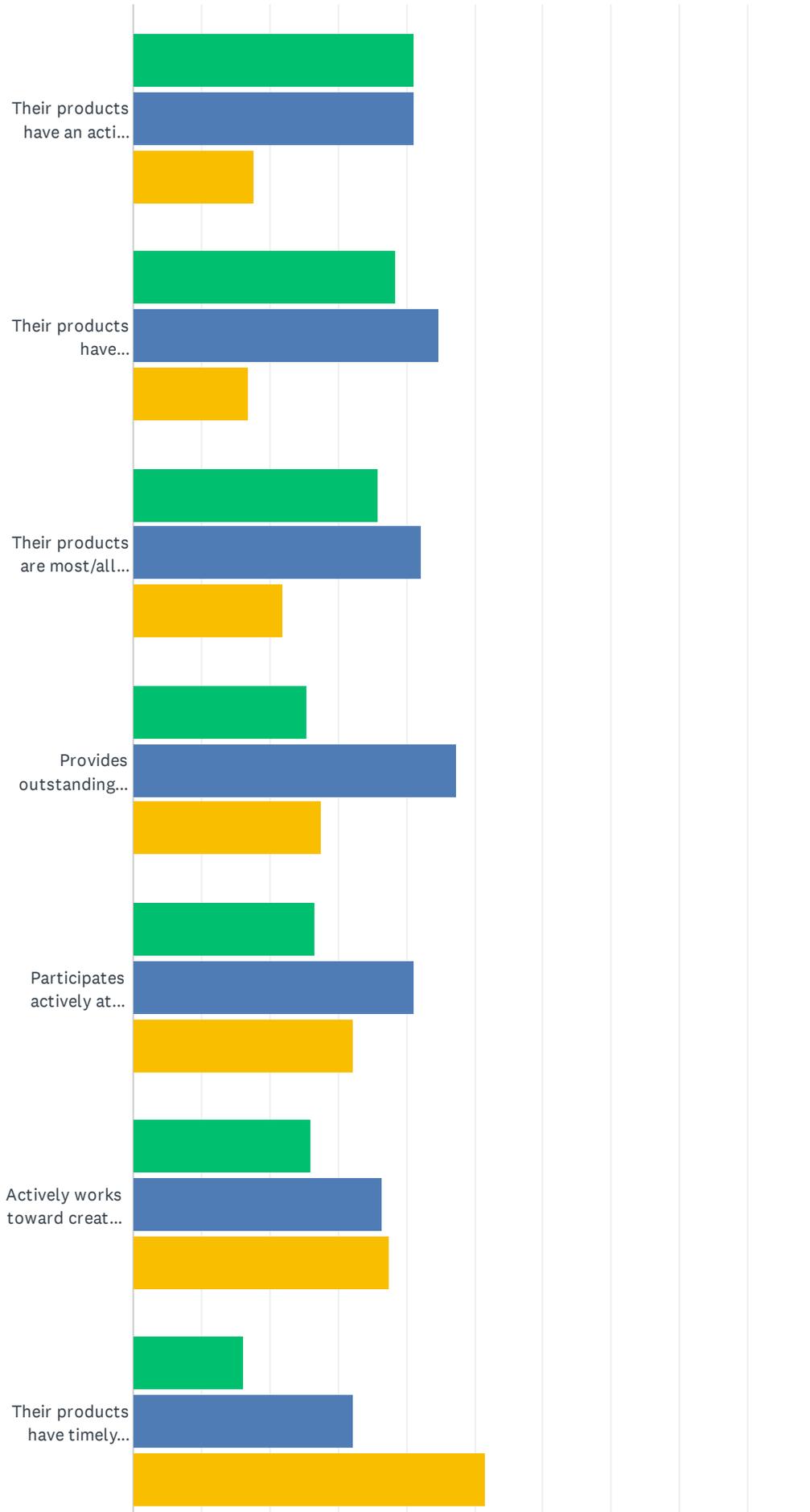
MS people language seem C standard support Apple committee Google Facebook  
Microsoft Google Facebook companies Epic Games organizations Game development  
due Intel Nvidia Intel Apple clang tools main conference codebase None vcpkg  
GCC others Facebook various development projects talks Adobe  
Nvidia Yandex libraries major JetBrains mostly Qt Microsoft Apple  
Bloomberg Kitware isocpp etc LLVM ISO C committee  
Boost C libraries C make Microsoft Google NVidia  
Google think ISO Windows CppCon know use many  
compiler one work isocpp.org standard committee  
LLVM clang Microsoft Google standardization Google Microsoft  
studios ISO C IBM standard msvc Microsoft Visual Studio products  
clang developers C Standard Committee ISO standards WG21 Qt Company  
use C Google Apple best Boost library C Foundation lot C GNU Foundation  
Microsoft MSVC contribute game since community vs ISO committee great CPP  
lot

# Q18 How important is each of these to you when you think about an organization's involvement in C++?

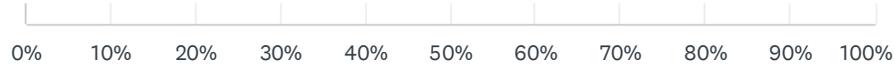
Answered: 1,089 Skipped: 100



# 2022 Annual C++ Developer Survey "Lite"



## 2022 Annual C++ Developer Survey "Lite"

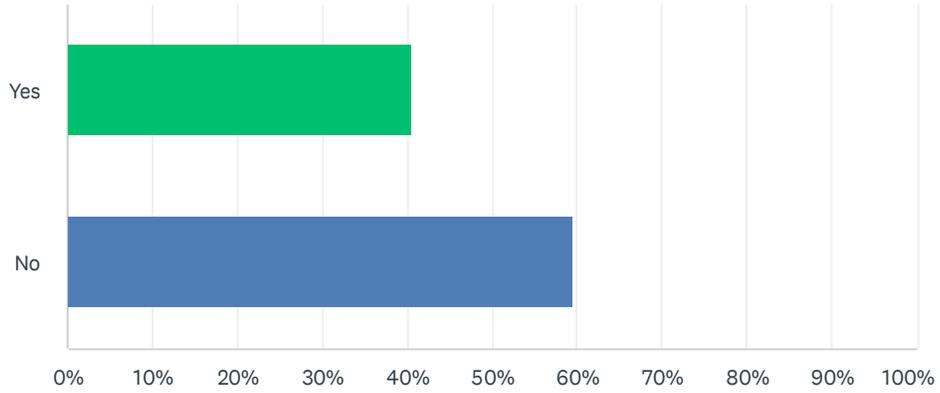


■ Very import...  
 ■ Somewhat i...  
 ■ Not import...

	VERY IMPORTANT	SOMEWHAT IMPORTANT	NOT IMPORTANT	TOTAL	WEIGHTED AVERAGE
Their products have timely support for the latest C++ standard	54.43% 584	34.58% 371	11.00% 118	1,073	2.43
Promotes C++ language evolution in directions that help my work	54.30% 581	33.55% 359	12.15% 130	1,070	2.42
Their products are reliable and backward-compatible	50.76% 537	38.37% 406	10.87% 115	1,058	2.40
Their products have an active user community	46.99% 499	42.84% 455	10.17% 108	1,062	2.37
Delivers exciting solutions to common development problems	41.83% 440	41.92% 441	16.25% 171	1,052	2.26
Drives new innovations	39.03% 411	45.49% 479	15.48% 163	1,053	2.24
Their products have an active open source community	41.17% 438	41.07% 437	17.76% 189	1,064	2.23
Their products have high-quality training materials	38.28% 405	44.80% 474	16.92% 179	1,058	2.21
Their products are most/all open-sourced	35.87% 382	42.25% 450	21.88% 233	1,065	2.14
Provides outstanding customer support and servicing	25.33% 266	47.24% 496	27.43% 288	1,050	1.98
Participates actively at conferences (e.g., talks, booths)	26.60% 283	41.17% 438	32.24% 343	1,064	1.94
Actively works toward creating a more diverse and inclusive C++ community	26.01% 277	36.43% 388	37.56% 400	1,065	1.88
Their products have timely support for the latest C standard	16.21% 172	32.14% 341	51.65% 548	1,061	1.65

### Q19 Is an organization's involvement in the C++ community a major factor in choosing their developer tools and services?

Answered: 1,098 Skipped: 91



ANSWER CHOICES	RESPONSES	
Yes	40.44%	444
No	59.56%	654
TOTAL		1,098

Q20 If you could wave a magic wand and change one thing about any part of C++, what would it be, and how would that change help your daily work?

Answered: 778 Skipped: 411

template object write force new introduce even reason also think something  
 large issues means reflection simple many still Break ABI include  
 builds go Rust take change dependencies modules epochs one  
 years project long compiler help things syntax support lot  
 Remove hard features example library exceptions  
 code developers use etc make default C want  
 language class better e.g work version  
 standard dependency management way maybe add  
 packages allow rules need system standard library solution  
 time errors types STL improve std build system worse  
 package manager simplify fix every will understand easy users tools  
 reduce less know without able build times functions ABI C standard  
 compile times Python much really