



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Zaawansowane programowanie w języku C++ Wstęp

Prezentacja jest współfinansowana przez
Unię Europejską w ramach
Europejskiego Funduszu Społecznego w projekcie pt.

*„Innowacyjna dydaktyka bez ograniczeń - zintegrowany rozwój Politechniki Łódzkiej -
zarządzanie Uczelnią, nowoczesna oferta edukacyjna i wzmocnienia zdolności do
zatrudniania osób niepełnosprawnych”*

Prezentacja dystrybuowana jest bezpłatnie





dr inż. Bartłomiej Świercz
swierczu@dmcs.pl

Katedra Mikroelektroniki i Technik Informatycznych, PŁ

- Godziny przyjęć w pokoju nr 28 w budynku DMCS przy ulicy Wólczańskiej:
 - Wtorek 9.30 – 11.00
- Strona katedry: www.dmcs.pl

Proximetry Poland sp. z o.o.

- Strona firmy: www.proximetry.com





Parę słów o Was ...

- Jaką szkołę i kierunek skończyliście?
- Jakie macie doświadczenie w C++?
- Czy programujecie w C++ komercyjnie?
- Czy bierzecie udział w projektach Open Source?
- Jak oceniacie swój stan wiedzy na temat programowania i języka C++?
- Czego oczekujecie od prowadzonego przedmiotu z C++?
- Czy są szczególnie interesujące Was zagadnienia z C++?





Zaliczenie przedmiotu

- Obecność na wykładach i laboratoriach (min. 80%).
- Samodzielne wykonanie ćwiczeń laboratoryjnych.
- *Aktywność na zajęciach i zabieranie głosu w dyskusji oraz zadawanie pytań będzie bardzo mile widziane i doceniane!*
- *Pytania zadajemy w trakcie zajęć – proszę się nie bać przerwać monologu prowadzącemu.*
- Wszelkie dodatkowe wyjaśnienia jak również nadrobienie ewentualnych braków na godzinach przyjęć.

- Podstawowa zasada: **nie ściemniamy!**





Cel prowadzonego przedmiotu

- Nauka C++ i zrozumienie mechanizmów języka, ich zastosowania oraz przyczyny powstania.
 - **Który z oferowanych przez język mechanizmów należy zastosować i kiedy ?**
 - **Jaki jest koszt użycia danego mechanizmu?**
 - **Czy mechanizm, który używam jest przenośny pomiędzy platformami i kompilatorami?**
- Znajdowanie błędów i zrozumienie komunikatów o błędach drukowanych przez kompilatory
- Optymalizacja kodu
- Łączenie z asemblerem i innymi językami programowania





Czy jest jakaś różnica?

```
for( int i = 0; i < vec.size(); i++ )  
{  
    ....  
}
```

```
for( size_t i = 0; i < vec.size(); ++i )  
{  
    ....  
}
```





Komunikaty o błędach – kod programu

```
#include<vector>
#include<iostream>

int main()
{
    std::vector<int> a;
    std::cout << a << std::endl;
    return 0;
}
```





Komunikaty o błędach – wydruk kompilatora

```
C:\Temp>c:/local/MinGW/bin/g++ a.cc
```

```
a.cc: In function `int main()':
```

```
a.cc:7: error: no match for 'operator<<' in 'std::cout << a'
```

```
c:/local/MinGW/bin/./lib/gcc/mingw32/3.4.5/./././././include/c++/3.4.5/bits/ostream.tcc:63: note: candidates are: std::basic_ostream<_CharT, _Traits>&
std::basic_ostream<_CharT, _Traits>::operator<<(std::basic_ostream<_CharT, _Traits>&*)(std::basic_ostream<_CharT, _Traits>&)) [with _CharT = char,
_Traits = std::char_traits<char>]
```

```
c:/local/MinGW/bin/./lib/gcc/mingw32/3.4.5/./././././include/c++/3.4.5/bits/ostream.tcc:74: note: std::basic_ostream<_CharT, _Traits>&
std::basic_ostream<_CharT, _Traits>::operator<<(std::basic_ios<_CharT, _Traits>&*)(std::basic_ios<_CharT, _Traits>&)) [with _CharT = char, _Traits =
std::char_traits<char>]
```

```
c:/local/MinGW/bin/./lib/gcc/mingw32/3.4.5/./././././include/c++/3.4.5/bits/ostream.tcc:86: note: std::basic_ostream<_CharT, _Traits>&
std::basic_ostream<_CharT, _Traits>::operator<<(std::ios_base&*)(std::ios_base&)) [with _CharT
= char, _Traits = std::char_traits<char>]
```

```
c:/local/MinGW/bin/./lib/gcc/mingw32/3.4.5/./././././include/c++/3.4.5/bits/ostream.tcc:121: note: std::basic_ostream<_CharT, _Traits>&
std::basic_ostream<_CharT, _Traits>::operator<<(long int) [with _CharT = char, _Traits = std::char_traits<char>]
```

```
c:/local/MinGW/bin/./lib/gcc/mingw32/3.4.5/./././././include/c++/3.4.5/bits/ostream.tcc:155: note: std::basic_ostream<_CharT, _Traits>&
std::basic_ostream<_CharT, _Traits>::operator<<(long unsigned int) [with _CharT = char, _Traits = std::char_traits<char>]
```

```
c:/local/MinGW/bin/./lib/gcc/mingw32/3.4.5/./././././include/c++/3.4.5/bits/ostream.tcc:98: note: std::basic_ostream<_CharT, _Traits>&
std::basic_ostream<_CharT, _Traits>::operator<<(bool) [with _CharT = char, _Traits = std::char_traits<char>]
```

```
c:/local/MinGW/bin/./lib/gcc/mingw32/3.4.5/./././././include/c++/3.4.5/ostream:178: note: std::basic_ostream<_CharT, _Traits>&
std::basic_ostream<_CharT, _Traits>::operator<<(short int) [with _CharT = char, _Traits = std::char_traits<char>]
```

```
c:/local/MinGW/bin/./lib/gcc/mingw32/3.4.5/./././././include/c++/3.4.5/ostream:189: note: std::basic_ostream<_CharT, _Traits>&
std::basic_ostream<_CharT, _Traits>::operator<<(short unsigned int) [with _CharT = char, _Traits = std::char_traits<char>]
```

```
c:/local/MinGW/bin/./lib/gcc/mingw32/3.4.5/./././././include/c++/3.4.5/ostream:193: note: std::basic_ostream<_CharT, _Traits>&
std::basic_ostream<_CharT, _Traits>::operator<<(int) [with _CharT = char, _Traits = std::char_traits<char>]
```

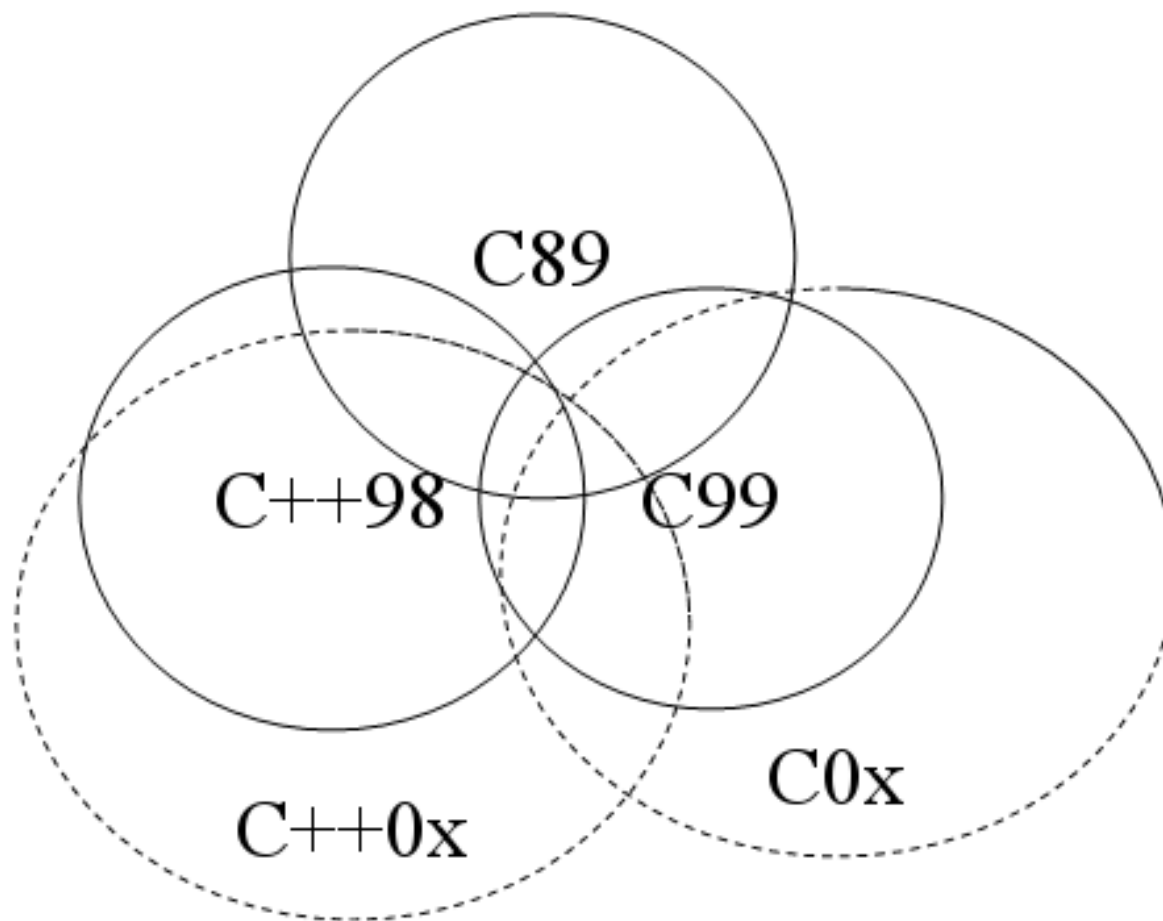
```
c:/local/MinGW/bin/./lib/gcc/mingw32/3.4.5/./././././include/c++/3.4.5/ostream:204: note: std::basic_ostream<_CharT, _Traits>&
std::basic_ostream<_CharT, _Traits>::operator<<(unsigned int) [with _CharT = char, _Traits = std::char_traits<char>]
```

```
c:/local/MinGW/bin/./lib/gcc/mingw32/3.4.5/./././././include/c++/3.4.5/bits/ostream.tcc:170: note: std::basic_ostream<_CharT, _Traits>&
std::basic_ostream<_CharT, _Traits>::operator<<(long long int) [with _CharT = char, _Traits = std::char_traits<char>]
```

```
c:/local/MinGW/bin/./lib/gcc/mingw32/3.4.5/./././././include/c++/3.4.5/bits/ostream.tcc:214: note: std::basic_ostream<_CharT, _Traits>&
std::basic_ostream<_CharT, _Traits>::operator<<(long long unsigned int) [with _CharT = char, _Traits = std::char_traits<char>]
```




Przyszłość języka C++





Standard języka C++ - podstawowy dokument referencyjny dla wszelkich dyskusji na temat języka C++!

- JTC1/SC22/WG21 - The C++ Standards Committee
 - <http://www.open-std.org/jtc1/sc22/wg21/>
- Dokument standaryzujący jest udostępniany odpłatnie, lecz możliwe jest darmowe ściągnięcie tzw. draftu standardu języka C++:
 - <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2005/n1905.pdf>





Literatura dla początkujących

- B. Eckel, „Thinking in C++”, edycja polska, Helion 2002
- B. Stroustrup, „Język C++”, WNT 2002
- N. Josuttis, „C++. Biblioteka standardowa. Podręcznik programisty”, Helion 2003
- K. Dattatri, „Język C++. Efektywne programowanie obiektowe”, Helion 2005





Literatura dla zaawansowanych

- B. Stroustrup, „Język C++”, WNT 2002
- B. Stroustrup, „**Projektowanie i rozwój języka C++**”, WNT 1996
- D. Vandevorode, N. Josuttis, „**C++ szablony**”, Helion 2003
- D. Abrahams, A. Gurtovoy, „Język C++. Metaprogramowanie za pomocą szablonów”, Helion 2005
- S. Lippman, „**Model obiektu w C++**”, WNT
- S. Oualline, „Jak nie programować w C++”, Mikom 2003
- H. Sutter, „**Wyjątkowy język C++**”, WNT 2002
- R. Alexander, G. Bensley, „C++. Optymalizacja oprogramowania”, RM 2001





Literatura – pliki nagłówkowe

- Pliki nagłówkowe są prawdziwą skarbnicą wiedzy i najpewniejszym źródłem wiedzy o interfejsach!
- Nie należy bać się ich czytać!
- Przykład ze standardowego pliku limits.h:

```
/*  
 * Maximum and minimum values for shorts.  
 */  
#define SHRT_MAX      32767  
#define SHRT_MIN      (-SHRT_MAX-1)  
  
#define USHRT_MAX     0xffff
```





- Wikipedia:
 - <http://pl.wikipedia.org/wiki/C%2B%2B>
 - <http://en.wikipedia.org/wiki/C%2B%2B>
- C++ FAQ Lite:
 - <http://www.parashift.com/c++-faq-lite/>
- The C++ Programming Language:
 - <http://www.research.att.com/~bs/C++.html>
- C++ bez cholesterolu:
 - <http://www.intercon.pl/~sektor/cbx/>





- Grupy dyskusyjne:
 - pl.comp.lang.c
 - pl.comp.os.linux.programowanie (podobne grupy są też dla innych systemów, np. MS Windows)
 - comp.lang.c++
 - comp.lang.c++.moderated
 - pl.comp.programming
 - comp.realtime
- Do czytania grup dyskusyjnych można skorzystać z:
 - <http://groups.google.com/>





Tonący brzytwy się chwyta ...

www.google.com

:)





Narzędzia i kompilatory

- Kompilator GCC (g++)
 - Standardowo instalowany w środowisku Linux
 - MinGW dla systemu Windows
 - Cygwin dla systemu Windows
- Visual Studio 2008 Express Editions
- Debugery: gdb + ddd/cgdb
- Narzędzia GNU binutils i textutils
- System budowania GNU Make
- Edytory i IDE: gvim, pspad, Eclipse + CDT, Dev-C++, dowolny edytor tekstowy
- Profilery i analizatory kodów źródłowych





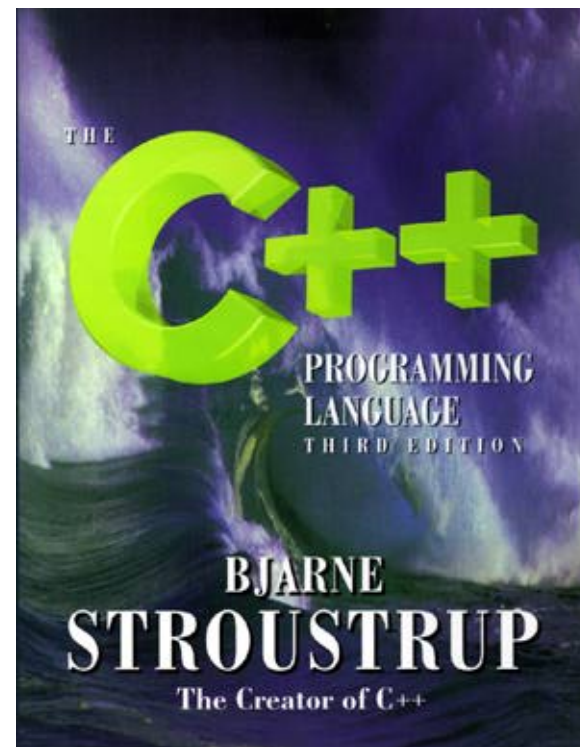
Biblioteki języka C++

- Biblioteki graficznego interfejsu użytkownika:
 - QT: <http://www.qtsoftware.com/>
 - GTKmm: <http://www.gtkmm.org/>
 - WxWidgets: <http://www.wxwidgets.org/>
 - FLTK: <http://www.fltk.org/>
- Biblioteki ogólnego stosowania:
 - Boost: <http://www.boost.org/>
- Biblioteki wyspecjalizowane:
 - VTK: <http://www.vtk.org/>
 - Boehm: http://www.hpl.hp.com/personal/Hans_Boehm/gc/





Bjarne Stroustrup



- Strona domowa: <http://www.research.att.com/~bs/>
- C++: <http://www.research.att.com/~bs/C++.html>

Bartłomiej Świercz – Katedra Mikroelektroniki i Technik Informatycznych





- Historia
- Co nam zapewnia?
- Czego nie oferuje?
- C++ a inne języki programowania (assembler, Java, języki skryptowe, języki deklaratywne)
- Czemu ma opinię trudnego i skomplikowanego? Czy faktycznie tak jest?





```
#include<iostream>
```

```
int main()
```

```
{
```

```
    float res;
```

```
    res = 1/3;
```

```
    std::cout << "Wynik: " << res;
```

```
    return 0;
```

```
}
```





KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Zaawansowane programowanie w języku C++ Wstęp

Prezentacja jest współfinansowana przez
Unię Europejską w ramach
Europejskiego Funduszu Społecznego w projekcie pt.

*„Innowacyjna dydaktyka bez ograniczeń - zintegrowany rozwój Politechniki Łódzkiej -
zarządzanie Uczelnią, nowoczesna oferta edukacyjna i wzmocnienia zdolności do
zatrudniania osób niepełnosprawnych”*

Prezentacja dystrybuowana jest bezpłatnie

