

1. Вовед во C++

Структурирано програмирање



Д-р Рамона Маркоска, вонр.проф

Преглед на Предмети

Имате вкупно 5 предмети за прегледување

Сите проблеми со системот за анкетирање, пријавете ги на iknow@uklo.edu.mk.

Летен / 2018

↓

Барај курсеви...

Пребарај

Предмети

Структурно програ...

Факултет: ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ И КОМУНИКАЦИ...

Семестар: Летен

Код: МИО 202

Циклус на студии: Прв циклус

Прооек: 5.00

Структурно програ...

Факултет: ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ И КОМУНИКАЦИ...

Семестар: Летен

Код: ИМСА 202

Циклус на студии: Прв циклус

Прооек: 4.00

Вовед во програми...

Факултет: ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ И КОМУНИКАЦИ...

Семестар: Летен

Код: ИКТ-122

Циклус на студии: Прв циклус

Прооек: 4.17

Напредно програми...

Факултет: ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ И КОМУНИКАЦИ...

Семестар: Летен

Код: ИИКИ-202

Циклус на студии: Прв циклус

Прооек: 4.13

Софтверски барања...

Факултет: ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ И КОМУНИКАЦИ...

Семестар: Летен

Код: ИИКИ-611

Циклус на студии: Прв циклус

Прооек: 4.24

Основни информации

- Начин на работа- предавања, аудиториски вежби, лабораториски вежби
- Правила за работа – ресурси, страници, информации, **!!! Комуникација !!!**
- Организација на активностите – редовност, чекирање присуство, проекти, тестови, тип на задачи.
- Просторно-временска организација- тајминг, измени, распоред, групи.

Запознавање со предметот и начинот на работа

- Препорачани ресурси:
- www.ramona-markoska.info со линкови до сите релевантни страни
- www.cplusplus.com
- Bjarne Stroustrup's homepage
www.stroustrup.com/
- C++ како се програмира – Деител, библиотека ТФБ
- Збирка задачи по Вовед во програмирање 1-ТФБ, Рамона Маркоска

Во ова поглавје!



- Запознавање со предметот и правилата на работа
- Презентирање на ресурси и договор за користење
- Презентирање на содржината која ќе се изучува.
- Историјат на програмските јазици
- Вовед во C/C++
- Вовед во практична работа: Демонстрација на користење на работна околина Dev C++ 4.9.9.2

Принципи за учење на програмирање

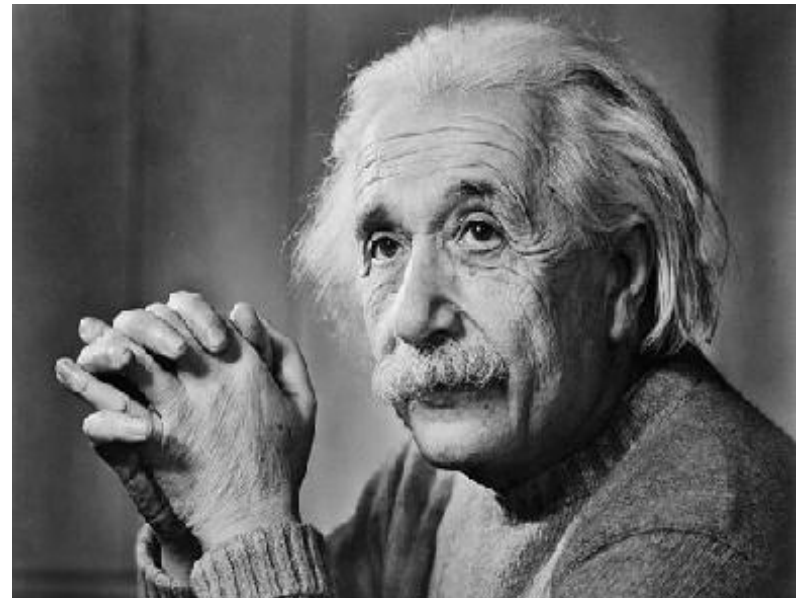
Einstein : Дел од мисли посебно згодни за корелација со програмирање и C++

“Imagination is more important than knowledge “

“ Logic will get you from A to B.
Imagination will take you
everywhere “ -

“Anyone can know - the point is to
understand “.

“Education is what remains after one
has forgotten what one has learned
in school.”



Вовед во програмирање како концепт - од C до C++ 11

- Што ?
- Зошто ?
- Како ?

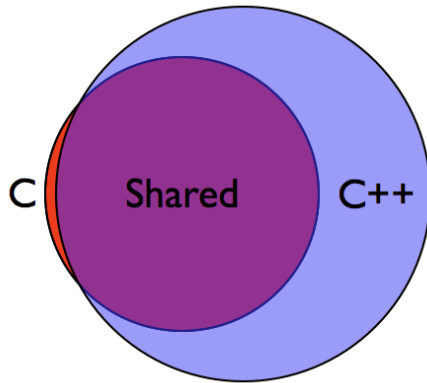
Со која цел ?

”Кога мапите не се согласуваат со теренот- верувајте му на теренот” – Швајцарска Армија, поговорка

“Не може да научите да возите велосипед во допишувајќи се со инструкторот”



C++ v.s. C



C++

bike->color
bike->brand
person->name

bike

person



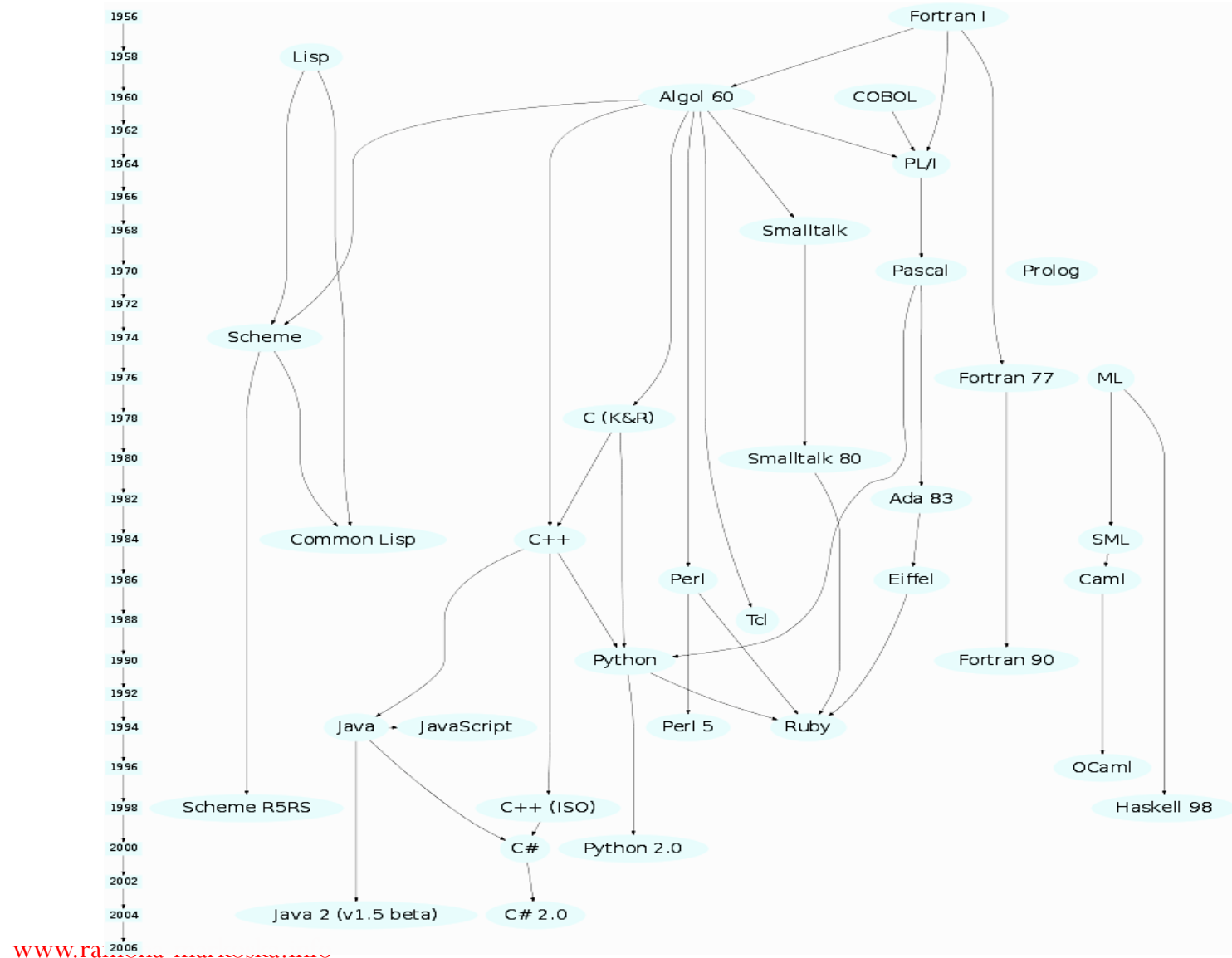
You have no class.



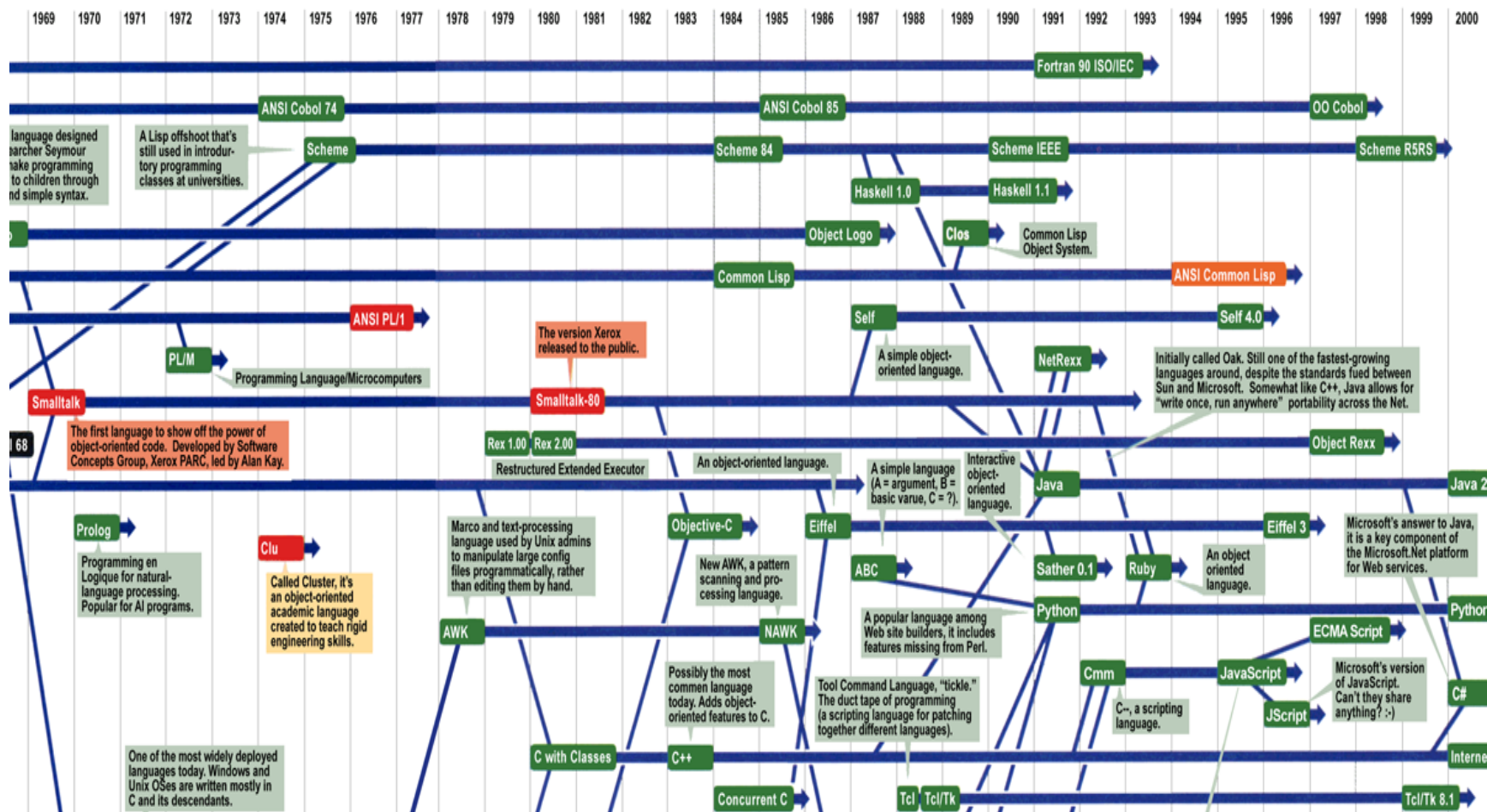
++



Историски развој на програмските јазици и софтверот

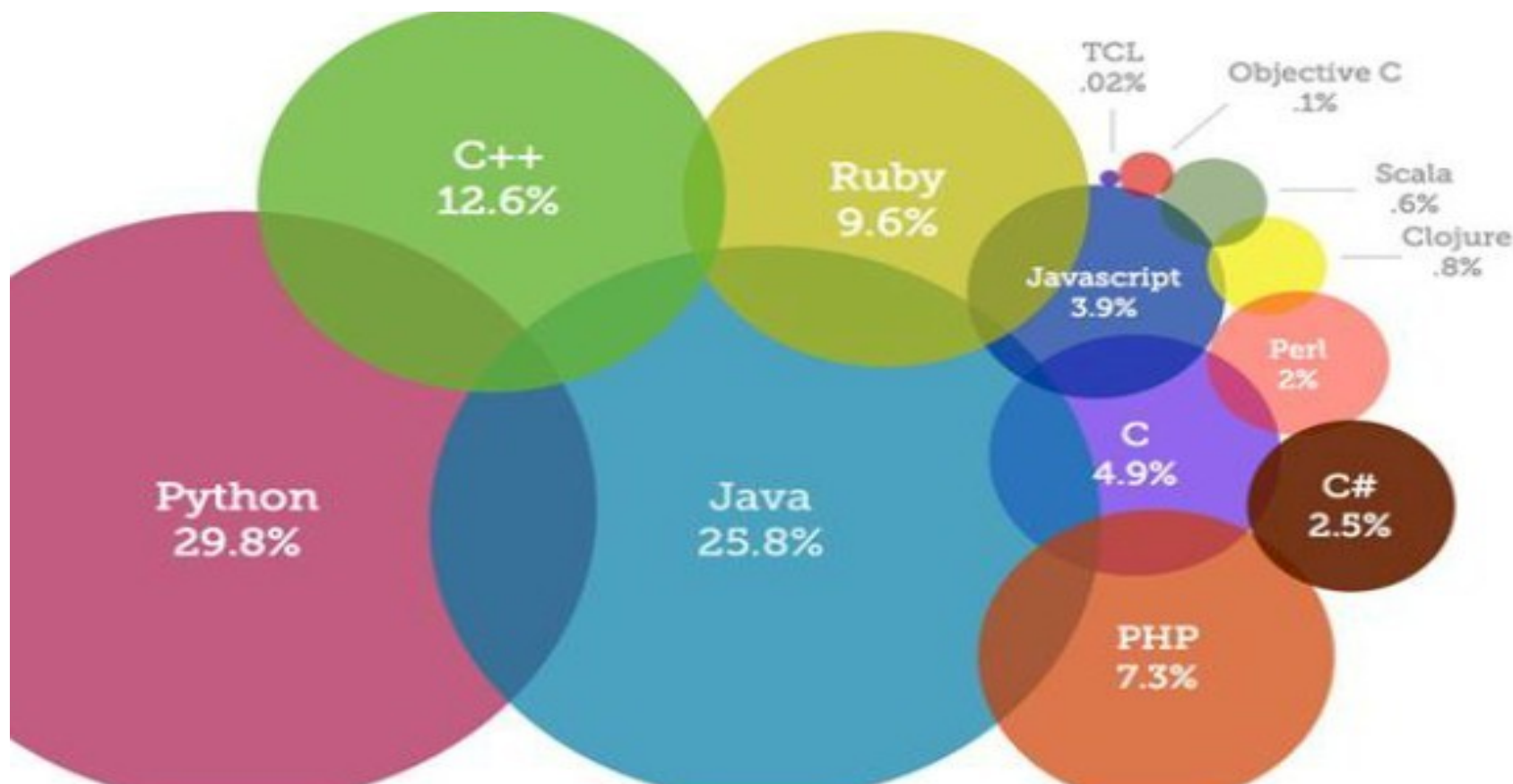


Историски развој на програмските јазици и софтверот

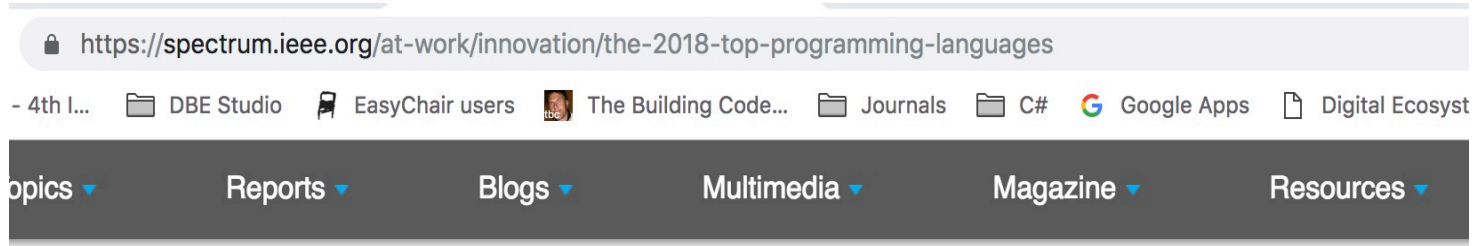


Популарност на програмските јазици























<https://www.tiobe.com/tiobe-index/>



<https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages>
























So what are the Top Ten Languages of 2018, as ranked for the typical IEEE member and *Spectrum* reader?


| Language Rank | Types | Spectrum Ranking |
|---------------|---|------------------|
| 1. Python |    | 100.0 |
| 2. C++ |    | 99.7 |
| 3. Java |    | 97.5 |
| 4. C |    | 96.7 |
| 5. C# |    | 89.4 |
| 6. PHP |  | 84.9 |
| 7. R |  | 82.9 |
| 8. JavaScript |   | 82.6 |
| 9. Go |   | 76.4 |
| 10. Assembly |  | 74.1 |


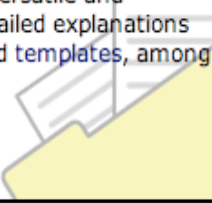
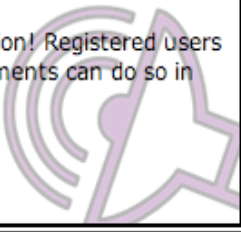
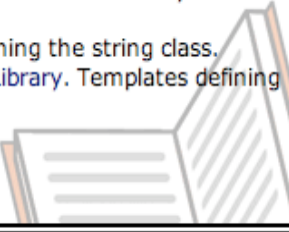

Python has maintained its grip on the No. 1 spot. Last year it came out on top by just barely beating out C, with Python's score of 100 to C's 99.7. Now C++

<http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages>

| Language Rank | Types | Spectrum Ranking | Spectrum Ranking |
|---------------|---|------------------|------------------|
| 1. Java |    | 100.0 | 100.0 |
| 2. C |    | 99.9 | 99.3 |
| 3. C++ |    | 99.4 | 95.5 |
| 4. Python |   | 96.5 | 93.5 |
| 5. C# |    | 91.3 | 92.4 |
| 6. R |  | 84.8 | 84.8 |
| 7. PHP |  | 84.5 | 84.5 |
| 8. JavaScript |   | 83.0 | 78.9 |
| 9. Ruby |   | 76.2 | 74.3 |
| 10. Matlab |  | 72.4 | 72.8 |

C++ pecypc online

Welcome to **cplusplus.com**  © The C++ Resources Network, 2011

| | |
|--|--|
| <h3>Information</h3> <p>General information about the C++ programming language, including non-technical documents and descriptions:</p> <ul style="list-style-type: none">• Description of the C++ language• History of the C++ language• F.A.Q., Frequently Asked Questions  | <h3>C++ Documentation</h3> <p>Documentation pages and tutorials describing specifics about the different features of the C++ language.</p> <ul style="list-style-type: none">• C++ Language Tutorial: Collection of tutorials covering all the features of this versatile and powerful language. Including detailed explanations of pointers, functions, classes and templates, among others...• more...  |
| <h3>Forum</h3> <p>Message boards where members can exchange knowledge and comments. Ordered by topics:</p> <ul style="list-style-type: none">• General C++ Programming• Beginners• Windows• UNIX/Linux <p>This section is open to user participation! Registered users who wish to post messages and comments can do so in this section.</p>  | <h3>Library Reference</h3> <p>Description of the most important classes, functions and objects of the Standard Language Library, with descriptive fully-functional short programs as examples:</p> <ul style="list-style-type: none">• C library: The popular C library, is also part of the of C++ language library.• IOStream library. The standard C++ library for Input/Output operations.• String library. Library defining the string class.• STL: Standard Template Library. Templates defining containers, algorithms...• more...  |
| <h3>C++ Search</h3> <p>Search this website:</p> <input type="text"/> <input type="button" value="Search"/> | <h3>Sourcecodes</h3> <p>Source code of fully-functional programs made available through this website by our users.</p>  |

Видови на софтвер - потсетување



Историјат на програмски јазик C/C++

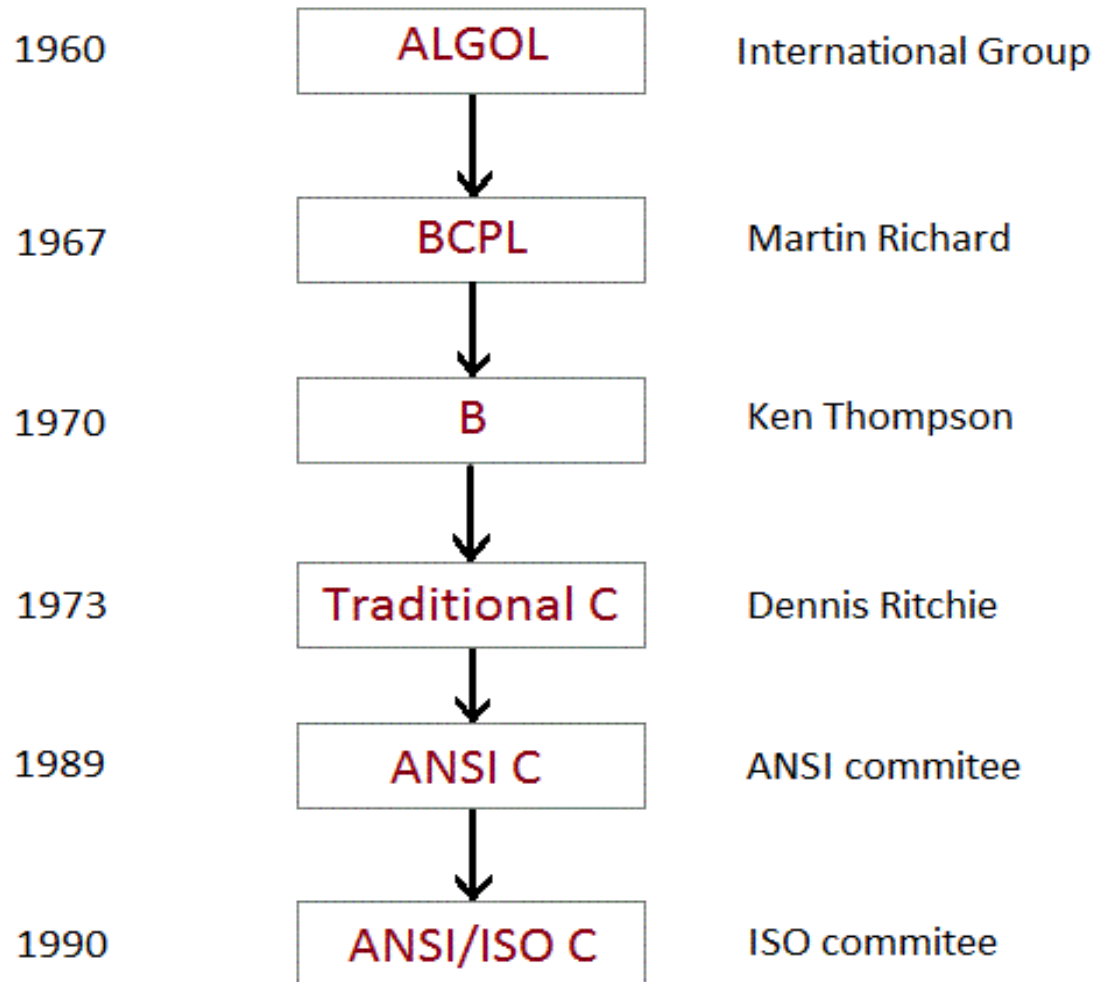
Timeline C

Настанок: 70 години од минат век: Ken Thompson, “B” и Denis Ritchie, “C”, стандардизација, ANSI 1980. Првенствена намена одржување на UNIX o.s. (Постоеа и “A”, а од 2007 постои и “Д” јазик=.

Барања кои требало да ги исполни:

- Виш програмски јазик со имплементирани можности на машински јазик.
 - Директен пристап до меморија, поголемо множество на оператори
 - Погодности на лесна разбирливост на виш програмски јазик
 - Моќ, стабилност, флексибилност и брзина на извршување на машински јазик
- Релативно едноставна синтакса, и ограничен сет на наредби, но надоградлив и со мноштво развојни библиотеки на функции, што овозможува едноставно програмирање, компајлирање и голема флексибионост.

Фази на развој на C

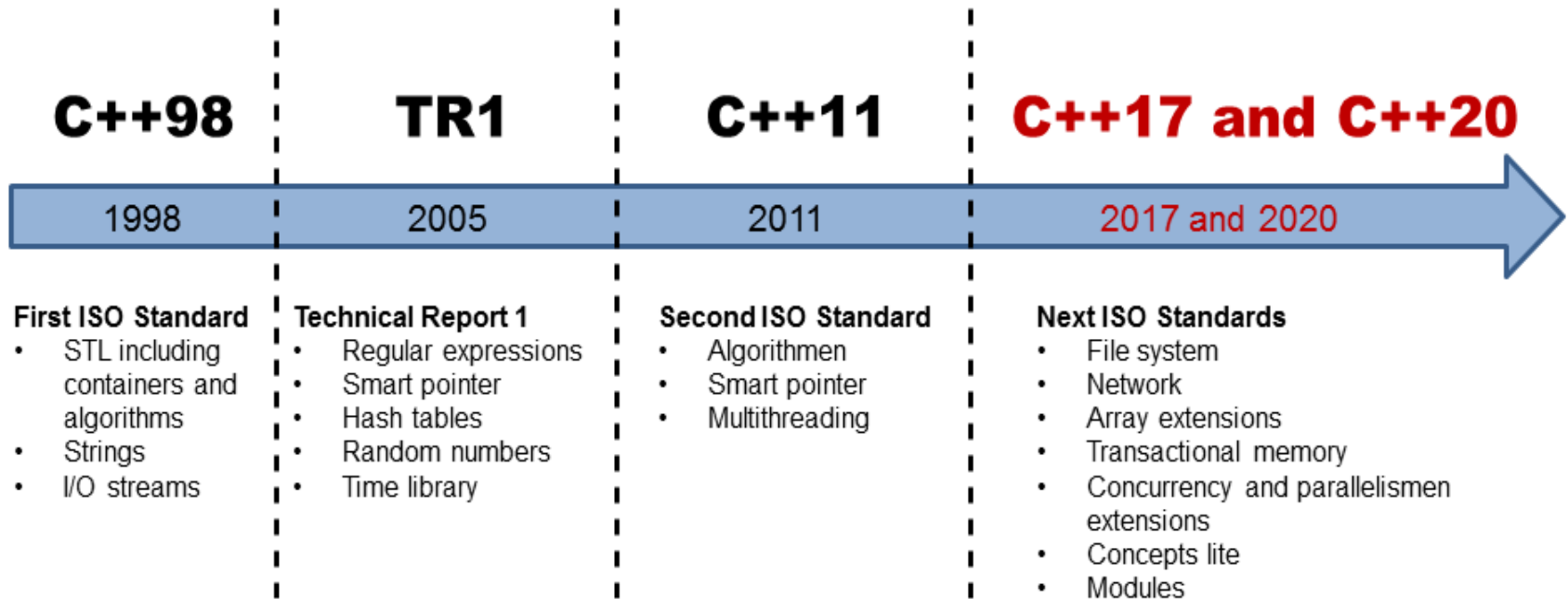


Timeline C++

- Со појавата на објектно ориенираните концепти во програмирањето, се јавува потреба од надоградба и измена на C:
- **1980 Појава на C со класи:**
 - Структурите овозможуваат податоците да имаат само вредности
 - Класите овозможуваат и дефинирање на операции над податоците
- **1983, настанок на C++ :** Екстензијата значи дека станува збор за надоградба на C, со тоа што се додадени виртуелни функции , овозможено е преклопување на операторите Bjarne Stroustrup. Стандардизација ANSI 1995.

Развој и стандардизација на C++

извор: <http://www.modernescpp.com/>



1.3. Елементи на еден програмски јазик

- Синтакса и семантика на програмскиот јазик
- Синтаксни елементи:
 - Знаци
 - Имиња
 - Резервирни зборови
 - Идентификатори
 - Типови на податоци- класификација.
 - Наредби- едноставни и сложени,
 - Контролни структури
 - Алгоритми (псевдокод и негово значење)

1.5. Постапка на програмирање, чекор по чекор

- Мала дискусија кога вреди некој проблем од техника, математика или друга област да се решава со програмирање.
- Предности и недостатоци на програмирањето в.с. класично решавање на проблемите.
 - Покрај математичките формулации- замена и на логиката на одлучување , кога се решава на лице место.
 - Организација на влез-излез
 - Користење на работна развојна околина (IDE(Integrated Development Environment)) и познавање на синтакса.

С ++ организација



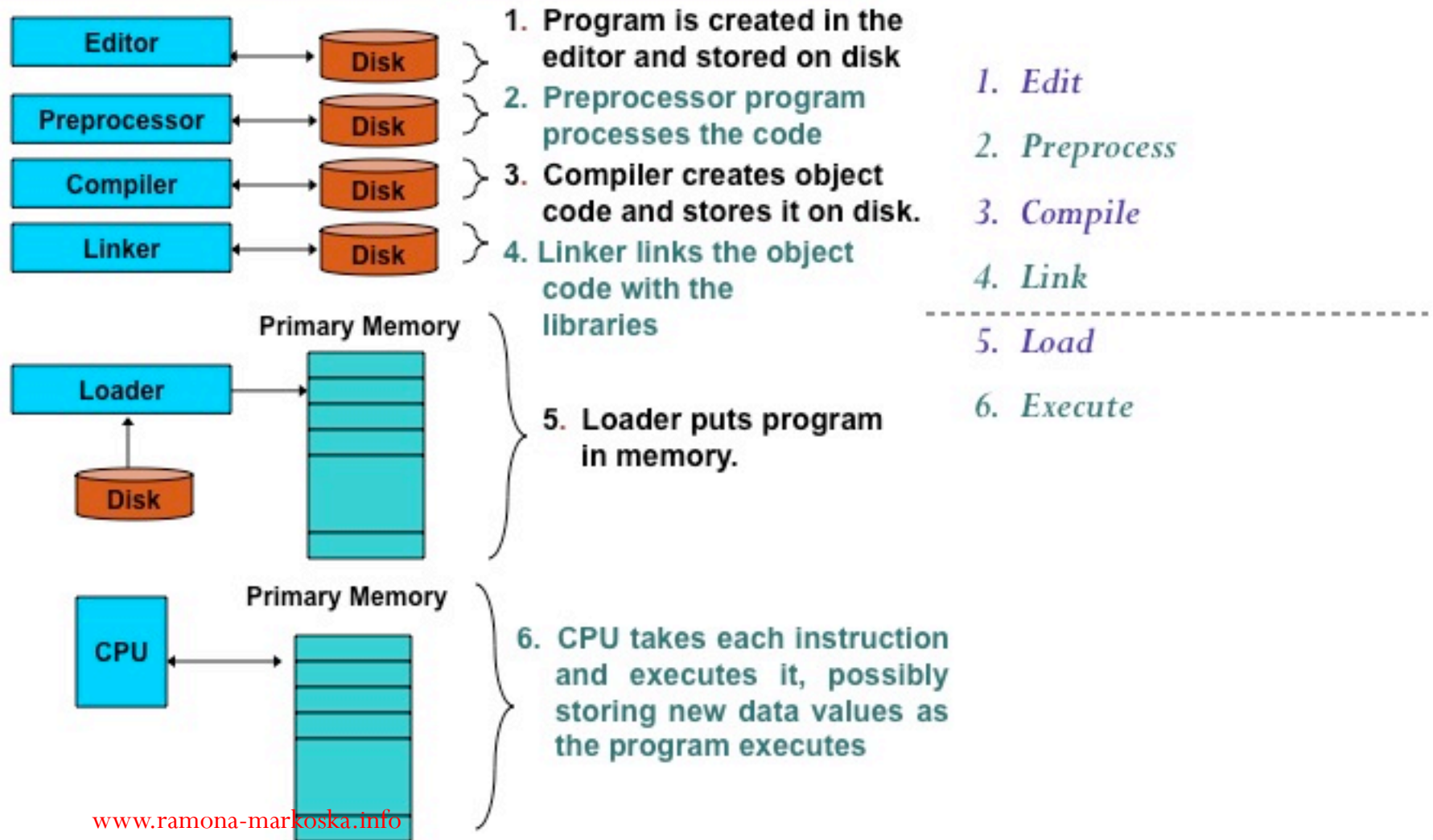
!! Да се коментира потреба од навраќање на содржините :
Зошто и со која крајна цел

1.5.1. Интегрирана развојна околина- IDE(Integrated Development Environment)

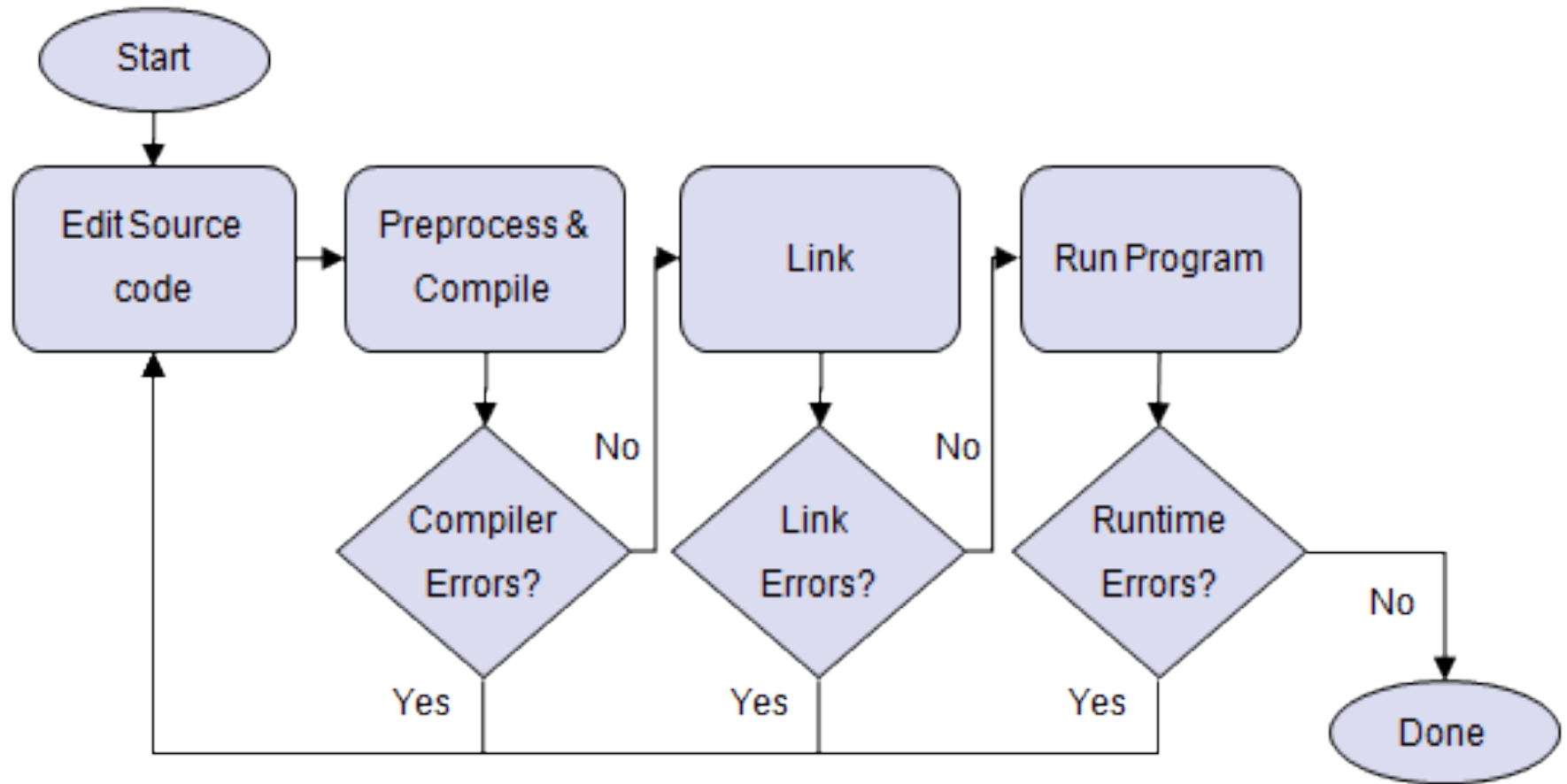
- **Едитор**- околина во која се пишува програмскиот код
- **Изворен код**- напишан според синтакса и семантика на програмскиот јазик
- **Машински јазик**- начин на кој процесорот ги “разбира” и решава проблемите.
- **Компајлер**- преведувач, кој го обработува изворниот код во облик разбирлив за процесорот, .т.н. објектен или извршен код.
- **Дебагер**-дел кој овозможува детектирање на грешките во изворниот код, со цел да може да се компајлира
- **Линкер** – во случај на потреба поврзува со библиотеки и ф-ции.
- **Предпроцесор**

A Typical C Program Development Environment

• Phases of C Programs:



Развоен циклус и алатки во C++



C++ - ОСНОВИ

- Структура на C++ програма
- Работна Dev C++ околина
- Константи и променливи
- Оператори
- Базични наредби за Влез-Излез

1.5.2. Структура на C++ програма

```
1 // mojata prva programa vo C++
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     cout << "Здраво студенти на ФИКТ ";
8     cout << "Јас сум Вашиот прв пример";
9     return 0;
10 }
```

Get URL

options

compilation

execution

Здраво студенти на ФИКТ Јас сум Вашиот прв пример

www.fakultetna-ima-koska.mk

1- // Коментар видлив само за програмерот, се поставува на почеток на секој ред или пак содржината се ограничува помеѓу /* и */. При компајлирање и извршување вака напишаните коментари не се појавуваат на конзола.

2. #include- директива на предпроцесор за вклучување на разни библиотеки наредби чии имиња се пишуваат во продолжение. Во случајов фајл <iostream> - во кој се содржани ба зични i/o декларации. Често се среќава cstdlib – библиотека на стандардни наредби 6 и 10. Почеток и крај на главната програма

7,8 – наредби за печатење. Постари компајлери, т.е. Нивни верзии не поддржуваат секогаш кирилица.

1.5.3. Структура на C++ програма

```
1 // mojata prva programa vo C++
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     cout << "Здраво студенти на ФИКТ ";
8     cout << "Јас сум Вашиот прв пример";
9     return 0;
10 }
```

Get URL

options

compilation

execution

Здраво студенти на ФИКТ Јас сум Вашиот прв пример

www.fanona-ma.com.mk

3. Користење на namespace std- имплицира користење на стандардни именувања на наредбите. Со тоа се спречува ситуација на забуна и нееднозначност даво некоја итерација различни библиотеки во себе имаат две наредби со исто име а различни намени.

5. Почеток на главна main функција (содржина) – напишаните делови пред main се напатствија за преведувачот (компајлерот) како да ја преведува програмата која следи, и кои библиотеки да ги користи.

9. Означува крај на активностите. Во зависност од облик на main, некои компајлери работат

без return 0;

1.5.3. Структура на C++ програма

```
1 // vtor primer ili varijanta na prvata programa C++
2 #include <iostream>
3
4 int main ()
5 {
6     std::cout << "Здраво студенти ! ";
7     std::cout << "Можам и без using namespace .. :) , ама тогаш ми треба std пред секој ред :( !";
8 }
```

Доколку не се користи `using namespace std`; треба на почетокот на секој ред каде се користат наредби чие значење бара “појаснување” да се пишува **`std ::`** .
Наведувањето на правилата и директивите пред `main` е со цел да се избегне нивно повеќекратно вметнување и повикување во кодот

Get URL

Run

options

compilation

execution

Здраво студенти ! Можам и без using namespace .. :) , ама тогаш ми треба std пред секој ред :(!

1.5.4 Базични елементи на C++ програма - појаснување

1. Коментари

- `// komentar vo eden red`
- `/* komentar vo dadeni granici */`
- `cout<<"zdravo !!"<<endl;`
- Специфики

2. Резервирани имиња и зборови

`asm, auto, bool, break, case, catch, case, continue, default, delete, ... goto, if, namespace, new, ..typedef, enum ...`

Види:

<http://cplusplus.com/doc/tutorial/variables/>

Вежби и практични примери

- Дел со појаснување за околината и прв дел од задачите со вежби.
- Општи и почетни содржини за кои се очекува ниво на преднаења од претходно образование. Повторување и дополнување на линкови
- <http://cplusplus.com/doc/tutorial/variables/>
- <http://mendo.mk/Lecture.do?id=6>
- <http://mendo.mk/Lecture.do?id=7>

